

**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA – EST**

THIERRY-YVES ALVES ARAÚJO

**DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO DE
MONITORAMENTO DE TEMPERATURA E GEOLOCALIZAÇÃO DE
VACINAS DURANTE O TRANSPORTE E EM UNIDADES BÁSICAS DE
SAÚDE POR MEIO DE TELEMETRIA PARA DISPOSITIVOS MÓVEIS**

Manaus

2023

THIERRY-YVES ALVES ARAÚJO

**DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO DE
MONITORAMENTO DE TEMPERATURA E GEOLOCALIZAÇÃO DE
VACINAS DURANTE O TRANSPORTE E EM UNIDADES BÁSICAS DE
SAÚDE POR MEIO DE TELEMETRIA PARA DISPOSITIVOS MÓVEIS**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletrônico.

Orientador: Jozias Parente de Oliveira, Dr.

Manaus

2023

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia – EST

Reitor:

André Luiz Nunes Zogahib

Vice-Reitor:

Kátia do Nascimento Couceiro

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Eletrônica:

Bruno da Gama Monteiro

Banca Avaliadora composta por: Data da defesa: 09/03/2023.

Prof. Jozias Parente de Oliveira, Dr (Orientador)

Prof. Fábio de Sousa Cardoso, Dr

Prof. Bruno da Gama Monteiro, Me

CIP – Catalogação na Publicação

Araújo, Thierry-yves Alves

Desenvolvimento de protótipo de dispositivo de monitoramento de temperatura e geolocalização de vacinas durante o transporte e em unidades básicas de saúde por meio de telemetria para dispositivos móveis/ Thierry-yves Alves Araújo; [orientado por] Jozias Parente de Oliveira. – Manaus: 2023.

72 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2023.

1. Monitoramento de vacinas. 2. GPS. 3. GSM. 4. Cartão de memória SD. 5. ESP32. 6. BMP280. I. Oliveira, Jozias Parente de.

THIERRY-YVES ALVES ARAÚJO

**DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO DE
MONITORAMENTO DE TEMPERATURA E GEOLOCALIZAÇÃO DE VACINAS
DURANTE O TRANSPORTE E EM UNIDADES BÁSICAS DE SAÚDE POR MEIO
DE TELEMETRIA PARA DISPOSITIVOS MÓVEIS**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletrônico.

Nota obtida: 10 (Dez)

Aprovada em 09 / 03 / 2023

Área de concentração: Processamento Digital de Sinais

BANCA EXAMINADORA

Jozias Parente de Oliveira
Orientador: Jozias Parente de Oliveira, Dr.

Fábio de Sousa Cardoso
Avaliador: Fábio de Sousa Cardoso, Dr.

Bruno da Gama Monteiro
Avaliador: Bruno da Gama Monteiro, Me.

Manaus 2023

AGRADECIMENTOS

A Deus, por todos os dons e bençãos que me deu e que me fizeram chegar até aqui. À minha família, a quem eu amo incondicionalmente e sempre tenho em meus pensamentos. À Bárbara Cortez, por todo o amor, apoio, incentivo e direcionamento que foram essenciais. Aos meus amigos da faculdade, que me ajudaram no decorrer da graduação, em especial ao meu amigo Gabriel Rocha. Aos professores que contribuíram durante a formação acadêmica, em especial ao professor e orientador Jozias pelos ensinamentos e conselhos.

O verdadeiro TCC são os amigos que fizemos na jornada...

RESUMO

Durante a pandemia de Covid-19, o mundo precisou parar e direcionar seus esforços para a produção de vacinas. As vacinas produzidas por laboratórios internacionais e nacionais são distribuídas entre os entes federativos pelo Programa Nacional de Imunizações (PNI) por vias aéreas, terrestres ou aquáticas. As vacinas precisam de cuidados especiais para manter sua capacidade de resposta, porém relatórios de unidades básicas de saúde apontaram que quantidades significativas de vacinas são perdidas por falhas nos equipamentos de refrigeração ou quedas de energia que ocorrem fora do período de expediente. Em vista do problema, o objetivo deste trabalho foi construir um protótipo de dispositivo de monitoramento capaz de capturar dados do sensor de temperatura BMP280 e do módulo *Global Positioning System* (GPS), para armazená-los num cartão de memória *Secure Digital* (SD), e enviar mensagens de *Short Messaging Service* (SMS) de alerta de temperaturas não ideais, se necessário, para um dispositivo móvel usando um módulo *Global System for Mobile Communications* (GSM). Com o auxílio de uma caixa de isopor e uma bolsa térmica em gel congelada, o dispositivo foi testado em 2 cenários: no primeiro cenário o dispositivo foi deixado estático num local para coleta de dados. No segundo cenário, o dispositivo foi levado em um trajeto pela cidade. Como resultado dos testes, foram gerados os arquivos de registros de dados, nos quais estavam os dados de geolocalização e temperatura. Os dados de geolocalização apontavam para o mesmo local do cenário 1 e trajeto feito no cenário 2 quando comparados com dados no aplicativo Google Maps e Google Earth Pro. Os dados de temperatura apresentaram um maior erro absoluto de 2,32 °C acima do medido pelo multímetro no primeiro cenário, já no segundo cenário foi de 2,06 °C. Também foram recebidos os SMS com os dados de monitoramento quando a temperatura não estava na faixa ideal. Por fim, analisando os dados obtidos, foi possível concluir que o protótipo de dispositivo desenvolvido é capaz de monitorar temperatura e geolocalização, registrar estes dados em dispositivo de armazenamento e enviar alertas de temperatura para um dispositivo móvel.

Palavras-chave: Monitoramento de vacinas, GPS, GSM, Cartão de memória SD, ESP32, BMP280.

ABSTRACT

During the Covid-19 pandemic, the world needed to stop and direct its efforts towards vaccine production. Vaccines produced by international and national laboratories are distributed among federative entities by the National Immunization Program (PNI) by air, land or water routes. Vaccines need special care to maintain their responsiveness, but reports from basic health units point out that significant amounts of vaccines are lost due to failures in refrigeration equipment or power outages that occur outside working hours. In view of the problem, the aim of this work was to build a prototype of a monitoring device capable of capturing data from the BMP280 temperature sensor and the Global Positioning System (GPS) module, to store them on a Secure Digital (SD) memory card, and send Short messages Messaging Service (SMS) alerting of non-ideal temperatures, if needed, to a mobile device using a Global System for Mobile Communications (GSM) module. With the aid of a styrofoam box and a frozen thermal gel bag, the device was tested in 2 scenarios: in the first scenario, the device was left static in a place for data collection. In the second scenario, the device was taken on a journey through the city. As a result of the tests, data log files were generated, which contained geolocation and temperature data. The geolocation data pointed to the same location in scenario 1 and the route taken in scenario 2 when compared with data in the Google Maps and Google Earth Pro application. The temperature data showed a greater absolute error of 2,32 °C above that measured by the multimeter in the first scenario, while in the second scenario it was 2,06 °C. SMS with monitoring data were also received when the temperature was not within the ideal range. Finally, analyzing the data obtained, it was possible to conclude that the developed device prototype is capable of monitoring temperature and geolocation, recording this data in a storage device and sending temperature alerts to a mobile device.

Keywords: Vaccine monitoring, GPS, GSM, SD memory card, ESP32, BMP280.

SUMÁRIO

INTRODUÇÃO	8
1 REFERENCIAL TEÓRICO	10
1.1 VACINAS	10
1.2 DISTRIBUIÇÃO DE VACINAS	11
1.3 PRECAUÇÕES DURANTE O TRANSPORTE DE VACINAS	13
1.4 TRANSPORTE DE IMUNOBIOLOGICOS DENTRO DA CADEIA DE FRIO	14
1.5 PERDAS	15
1.6 MICROCONTROLADOR ESP32	16
1.7 SISTEMAS EMBARCADOS	17
1.8 GPS	18
1.9 GSM/GPRS	20
1.10 SD CARD	21
1.11 SENSOR DE TEMPERATURA BMP280	23
2 MATERIAIS E MÉTODOS	25
2.1 AMBIENTE DE DESENVOLVIMENTO	25
2.2 MATERIAIS	25
2.3 MÉTODOS	26
2.3.1 Levantamento de requisitos	27
2.3.2 Definição da arquitetura de hardware	27
2.3.3 Desenho do esquema elétrico	27
2.3.4 Desenho do layout	28
2.3.5 Montagem da placa	28
2.3.6 Definição da arquitetura de firmware	28
2.3.7 Desenvolvimento de firmwares de teste dos periféricos	28
2.3.8 Integração do firmware	29
2.3.9 Integração entre hardware e firmware	29
2.3.10 Definição dos cenários de testes	29
2.3.11 Modelagem dos itens mecânicos	29
2.3.12 Prototipagem dos itens mecânicos	30
2.3.13 Testes de validação	30
2.3.14 Avaliação de desempenho	30
3 IMPLEMENTAÇÃO DO PROJETO	31
3.1 PREPARAÇÃO DO AMBIENTE NA ARDUINO IDE	31
3.2 DESENVOLVIMENTO DO PROTÓTIPO DE HARDWARE	34
3.3 DESENVOLVIMENTO DOS SCRIPTS DE TESTE DE CADA PERIFÉRICO	39
3.4 DESENVOLVIMENTO DO SCRIPT GERAL DE INTEGRAÇÃO DO PROTÓTIPO	48
3.5 VALIDAÇÃO FINAL DO PROTÓTIPO DO DISPOSITIVO	54
4 ANÁLISE DOS RESULTADOS	57
4.1 PREPARAÇÃO DOS TESTES	57
4.2 PRIMEIRO CENÁRIO DE TESTES	57
4.3 SEGUNDO CENÁRIO DE TESTES	60
4.3.1 Trajetória no cenário de teste	62
4.4 ALERTAS DE TEMPERATURA	63

4.5 COMPARAÇÃO COM PRODUTOS DE MERCADO	64
CONCLUSÃO.....	65
REFERÊNCIAS	67

INTRODUÇÃO

Durante a pandemia de Covid-19, o mundo precisou parar e focar seus esforços em achar uma resposta eficaz e duradoura, que foram as vacinas. As vacinas foram produzidas por laboratórios espalhados pelo mundo e chegaram ao Brasil através do PNI, que tem o dever de adquirir e coordenar as atividades de vacinação da população. As vacinas são distribuídas entre os entes federativos de acordo com sua demanda, usando vias aéreas, terrestres ou aquáticas, e precisam de cuidados especiais para manter sua capacidade de resposta à doença.

No Brasil, várias unidades básicas de saúde relatam perda de vacinas durante todos os meses do ano e, em sua maioria, são causadas por falhas nos equipamentos de refrigeração ou quedas de energia que ocorrem fora do período de expediente, durante a noite ou em finais de semana (SANTOS *et al.*, 2016), fazendo com que as vacinas passem longos períodos fora da faixa de temperatura indicada para armazenamento. Outro ponto de insegurança é durante o transporte das vacinas, que podem ser alvo de bandidos que roubam as cargas.

Tendo em vista o problema supracitado, o objetivo deste trabalho foi construir um protótipo de dispositivo capaz de capturar dados de temperatura e geolocalização, para guardá-los em dispositivo de armazenamento e enviar esses e alertas sobre temperaturas não ideais, se necessário, para um dispositivo móvel. Este dispositivo foi anexado a uma caixa térmica de onde foram coletados os dados de entrada, temperatura e geolocalização, para serem monitorados e registrados durante trajetos pela cidade. Os resultados de saída são disponibilizados por meio dos registros de geolocalização e temperatura salvos no cartão SD e, também, por meio do envio de mensagem de SMS com os dados para um dispositivo móvel em tempo real. Portanto, o presente trabalho se propôs a mitigar os efeitos da falta de monitoramento em tempo real dos lotes de vacina, que inibe a possibilidade de ações reparadoras contra intercorrências que causam a perda dos lotes de vacina, como extravios, falhas em equipamentos de refrigeração e quedas de energia.

Uma das justificativas para o desenvolvimento deste trabalho foi fornecer um meio de obter informações em tempo real de status de lotes das vacinas, tendo em mente que grande parte da perda de vacinas acontece por problemas de alteração de temperatura, em decorrência de quedas de energia ou falhas no equipamento, ou por roubos durante o seu transporte. Além disso, ao longo do desenvolvimento deste trabalho foram utilizados diversos conhecimentos abordados nas seguintes disciplinas de Engenharia Eletrônica: Fundamentos de Eletrônica, Análise de Circuitos Elétricos, Técnicas de Programação em Engenharia Eletrônica,

Microprocessadores e Microcontroladores, Projetos de Sistemas Microprocessados, Processamento Digital de Sinais, Sensores e Instrumentação Eletrônica.

Este trabalho está dividido em quatro capítulos, quais sejam: referencial teórico, materiais e métodos, implementação do projeto e análise dos resultados.

No primeiro capítulo, referencial teórico, são apresentados os conceitos de vacinas e imunobiológicos, os cuidados necessários para a conservação de sua eficácia, a cadeia de transporte sistematizada e regulamentada pelo PNI que faz o produto chegar até o usuário final nos postos de vacinação, e também dos módulos GSM/GPRS, GPS, leitor de SD Card e sensor de temperatura que foram utilizados em conjunto com o microcontrolador para fazer a coleta de dados, registro e monitoramento por telemetria do protótipo de dispositivo.

No segundo capítulo, é apresentada a metodologia e os materiais utilizados para o desenvolvimento do protótipo. A metodologia é descrita de forma sequencial desde a concepção do dispositivo até os testes finais, com ilustração por meio de um diagrama de blocos.

No terceiro capítulo, é descrita a implementação, de forma detalhada, em cada uma das áreas que envolvem a construção do protótipo, e usando a metodologia descrita e materiais listados no capítulo anterior.

No quarto capítulo, é realizada a demonstração e análise dos resultados obtidos através da comparação entre os dados coletados e registrados, no cartão de memória SD e nas mensagens de SMS enviadas. Os dados de temperatura são comparados com os registros anotados de um termômetro digital, e os dados de geolocalização são pesquisados no software Google Earth Pro, para verificação do trajeto percorrido.

Por fim, na conclusão são feitas considerações finais sobre a pesquisa desenvolvida, trazendo o objetivo para uma comparação com os resultados obtidos e fazendo sugestões para trabalhos futuros.

1 REFERENCIAL TEÓRICO

1.1 VACINAS

Vacinas são medicamentos imunobiológicos que contêm uma ou mais substâncias antigênicas que são capazes de induzir imunidade específica ativa, no organismo no qual foram inoculadas, com objetivo de combater, reduzir a severidade ou proteger contra doenças causadas pelo agente que originou o antígeno (BRASIL, 2010).

Os imunobiológicos são produtos termolábeis e fotossensíveis. Assim, seu transporte, armazenamento e aplicação devem ser feitos de forma controlada e cuidadosa para manter sua capacidade de resposta (BRASIL, 2017).

A primeira vacina de que se tem notícia foi criada por Edward Jenner no século XVIII, ao testar a hipótese de que pessoas que já haviam contraído varíola bovina se tornavam imunes à varíola, depois de ouvir histórias de trabalhadores rurais. Ele coletou o pus de um ferimento de um trabalhador rural com varíola bovina, inoculou-o em uma criança, como mostrado na figura 1, que contraiu a doença e se curou e depois inoculou varíola na mesma, que nada sofreu (INSTITUTO BUTANTAN, 2021).

Figura 1 – Experimento da primeira vacina aplicada em criança



Fonte: (INSTITUTO BUTANTAN, 2021)

1.2 DISTRIBUIÇÃO DE VACINAS

O PNI é o responsável por toda elaboração da política de vacinação do país, desde a aquisição até a definição do público que receberá os imunizantes. Ele também é responsável pela definição de uma diretriz que acompanha os brasileiros desde o nascimento, o Calendário Nacional de Vacinação, que contém todas as vacinas de rotina e os períodos em que devem ser tomadas (UNIVERSIDADE ABERTA DO SUS, 2022).

Um dos objetivos do PNI é promover a garantia da qualidade dos imunobiológicos adquiridos e ofertados à população. Para isso, ela utiliza a Rede de Frio, uma estrutura técnico-administrativa que faz a manutenção da Cadeia de Frio, que é o processo logístico de conservação dos imunobiológicos, desde o laboratório até a população, ele inclui todas as etapas de armazenamento e movimentação, fazendo-as de forma eficiente e segura, visando a preservação das características originais do produto (BRASIL, 2017).

A estrutura organizacional da Rede de Frio está inserida nas 3 esferas da gestão pública e é distribuída em instâncias, cada uma com fluxos de distribuição e armazenamento, são elas: Instância Nacional, Instância Estadual, Instância Regional, Instância Municipal, Instância Local. A figura 2 ilustra o fluxo.

A Coordenação-Geral do Programa Nacional de Imunizações (CGPNI) representa a Instância Nacional, ela fica responsável pela aquisição, controle de qualidade, distribuição para as instâncias estaduais, interlocução com as instâncias, normatizações e definições gerais. A Instância Nacional tem uma área física para estrutura gestora do CGPNI e também conta com um complexo logístico com ambientes e câmaras com temperatura controlada, a Central Nacional de Armazenamento e Distribuição de Insumos (Cenadi). Os insumos e imunobiológicos são primeiramente destinados ao Cenadi para passar por inspeção no Instituto Nacional de Controle de Qualidade em Saúde (INCQS), e depois seguem para as instâncias estaduais (BRASIL, 2017).

A Instância Estadual é formada por 27 centrais estaduais de armazenamento e distribuição de imunobiológicos, geralmente, localizadas nas capitais das unidades federadas do Brasil. Ela faz um planejamento da necessidade de imunobiológicos, de acordo com o Calendário de Vacinação Nacional e com a situação epidemiológica, e compartilha com a Instância Nacional. Este planejamento ainda leva em conta a capacidade de armazenamento da Central Estadual de Rede de Frio (Cerf) e a distribuição na logística da cadeia de frio às centrais vinculadas (BRASIL, 2017).

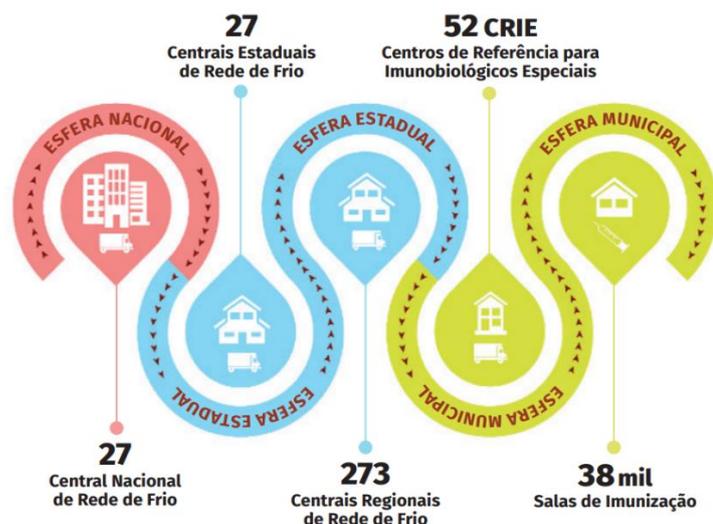
A Instância Regional surge nos estados que assim acharem melhor para sua organização, exercendo atividades compatíveis com as centrais estaduais, e incorporam as Centrais Regionais de Rede de Frio (CRRFs) que ocupam posições estratégicas para distribuição nas unidades federadas e são subordinadas às Secretarias Estaduais de Saúde. Também dispõem de áreas para apoio logístico de imunobiológicos e insumos e para estruturas para atividades de apoio técnico e administrativo (BRASIL, 2017).

A Instância Municipal é representada pela Central Municipal de Rede de Frio (CMRF), subordinada à Secretaria Municipal de Saúde, tem como objetivo o armazenamento e planejamento integrado de insumos e imunobiológicos recebidos das centrais estaduais e regionais, que serão usados nas salas de imunização. A depender do quantitativo de imunobiológicos, podem necessitar de câmaras frias. Suas estruturas também devem prever áreas para atividades de todo apoio logístico, administrativo e técnico (BRASIL, 2017).

