



CURSO DE ENGENHARIA MECATRÔNICA

**RUBEM DIEGO DE MELO SILVA**

**SISTEMA DE CONTROLE DE TEMPERATURA PARA CARGAS  
CRÍTICAS COM MONITORAMENTO VIA GPS USANDO  
CONEXÃO GPRS**

Recife, PE  
2021

**RUBEM DIEGO DE MELO SILVA**

**SISTEMA DE CONTROLE DE TEMPERATURA PARA CARGAS  
CRÍTICAS COM MONITORAMENTO VIA GPS USANDO  
CONEXÃO GPRS**

Trabalho de Conclusão de Curso desenvolvido como pré-requisito para obtenção do título de Engenheiro Mecatrônico pelo Centro universitário Tiradentes (UNIT-PE).

Orientador: Prof. Ms. Pedro Henrique Meira de Araújo

Recife, PE  
2021

**RUBEM DIEGO DE MELO SILVA**

**SISTEMA DE CONTROLE DE TEMPERATURA PARA CARGAS  
CRÍTICAS COM MONITORAMENTO VIA GPS USANDO  
CONEXÃO GPRS**

---

Orientador

---

Avaliador 1

---

Avaliador 2

Recife, \_\_\_\_ de \_\_\_\_\_ de 2021.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por me abençoar sempre com luz e paz e me dar forças nas horas mais difíceis.

À minha mãe Ana Gláucia, por ser meu porto seguro e minha fonte de ensinamentos morais.

Ao meu padrasto Sandro Machado pela ajuda e prestatividade.

À Luis Felipe, por ceder conhecimento e materiais necessários para a conclusão deste trabalho, além de toda a sua gentileza e paciência.

À Ezequiel Reis e Diogo Moraes, pela amizade e prestatividade.

À Ivanilda Melo, também por ceder seu conhecimento e me ajudar na consolidação deste trabalho.

À empresa Tomus Soluções em Eletrônica e Telecomunicações e colegas de trabalho, por todos os ensinamentos adquiridos e compreensão.

Finalmente agradeço a todos os professores pelos ensinamentos recebidos durante o curso.

*“Desenvolver força, coragem e paz interior demanda tempo. Não espere resultados rápidos e imediatos, sob o pretexto de que decidiu mudar. Cada ação que você executa permite que essa decisão se torne efetiva dentro de seu coração.”*

*Dalai Lama*

## RESUMO

SILVA, Rubem Diego de Melo. Sistema de Controle de Temperatura para Cargas Críticas com Monitoramento via GPS Usando Conexão GPRS. Monografia (Graduação em Engenharia Mecatrônica – Centro Universitário Tiradentes (UNIT), Recife, 2021).

Em vista da necessidade de um rigoroso controle de temperatura para serem utilizadas no transporte de cargas críticas, foi desenvolvido um sistema que permitirá o acondicionamento dessas cargas sob uma temperatura controlada com precisão. Assim como, este sistema irá auxiliar na gestão do transporte e armazenamento dessas cargas de alto valor cuja estabilidade térmica deve ser cuidadosamente controlada. Neste trabalho, apresento como foi o desenvolvimento desse sistema, no qual, consistirá a priori, no controle e monitoramento de temperatura de uma caixa térmica comercial. Também nesse sistema será possível efetuar configurações, registrar parâmetros e permitir geração de alarmes em caso de anormalidades. Esse sistema também irá informar os dados a uma página *WEB* a localização global da carga crítica a ser transportada, com as informações indicadas em tempo real. Neste trabalho também serão conceituados elementos de Sistemas Embarcados, além de apresentar o estudo dessa aplicação, mostrando os seus respectivos resultados.

**Palavras-chave:** transporte de cargas críticas; temperatura controlada; sistemas embarcados; página *WEB*; localização global.

## ABSTRACT

In view of the need for strict temperature control to be used in the transport of critical loads, a system was developed that will allow these loads to be stored under a precisely controlled temperature. As well, this system will assist in the management of transport and storage of these high value loads whose thermal stability must be accumulated. In this work, a present how was the development of this system, which will consist, first, in the control and monitoring of the temperature of a commercial thermal box. This system will also be able to make configurations, record parameters and allow the generation of alarms in case of abnormalities. This system will also inform the data to a WEB page the global location of the critical load to be transported, with the information indicated in real time. In this work, elements of Embedded Systems will also be conceptualized, in addition to presenting the study of this application, showing their respective results.