A Instância Local é a que vai finalizar a cadeia de frio e concretizar a Política Nacional de Imunizações, entrando em contato com o usuário final, fazendo administração segura e dando assistência básica (BRASIL, 2017).

A instância final da Rede de Frio é representada pelas Salas de Imunização (SI), sua responsabilidade exclusiva é a de procedimentos de vacinas, campanhas, bloqueios e intensificações. Para a realização, é fundamental que os equipamentos de refrigeração que guardam os imunobiológicos sejam apropriados e estejam em condições ideais. O dimensionamento dos equipamentos deve prever idealmente 30 dias de prazo, levando em conta aspectos de atendimento à população e situações de emergência (BRASIL, 2017).

Figura 2 - Organização da Rede de Frio Nacional



Fonte: (BRASIL, 2022, p.58)

Por último, os Centros de Referência para Imunobiológicos Especiais (Crie), são voltados para atendimento à população, em especial aos portadores de imunodeficiência ou condição de morbidade ou risco, assim como acompanhamento e esclarecimentos de eventos adversos após a vacinação. Como ele tem caráter de atendimento à população, estão previstas instalações mínimas como: recepção, consultório, sala de imunização e sanitário (BRASIL, 2017).

1.3 PRECAUÇÕES DURANTE O TRANSPORTE DE VACINAS

O planejamento logístico do transporte de vacinas dentro da Rede de Frio deve fazer uma análise cuidadosa e multifatorial levando em conta: tipo dos imunobiológicos, demanda, capacidade de armazenamento, período de reposição, perdas técnicas e físicas, em cada instância da rede, pois a temperatura e o tempo de armazenamento são fatores cruciais na garantia da qualidade do produto.

Figura 3 – Período e temperatura de armazenamento dos imunobiológicos nas instâncias da Rede de Frio

		Cenadi Centrais estaduais 6 a 12 meses	Centrais regionais, distrital ou municipais – Crie 3 a 6 meses	Local Sala de vacinação 1 mês	
POSITIVO	+8°C	Vacinas diluentes* ¹ FA* ²	Vacinas diluentes* ¹ FA* ² VOP* ³	Vacinas diluentes* ¹ FA VOP	
	+2°C	Soros Imunoglobulinas	Soros Imunoglobulinas	Soros Imunoglobulinas	
NEGATIVO	-15°C	FA VOP	FA VOP		
	-25°C				
		 Vacinas diluentes Vacinas: BCG, dT, DT, HB, DTP, VRH, SCR, VRC, FTp, HiB, VIP, DTPa, HA, Meningo conj C, penta, Pncc 23V, Pncc10V, raiva embrião galinha, raiva vero, Varc, VRC, cólera, tetra viral, FLU3V.	 Soros Imunoglobulinas Soros: SARC, SBOTR, SBOCR, SBOLAQ, SBOTULBI, SCROT, SAD, SELAP, SESCOR, SLONO, SLOXO, SARH, SAT. Imunoglobulinas: IGHB, IGRH, IGTH, IGZV.	 FA VOP  FA VOP	FA: Vacina Febre amarela VOP: Vacina poliomielite oral.

Fonte: (BRASIL, 2017, p.25)

A figura 3 apresenta uma referência do volume e tempo de armazenamento em cada instância. Importante ressaltar que o tempo descrito é somente uma referência, desse modo, deve ser levado em conta as especificidades de cada imunobiológico.

O laboratório produtor estipula prazo de validade e faixas de temperatura às quais devem ser mantidas as vacinas para conservar suas propriedades físicas, químicas e biológicas. A alteração de temperatura pode comprometer não só a potência imunogênica da vacina como outras características verificadas pelo laboratório produtor quando a produziu como: prazo de validade, faixa de temperatura de conservação, umidade etc. (BRASIL, 2017).

Por isto, é dever do PNI alertar às instâncias quanto aos cuidados recomendados pela OMS, seus estudos demonstram que os erros mais comuns são os de manuseio durante o armazenamento e distribuição como: acondicionamento sob temperaturas inadequadas, sejam acima ou abaixo do recomendado, aparelhos de refrigeração sem controle de temperatura, armazenamento junto com outros materiais, alimentos e bebidas. Qualquer exposição fora da temperatura preconizada, ocorre perda de potência imunogênica, portanto toda a exposição deve ser reportada pelas instâncias estaduais ao PNI, que fará uma avaliação, através do INCQS, diante dos registros enviados. Durante este período de análise, as amostras expostas devem ser separadas, identificadas e conservadas em temperaturas entre +2° C a +8° C (BRASIL, 2017).

1.4 TRANSPORTE DE IMUNOBIOLOGICOS DENTRO DA CADEIA DE FRIO

O PNI oferta à população produtos de origem nacional e/ou internacional e o transporte ocorre por vias aéreas, terrestres e aquáticas levando em consideração: volume para transporte, facilidade de acesso da via em relação ao local e origem/destino. Além do monitoramento da temperatura, checar a via que oferece menos riscos de impactos mecânicos é outro fator importante, pois estes choques podem causar microfissuras que causam vazamentos ou expõem o produto à contaminação de microrganismos (BRASIL, 2017).

Figura 4 – Transporte do laboratório produtor X PNI/Cenadi



Fonte: (BRASIL, 2017, p.28)

Os produtos de origem internacional vêm, obrigatoriamente, por vias aéreas até a alfândega, de onde são despachados por vias terrestres até o Cenadi. Já os de origem nacional, somente por vias terrestres, como ilustrado na figura 4. Ao chegarem, os produtos são conservados de acordo com suas especificidades e depois avaliados e registrada sua entrega.

Figura 5 – Transporte do PNI/Cenadi X Instância Estadual



Fonte: (BRASIL, 2017, p.29)

Das instâncias estaduais às instâncias regionais e municipais, o transporte é feito prioritariamente por via terrestre, mas pode também ser utilizada via aquática ou aérea (figura 5).

Em todo trajeto, a temperatura é monitorada e deve ser controlada, transportando-as em veículos refrigerados, e na ausência destes veículos climatizados e com apoio de bobinas reutilizáveis congeladas, longe de fontes de calor e luminosidade. Também acompanham documentos de transferência para facilitar a conferência na hora da chegada e promover a rastreabilidade destas cargas. Nas instâncias, os imunobiológicos são armazenados agrupando por temperatura ideal de conservação.

1.5 PERDAS

Foi observado nos relatórios de perda de vacinas que os principais motivos para ocorrência de tais perdas são: por interrupções de energia em horários fora do expediente das unidades de saúde, como finais de semana ou à noite, o que aumentava o tempo de exposição das vacinas em temperaturas acima da faixa pré-estabelecida pelo PNI, e por falhas nos equipamentos de refrigeração, levando ao acondicionamento inapropriado em temperaturas acima ou abaixo da faixa ideal para a conservação das vacinas (SANTOS *et al.*, 2016).

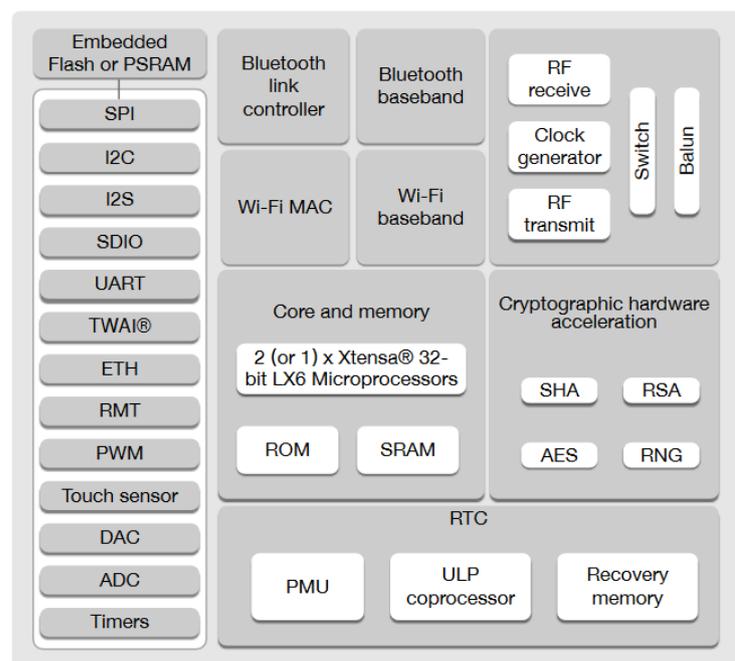
Ainda em seu estudo, Santos *et al.* (2016) aponta que o tempo médio de exposição a temperaturas não-ideias era de 40,25h, sendo que o PNI recomenda que caso o imunobiológico tenha ficado por mais de 24h sob temperaturas acima de 25 °C, o produto seja descartado.

1.6 MICROCONTROLADOR ESP32

Os microcontroladores são circuitos integrados cuja principal característica é possuir um hardware com elementos essenciais para implementação de uma aplicação eletrônica, contendo um microprocessador, que é um circuito integrado onde são processados os dados e executadas as instruções, e um conjunto de periféricos com os quais o microprocessador irá se comunicar e comandar, como por exemplo as interfaces de entradas e saídas de dados, temporizadores e conversores analógico/digitais (OLIVEIRA JÚNIOR; DUARTE, 2010).

Os microcontroladores da família ESP32, foram projetados para aplicações em dispositivos móveis, eletrônicos vestíveis e aplicações em Internet das Coisas. Possuem integração com várias interfaces de entrada e saída programáveis como I2C, SPI, I2S, UART. (ESPRESSIF, 2022). Sua programação pode ser feita através do Arduino IDE, o que o torna ainda mais atrativo para projetos eletrônicos por sua simplicidade e variedade de aplicações, seu diagrama de blocos é mostrado na figura 6.

Figura 6 – Diagrama blocos funcionais do ESP32



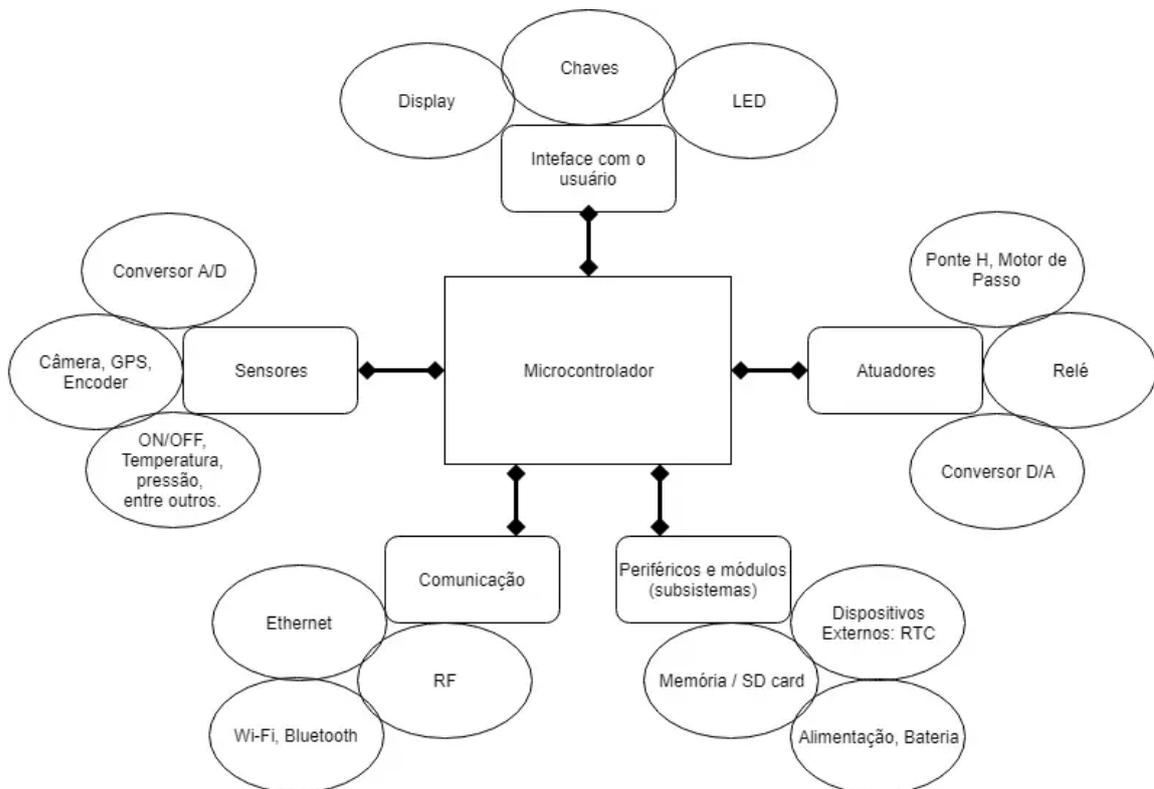
Fonte: (ESPRESSIF, 2022, p. 12).

1.7 SISTEMAS EMBARCADOS

Os sistemas embarcados são sistemas computacionais projetados para executar uma tarefa específica dentro um sistema maior usando conjunto de hardware e firmware. Pelo intuito de serem projetados para uma tarefa, geralmente são compactos e possuem recursos específicos para executá-la de forma confiável e eficiente em termos de energia (GARCIA, 2018).

Também são caracterizados como sistemas reativos pois a aplicação reage aos estímulos do ambiente. Então o sistema embarcado é formado pelas entradas, área de processamento e as saídas. As entradas podem ser representadas por sensores e chaves, a área de processamento é representada pelos microcontroladores, e as saídas representadas por atuadores, displays e outros. Estão presentes em praticamente todos os dispositivos eletroeletrônicos (GARCIA, 2018). A figura 7 mostra uma visão geral de um sistema embarcado com microcontrolador.

Figura 7 - Visão geral de um sistema embarcado com microcontrolador



Fonte (GARCIA, 2018).

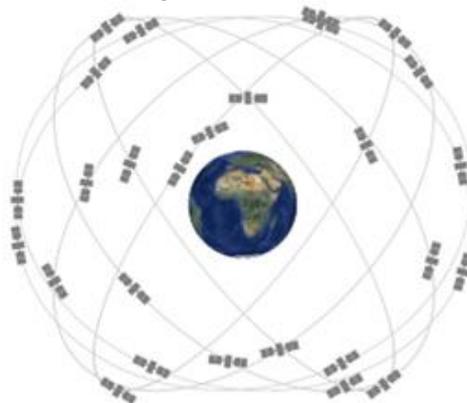
1.8 GPS

O GPS é um sistema de navegação por satélite, desenvolvido pelo Departamento de Defesa dos Estados Unidos, que fornece informações de tempo e posicionamento continuamente e em qualquer lugar do mundo. Para garantir cobertura global, desde julho de 1993, existem 24 satélites orbitando a Terra (EL-RABBANY, 2002).

Os satélites enviam constantemente um sinal de rádio para a Terra, abrangendo uma área circular, onde qualquer dispositivo com um receptor ligado à uma antena de GPS consegue receber este sinal. Com o sinal de 4 satélites no mesmo receptor, é possível obter a localização. O sistema é dividido em 3 segmentos: segmento espacial, segmento de controle e segmento dos usuários.

O segmento espacial é representado pela constelação de satélites que orbitam a Terra a aproximadamente 20.200 km na MEO (*Medium Earth Orbits* – Órbitas Terrestre Média). Na constelação principal, existem 4 satélites posicionados em 6 planos orbitais do planeta, totalizando uma constelação de 24 satélites, ilustrado na figura 8, eles posicionados de tal forma que os usuários consigam o sinal de pelo menos 4 satélites em qualquer ponto da Terra. Em 2022, o número de satélites operacionais era de 31, que podem até aprimorar a performance do sistema, porém não fazem parte da constelação principal (NATIONAL COORDINATION OFFICE, 2022).

Figura 8 – Constelação de satélites ao redor da Terra

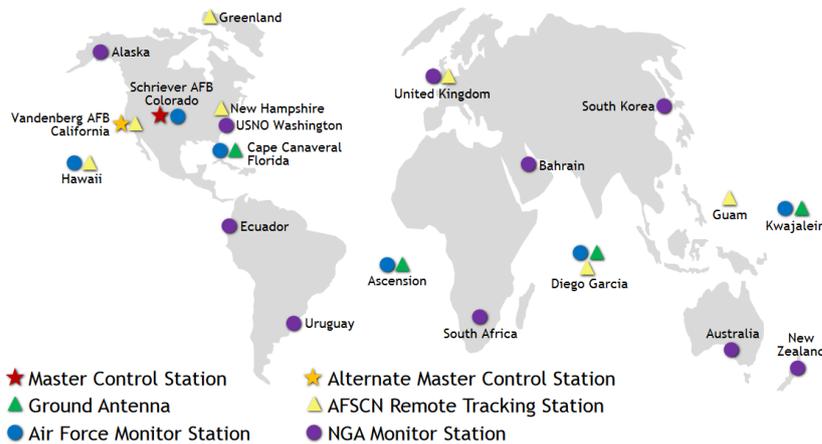


Fonte: (NATIONAL COORDINATION OFFICE, 2022)

O segmento de controle é uma rede global de instalações terrestres que localizam os satélites de GPS, monitoram e analisam suas transmissões e enviam atualizações para a constelação. O segmento de controle atualmente inclui uma estação de controle mestre, uma estação de controle mestre alternativa, 11 antenas de comando e controle e 16 locais de

monitoramento, suas localizações podem ser vistas na figura abaixo (NATIONAL COORDINATION OFFICE, 2021).

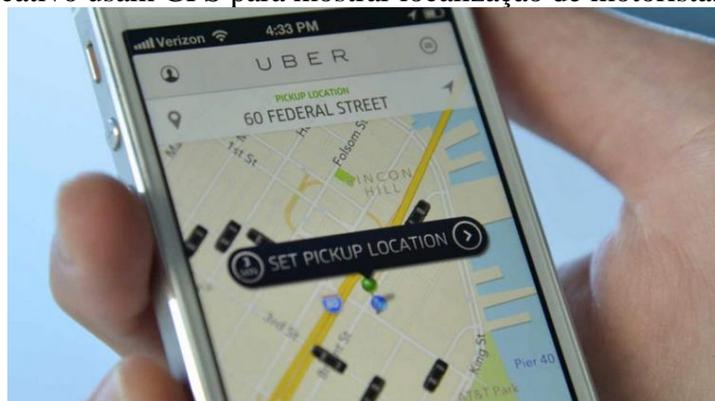
Figura 9 – Mapa com localizações de instalações terrestres do segmento de controle do GPS
GPS Control Segment



Fonte: (NATIONAL COORDINATION OFFICE, 2021)

O segmento dos usuários inclui civis e militares e é representado pelos receptores e suas aplicações em diversas atividades (EL-RABBANY, 2002). O receptor recebe o sinal dos satélites e calcula as coordenadas do usuário, estes dados de localização constantes têm muitas aplicações no cotidiano como nas áreas de agricultura, monitoramento de objetos e pessoas, automação de veículos e entre outras, ilustrado na figura 10.

Figura 10 - Aplicativo usam GPS para mostrar localização de motoristas perto do usuário



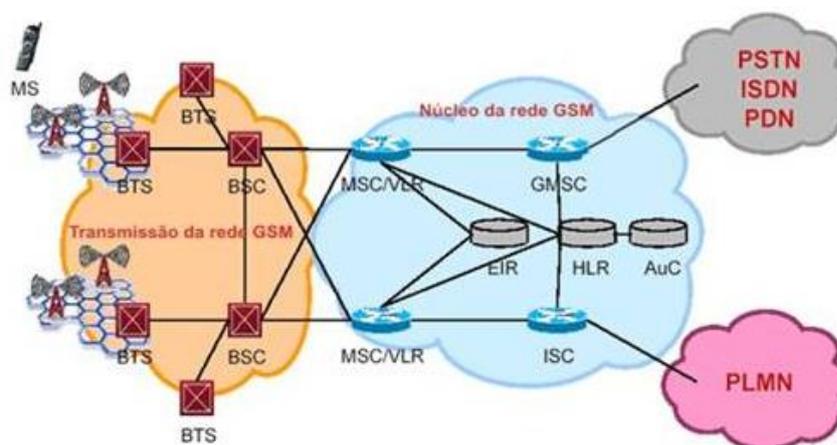
Fonte: (LINDOMAR, 2020)

Em seu trabalho, Ferreira (2010) descreve como utilizou a tecnologia GPS para fazer um sistema de alarme integrando GPS e celular. Quando o carro era estacionado, o sistema começava a monitorar e se esse movimentava-se em 15 metros da posição inicial, o sistema faz uma ligação para o usuário, para avisá-lo do ocorrido e possa tomar providências.

1.9 GSM/GPRS

O GSM faz parte da segunda geração (2G) de tecnologia para telefonia móvel e utilizava um padrão digital para fazer a comunicação, o que popularizou ainda mais o acesso já que suportava mais usuários e tinha mais estabilidade. Foi desenvolvido na Europa, pode utilizar várias frequências sendo as mais comuns as de 900MHz, 1800MHz e 1900MHz, variando de acordo com a operadora e o país (ALECRIM, 2013).

Figura 11 - Arquitetura do sistema GSM



Fonte: (PIMENTA, 2006b)

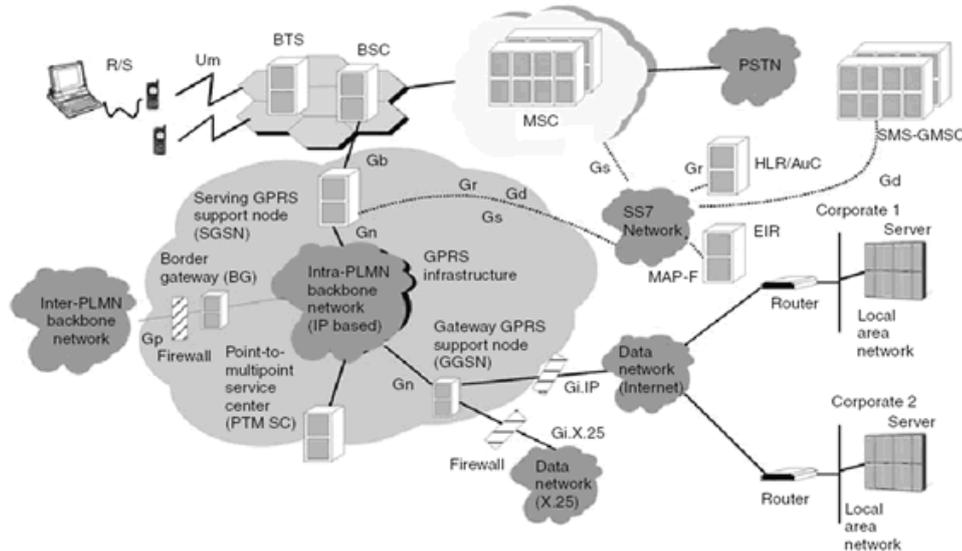
A telefonia móvel foi inicialmente projetada para ser um serviço de comunicação por voz, porém, com o sucesso do GSM e a flexibilidade da rede (figura 11), em 1995 foi introduzido o serviço de envio e recepção de mensagens de até 160 caracteres, o SMS (EBERSPÄCHER, 2009).

Uma das características do GSM é o uso do cartão *Subscriber Identity Module* (SIM) para guardar informações da linha telefônica do usuário como número e operadora, assim mesmo que o telefone se perca, o usuário pode colocar o cartão SIM em outro telefone e voltar a usar normalmente (ALECRIM, 2013).

O GPRS foi uma evolução muito importante para o GSM, pois utilizava técnicas para comutar os pacotes para atender assinantes com altas taxas de bits para transmitir. Como a comunicação se dá por altas taxas de bits para transmitir seguida de intervalos, o pacote é transmitido usando vários times slots do canal e depois este é liberado, assim ganha-se velocidade e pode-se atender mais pessoas usando o mesmo canal (figura 12). Com isto, e sua compatibilidade com protocolo IP, foi possível introduzir a comunicação com a internet.

Utilizando esta técnica, as operadoras podem cobrar por pacote de dados transmitido (PIMENTA, 2006a).

Figura 12 - Visão geral do sistema GPRS



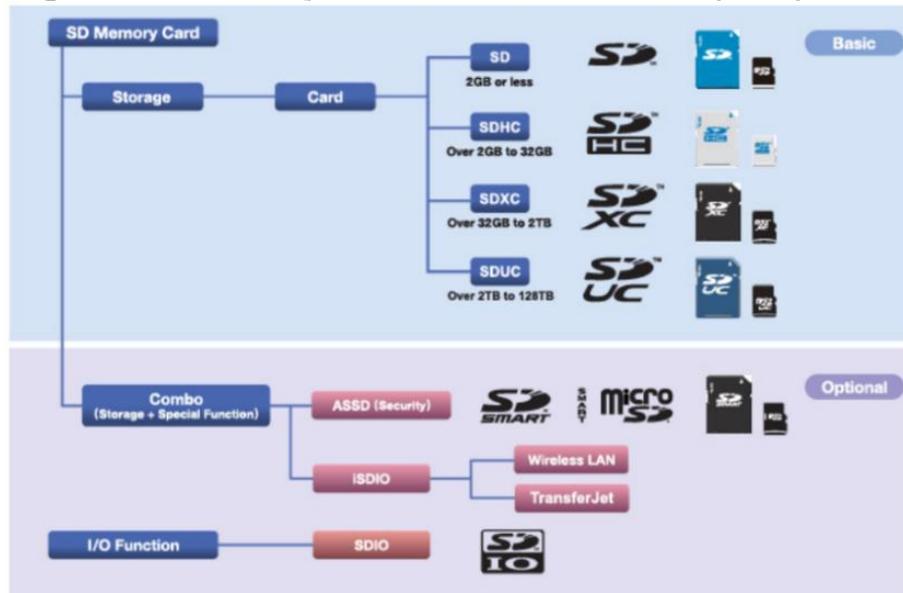
Fonte: (PIMENTA, 2006a)

Esta tecnologia é usada não só para comunicação entre pessoas, mas para monitoramento remoto de energia. Segundo Finkler e Fischer (2016), existe um protótipo de medição de energia com um chip acoplado onde o cliente faz a recarga no chip e a concessionária oferece informações detalhadas de consumo de energia. O protótipo usa um GSM/GPRS Shield SIM 900 para enviar as mensagens pela rede.

1.10 SD CARD

O cartão de memória SD foi criado para atender requisitos de gravação de dados de forma segura e eficiente, no contexto dos surgimentos de tecnologias de áudio e vídeo. Possui 4 variações quanto à sua capacidade de armazenamento o *Standard Capacity SD Memory Card* (SDSC), que armazena até 2 GB, o *High Capacity SD Memory Card* (SDHC) que armazena de 2 GB até 32 GB, o *Extended Capacity SD Memory Card* (SDXC), que armazena de 32GB até 2TB e o *Secure Digital Ultra Capacity* (SDUC), que armazena de 2TB até 128TB, como mostrado na figura 13. Ainda possui uma categoria com ferramentas de segurança de conteúdo, o que o habilita para armazenar dados protegidos (SD ASSOCIATION, 2000).

Figura 13 – Classificação dos cartões de memória SD por capacidade

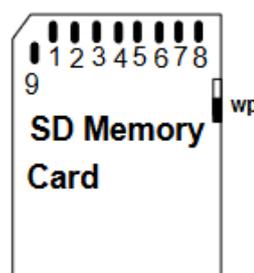


Fonte: (SD ASSOCIATION, 2000)

A comunicação básica do cartão de memória SD utiliza uma interface de 9 pinos feita para operar numa frequência máxima de 208MHz e faixa de baixa tensão. Esta configuração de pinos dá suporte para 2 protocolos de comunicação: protocolo SD e protocolo SPI. A figura 14 mostra o formato geral de um cartão de tamanho padrão e os contatos da interface. A figura 15 mostra a função de cada pino para cada protocolo. O protocolo SPI é uma opção secundária para comunicação com o cartão de memória e é utilizada em projetos com microcontroladores (SD CARD ASSOCIATION, 2022).

Um dos exemplos de utilização do armazenamento seguro e robusto do SD Card é feito por Vargas 2016 no desenvolvimento de um *data logger* aplicado ao monitoramento de encostas. Em sua dissertação, ele utilizou um circuito com microcontrolador para criar um arquivo no formato *.csv, e salvar dados de umidade de solo, pluviômetros em um SD Card de 1Gb.

Figura 14 – Vista superior de um cartão de memória SD de tamanho padrão e sua interface.



Fonte: (SD CARD ASSOCIATION, 2022, p. 12)

Figura 15 – Funções de cada pino para protocolos SD e SPI.

Pin #	SD Mode			SPI Mode		
	Name	Type ¹	Description	Name	Type ¹	Description
1	CD/DAT3 ²	I/O/PP ³	Card Detect/ Data Line [Bit 3]	CS	I ³	Chip Select (neg true)
2	CMD	I/O/PP	Command/Response	DI	I	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1 ⁴	I/O/PP	Data Line [Bit 1]	RSV		
9	DAT2 ⁵	I/O/PP	Data Line [Bit 2]	RSV		

Fonte: (SD CARD ASSOCIATION, 2022, p. 12)

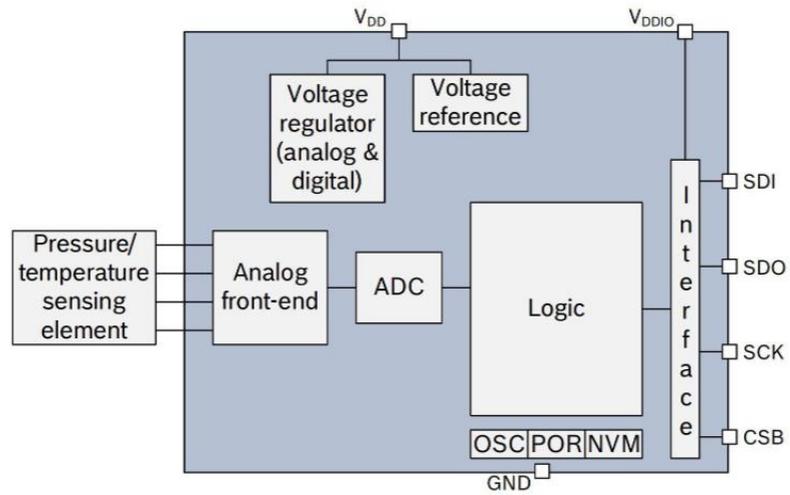
1.11 SENSOR DE TEMPERATURA BMP280

Sensores são elementos que fazem o contato da aplicação eletrônica com o mundo externo. Dentro do mundo eletrônico, um sensor já vem acompanhado de um circuito transdutor, que faz a transformação de uma grandeza física, como luz ou a temperatura, em uma grandeza elétrica que pode ser trabalhada na aplicação (BRAGA, 2021).

Os sensores de temperatura podem ser divididos em duas classes: absolutos, quando medem a temperatura com referência a um ponto fixo na escala de temperatura absoluta, como o 0 °C, e relativos, quando medem a diferença de temperatura entre dois objetos, ou o gradiente térmico, por exemplo o termopar. Para medição de temperatura, é necessário que o sensor seja acoplado termicamente ao objeto, este acoplamento pode ser de forma física ou remota, mas é necessário para que o sensor produza uma resposta elétrica mensurável ao estímulo (FRADEN, 2016).

O BMP280 é um sensor digital de pressão e temperatura que oferece interface de comunicação para protocolos I²C e SPI e já vem equipado com filtros para evitar flutuações nas leituras e conversor analógico para digital, como mostra a figura 16. Apresenta uma faixa de leitura de temperatura de -40 °C a +85 °C com precisão de ± 1 °C (BOSCH SENSORTEC, 2018).

Figura 16 – Diagrama de blocos do BMP280.



Fonte: (BOSCH SENSORTEC, 2018, p. 11)

2 MATERIAIS E MÉTODOS

A seguir são abordados os aspectos metodológicos da pesquisa realizada, escrevendo-se os procedimentos necessários para desenvolver um dispositivo de monitoramento e registro de dados de temperatura e geolocalização. Esse estudo teve por finalidade realizar uma pesquisa de natureza aplicada. Para alcançar os objetivos propostos e melhor apreciação deste trabalho, foi utilizada uma abordagem quali-quantitativa. Com intuito de conhecer a problemática sobre a área de estudo foi realizada uma pesquisa exploratória e bibliográfica (GIL, 2019). A seguir são descritos os materiais utilizados ao longo da pesquisa, bem como os procedimentos metodológicos.

2.1 AMBIENTE DE DESENVOLVIMENTO

O hardware utilizado nesta pesquisa tem a seguinte configuração: Processador AMD Ryzen 5 2600 Six-Core Processor 3.85 GHz; RAM 16 GB; Sistema operacional de 64 bits, Windows 10 Pro.

2.2 MATERIAIS

Para a realização de testes com o protótipo e validação de seu funcionamento, foram utilizados os seguintes itens:

- 1 Multímetro Digital ET-2042F com termopar tipo K;
- 1 caixa de isopor de 12L;
- 1 bolsa térmica gel média;
- Software Google Earth Pro;
- 1 Chip da Claro.

Para a construção do protótipo de dispositivo e verificação de funcionalidades, foram necessários os seguintes itens:

- 1 Módulo Serial I2C para Display LCD;
- 1 Display LCD 16x2;
- 1 Fonte ajustável Step Down Regulador de Tensão LM2596;
- 1 Módulo GSM/GPRS SIM800L;
- 1 Placa ESP-WROOM-32;

- 1 Bateria 12V 4800mAh ;
- 1 Módulo cartão Micro SD;
- 1 Módulo GPS GY-NEO6M V2;
- 1 Sensor de pressão atmosférica, umidade e temperatura BMP280;
- Cabos para conexão;
- Arduino IDE 1.8;
- Altium Designer 21.

2.3 MÉTODOS

No diagrama de blocos da figura 17 são apresentadas as etapas da metodologia utilizada no desenvolvimento deste trabalho, quais sejam: levantamento dos requisitos, definição da arquitetura de hardware, desenho do esquema elétrico, desenho do layout, montagem da placa, definição da arquitetura de firmware, desenvolvimento de firmwares de teste dos periféricos, integração do firmware, integração entre hardware e firmware, definição dos cenários de testes, modelagem dos itens mecânicos, prototipagem dos itens mecânicos, testes de validação, avaliação de desempenho.

Figura 17 - Metodologia utilizada para desenvolvimento do projeto



Fonte: Autoria própria.

2.3.1 Levantamento de requisitos

Para cumprir as demandas apresentadas na hipótese o dispositivo deve:

- Enviar e receber mensagens de SMS;
- Medir temperatura dentro de um recipiente de isopor;
- Receber dados de localização do sistema GPS;
- Registrar dados de temperatura e geolocalização no cartão SD;
- Enviar alertas de temperatura para o número programado quando a temperatura sair da faixa de 2 °C a 8 °C;

2.3.2 Definição da arquitetura de hardware

Para escolher os componentes do dispositivo, foram levados em consideração os requisitos, a disponibilidade do mercado e a utilização pela comunidade eletrônica. Foram escolhidos o módulo GSM/GPRS SIM800L e o módulo GPS GY-NEO6M V2, ambos com interface serial. O sensor de temperatura BMP280 com interface I2C, e o módulo leitor de Micro SD Card por estender os pinos de comunicação SPI e de alimentação do cartão SD. O módulo regulador de tensão LM2596 foi escolhido para baixar a tensão da bateria para o nível do sistema, e fornecer mais corrente. O display LCD 16x2 fornece ao usuário a resposta instantânea sobre o status geral do dispositivo, o módulo I2C reduz o número de conexões e controla-o via interface I2C. O microcontrolador ESP32 oferece suporte à todas as interfaces usadas pelos módulos sejam elas: serial, I2C e SPI, e na quantidade necessária. Este microcontrolador também dá suporte ao sistema operacional com processamento paralelo, importante para que a recepção de mensagens de SMS não interrompa coleta de dados. A bateria de 12V e 4800mAh garante ao dispositivo um bom tempo de vida e boa quantidade de carga

2.3.3 Desenho do esquema elétrico

O desenho do esquemático foi feito no software Altium Designer 21, onde foram feitas conexões diretas entre os pinos do microcontrolador e os pinos do periférico levando em conta a interface de comunicação utilizada. A saída de 12V foi conectada ao ponto de entrada do regulador de tensão, o qual foi regulado para sair 5V. Da saída do regulador foi distribuído para o sistema.

2.3.4 Desenho do layout

Ainda no Altium Designer, depois de fazer as conexões, o próximo passo foi fazer o software gerar o layout 3D da placa para melhor posicionar os componentes conforme o tamanho e formato na vida real. Após o reposicionamento o arquivo Gerber foi gerado para impressão na prototipadora.

2.3.5 Montagem da placa

A montagem da placa foi feita no Laboratório de Sistemas Embarcados (LSE) localizado no Hub Tecnologia e Inovação. O arquivo Gerber gerado anteriormente foi exportado para a confecção da placa de circuito impresso no equipamento ProtoMat M60. Depois da impressão, foi feita uma limpeza da placa, inspeção visual e teste de continuidade com o multímetro digital ET-2042F. Depois do teste foi utilizada a estação de solda HK-936A e um rolo de solda estanho para soldar os componentes nos locais planejados no esquema elétrico.

2.3.6 Definição da arquitetura de firmware

O firmware foi desenhado para inicialmente verificar as conexões com os módulos e criar a estrutura e o arquivo de registro de dados. Depois da inicialização, ele repetidamente faz a leitura do sensor de temperatura e do módulo GPS, registra estes dados no arquivo criado no cartão SD, enquanto tenta receber mensagens de SMS pelo módulo GSM. Quando detectar uma temperatura fora da faixa ideal ele enviará uma mensagem de alerta com os dados do GPS e da temperatura a cada 5 minutos. também enviará uma mensagem de SMS quando receber um SMS com a palavra “*request*”.

2.3.7 Desenvolvimento de firmwares de teste dos periféricos

Os testes de periféricos foram desenvolvidos no software Arduino IDE criando scripts separados para cada um deles, onde foram testadas funcionalidades básicas. O módulo SD card deveria criar e escrever arquivos e pastas, o módulo GSM deveria enviar e receber mensagens de SMS para o número programado e o módulo GPS deveria fornecer a geolocalização.

2.3.8 Integração do firmware

Ainda usando o Arduino IDE, os scripts de teste de periférico foram utilizados para criar um script que integrava todas as funções desenvolvidas. Como um SMS pode chegar a qualquer momento, para não afetar a coleta de dados foi embarcado um script que habilitava o FreeRTOS, um sistema operacional que permite o processamento de tarefas em paralelo. Foram criadas 2 tarefas para serem processadas concorrentemente, uma responsável pela coleta de dados e atualização das informações no display LCD e outra cuidando da comunicação com módulo GSM.

2.3.9 Integração entre hardware e firmware

Quando o firmware de integração foi finalizado, ele foi embarcado e testado com todas as funcionalidades desenvolvidas trabalhando em conjunto.

2.3.10 Definição dos cenários de testes

Para testar o pleno funcionamento do dispositivo dois cenários de teste foram criados, neles o dispositivo é anexado à uma caixa de isopor com uma bolsa térmica congelada dentro, o sensor de temperatura foi introduzido através de um orifício na tampa da caixa de isopor, junto do dispositivo onde foi colocado o multímetro digital ET-2042F e o termopar do multímetro. O sensor de temperatura e o termopar foram colocados na mesma posição, dentro da caixa de isopor.

No primeiro cenário, o dispositivo ficou parado coletando os dados. No segundo cenário, o dispositivo foi conduzido em trajeto desde a UEA até o local onde foi programado. Depois de completar a tarefa, os resultados foram coletados do cartão SD e do *smartphone* programado para ser o destinatário dos alertas de temperatura.