**Keywords:** transport of critical loads; controlled temperature; embedded systems; Web page; global location.

## LISTA DE FIGURAS

Figura 1 - Diagrama Simplificado de Máquina Térmica.....	18
Figura 2 - Diagrama Simplificado de um Refrigerador a gás.....	19
Figura 3 - Pastilha Termoelétrica ou Pastilha Peltier.....	21
Figura 4 - Esquema Simplificado Efeito Seebeck-Peltier .....	22
Figura 5 - Esquemática interna de uma Pastilha Peltier .....	23
Figura 6 – Respostas do Controle Proporcional para diferentes $K_p$ .....	31
Figura 7 - Respostas do Controle Integral para diferentes $K_i$ .....	32
Figura 8 - Respostas do Controle Derivativo para diferentes $K_d$ .....	33
Figura 9 - Caixa térmica simples da Soprano 12L.....	60
Figura 10 - Caixa térmica com medidor de temperatura comercial .....	61
Figura 11 - Dissipador de Calor (Alumínio). .....	62
Figura 12 - Exaustor para auxiliar a refrigeração da Pastilha Peltier.....	62
Figura 13 - Conjunto Refrigeração. ....	62
Figura 14 - Parte interna - Instalação Pastilha Peltier .....	63
Figura 15 - Parte externa - Instalação Dissipador e Ventoinha .....	63
Figura 16 - Diagrama de Contexto - Unidade Controle .....	64
Figura 17 - Caixa Térmica com IHM frontal.....	65
Figura 18 - Sensor LM35 da Texas Instruments. ....	65
Figura 19 - Arduino Nano com Microcontrolador ATMEGA328p.....	66
Figura 20 - Display LCD alfanumérico 16 caracteres e 2 linhas.....	66
Figura 21 - LED como indicador visual.....	67
Figura 22 - Buzzer como indicador sonoro.....	67
Figura 23 - Micro Chave tipo tátil.....	67
Figura 24 - Esquemática interna de um Relé eletromecânico .....	68
Figura 25 - Relé Metaltex para acionar a pastilha Peltier .....	68
Figura 26 - Exemplo de Programação feita no ambiente do Arduino .....	69
Figura 27 - Montagem da Unidade Central em Placa Padrão .....	70
Figura 28 - Testes Iniciais com a Unidade de Controle.....	70
Figura 29 - Montagem Integração Unidade de Refrigeração e Controle .....	71
Figura 30 - Diagrama de Contexto - Unidade de Rastreamento .....	72
Figura 31 - Arduino PRO Mini com microcontrolador ATMEGA2560 .....	72
Figura 32 - Módulo GPS uBlox Neo 6m com antena.....	73
Figura 33 - Modem SIM800L.....	74
Figura 34 - Antena para modem GSM/GPRS 3dBi .....	74
Figura 35 - LOG da Unidade de Rastreamento.....	75
Figura 36 - Protótipo da Unidade de Rastreamento .....	76
Figura 37 - Página WEB que receberá pacote de mensagens .....	77
Figura 38 - Tela do Banco de Dados.....	77
Figura 39 - Exemplo real da página inicial .....	78
Figura 40 - Simulação da Unidade Central usando o programa Proteus. ....	79
Figura 41 - Exemplo de Trajeto sendo mostrado no Sistema WEB .....	81
Figura 42 - Primeiras medições de Temperatura - Sistema WEB .....	81

Figura 43 - Teste de validação do Sensor e Leituras pelo Sistema WEB .....	82
Figura 44 - Segunda etapa de testes com Carga Fria.....	83