2.3.11 Modelagem dos itens mecânicos

Para acomodar todos os componentes do dispositivo, foi modelada uma caixa para impressão 3D no site TinkerCAD. Para definir as medidas, os componentes foram dispostos em formato que formasse um quadrado, as medidas foram tiradas com régua, a espessura foi posta em 2mm e o arquivo “.stl” foi gerado. Este arquivo foi enviado para um serviço de impressão 3D.

2.3.12 Prototipagem dos itens mecânicos

O case foi impresso usando filamento ABS na cor preta, as medidas da caixa impressa seguiram o modelo enviado com precisão. Os componentes foram posicionados conforme a disposição planejada na etapa anterior e foram fixados com fita dupla face. Foram feitos furos na lateral para fixar o display LCD como parafusos, e na base para passar o sensor de temperatura.

2.3.13 Testes de validação

Seguindo os cenários de teste criados, a caixa impressa contendo a placa montada foi anexada à uma caixa de isopor de 12L com 1 bolsa térmica congelada dentro. Foi colocado ao lado da caixa impressa o multímetro digital com seu termopar passando pelo mesmo furo na tampa do sensor de temperatura do dispositivo. Este conjunto foi levado para os locais de teste. O primeiro cenário foi realizado no local onde o dispositivo foi programado durante 57 minutos, o segundo foi realizado durante 34 minutos no trajeto feito de carro da UEA de volta ao local de programação. Os registros foram copiados do cartão SD. Os dados de temperatura foram comparados com as medições do multímetro anotadas durante o teste. Os dados de geolocalização foram pesquisados no software Google Earth Pro e comparados com o trajeto registrado pelo *smartphone* no aplicativo Google Maps para verificar se estavam corretos.

2.3.14 Avaliação de desempenho

As métricas utilizadas na validação e nos testes foram as seguintes: erro absoluto e o erro relativo. O erro absoluto corresponde à diferença algébrica entre o valor obtido e o valor verdadeiro, conforme a equação 1, e o erro relativo corresponde à normalização do erro absoluto com o valor verdadeiro que pode ser expresso em porcentagem, conforme à equação 2 (CHAPRA; RAYMOND, 2011).

$$\text{Erro absoluto} = \text{Valor medido} - \text{Valor verdadeiro} \quad (\text{Equação 1})$$

$$\text{Erro relativo} = \frac{\text{Erro absoluto}}{\text{Valor verdadeiro}} * 100\% \quad (\text{Equação 2})$$

3 IMPLEMENTAÇÃO DO PROJETO

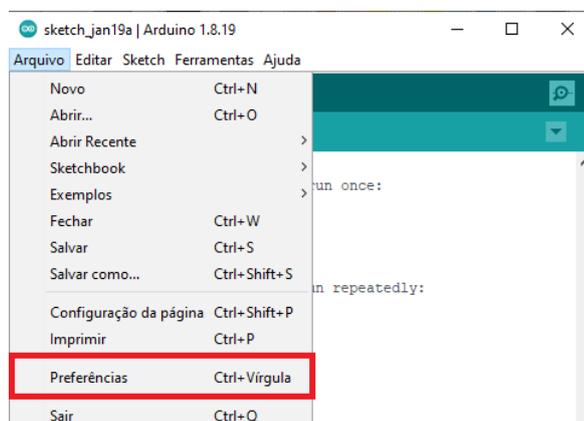
Neste capítulo são apresentados os procedimentos detalhados para o desenvolvimento de um protótipo de dispositivo capaz de monitorar e registrar dados de temperatura e geolocalização e enviá-los via SMS para um dispositivo móvel. Os dados coletados pelo sensor de temperatura e pelo módulo GPS serão processados pelo microcontrolador ESP32, registrados em um cartão de memória SD e enviados por um módulo GSM caso sejam requisitados ou a temperatura atinja valores fora da faixa de 2 °C a 8 °C. Este capítulo apresenta os seguintes tópicos:

- a) Preparação do ambiente na Arduino IDE;
- b) Desenvolvimento do protótipo de hardware;
- c) Desenvolvimento dos scripts de teste de cada periférico;
- d) Desenvolvimento do script geral de integração do protótipo;
- e) Validação final do protótipo do dispositivo.

3.1 PREPARAÇÃO DO AMBIENTE NA ARDUINO IDE

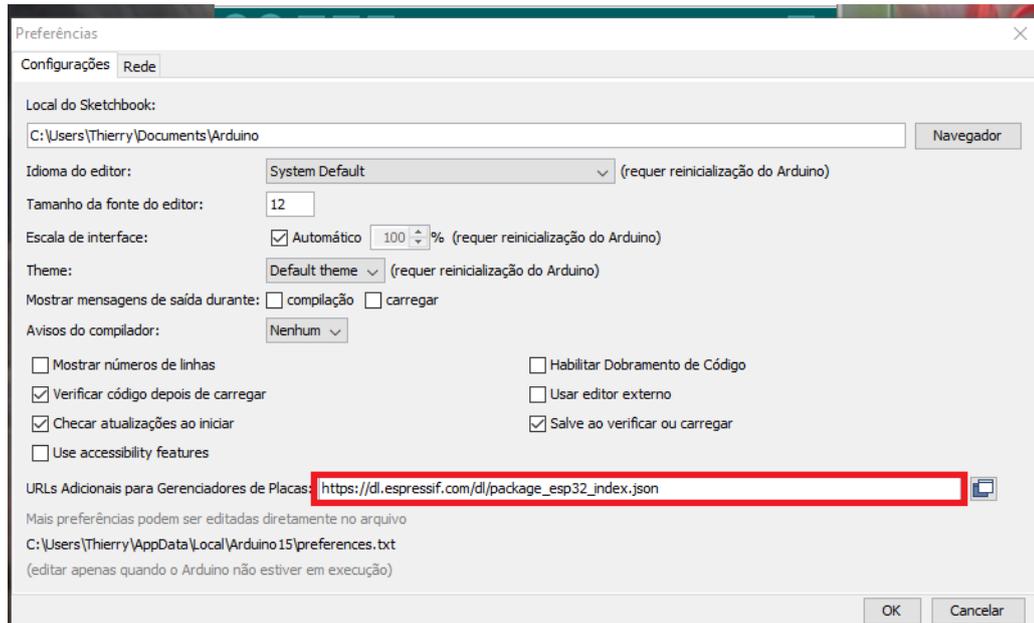
Para o desenvolvimento dos scripts, tanto os de teste como o de integração, foi escolhido o Arduino IDE por ser ambiente de programação de fácil instalação e uso. Porém, o Arduino não apresenta o microcontrolador ESP32 em sua biblioteca de placas padrão para programação, por isso é necessário adicionar o endereço https://dl.espressif.com/dl/package_esp32_index.json no campo URLs Adicionais para Gerenciadores de Placa (figura 19) na aba Arquivo > Preferências (figura 18) do software.

Figura 18 – Seleção de aba Preferências



Fonte: Autoria própria.

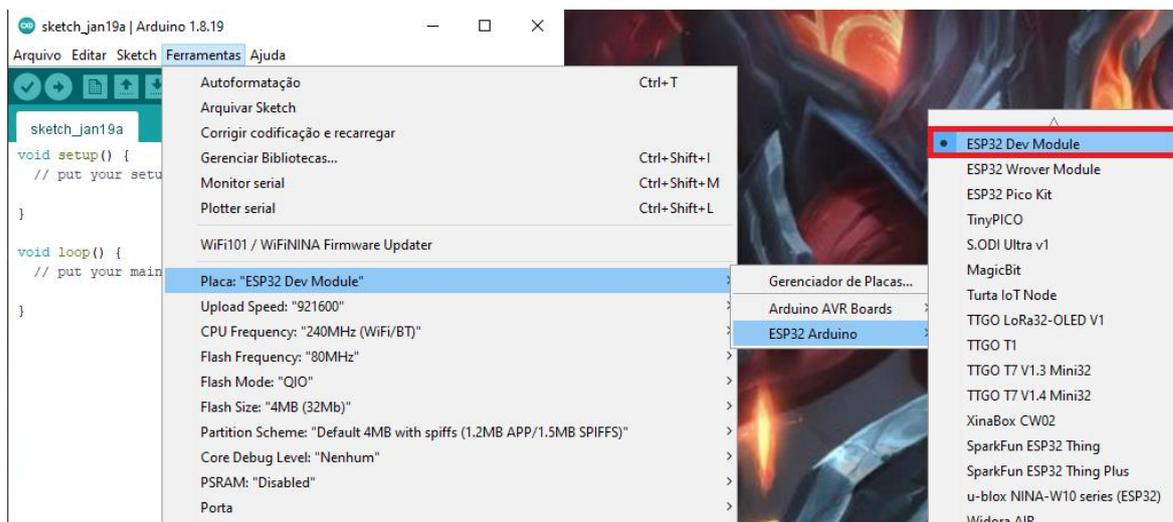
Figura 19 – Adição de link no campo URLs Adicionais



Fonte: Autoria própria.

Depois disto, reiniciou-se o software para que este fizesse o download das placas e assim, a placa fosse selecionada para programação figura 20 e o script pudesse ser embarcado sem problemas.

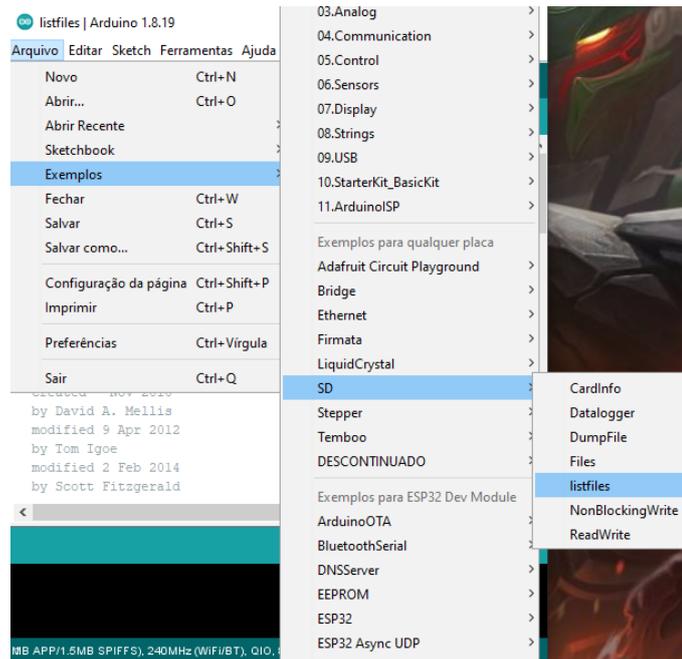
Figura 20 – Seleção de placa ESP32 Dev Module para escrever o script



Fonte: Autoria própria.

Para controlar ou ler alguns periféricos, é preciso utilizar bibliotecas que podem já estar instaladas no software, que é o caso da biblioteca do módulo leitor de SD Card (figura 21), tendo apenas que incluir um comando para importar a biblioteca no script.

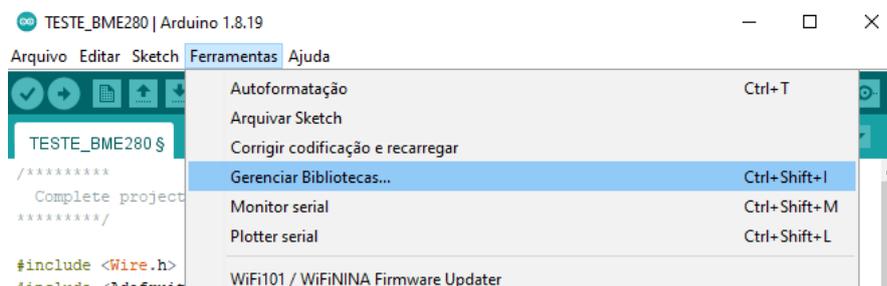
Figura 21 – Seleção de exemplo de uso de SD Card



Fonte: Autoria própria.

Já no caso do módulo GPS, sensor de temperatura e display LCD 16x2, foi preciso baixar bibliotecas auxiliares, essas bibliotecas foram baixadas pela aba Ferramentas > Gerenciar bibliotecas (figura 22), onde abre uma janela de pesquisa e aquelas são pesquisadas pelo nome. Todas as bibliotecas baixadas e usadas foram escolhidas depois de vários exemplos de seu uso pela comunidade eletrônica na internet.

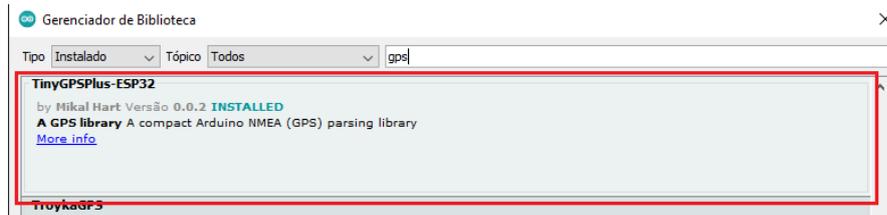
Figura 22 – Seleção de janela para baixar bibliotecas novas



Fonte: Autoria própria.

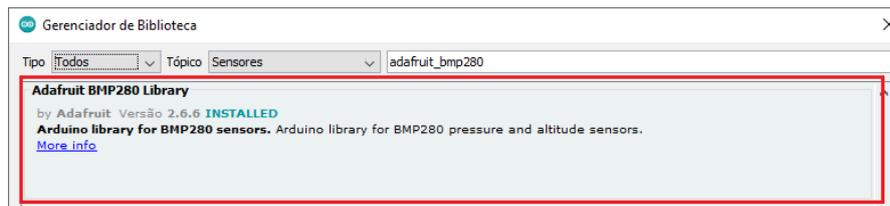
Para ler dados do módulo GPS, dados do sensor de temperatura e escrever no display LCD 16x2 foram baixadas e usadas, respectivamente, as bibliotecas TinyGPSPlus-ESP32 (figura 23), Adafruit BMP 280 (figura 24), e LiquidCrystal_I2C (figura 25).

Figura 23 – Biblioteca do GPS instalada



Fonte: Autoria própria.

Figura 24 – Biblioteca do sensor de temperatura instalada



Fonte: Autoria própria.

Figura 25 – Biblioteca do display 16x2 controlado por I2C instalada



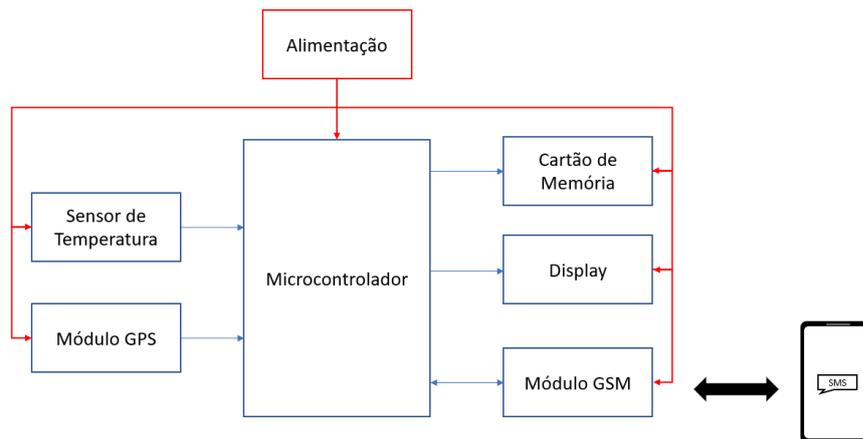
Fonte: Autoria própria.

Com isso, conclui-se a preparação do ambiente de desenvolvimento de firmware.

3.2 DESENVOLVIMENTO DO PROTÓTIPO DE HARDWARE

Para o desenvolvimento do protótipo de hardware, foi usado o diagrama de blocos da figura 26, onde já se tinha esquematizado quais eram os periféricos de entrada e saída que estariam presentes na placa.

Figura 26 - Diagrama de blocos da arquitetura do protótipo



Fonte: Autoria própria.

Com os periféricos já selecionados, passou-se para o desenvolvimento do esquemático do protótipo e seu layout, ambos foram gerados no software Altium Designer 21. Como a maioria dos periféricos foram adquiridos no formato de módulos, e tais módulos funcionam com faixas de tensão parecidas senão iguais, o esquemático ficou focado em fazer a ligação correta com os canais de comunicação do microcontrolador e distribuir a alimentação pelo circuito.

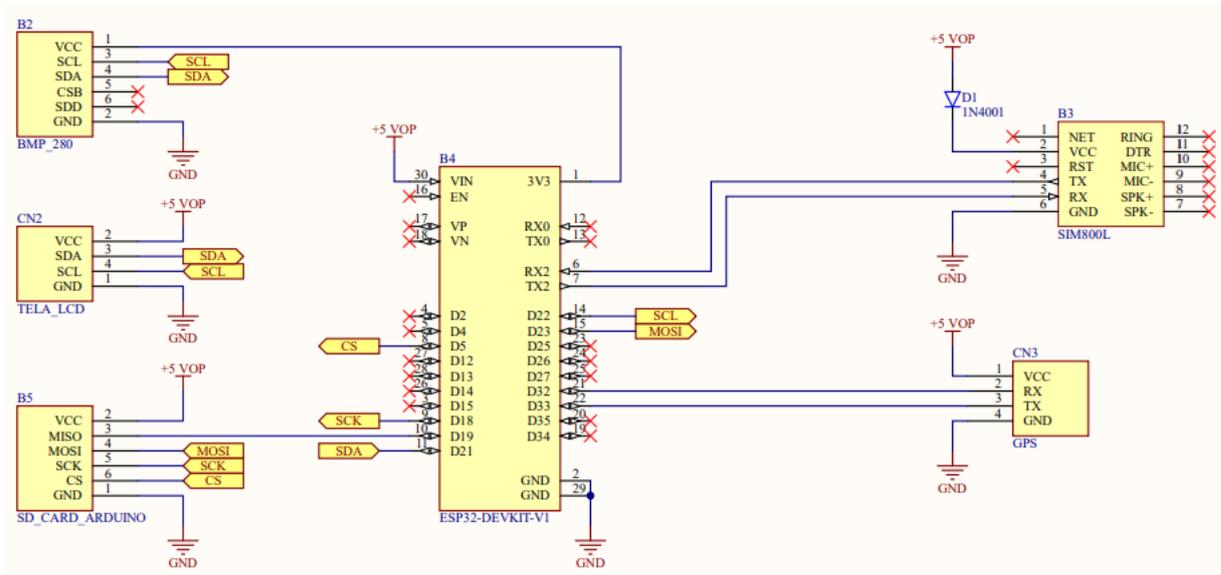
Como o circuito deve ter capacidade de funcionar acompanhando uma caixa térmica, para a alimentação dele foi necessário o uso de uma bateria de lítio 12V. Esta bateria foi escolhida por estar em melhor preço e já vir com fonte para sua recarga. Porém, a maioria dos componentes do projeto funciona com 5V, então foi adicionado ao projeto o regulador de tensão LM2596 com *step down*, este foi ajustado para fornecer 5V na saída e essa alimentação foi distribuída entre todos os componentes do circuito. O ESP32 recebe uma alimentação padrão de 3.3V porém sua placa suporta 5V através do pino de entrada Vin. O módulo GSM recebe uma alimentação de 4.2V, então foi colocado um diodo de silício antes da alimentação para fazer cair a tensão em 0.7V, e fazê-la ficar na faixa de tensão próxima a que o módulo pede.

Com a alimentação distribuída entre os componentes, partiu-se para a conexão de comunicação entre o microcontrolador e os periféricos. O sensor de temperatura e o módulo serial I2C, que faz interface com display LCD 16x2, utilizam comunicação I2C, então ambos tiveram seus pinos SDA e SCL conectados nos pinos D21 e D22 do microcontrolador. Já o módulo GPS e o módulo GSM têm interface serial, então o pino TX do primeiro foi conectado ao D33 do microcontrolador e seu RX, no D32. O pino TX do segundo foi conectado no RX2

do microcontrolador e seu RX, no TX2. Por último, o módulo leitor SD Card usa interface SPI, então seu pino MOSI foi no pino D23, o MISO no D19, o SCK no D18 e o CS no D5.

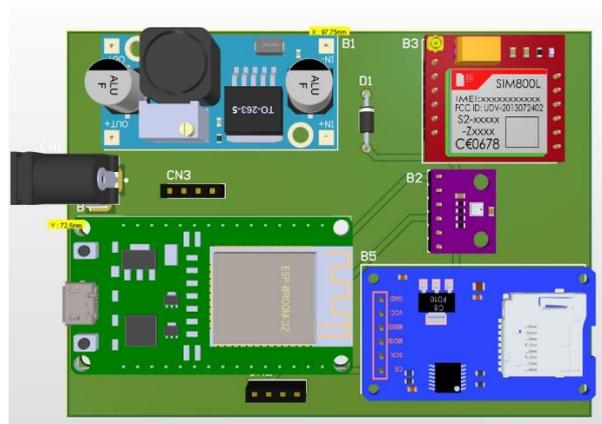
Com isto, as conexões entre os componentes foram finalizadas. A figura 27 a seguir mostra como ficou o esquemático.

Figura 27 – Esquemático do protótipo de hardware



Fonte: Autoria própria.

Figura 28 – Visualização 3D do layout do protótipo de hardware



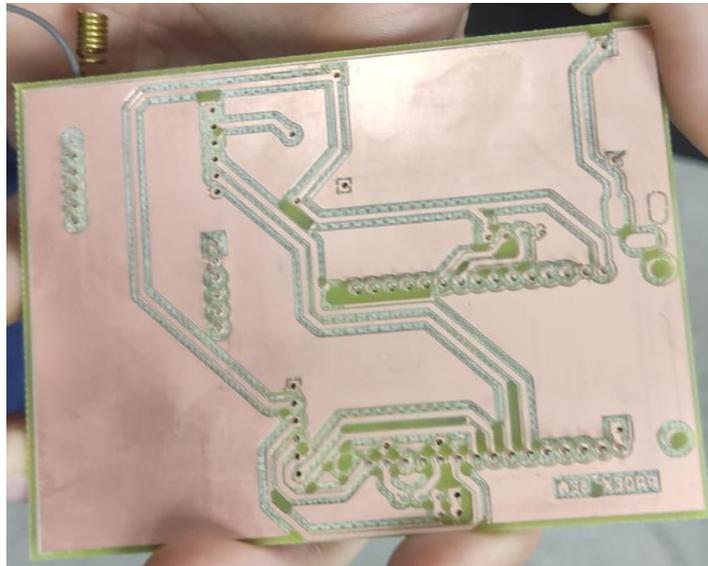
Fonte: Autoria própria.

O layout 3D final da placa é mostrado na figura 28. Com o esquemático finalizado, utilizou-se o programa para gerar o layout do protótipo. Esta etapa consiste em utilizar o esquemático montado para fazer o dimensionamento e posicionamento dos componentes na

placa de circuito impresso. Depois o software vai utilizar as conexões e posições dos componentes para desenhar trilhas, dispondo-as de forma mais eficiente.

Finalizando o layout da placa, o próximo passo é a fabricação utilizando o arquivo gerado pelo software. A fabricação ocorreu no Laboratório de Sistemas Embarcados (LSE) localizado no Hub Tecnologia e Inovação, na Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (UEA), onde o arquivo foi aberto em uma máquina fresadora CNC semiautomática, que gravou o circuito em uma placa de fenolite cobreada de duas faces. O resultado é visto na figura 29 a seguir.

Figura 29 – Resultado da impressão do protótipo em uma placa de fenolite



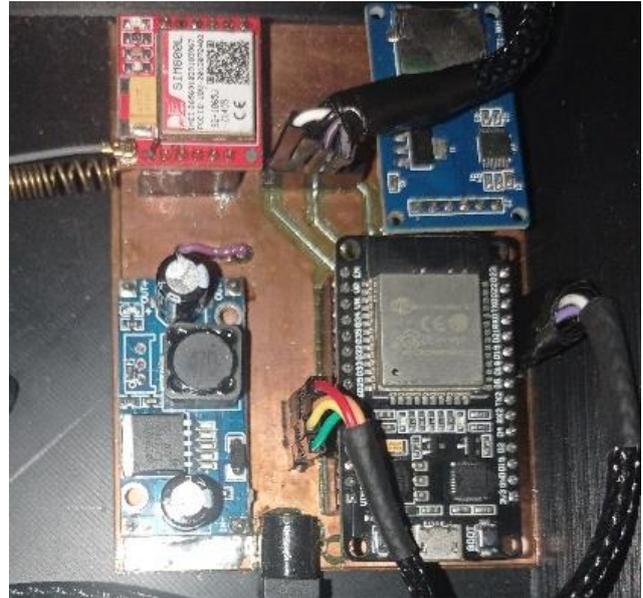
Fonte: Autoria própria.

Depois da impressão, foi feito um teste de continuidade por toda a placa em todos os pontos de conexão, trilhas e ilhas para verificar se não havia algum curto-circuito ou conexão indevida. Passado este teste, utilizou-se a placa de circuito impresso, para fazer a montagem, onde se utilizou ferro de solda, tubo com solda estanho e os componentes do circuito citados no capítulo 2 para realizar a montagem do protótipo em bancada. As figuras 30.a e 30.b demonstram o protótipo de hardware finalizado e montado.

Figura 30 – Resultado da impressão do protótipo em uma placa de fenolite



(a) Placa montada com header



(b) Placa montada com componentes posicionados

Depois da montagem, a validação final da placa consistiu em fazer ligações gradativas, colocando um componente por vez nos conectores. Na figura 31, é demonstrado o final do teste, onde todos os componentes ligaram utilizando a alimentação da bateria, concluindo assim a fase de montagem do protótipo de hardware.

Figura 31 – Resultado da impressão do protótipo em uma placa de fenolite



Fonte: Autoria própria.

3.3 DESENVOLVIMENTO DOS SCRIPTS DE TESTE DE CADA PERIFÉRICO

Primeiramente foi feito o teste de funcionalidade do sensor de temperatura BMP280. Para a programação de scripts foi utilizado o Arduino IDE, escritos separadamente em linguagem C, no ambiente preparado anteriormente.

Inicialmente, faz-se a chamada das bibliotecas que serão necessárias para acionar funções nativas da linguagem C e funções que foram escritas nas bibliotecas baixadas para controlar ou ler dados dos periféricos. Depois ocorre a definição de variáveis para receber valores dos periféricos ou serem utilizadas dentro das funções.

A função `setup()` é executada quando o microcontrolador liga e, geralmente, faz toda a configuração para utilizar no restante do programa (figura 32). Neste caso, está se inicializando a comunicação serial, e depois faz-se um teste de comunicação com o periférico que usa comunicação I2C e usa endereço hexadecimal 0x76, se essa comunicação falhar, o programa imprime uma mensagem de erro e fica preso na linha 21, senão ele imprime uma mensagem e dá prosseguimento na função `loop()`.

Figura 32 – Função `setup()` e configurações do script de teste de sensor de temperatura

```

1  #include <Wire.h>
2  #include <Adafruit_BMP280.h>
3
4  #define PRESSAOINIVELDOMAR 1013.25
5
6  Adafruit_BMP280 sensor; // I2C
7
8  unsigned long delayTime;
9
10 void setup() {
11     Serial.begin(9600);
12     Serial.println(F("Teste modulo BMP280"));
13
14     bool status;
15
16     // default settings
17     // (you can also pass in a Wire library object like
18     status = sensor.begin(0x76);
19     if (!status) {
20         Serial.println("Sensor não encontrado no endereço
21         while (1);
22     } else Serial.println(" - testando");
23
24     delayTime = 1000;
25
26     Serial.println();
27 }

```

Fonte: Autoria própria.

A função `loop()` é executada constantemente enquanto o microcontrolador estiver ligado (figura 33). Neste script de teste, ela chama a função `printValues()` e depois executa um

delay definido na função setup em 1000ms. A função printValues() imprime na janela serial os dados retornados pelas funções de leitura de temperatura, pressão e altitude. Essas funções foram escritas na biblioteca Adafruit, chamada no início do script, e elas acessam as informações coletadas pelo sensor.

Figura 33 – Função loop() do script de teste de sensor de temperatura

```

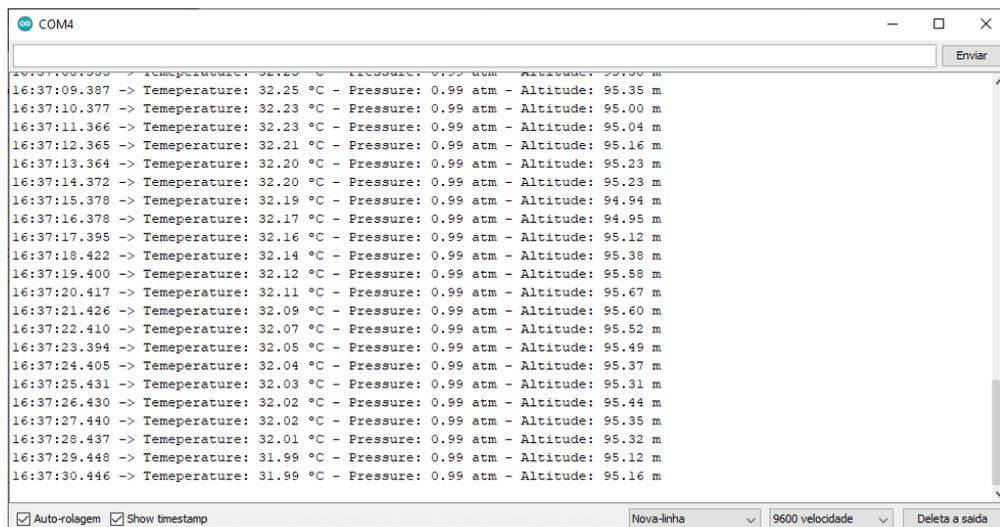
30 void loop() {
31   printValues();
32   delay(delayTime);
33 }
34
35 void printValues() {
36
37   Serial.print("Temperatura: ");
38   Serial.print(sensor.readTemperature(), 2);
39
40   Serial.print(" °C - Pressure: ");
41   Serial.print(sensor.readPressure() / (PRESSAONIVELDOMAR*100), 2)
42
43   Serial.print(" atm - Altitude: ");
44   Serial.print(sensor.readAltitude(PRESSAONIVELDOMAR), 2);
45
46   Serial.println(" m");
47 }

```

Fonte: Autoria própria.

A figura 34 a seguir mostra os valores retornados na janela serial, isso demonstra o funcionamento do sensor.

Figura 34 – Dados capturados do sensor de temperatura no script de teste



The screenshot shows a serial terminal window titled 'COM4'. The output consists of multiple lines of data, each starting with a timestamp and followed by the sensor readings: Temperature in °C, Pressure in atm, and Altitude in m. The data is as follows:

Timestamp	Temperature (°C)	Pressure (atm)	Altitude (m)
16:37:09.387	32.25	0.99	95.35
16:37:10.377	32.23	0.99	95.00
16:37:11.366	32.23	0.99	95.04
16:37:12.365	32.21	0.99	95.16
16:37:13.364	32.20	0.99	95.23
16:37:14.372	32.20	0.99	95.23
16:37:15.378	32.19	0.99	94.94
16:37:16.378	32.17	0.99	94.95
16:37:17.395	32.16	0.99	95.12
16:37:18.422	32.14	0.99	95.38
16:37:19.400	32.12	0.99	95.58
16:37:20.417	32.11	0.99	95.67
16:37:21.426	32.09	0.99	95.60
16:37:22.410	32.07	0.99	95.52
16:37:23.394	32.05	0.99	95.49
16:37:24.405	32.04	0.99	95.37
16:37:25.431	32.03	0.99	95.31
16:37:26.430	32.02	0.99	95.44
16:37:27.440	32.02	0.99	95.35
16:37:28.437	32.01	0.99	95.32
16:37:29.448	31.99	0.99	95.12
16:37:30.446	31.99	0.99	95.16

The terminal window also shows control options at the bottom: 'Auto-rolagem' (checked), 'Show timestamp' (checked), 'Nova-linha' (dropdown), '9600 velocidade' (dropdown), and 'Deleta a saida' (button).

Fonte: Autoria própria.

Para o teste do módulo GPS, além das chamadas bibliotecas foi criado um novo canal de comunicação serial usando a biblioteca Hardware Serial, pois o módulo retorna os valores

coletados através de comunicação serial. A função `setup()` iniciou 2 seriais, uma para coletar os valores do módulo GPS e outra para imprimir os valores coletados na janela serial no Arduino IDE, e prosseguiu-se para a próxima função (figura 35).

Figura 35 – Função `setup()` e configurações do script de teste de módulo GPS

```

1  #include <TinyGPSPlus.h>
2  #include <HardwareSerial.h>
3
4  static const int RXPin = 33, TXPin = 32;
5  static const uint32_t GPSBaud = 4800;
6
7  TinyGPSPlus gps;
8
9  HardwareSerial ss(RXPin, TXPin);
10
11 void setup()
12 {
13     Serial.begin(115200);
14     ss.begin(GPSBaud);
15 }
16

```

Fonte: Autoria própria.

A função `loop()` faz verificações quanto aos valores retornados na serial do módulo GPS, caso não retorne nenhum valor dentro de 5 segundos depois da ligação do microcontrolador, é impresso uma mensagem de erro e o programa ficava travado (figura 36). Caso algum valor seja retornado pela serial do módulo GPS, este valor é passado por uma codificação e depois a função `displayInfo()` é chamada.

Figura 36 – Função `loop()` do script de teste de módulo GPS

```

17 void loop()
18 {
19     while (ss.available() > 0)
20         if (gps.encode(ss.read()))
21             displayInfo();
22
23     if (millis() > 5000 && gps.charsProcessed() < 10)
24     {
25         Serial.println(F("No GPS detected: check wiring."));
26         while(true);
27     }
28 }
29

```

Fonte: Autoria própria.

A função `displayInfo()`, ilustrada na figura 37, é responsável por verificar se os valores codificados são válidos, se forem, eles são impressos na janela serial, se não, é impressa uma mensagem de erro.

Figura 37 – Função displayInfo() do script de teste de módulo GPS

```

30 void displayInfo()
31 {
32   Serial.print(F("Location: "));
33   if (gps.location.isValid())
34   {
35     Serial.print(gps.location.lat(), 6);
36     Serial.print(F(", "));
37     Serial.print(gps.location.lng(), 6);
38   }
39   else
40   {
41     Serial.print(F("INVALID"));
42   }
43
44   Serial.print(F(" Date/Time: "));
45   if (gps.date.isValid())
46   {
47     Serial.print(gps.date.month());
48     Serial.print(F("/"));
49     Serial.print(gps.date.day());
50     Serial.print(F("/"));
51     Serial.print(gps.date.year());
52   }
53   else
54   {
55     Serial.print(F("INVALID"));
56   }
57
58   Serial.print(F(" "));
59   if (gps.time.isValid())
60   {
61     if (gps.time.hour() < 10) Serial.print(F("0"));
62     Serial.print(gps.time.hour());

```

Fonte: Autoria própria.

Os valores coletados foram escritos em uma planilha para comparação e verificação, como mostra a figura 38 a seguir, isso concluiu o teste do módulo GPS.

Figura 38 – Dados capturados do módulo GPS no script de teste

latitude, longitude, dia, mes, ano, hora, minutos, segundos, cent. de seg.		
-3.059710, -60.055300, 7, 5, 2022, 4, 18, 16, 0		
-3.059710, -60.055300, 7, 5, 2022, 4, 18, 16, 0		
-3.059712, -60.055295, 7, 5, 2022, 4, 18, 17, 0		
-3.059712, -60.055295, 7, 5, 2022, 4, 18, 17, 0		
-3.059709, -60.055287, 7, 5, 2022, 4, 18, 18, 0		
-3.059709, -60.055287, 7, 5, 2022, 4, 18, 18, 0		
-3.059712, -60.055279, 7, 5, 2022, 4, 18, 19, 0		
-3.059712, -60.055279, 7, 5, 2022, 4, 18, 19, 0		

Fonte: Autoria própria.

Para o teste do modo leitor de SD Card, foram utilizadas as bibliotecas já instaladas no Arduino IDE, depois disso foram criadas variáveis para sinalizar se for possível a inicialização do SD card e a criação dos diretórios, também foi criado uma estrutura especial,

o DadosGPS, para simular os dados que serão recebidos do módulo GPS e serão salvos no SD card (figura 39).

Figura 39 – Configurações do script de teste do módulo leitor de SD Card

```

1  #include "FS.h"
2  #include "SD.h"
3  #include "SPI.h"
4  #include <stdint.h>
5  #include <string.h>
6
7  #define DEBUG 1
8
9  uint8_t statusCartao = 0, statusDiretorioLogs = 0;
10
11  String diretorioLogAtual = "";
12
13  char dadosGPSString[80];
14
15  typedef struct __attribute__((__packed__)) {
16     double lat, lng;
17     uint16_t year;
18     uint8_t month, day;
19     uint8_t hour, minute, second, centisecond;
20 }DadosGPS;
21
22 DadosGPS dadosGPS;
23

```

Fonte: Autoria própria.

Para facilitar a utilização e escrita do código, dividiu-se o script em arquivos separados por funcionalidade onde, “SD_Card_ESP32” contém o código principal, “drive_GPS” contém as funções do GPS e “driveSD” contém as funções do módulo SD card, ilustrado na figura 40.

Figura 40 – Divisão de código por funcionalidade no script de teste do módulo leitor de SD Card



Fonte: Autoria própria.

O drive_GPS contém apenas a função displayInfo(), escrita durante o teste do módulo GPS, já o drive_SD contém além das funções usadas em exemplos de manipulação e leitura de arquivos e pastas, foram escritas funções que utilizam as funções dos exemplos para criar a estrutura de pastas e escrever arquivos nos formatos desejados para o projeto, ilustrado na figura 41.