## SUMÁRIO

1. INTRODUÇÃO.....	13
2. OBJETIVOS.....	16
2.1. OBJETIVOS GERAL.....	16
2.2. OBJETIVOS ESPECÍFICOS.....	16
3. FUNDAMENTAÇÃO TEÓRICA.....	17
3.1. PRINCÍPIOS DA TERMODINÂMICA.....	17
3.2. MÁQUINAS TÉRMICAS.....	18
3.2.1. Conceito Básico de Máquinas Térmicas.....	18
3.3. EFEITO PELTIER.....	21
3.3.1. Conceito Básico do Efeito Peltier.....	21
3.4. SISTEMAS EMBARCADOS.....	24
3.4.1. Conceito sobre Sistemas Embarcados.....	25
3.4.2. Interfaces do Utilizador.....	25
3.4.3. Processamento.....	26
3.4.4. Periféricos.....	26
3.4.5. Exemplos de Utilização de Sistemas Embarcados:.....	26
3.5. SISTEMAS DE CONTROLE.....	27
3.5.1. Controles Digitais.....	28
3.5.2. Controles Analógicos.....	28
3.5.3. Controle PID.....	29
3.6. TELEMETRIA.....	34
3.6.1. Conceito de Telemetria.....	34
3.6.2. Exemplos de Telemetria.....	35
3.7. GENERAL PACKET RADIO SERVICE ( <i>GPRS</i> ).....	37
3.7.1. Conceito Sobre <i>GPRS</i> .....	37
3.7.2. Principais vantagens da Utilização do <i>GPRS</i> .....	38
3.7.3. Disponibilidade imediata.....	38
3.7.4. Acesso ao serviço <i>GPRS</i> .....	39
3.7.5. Características principais da rede <i>GPRS</i> .....	39
3.7.6. Compatibilidade com a Internet.....	40
3.7.7. Suporte a <i>TDMA</i> e <i>GSM</i> .....	41
3.8. GLOBAL POSITIONING SYSTEM ( <i>GPS</i> ).....	41
3.8.1. Conceito Sobre <i>GPS</i> .....	41

3.8.2.	Funcionamento do <i>GPS</i> .....	41
3.8.3.	Precisão do <i>GPS</i> .....	42
3.8.4.	Sistema de Satélite <i>GPS</i> .....	43
3.8.5.	Sinal <i>GPS</i> .....	43
3.8.6.	Fatores de erros do <i>GPS</i> .....	44
3.9.	PÁGINA <i>WEB</i> E <i>PHP</i> .....	45
3.9.1.	Conceito de Página <i>WEB</i> .....	45
3.9.2.	Conceito do <i>PHP</i> .....	46
3.9.3.	Diferenças entre <i>PHP</i> e <i>JavaScript</i> .....	46
3.9.4.	Vantagens de Utilizar <i>PHP</i> .....	46
3.9.5.	Ambiente de Utilização do <i>PHP</i> .....	47
3.10.	BANCO DE DADOS E <i>SQL</i> .....	48
3.10.1.	Conceito Banco de Dados .....	49
3.10.2.	Conceito Sobre <i>SQL</i> .....	49
3.10.3.	Tipos de Banco de Dados .....	50
3.10.4.	Software Banco de Dados .....	52
3.10.5.	Gerenciamento de Banco de Dados .....	52
3.10.6.	Exemplos de Banco de Dados .....	52
3.10.7.	Banco de Dados <i>MySQL</i> .....	53
3.11.	USO DE <i>API</i> 's .....	53
3.11.1.	Conceito de <i>API</i> : .....	53
3.11.2.	Vantagens de utilizar <i>API</i> 's .....	54
3.11.3.	Inovação com <i>API</i> .....	55
3.11.4.	<i>API</i> 's remotas .....	56
3.11.5.	<i>API</i> 's da Google .....	57
3.11.6.	<i>API</i> do <i>Google Maps</i> .....	57
4.	METODOLOGIA .....	60
4.1.	UNIDADE DE REFRIGERAÇÃO .....	60
4.1.1.	Caixa Térmica .....	60
4.1.2.	Pastilha de <i>Peltier</i> .....	61
4.1.3.	Dissipador de Calor e Ventoinha .....	61
4.1.4.	Montagem do Protótipo – Unidade de Refrigeração .....	62
4.2.	Unidade Central de Controle .....	63
4.2.1.	Sonda – Sensor de Temperatura .....	65
4.2.2.	Microcontrolador <i>ATMEGA328P</i> .....	66