Figura 41 – Funções escritas para utilizar o módulo leitor de SD Card

```

202 uint8_t inicializaSD(){
203
204     if(DEBUG) Serial.println("\n[DEBUG] inicializaSD()");
205     if(!SD.begin()){
206         if(DEBUG) Serial.printf("\nCard Mount Failed\n");
207         return 0 ;
208     }
209     if(SD.cardType() == CARD_NONE){
210         if(DEBUG) Serial.printf("\nNo SD card attached\n");
211         return 0 ;
212     }
213
214     return 1;
215 }
216
217
218 int procuraNome(String diretorio, String nome, uint8_t isPasta){
219
220
221
222
223
224
225
226
227
228
229 uint8_t criaDiretorios(String diretorioLogGPS){
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fonte: Autoria própria.

A figura 42 mostra a função setup(), onde a serial de comunicação foi inicializada para visualizar o status do uso do cartão SD pelo microcontrolador, depois é feita uma operação para inicializar a comunicação com o cartão SD. Depois da inicialização, a função cria Diretorios() vai tentar criar a pasta “logsGPS” no cartão SD, a variável diretorioLogAtual salva o caminho gerado ou localizado. Depois dessa preparação, usa-se a variável dadosGPS, com todos seus campos zerados pelo comando na linha 40, e a variável diretorioLogAtual, com o caminho de pastas criado para guardar dados de log, para fazer uma escrita num arquivo .csv. Todas as operações são verificadas após sua execução. E as mensagens de status são imprimidas na tela da janela serial.

Figura 42 – Funções setup() e loop() do script de teste do módulo leitor de SD Card

```

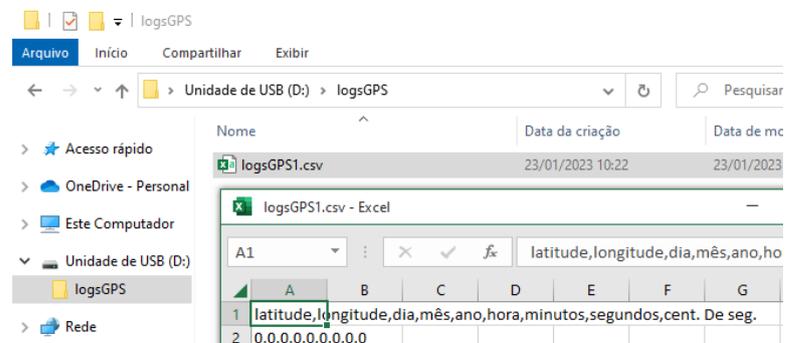
24 void setup() {
25
26     Serial.begin(115200);
27
28     statusCartao = inicializaSD();
29
30     if(statusCartao)
31         statusDiretorioLogs = criaDiretorios("logsGPS");
32     else
33         Serial.print("\n Não montou o cartão SD então não pôde continuar as operações.");
34
35     if(statusDiretorioLogs){
36         Serial.print("\ndiretorioLogAtual: ");
37         Serial.println(diretorioLogAtual);
38     }
39
40     memset(dadosGPS, 0x0, sizeof(dadosGPS));
41
42     if( escreveDadosArquivoCSV(diretorioLogAtual, dadosGPS) != 1 ){
43         if(DEBUG) Serial.println("Não conseguiu salvar os dados capturados no CSV");
44         break;
45     }else
46         if(DEBUG) Serial.println("Salvou os dados no SD Card");
47
48 }
49
50 void loop() {
51     //somente teste de criação de pasta e escrita de arquivo de teste
52 }

```

Fonte: Autoria própria.

Depois da operação com o protótipo, cartão SD foi retirado do leitor e colocado num computador onde foi possível ver o resultado. Como mostra a figura a seguir, o leitor detectou o SD Card, conseguiu salvar a pasta com o nome escrito no código e salvar o arquivo de teste com o conteúdo deseja. Concluindo o teste com o módulo leitor de SD Card.

Figura 43 – Dados capturados do módulo GPS no script de teste



Fonte: Autoria própria.

Para o teste do módulo GSM, foi adquirido um chip da Claro pois um dos planos dava acesso ao envio de mensagens de SMS. Na parte inicial do código, faz-se a chamada das

bibliotecas necessárias para a comunicação com o módulo e define-se o número para o qual será enviado o SMS de teste, como mostrado na figura 44.

Como o módulo GSM usa a comunicação serial, cria-se um canal de comunicação novo e depois utiliza-se função configuraGSM() para enviar comandos de configuração e enviar a mensagem de teste “alo mundo” para o número desejado, funções ilustradas na figura 45.

Figura 44 – Função setup() e configurações do script de teste do módulo GSM

```

2  #include <HardwareSerial.h>
3  #include <stdint.h>
4  #include <string.h>
5
6  #define SMS_TARGET  "+5592981520445"
7  #define GSM_RX_PIN  17
8  #define GSM_TX_PIN  16
9
10 HardwareSerial SerialGSM(2);
11
12 void setup() {
13
14     Serial.begin(115200, SERIAL_8N1, 3, 1); // rx tx
15     SerialGSM.begin(9600, SERIAL_8N1, GSM_RX_PIN, GSM_TX_PIN); // rx tx
16
17
18     configuraGSM();
19
20     char msg[] = "alo mundo";
21     enviaSMS(SMS_TARGET, msg);
22 }

```

Fonte: Autoria própria.

O módulo GSM SIM8001 é controlado através de comandos AT, usados no script e ilustrados na figura 46, onde o prefixo deve ser as letras “AT” não importando se são maiúsculas ou minúsculas, e para finalizar a linha de comando deve ser mandado um <CR> ao final. Os comandos AT básicos têm o formato “AT<x><n>” onde “<x>” é o comando e “<n>” são os argumentos para aquele comando (SHANGHAI SIMCOM WIRELESS SOLUTIONS LTD, 2015).

Figura 45 – Funções de envio de SMS e configuração de módulo GSM

```

29 void enviaSMS(String telefone, String mensagem) {
30     SerialGSM.print("AT+CMGS=\"" + telefone + "\"\n");
31     SerialGSM.print(mensagem + "\n");
32     SerialGSM.print((char)26);
33 }
34
35 void configuraGSM() {
36     SerialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0,0\n;ATX4\n;AT+COLP=1\n");
37 }

```

Fonte: Autoria própria.

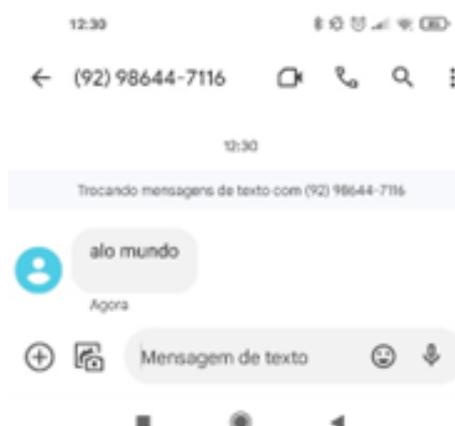
Figura 46 – Comandos utilizados no script de teste para configurar o módulo GSM

Comandos	Descrição
AT+CMGF=1	Ativa o modo de recepção e envio de mensagens de SMS
AT+CNMI=<mode>[,<mt>[,<bm>[,<ds>[,<bfr>]]]] AT+CNMI=2,2,0,0,0	mode=2 significa que os as mensagens são armazenadas em buffer em caso de modem ocupado. mt=2 significa que as mensagens recebidas são mostradas diretamente com um +CMT: . bm=0 significa sem opção de transmissão. ds=0 significa que o recebimento do relatório de status não é relatado e não é armazenado. bfr = 0 significa sem manipulação de códigos de resultado em buffer
ATX4	Código de resultado retornado, discagem e detecção de módulo ocupado estão ambos ativados
AT+COLP=1	Habilita a apresentação do COL (Linha Conectada) no TE para uma chamada originada em celular
AT+CMGS=<da>[,<toda>] <CR>text is entered <CR> AT+CMGS=<da>”\n”<toda>”\n”	Envia mensagem de SMS <toda> para o destinatário <da>

Fonte: Autoria própria.

A figura 47 mostra o resultado do código de teste embarcado no protótipo de hardware, a mensagem recebida no celular. Isso conclui a fase de testes dos periféricos.

Figura 47 – Funções de envio de SMS e configuração de módulo GSM



Fonte: Autoria própria.

3.4 DESENVOLVIMENTO DO SCRIPT GERAL DE INTEGRAÇÃO DO PROTÓTIPO

O script de integração vai fazer uso de todas as tecnologias que estão conectadas no microcontrolador para atender aos requisitos finais do projeto. Ao ligar, o protótipo deve poder comunicar-se com o módulo GSM, módulo GPS, módulo leitor de cartão SD, sensor de temperatura e com o display 16x2. O protótipo deve:

- Enviar mensagens periódicas sobre a localização e a temperatura, quando a temperatura estiver fora da faixa de +2 °C e +8 °C e quando receber a mensagem "request";
- Coletar dados de geolocalização do módulo GPS;
- Coletar dados de temperatura do sensor;
- Registrar os dados coletados de temperatura e geolocalização no cartão SD;
- Mostrar o status dos módulos no display LCD;

Para atingir os objetivos, primeiramente foi criada uma estrutura de arquivos para separar os códigos por funcionalidade (figura 48). As funcionalidades foram transcritas dos testes anteriores, algumas delas sofreram alterações para se encaixarem melhor neste novo código.

Figura 48 – Divisão de código por funcionalidade no script de integração



Fonte: Autoria própria.

O arquivo "Defines.h", ilustrado na figura 49, é uma biblioteca que foi criada para chamar todas as outras bibliotecas que dão acesso aos periféricos da placa, e também declarar as variáveis que vão receber e manipular os dados. São 2 variáveis para inicializar a comunicação serial usada pelo módulo GPS e módulo GSM, uma variável para guardar os dados recebidos do módulo GPS, uma para o sensor de temperatura, e outra para controlar o display LCD, também são definidos os endereços I2C utilizados por estes 2 últimos.

Os arquivos "drive_GPS" e "drive_SD", contendo as funções úteis para este projeto, foram copiados dos testes anteriores. O "driveGPS" tem a função para verificar a validade dos dados recebidos do módulo GPS e o "drive_SD" tem as funções para inicializar o cartão SD dentro do firmware, criar a estrutura de pastas e arquivos e escrever os dados de temperatura e geolocalização em um arquivo .csv.

Figura 49 – Imagem do script “Defines.h”

```

1  #ifndef DEFINES_H
2  #define DEFINES_H
3
4  #include "FS.h"
5  #include "SD.h"
6  #include "SPI.h"
7  #include "TinyGPSPlus.h"
8  #include <stdint.h>
9  #include <string.h>
10 #include <HardwareSerial.h>
11 #include <Wire.h>
12 #include <Adafruit_BMP280.h>
13 #include <LiquidCrystal_I2C.h>
14
15 /***** DEFINES *****/
16
17 #define DEBUG 1
18 #define TEMPO_RESET_LOOP 15
19
20 #define GPS_RX_PIN 33
21 #define GPS_TX_PIN 32
22
23 #define GSM_RX_PIN 17
24 #define GSM_TX_PIN 16
25
26 #define numeroCall "+5592981520445" //NUMERO A RECEBER SMS
27
28 uint8_t flag_lcd = 0;
29
30 /***** GPS *****/
31 char dadosGPSString[80];
32
33 typedef struct __attribute__((__packed__)) {
34     double lat, lng;
35     uint16_t year;
36     uint8_t month, day;
37     uint8_t hour, minute, second, centisecond;
38 }DadosGPS;
39
40
41 HardwareSerial SerialGPS(1);
42 String msg;
43
44 DadosGPS dadosGPS, ultimoDado;

```

Fonte: Autoria própria.

No caso do “DriverGSM.h”, foi adicionado uma nova função para poder ler a mensagem que chegasse, pois no protótipo final ele deve ser capaz de enviar uma mensagem sobre os status da caixa quando receber a mensagem “request”. As outras funcionalidades de envio e configuração foram copiadas do script de teste anterior. Também foram criadas flags para compor a mensagem que será escrita para envio. Como mostra a figura 50 a seguir.

Figura 50 – Imagem do script “DriverGSM.h”

```

1 #ifndef DRIVERGSM_H
2 #define DRIVERGSM_H
3
4 bool temSMS = false, flag_send_SMS = false;
5 String telefoneSMS;
6 String dataHoraSMS;
7 String mensagemSMS;
8 String comandoGSM = "";
9 String ultimoGSM = "";
10
11 #define senhaGsm "request"
12
13 void enviaSMS(String telefone, String mensagem) {
14     SerialGSM.print("AT+CMGS=\" + telefone + "\"\n");
15     SerialGSM.print(mensagem + "\n");
16     SerialGSM.print((char)26);
17 }
18
19 void configuraGSM() {
20     SerialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0,0\n;ATX4\n;AT+COLP=1\n");
21 }
22
23 void leGSM(){
24     static String textoRec = "";
25     static unsigned long delayl = 0;
26     static int count=0;
27     static unsigned char buffer[64];
28
29     if (SerialGSM.available()) {
30
31         while(SerialGSM.available()) {
32
33             buffer[count++] = SerialGSM.read();
34             if(count == 64)break;
35         }
36
37         textoRec += (char*)buffer;
38         delayl = millis();
39
40         for (int i=0; i<count; i++) {
41             buffer[i]=NULL;
42         }
43         count = 0;
44     }

```

Fonte: Autoria própria.

No arquivo SD_Card_ESP32, ilustrado na figura 51, são feitas as inclusões dos drivers auxiliares criados e utiliza-se a função setup() para fazer a configuração geral de todos os periféricos, inicializando em sequência os canais comunicação serial, o display LCD, o sensor de temperatura, o módulo leitor de cartão SD e criando uma task para gerenciar o envio e recebimento de SMS.

Figura 51 – Imagem do script “SD_Card_ESP32”

```

1 #include "Defines.h"
2 #include "DriverGSM.h"
3
4 TaskHandle_t task_GSM;
5
6 void setup() {
7
8     memset(dadosGPSString, 0x20, sizeof(dadosGPSString));
9     Serial.begin(115200, SERIAL_8N1, 3, 1); // rx tx
10    SerialGPS.begin(9600, SERIAL_8N1, GPS_RX_PIN, GPS_TX_PIN); // rx tx
11    SerialGSM.begin(9600, SERIAL_8N1, GSM_RX_PIN, GSM_TX_PIN); // rx tx
12
13    lcd.begin(); // inicio do LCD
14    lcd.backlight(); // ligando backlight
15
16    unsigned status;
17    status = bmp.begin(0x76);
18    if (!status) {
19        lcd.clear();
20        lcd.setCursor(6, 0);
21        lcd.print("ERRO");
22        lcd.setCursor(3, 1);
23        lcd.print("TERMOMETRO");
24        if(DEBUG) Serial.println(F("Sensor de temperatura nao encontrado"));
25        vTaskDelay(1000);
26    }
27
28    bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, // Operating Mode.
29                  Adafruit_BMP280::SAMPLING_X2, // Temp. oversampling
30                  Adafruit_BMP280::SAMPLING_X16, // Pressure oversampling
31                  Adafruit_BMP280::FILTER_X16, // Filtering.
32                  Adafruit_BMP280::STANDBY_MS_500); // Standby time.
33
34
35    statusCartao = inicializaSD();
36
37    if(statusCartao)

```

Fonte: Autoria própria.

A necessidade de utilizar a ferramenta de processamento paralelo do ESP32 por meio de *tasks* separadas, surgiu da necessidade da funcionalidade de coleta de dados não interferir com a comunicação via SMS do protótipo, desse jeito a *task* loop() cuida de toda a coleta de informação dos sensores de temperatura e do módulo GPS, registra os dados no cartão SD e atualiza o display LCD, como demonstra a figura 53.

Na figura 52 é possível ver a tela do display LCD preenchida com as informações de status de cada um dos periféricos.

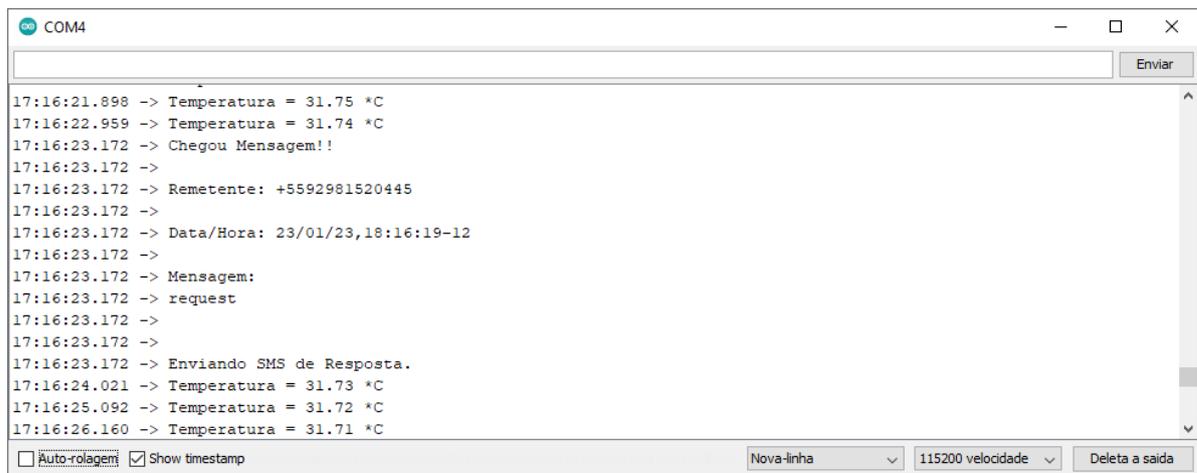
Figura 52 – Foto do display LCD 16x2



Fonte: Autoria própria.

Como teste, foi enviado uma mensagem de SMS para a caixa enquanto se observava a janela serial do Arduino IDE. Foi possível observar que o SMS foi recebido, o remetente e a hora puderam ser vistos também (figura 55). Como programado, o protótipo também manda um SMS de resposta para o remetente (figura 56), na mensagem utiliza-se a latitude e longitude capturadas para compor um link, que indica o local onde as coordenadas apontam segundo o site Google Maps.

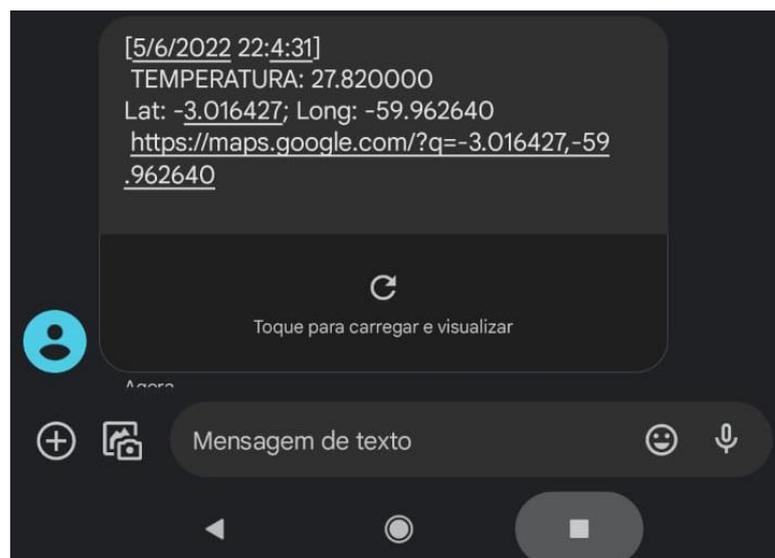
Figura 55 – Demonstração de mensagem recebida pelo módulo GSM



```
COM4
17:16:21.898 -> Temperatura = 31.75 *C
17:16:22.959 -> Temperatura = 31.74 *C
17:16:23.172 -> Chegou Mensagem!!
17:16:23.172 ->
17:16:23.172 -> Remetente: +5592981520445
17:16:23.172 ->
17:16:23.172 -> Data/Hora: 23/01/23,18:16:19-12
17:16:23.172 ->
17:16:23.172 -> Mensagem:
17:16:23.172 -> request
17:16:23.172 ->
17:16:23.172 ->
17:16:23.172 -> Enviando SMS de Resposta.
17:16:24.021 -> Temperatura = 31.73 *C
17:16:25.092 -> Temperatura = 31.72 *C
17:16:26.160 -> Temperatura = 31.71 *C
```

Fonte: Autoria própria.

Figura 56 – Mensagem recebida pelo celular enviada pelo protótipo

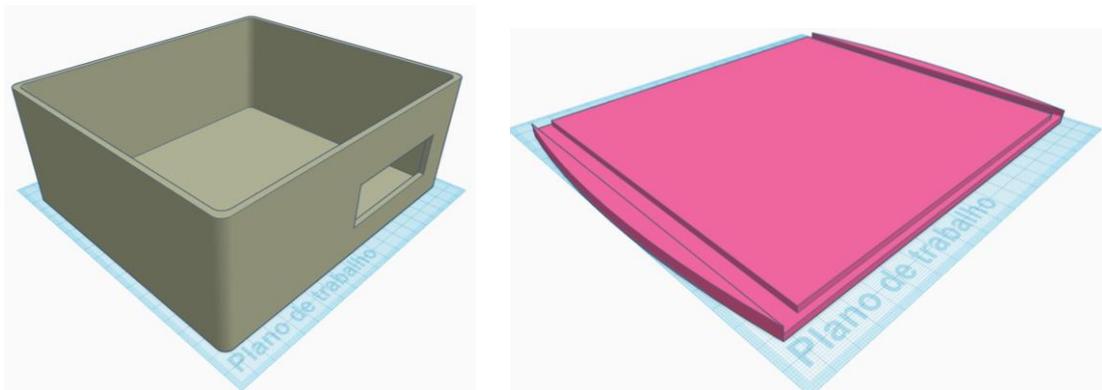


Fonte: Autoria própria.

3.5 VALIDAÇÃO FINAL DO PROTÓTIPO DO DISPOSITIVO

Para a validação final do protótipo de dispositivo, surgiu a necessidade de colocar o protótipo de hardware em uma caixa para que a placa e seus componentes ficassem protegidos. Como no mercado não havia uma caixa que fosse compacta, mas que coubessem todos os componentes do circuito, foi utilizado o software online Tinker CAD, para fazer uma caixa modelada 3D. A caixa e sua tampa projetadas são mostradas nas figuras a seguir.

Figura 57 – Caixa projetada no Tinker CAD



(a) Caixa 3D projetada no Tinker CAD (b) Tampa para a Caixa projetada no Tinker CAD

A caixa foi mandada para um serviço de impressão e o resultado é mostrado na figura 58 a seguir.

Figura 58 – Caixa e tampa impressas com base no projeto mandados



Fonte: Autoria própria.

Os componentes foram montados dentro da caixa com auxílio de fita dupla face e cola quente. Também foram feitos furos para passar o sensor de temperatura e fixar o display LCD. O resultado é mostrado na figura a seguir.

Figura 59 – Montagem do hardware e componentes dentro da caixa impressa



Fonte: Autoria própria.

A caixa contendo o protótipo de hardware foi anexada na parte de cima de uma caixa de isopor de 12 l, passando o sensor de temperatura por um furo feito na tampa da caixa de isopor formando um conjunto mostrado na figura 60. Logo em seguida, foi feito o primeiro teste de medição de temperatura, comparando com a medição o multímetro digital ET-2042F.

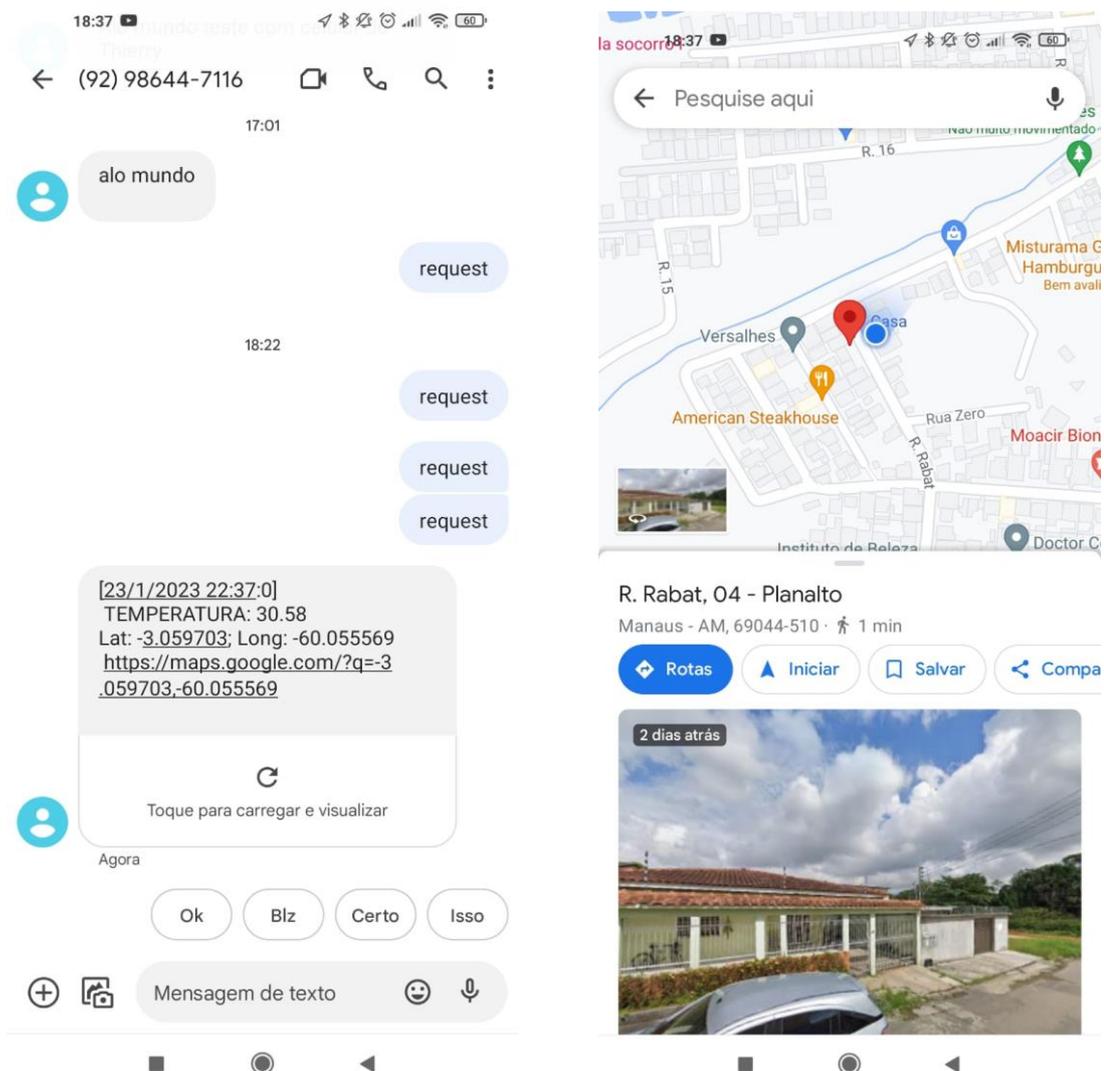
Figura 60 – Conjunto de caixa do protótipo e caixa de isopor em teste



Fonte: Autoria própria.

No segundo teste, o protótipo foi ligado e o conjunto foi deixado em um local fixo, sem gelo ou bolsa térmica dentro da caixa, na casa do desenvolvedor da pesquisa, para verificar os registros de geolocalização e temperatura salvos no cartão de memória assim como verificar o envio e recepção de mensagens com o status da caixa (figura 61).

Figura 61 – Verificação de envio de mensagem de SMS com localização coletada pelo protótipo comparada com a localização do GPS do smartphone



(a) Mensagem recebida após a requisição

(b) Visualização de onde as coordenadas apontam

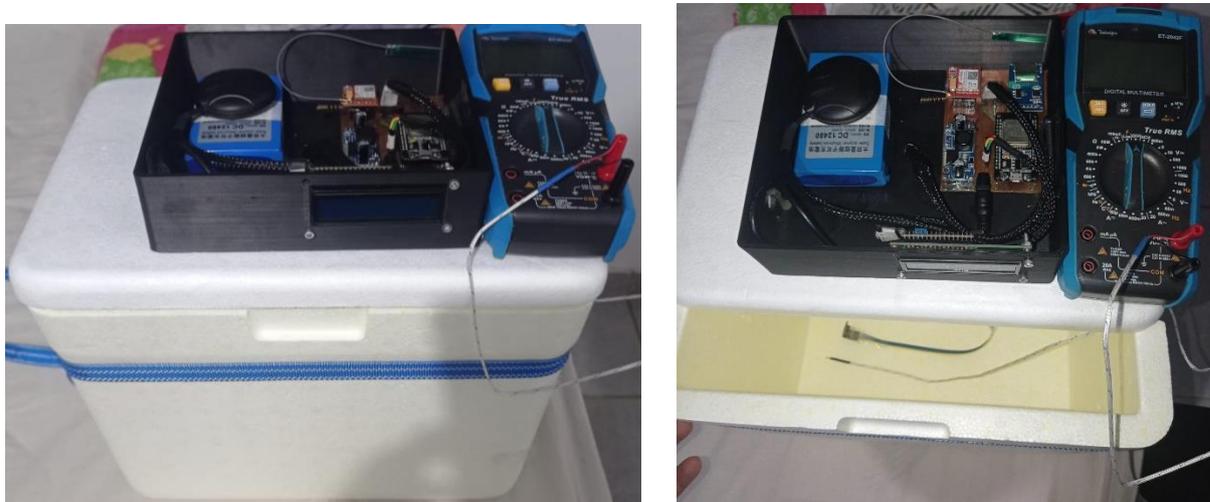
4 ANÁLISE DOS RESULTADOS

Este capítulo apresenta análise dos resultados obtidos com os testes de validação e está dividido em 5 seções onde serão detalhados: a preparação dos testes, os cenários de teste, os alertas de temperatura emitidos e uma comparação com produtos do mercado.

4.1 PREPARAÇÃO DOS TESTES

Para a preparação de teste, foi feito um furo na tampa da caixa de isopor, para passar o sensor de temperatura. O *case* foi fixado na tampa com fita dupla face, assim como o multímetro digital com o termopar. O termopar foi colocado na mesma posição que o sensor de temperatura dentro da caixa de isopor, conforme ilustrado na figura 62. Em ambos os cenários foi colocada uma bolsa térmica dentro da caixa de isopor, e durante o período de teste a temperatura vista no multímetro e o horário foram anotados para comparação posterior.

Figura 62 – Conjunto de caixa do protótipo e caixa de isopor em teste



Fonte: Autoria própria.

4.2 PRIMEIRO CENÁRIO DE TESTES

No primeiro cenário de teste, o protótipo foi ligado e o conjunto foi deixado estático por um período no local onde foi programado (com latitude = -3.059601° e longitude = -60.055361°) para verificação de temperatura medida e mensagens de SMS recebidas. Durante o teste, mensagens de “*request*” foram enviadas para verificar a recepção no protótipo.

Figura 63 – Dados de medição de temperatura do multímetro ET-2042F no cenário de teste 1

Data e Hora	Temperatura Anotada
25/01/2023 6:23	-14 °C
25/01/2023 6:37	-13 °C
25/01/2023 6:49	-12 °C
25/01/2023 7:08	-11 °C
25/01/2023 7:20	-10 °C

Fonte: Autoria própria.

Do cenário de teste 1, o arquivo de nome “log3.csv” (figura 64) foi copiado do cartão SD. Nele continham 5434 amostras, com os dados dos sensores temperatura e módulo GPS.

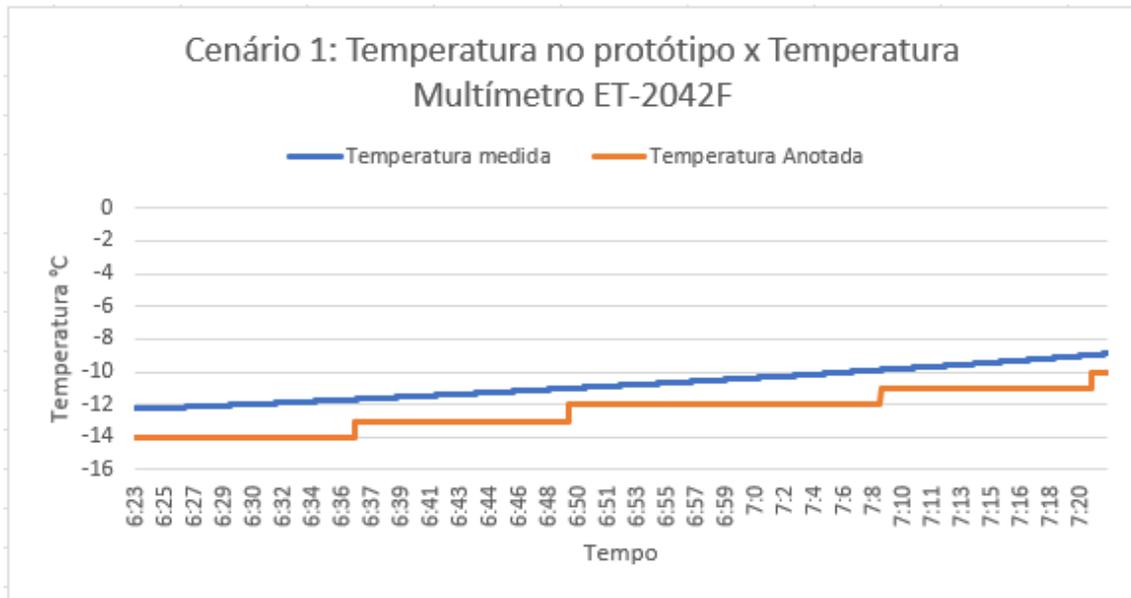
Figura 64 – Dados registrados no arquivo log3.csv copiado do cartão SD.

	A	B	C	D	E	F	G	H
1	latITUDE,longitude,dia,mes,ano,hora,minutos,segundos,cent. de seg.,temperatura							
2	-3.059615, -60.055371, 25, 1, 2023, 10, 23, 44, 0, -12.17							
3	-3.059615, -60.055371, 25, 1, 2023, 10, 23, 44, 0, -12.17							
4	-3.059615, -60.055371, 25, 1, 2023, 10, 23, 44, 0, -12.18							
5	-3.059609, -60.055377, 25, 1, 2023, 10, 23, 45, 0, -12.18							
6	-3.059598, -60.055382, 25, 1, 2023, 10, 23, 46, 0, -12.19							
7	-3.059595, -60.055386, 25, 1, 2023, 10, 23, 48, 0, -12.19							
8	-3.059595, -60.055386, 25, 1, 2023, 10, 23, 48, 0, -12.19							
9	-3.059595, -60.055386, 25, 1, 2023, 10, 23, 48, 0, -12.20							
10	-3.059595, -60.055386, 25, 1, 2023, 10, 23, 48, 0, -12.20							

Fonte: Autoria própria.

Os dados coletados pelo protótipo e os anotados do multímetro (figura 63) foram usados para criação de uma tabela comparativa e um gráfico para auxiliar a comparação, como mostram as figuras 65 e 66 a seguir.

Figura 65 – Gráfico de comparação de temperatura medidas no cenário 1



Fonte: Autoria própria.

Figura 66 – Tabela criada a partir dos dados de coleta do cenário de teste 1 para análise e comparação

	latitude	longitude	Data	Hora Corrigida	Segundos	Temperatura medida	Temperatura Anotada	Erro Absoluto	Erro Relativo
1	-3,059615	-60,055371	25/1/2023	6:23	44	-12,17	-14	-1,83	13,07%
2	-3,059615	-60,055371	25/1/2023	6:23	44	-12,17	-14	-1,83	13,07%
3	-3,059615	-60,055371	25/1/2023	6:23	44	-12,18	-14	-1,82	13,00%
4	-3,059609	-60,055377	25/1/2023	6:23	45	-12,18	-14	-1,82	13,00%
5	-3,059598	-60,055382	25/1/2023	6:23	46	-12,19	-14	-1,81	12,93%
6	-3,059595	-60,055386	25/1/2023	6:23	48	-12,19	-14	-1,81	12,93%
7	-3,059595	-60,055386	25/1/2023	6:23	48	-12,19	-14	-1,81	12,93%
8	-3,059595	-60,055386	25/1/2023	6:23	48	-12,19	-14	-1,81	12,93%
9	-3,059595	-60,055386	25/1/2023	6:23	48	-12,2	-14	-1,8	12,86%
10	-3,059595	-60,055386	25/1/2023	6:23	48	-12,2	-14	-1,8	12,86%
11	-3,059595	-60,055386	25/1/2023	6:23	48	-12,21	-14	-1,79	12,79%
12	-3,059593	-60,05538	25/1/2023	6:23	50	-12,21	-14	-1,79	12,79%
13	-3,059593	-60,05538	25/1/2023	6:23	50	-12,21	-14	-1,79	12,79%
14	-3,059589	-60,055384	25/1/2023	6:23	51	-12,21	-14	-1,79	12,79%
15	-3,059589	-60,055384	25/1/2023	6:23	51	-12,22	-14	-1,78	12,71%
16	-3,059583	-60,055386	25/1/2023	6:23	54	-12,22	-14	-1,78	12,71%
17	-3,059583	-60,055386	25/1/2023	6:23	54	-12,22	-14	-1,78	12,71%
18	-3,059583	-60,055386	25/1/2023	6:23	54	-12,23	-14	-1,77	12,64%
19	-3,059583	-60,055386	25/1/2023	6:23	54	-12,23	-14	-1,77	12,64%
20	-3,059579	-60,055384	25/1/2023	6:23	55	-12,23	-14	-1,77	12,64%
21	-3,059579	-60,05538	25/1/2023	6:23	56	-12,23	-14	-1,77	12,64%
22	-3,059576	-60,055375	25/1/2023	6:23	57	-12,23	-14	-1,77	12,64%
23	-3,05957	-60,055379	25/1/2023	6:23	58	-12,23	-14	-1,77	12,64%
24	-3,05957	-60,055379	25/1/2023	6:23	58	-12,23	-14	-1,77	12,64%
25	-3,05957	-60,055379	25/1/2023	6:23	58	-12,23	-14	-1,77	12,64%
26	-3,059563	-60,055374	25/1/2023	6:23	59	-12,24	-14	-1,76	12,57%
27	-3,05956	-60,055372	25/1/2023	6:24	0	-12,24	-14	-1,76	12,57%
28	-3,05956	-60,055372	25/1/2023	6:24	0	-12,24	-14	-1,76	12,57%
29	-3,059556	-60,05537	25/1/2023	6:24	1	-12,24	-14	-1,76	12,57%
30	-3,059556	-60,05537	25/1/2023	6:24	1	-12,24	-14	-1,76	12,57%

Fonte: Autoria própria.

4.3 SEGUNDO CENÁRIO DE TESTES

No segundo cenário de teste, o protótipo foi ligado e o conjunto foi levado em um trajeto desde a Escola Superior de Tecnologia da Universidade do Estado do Amazonas (com latitude = -3.091646° e longitude = -60.017234°) de volta ao local onde foi programado.

Figura 67 – Dados de medição de temperatura do multímetro ET-2042F no cenário de teste 2

Data e Hora	Temperatura Anotada
24/01/2023 19:31	2 °C
24/01/2023 19:32	3 °C
24/01/2023 19:36	4 °C
24/01/2023 19:44	5 °C
24/01/2023 19:55	6 °C
24/01/2023 20:02	7 °C
24/01/2023 20:04	8 °C
24/01/2023 20:05	9 °C

Fonte: Autoria própria.

Do cenário de teste 2, o arquivo de nome “log16.csv” (figura 68) foi copiado do cartão SD. Nele continham 3211 amostras, com os dados dos sensores temperatura e módulo GPS.

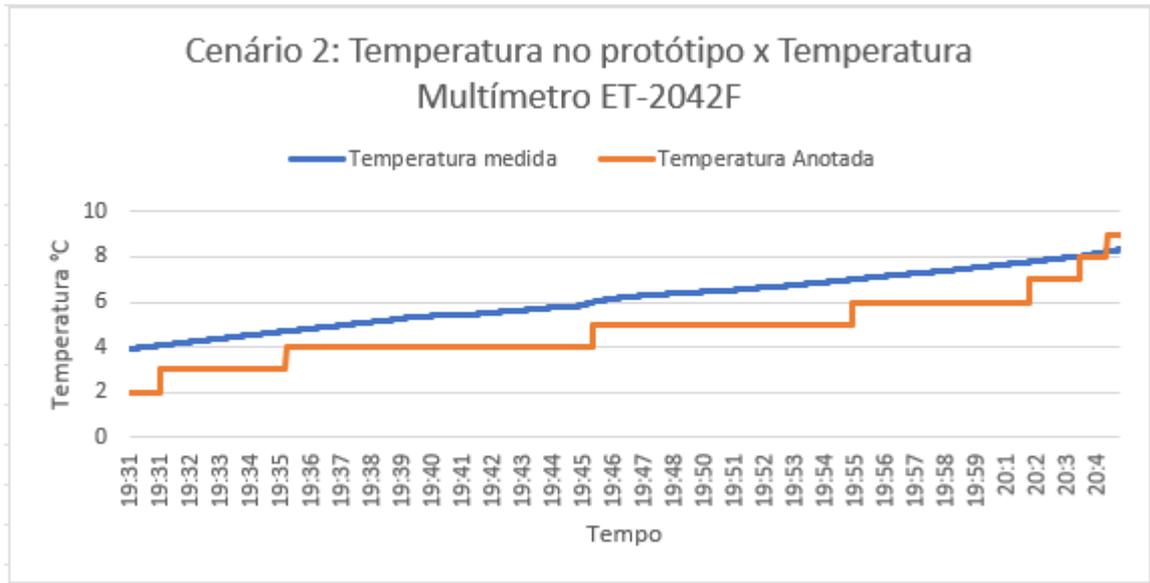
Figura 68 – Dados registrados no arquivo log16.csv copiado do cartão SD.

	A	B	C	D	E	F	G	H
1	latitude,longitude,dia,mes,ano,hora,minutos,segundos,cent. de seg.,temperatura							
2	-3.091380, -60.017947, 24, 1, 2023, 23, 31, 2, 0, 3.91							
3	-3.091380, -60.017947, 24, 1, 2023, 23, 31, 2, 0, 3.91							
4	-3.091377, -60.017944, 24, 1, 2023, 23, 31, 4, 0, 3.92							
5	-3.091377, -60.017944, 24, 1, 2023, 23, 31, 4, 0, 3.92							
6	-3.091379, -60.017943, 24, 1, 2023, 23, 31, 5, 0, 3.92							
7	-3.091379, -60.017943, 24, 1, 2023, 23, 31, 5, 0, 3.92							

Fonte: Autoria própria.

Os dados coletados pelo protótipo e os anotados do multímetro (figura 67) foram usados para criação de uma tabela comparativa e um gráfico para auxiliar a comparação, como mostram as figuras 69 e 70 a seguir.

Figura 69 – Gráfico de comparação de temperatura medidas no cenário 2



Fonte: Autoria própria.

Figura 70 – Tabela criada a partir dos dados de coleta do cenário de teste 2 para análise e comparação

	latitude	longitude	Data	Hora Corrigida	Segundos	Temperatura medida	Temperatura Anotada	Erro Absoluto	Erro Relativo
1									
2	-3,09138	-60,017947	24/1/2023	19:31	2	3,91	2	1,91	95,50%
3	-3,09138	-60,017947	24/1/2023	19:31	2	3,91	2	1,91	95,50%
4	-3,091377	-60,017944	24/1/2023	19:31	4	3,92	2	1,92	96,00%
5	-3,091377	-60,017944	24/1/2023	19:31	4	3,92	2	1,92	96,00%
6	-3,091379	-60,017943	24/1/2023	19:31	5	3,92	2	1,92	96,00%
7	-3,091379	-60,017943	24/1/2023	19:31	5	3,92	2	1,92	96,00%
8	-3,091379	-60,017943	24/1/2023	19:31	5	3,92	2	1,92	96,00%
9	-3,091379	-60,017943	24/1/2023	19:31	5	3,93	2	1,93	96,50%
10	-3,091381	-60,017941	24/1/2023	19:31	8	3,93	2	1,93	96,50%
11	-3,091381	-60,017941	24/1/2023	19:31	8	3,93	2	1,93	96,50%
12	-3,091381	-60,017941	24/1/2023	19:31	8	3,93	2	1,93	96,50%
13	-3,091381	-60,017937	24/1/2023	19:31	9	3,93	2	1,93	96,50%
14	-3,091381	-60,017937	24/1/2023	19:31	9	3,94	2	1,94	97,00%
15	-3,091381	-60,017933	24/1/2023	19:31	10	3,94	2	1,94	97,00%
16	-3,091383	-60,017932	24/1/2023	19:31	11	3,94	2	1,94	97,00%
17	-3,091385	-60,017932	24/1/2023	19:31	12	3,94	2	1,94	97,00%
18	-3,091385	-60,017932	24/1/2023	19:31	12	3,94	2	1,94	97,00%
19	-3,091385	-60,017932	24/1/2023	19:31	12	3,94	2	1,94	97,00%
20	-3,091385	-60,017932	24/1/2023	19:31	12	3,95	2	1,95	97,50%
21	-3,091388	-60,017934	24/1/2023	19:31	13	3,95	2	1,95	97,50%
22	-3,091392	-60,017936	24/1/2023	19:31	14	3,95	2	1,95	97,50%
23	-3,091395	-60,017942	24/1/2023	19:31	15	3,95	2	1,95	97,50%
24	-3,091395	-60,017942	24/1/2023	19:31	15	3,95	2	1,95	97,50%
25	-3,091398	-60,017943	24/1/2023	19:31	16	3,95	2	1,95	97,50%