4.2.3.	Display <i>LCD</i> .....	66
4.2.4.	Alertas Visuais e Sonoros.....	66
4.2.5.	Botões de Acionamento.....	67
4.2.6.	Relé de Acionamento .....	67
4.2.7.	Reguladores de Tensão .....	68
4.2.8.	Fonte de Alimentação.....	68
4.2.9.	Desenvolvimento do <i>Firmware</i> – Unidade de Controle.....	68
4.2.10.	Montagem do Protótipo – Unidade de Controle .....	69
4.3.	UNIDADE DE RASTREAMENTO .....	71
4.3.1.	Microcontrolador ATMEGA2560.....	72
4.3.2.	Módulo GPS .....	73
4.3.3.	Modem <i>GPRS</i> .....	73
4.3.4.	Desenvolvimento do <i>Firmware</i> – Unidade Rastreamento.....	74
4.3.5.	Montagem do Protótipo – Unidade de Rastreamento.....	76
4.4.	Sistema <i>WEB</i> para este dispositivo.....	76
5.	RESULTADOS E DISCUSSÃO .....	79
6.	CONSIDERAÇÕES FINAIS.....	84
7.	REFERÊNCIAS .....	85

## 1. INTRODUÇÃO

Na cadeia de logística destacam-se a cadeia de transporte de frios, responsável pela condução de produtos perecíveis e/ou críticos que precisam ser mantidos em ambientes de temperatura e umidade controlados. Nesta cadeia destaca-se a indústria farmacêutica associada a rede hospitalar com grandes desafios no transporte adequado de vacinas, biomedicamentos, hemoderivados, órgãos e outros produtos que necessitam de controle de temperatura com alta precisão.

Contudo, há grandes desafios no transporte de produtos de alto valor agregados, assim as entidades também sofrem com a carência em infraestrutura e adoção e manutenção de boas práticas em todos os pontos da cadeia logística.

A Organização Mundial de Saúde – OMS (2015) estima que aproximadamente 50% das vacinas produzidas em todo o mundo atinge seu destino deteriorada, devido a quebra da cadeia de frio. A maior falha está no controle da temperatura (cerca de 40%), comprometendo a qualidade do produto e causando possíveis riscos à saúde. (SANCHES, C. 2015).

Para o transporte eficiente são necessários equipamentos e insumos adequados, como local de armazenamento esterilizado e certificado; controle de temperatura; espaço suficiente para o armazenamento de todos os produtos de forma a garantir que eles estejam sendo refrigerados homoganeamente; utilização de caixas térmicas (isopor ou poliestileno) resistentes, em bom estado e que sejam capazes de conservar a temperatura interna considerando a temperatura externa e o tempo de transporte; termômetros ou *Datalogers* para monitoramento da temperatura durante todo o transporte auxiliando em uma tomada de decisão rápida durante o percurso, registrando desvios de temperatura e provendo meios para a identificação da causa raiz desse desvio.

“Não menos importante do que os equipamentos e insumos são as pessoas envolvidas na cadeia de frio. Elas precisam estar capacitadas e ter conhecimento sobre o que é cadeia de frio, qual sua importância, quais insumos e equipamentos devem ser utilizados e como os produtos devem ser armazenados e transportados”, comenta Mayra Moura, colaboradora da Assessoria Clínica de Bio-Manguinhos (ASCLIN, 2021).

Quando as normas de armazenamento e transporte não são cumpridas, o produto fica exposto à perda da eficácia ao longo de sua vida útil, chegando a haver a necessidade de destruí-lo. O quanto de potência – ou eficácia – ele perde depende de vários fatores: a qual temperatura foi exposta? Por quanto tempo? Qual a validade deste produto? Ele já foi submetido anteriormente a um desvio de temperatura? Outro problema que pode ocorrer é o aumento de eventos adversos (MOURA, 2015).

Da fabricação do produto até sua chegada ao consumidor final, são rígidas as normas que devem ser cumpridas para garantir que a qualidade seja mantida satisfatória. Como apontado no portal da Anvisa (2015), no Brasil, a norma RDC66/99, por exemplo, visa melhorar significativamente as condições sanitárias de transporte dos órgãos humanos destinados a transplantes, desde a captação até a chegada aos milhares de pacientes que estão na fila de espera por um transplante no Brasil.

A norma RDC 38/10, que trata da importação de produtos biológicos, determina, no parágrafo seis, que a empresa importadora (detentora do registro) deve possuir registros contínuos de temperatura da cadeia de transporte que comprovem que o produto foi mantido dentro das condições de armazenamento e de transporte preconizadas pelo fabricante. Os registros de temperatura devem identificar o nome do produto, número de lote, hora e data de envio e recepção.