26	-3,091398	-60,017943	24/1/2023	19:31	16	3,95	2	1,95	97,50%
27	-3,091398	-60,017943	24/1/2023	19:31	16	3,96	2	1,96	98,00%
28	-3,091398	-60,017943	24/1/2023	19:31	16	3,96	2	1,96	98,00%
29	-3,091411	-60,017958	24/1/2023	19:31	19	3,96	2	1,96	98,00%
30	-3,091411	-60,017958	24/1/2023	19:31	19	3,96	2	1,96	98,00%

Fonte: Autoria própria.

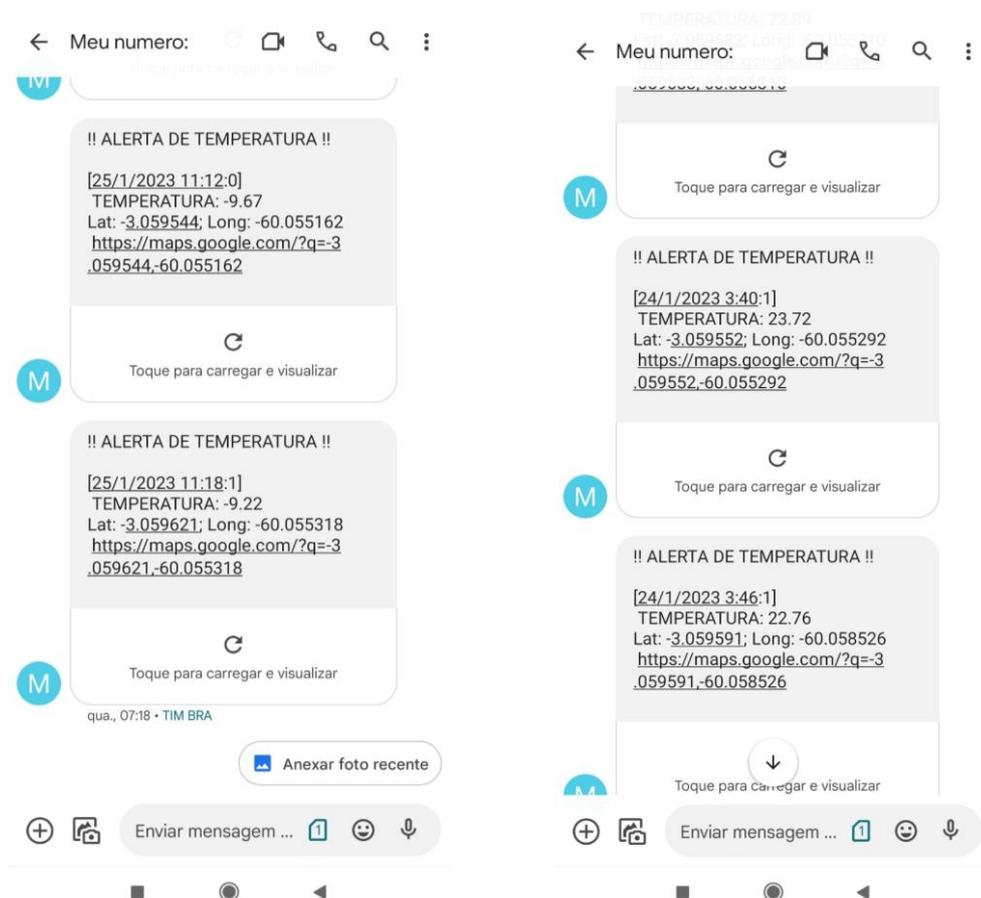
Foi utilizado o aplicativo Google Maps instalado em um *smartphone* que estava junto do protótipo para verificar a rota traçada no trajeto, a figura 72 mostra esta rota.

4.4 ALERTAS DE TEMPERATURA

Durante o período de funcionamento do dispositivo, ele enviou mensagens de SMS alertando sobre a temperatura detectada pelo sensor para o número indicado a cada 6 minutos (figura 73), e também recebeu mensagens e enviou mensagens de resposta com status de localização e temperatura, conforme o programado.

O conteúdo da mensagem é baseado em dados do módulo GPS e do sensor de temperatura. Do módulo GPS foi possível retirar não só a geolocalização como data e hora da recepção do sinal. Comparando com o horário de chegada no *smartphone*, percebe-se que a hora precisa apenas levar em conta o fuso horário de -4 horas de Manaus, mas os minutos apontam que a mensagem chegou em menos de 1 minuto desde o seu envio pelo dispositivo.

Figura 73 – Alertas de temperatura recebidos via SMS no *smartphone*



(a) Temperatura abaixo do ideal

(b) Temperatura acima do ideal

4.5 COMPARAÇÃO COM PRODUTOS DE MERCADO

O custo total do protótipo foi de R\$ 493,80, que apresenta valor mais baixo que os produtos comerciais que têm funcionalidades similares como o Testo 174H TESTO 0572 6560, custando R\$ 1105,99, ou o Testo 184 T3 TESTO 0572 1843, que custa R\$ 1402,99. Porém o dispositivo desenvolvido ainda apresenta as vantagens de gravar geolocalização, fazer monitoramento remoto via comunicação GSM e memória removível com o SD Card, isso dá a vantagem de poder trocar de cartão SD sem ter que trocar de aparelho. As desvantagens são com relação ao tamanho (o protótipo é maior), precisão da temperatura.

CONCLUSÃO

Neste trabalho foi desenvolvido um protótipo de dispositivo de monitoramento de temperatura e geolocalização de vacinas durante o transporte e em unidades básicas de saúde por meio de telemetria para dispositivos móveis.

Para validar as funcionalidades do protótipo, os testes foram feitos em dois cenários. No primeiro cenário, o dispositivo foi deixado estático no mesmo local onde foi programado. No segundo cenário, o dispositivo foi levado em um trajeto desde a UEA de volta ao local de programação. Durante o primeiro cenário, o dispositivo detectou temperaturas abaixo do ideal e enviou mensagens de alerta para o número programado. No segundo cenário, o dispositivo não enviou alertas pois a temperatura detectada estava na faixa ideal. Depois de cada cenário, os dados armazenados no cartão SD foram coletados para análise.

Analisando os dados, o maior erro absoluto de temperatura detectado no primeiro cenário foi de 2,32 °C acima do medido pelo multímetro, já no segundo cenário foi de 2,06 °C acima. Esse erro pode ser explicado pela precisão do sensor digital de ± 1 °C. Outro fator que pode afetar a medição é a posição do sensor em relação ao que vai se medir, pois o multímetro usa um termopar que entra em contato direto com o objeto, já no sensor digital existe um elemento sensível dentro de um chip que está numa placa, todo este aparato pode influenciar a precisão da medição. Observa-se então que o sensor digital de temperatura precisa ter uma posição fixa e o mais próxima possível do objeto a ser medido dentro do recipiente, a fim de garantir uma leitura mais perto da temperatura real. Também é preciso checar a precisão do sensor previamente para adição de *offset*.

O módulo leitor de SD Card funcionou conforme o programado, registrando dados em um arquivo no formato especificado e sem corromper os arquivos, o que resultaria na perda total dos registros.

Os módulos GSM e GPS também funcionaram de acordo com o programado, enviando e recebendo mensagens de SMS e apontando as coordenadas geográficas corretas, porém durante os testes percebeu-se uma demora de alguns minutos em ambos para entrar em pleno funcionamento quando colocados em ambientes internos de edifícios, isso pode ser explicado pela perda de potência do sinal dentro destes ambientes combinado com antenas de baixa sensibilidade. Isto pode causar falhas de comunicação com o protótipo e não envio de alertas de temperatura.

Por fim, com base na hipótese proposta na introdução e nos resultados obtidos, conclui-se que é possível construir um dispositivo capaz de monitorar lotes de vacina,

capturando dados de temperatura e geolocalização, gravando-os em dispositivo de armazenamento e fornecendo-os para um dispositivo móvel em tempo real.

Para a realização de trabalhos futuros, recomenda-se a implementação de um módulo GSM e módulo GPS com maior sensibilidade ou equipados com antenas mais sensíveis, para ser capaz de sincronizar-se com a rede de telefonia e rede de satélites de dentro de locais mais fechados e assim não comprometer o monitoramento remoto. Outra recomendação é verificar o posicionamento do sensor digital dentro da caixa para que o dispositivo não sugira medições erradas, ou usar um sensor que usa termopares para melhor entrar em contato com o objeto a se medir.

REFERÊNCIAS

- ALECRIM, E. **Tecnologias 2G e 2,5G**: TDMA, CDMA, GSM, GPRS e EDGE. [S. l.], 2013. Disponível em: <<https://www.infowester.com/2g.php>>. Acesso em 15 jan. 2023.
- BOSCH SENSORTEC. **BMP280**: Data sheet. Reutlingen, 2018. Disponível em: <https://datasheet.lcsc.com/lcsc/2007021612_Bosch-Sensortec-BMP280_C83291.pdf>. Acesso em 9 fev. 2023.
- BRAGA, C. N. **Como Funcionam os Sensores**. 1. ed. São Paulo: Instituto Newton C Braga, 2021.
- BRASIL. MINISTÉRIO DA SAÚDE; AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA. **Resolução RDC nº 55, de 16 de dezembro de 2010**. Dispõe sobre o registro de produtos biológicos novos e produtos biológicos e dá outras providências. [S. l.], 2010. Disponível em: <https://bvsms.saude.gov.br/bvs/saudelegis/anvisa/2010/anexo/anexo_res0055_16_12_2010.pdf>. Acesso em 8 jan. 2023.
- BRASIL. MINISTÉRIO DA SAÚDE; SECRETARIA DE VIGILÂNCIA EM SAÚDE; DEPARTAMENTO DE VIGILÂNCIA DAS DOENÇAS TRANSMISSÍVEIS. **Manual de Rede de Frio do Programa Nacional de Imunizações**. 5. ed. Brasília: Ministério da Saúde, 2017. Disponível em: <https://bvsms.saude.gov.br/bvs/publicacoes/manual_rede_frio_programa_imunizacoes_5ed.pdf>. Acesso em 9 jan. 2023.
- BRASIL. MINISTÉRIO DA SAÚDE; SECRETARIA DE VIGILÂNCIA EM SAÚDE; DEPARTAMENTO DE IMUNIZAÇÃO E DOENÇAS TRANSMISSÍVEIS. **Plano Nacional de Operacionalização da Vacinação contra a Covid-19**. 2. ed. Brasília: Ministério da Saúde, 2022. Disponível em: <<https://www.gov.br/saude/pt-br/centrais-de-conteudo/publicacoes/publicacoes-svs/coronavirus/plano-nacional-de-operacionalizacao-da-vacinacao-contr-a-covid-19-pno-2a-edicao-com-isbn>>. Acesso em 9 jan. 2023.
- CHAPRA, S. C.; RAYMOND P. C. **Métodos Numéricos para Engenharia**. 5. ed. Porto Alegre: AMGH, 2011.
- EBERSPÄCHER, J. *et al.* **GSM – Architecture, Protocols and Services**. 3rd ed. United Kingdom: John Wiley & Sons Ltd, 2009.
- EL-RABBANY, A. **Introduction to GPS: The Global Positioning System**. Boston, Londres: Artech House, 2002.
- ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**. [S. l.], 2022. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 15 jan. 2023.
- FERREIRA, D. A. **Sistema de alarme integrando GPS e celular**. Brasília, 2010. Disponível em: <<https://repositorio.uniceub.br/jspui/handle/123456789/3336>>. Acesso em 16 fev. 2023.

FINKLER A. L.; FISCHER, D. Utilização Da Plataforma Gsm Em Um Protótipo De Medidor De Energia Elétrica Pré-Pago Em Conexão Com Redes Inteligentes. **Revista Científica Semana Acadêmica**, Fortaleza, n. 84, jun. 2016. Disponível em: <<https://semanaacademica.org.br/artigo/utilizacao-da-plataforma-gsm-em-um-prototipo-de-medidor-de-energia-eletrica-pre-pago-em>>. Acesso em 16 fev. 2023.

FRADEN, J. **Handbook of Modern Sensors: Physics, Designs, and Applications**. 5th. ed. New York: Springer, 2016.

GARCIA, F. D. **Introdução aos sistemas embarcados e microcontroladores**. [S. l.], 2018. Disponível em: <<https://embarcados.com.br/sistemas-embarcados-e-microcontroladores/>>. Acesso em: 13 fev. 2023.

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 6. ed. São Paulo: Atlas, 2019.

INSTITUTO BUTANTAN. **Imunização, uma descoberta da ciência que vem salvando vidas desde o século XVIII**. [S. l.], 2021. Disponível em <<https://butantan.gov.br/noticias/imunizacao-uma-descoberta-da-ciencia-que-vem-salvando-vidas-desde-o-seculo-xviii>>. Acesso em 8 jan. 2023.

LINDOMAR. **Sua conta requer atenção Uber – Motivos, Como Resolver**. [S. l.], 2020. Disponível em: <<https://motoristaelite.com/sua-conta-requer-atencao-uber/>>. Acesso em 16 jan. 2023.

NATIONAL COORDINATION OFFICE. **Control Segment**. [S. l.], 2021. Disponível em: <<https://www.gps.gov/systems/gps/control/>>. Acesso em 15 jan. 2023.

_____. **Space Segment**. [S. l.], 2022. Disponível em: <<https://www.gps.gov/systems/gps/space/>>. Acesso em 15 jan. 2023.

OLIVEIRA JÚNIOR, M.; DUARTE, R. O. **Introdução ao Projeto com Microcontroladores e Programação de Periféricos**. Minas Gerais, 2010. Disponível em: <<https://docplayer.com.br/656429-Apostila-sobre-introducao-ao-projeto-commicrocontroladores-e-programacao-de-perifericos.html>>. Acesso em: 20 jan. 2022.

PIMENTA, B. P. **Recurso Rádio em GSM/GPRS: GPRS**. [S. l.], 2006a. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialgprsaloc/pagina_3.asp>. Acesso em 16 jan. 2023.

_____. **Recurso Rádio em GSM/GPRS: GSM**. [S. l.], 2006b. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialgprsaloc/pagina_2.asp>. Acesso em 16 jan. 2023.

SANTOS, J. A. S. *et al.*, **Fatores determinantes das perdas das vacinas em Unidades Básicas de Saúde**. Id on Line Revista Multidisciplinar e de Psicologia, Nov-Dez. de 2016, vol.10, n.32, p. 41-53. ISSN: 1981-1179.

SD ASSOCIATION. **SD Standards Family**. [S. l.], 2000. Disponível em: <<https://www.sdcard.org/developers/sd-standard-overview/sd-family/>>. Acesso em 16 jan. 2023.

SD CARD ASSOCIATION. **Physical Layer Simplified Specification Version 9.00**. San Ramon, 2022. Disponível em: <https://www.sdcard.org/downloads/pls/pdf/?p=Part1_Physical_Layer_Simplified_Specificatio_n_Ver9.00.jpg&f=Part1_Physical_Layer_Simplified_Specification_Ver9.00.pdf&e=EN_SS1_9>. Acesso em 9 fev. 2023.

SHANGHAI SIMCOM WIRELESS SOLUTIONS LTD. **SIM800 Series AT Command Manual**. Xangai, 2015. Disponível em: <https://portal.vidadesilicio.com.br/wp-content/uploads/2018/05/SIM800-Series_AT-Command-Manual_V1.09.pdf>. Acesso em 20 jan. 2023.

UNIVERSIDADE ABERTA DO SUS. **PNI: entenda como funciona um dos maiores programas de vacinação do mundo**. [S. l], 2022. Disponível em: <<https://www.unasus.gov.br/noticia/pni-entenda-como-funciona-um-dos-maiores-programas-de-vacinacao-do-mundo>>. Acesso em 9 jan. 2023.

VARGAS, M. T. **Desenvolvimento de um data logger aplicado ao monitoramento de encostas**. Dissertação (Mestrado em Engenharia Civil e Ambiental) - Universidade de Passo Fundo, Passo Fundo, 2016. Disponível em: <<http://tede.upf.br/jspui/handle/tede/1416>>. Acesso em 16 fev. 2023.