A situação com as vacinas atualmente, por exemplo, não é muito diferente desta relatada em 2015. Ainda mais quando a pesquisa foi refeita no momento, em 2021, plena época da pandemia ocasionado pela Covid-19. A situação continua a mesma.

De acordo com Emerson Lopes, gerente regional de Supply Chain e S&OP da Alcon Laboratórios, o problema está na falta de padronização, descumprimento das exigências legais de prestadores de serviços e falta de efetividade das autoridades (VACINAS, 2021).

A indústria é capaz de controlar os requisitos técnicos até certo ponto, porém não é possível garantir a manutenção das condições de armazenagem exigidas por lei em cidades com poucos recursos, por exemplo. Para isso, devemos definir estratégias pertinentes para o Brasil e o mundo. “A infraestrutura logística em países desenvolvidos permite uma distribuição infinitamente mais eficiente. Além disso, as condições climáticas são mais extremas nos outros

países, o que faz com que a infraestrutura deles seja naturalmente melhor estabelecida que a nossa. A falta de um sistema de transporte multimodal eficiente também limita muito as alternativas de transporte, deixando o modelo limitado a rodoviário e aéreo. Pensando em abrangência nacional, a falta de estradas de boa qualidade e de armazéns com as adequações exigidas pela cadeia farmacêutica limitam ainda mais nossas opções, incrementando assim o custo e a complexidade deste controle” – Explica Lopes (VACINAS, 2021).

Neste trabalho, será apresentado uma solução que pode auxiliar na demanda da cadeia logística na cadeia de frios, que poder ser acompanhada por profissionais do ramo e também poderá servir para diversas outras aplicações. Usando a tecnologia de IoT (*Internet of Things*), sistemas de geolocalização, sistemas de controle e armazenamento de dados como ferramentas aliadas durante o processo de logística da cadeia de frios, será possível além de facilidade de implementação e usabilidade, transpor informações necessárias para corrigir quaisquer eventualidades anormais e assim, auxiliar nas tomadas de decisões necessárias de modo que evitem perdas e desperdícios de um bem tão valioso.

## 2. OBJETIVOS

### 2.1. OBJETIVOS GERAL

Construir um sistema fechado e controlado que se propõe condicionar materiais de alta criticidade mantendo a temperatura estável com precisão e rapidez, além de transpor essas informações, o sistema irá coletar informações de configurações, alarmes e posicionamento global para que possam ser mostradas em uma página *WEB*.

### 2.2. OBJETIVOS ESPECÍFICOS

- Estabelecer o escopo do problema e o ambiente acerca do transporte de cargas críticas;
- Montar e testar um sistema de refrigeração baseado no Efeito *Peltier*;
- Montar e testar um sistema de controle de temperatura baseado na técnica de Controle *PID* (Proporcional, Integral e Derivação);
- Montar e testar um sistema de geolocalização usando um módulo *GPS* comercial;
- Testar conexão com um modem *GPRS* com redes de telefonia 2G para aplicação em telemetria;
- Construir uma página *WEB* usando *PHP* para receber dados gerados dos sistemas de rastreamento e controle de temperatura;
- Testar a utilização do Banco de Dados e *SQL* para armazenamento dos dados gerados no sistema *WEB*;
- Testar a utilização de *API* da *Google Maps* no sistema *WEB*, para auxiliar na visualização do posicionamento global da carga em um mapa dinâmico.
- Mostrar uma aplicação de um *Datalog* no sistema *WEB* para auxiliar no acompanhamento da temperatura da carga crítica.

### 3. FUNDAMENTAÇÃO TEÓRICA

#### 3.1. PRINCÍPIOS DA TERMODINÂMICA

O princípio básico para o acondicionamento das cargas críticas e decompor o funcionamento do Sistema de Refrigeração, é controlar na prática a segunda Lei da Termodinâmica.

A segunda lei da termodinâmica ou segundo princípio da termodinâmica expressa, de uma forma concisa, diz que "A quantidade de entropia (equação 1) de qualquer sistema isolado termodinamicamente tende a incrementar-se com o tempo, até alcançar um valor máximo". Mais sensivelmente, quando uma parte de um sistema fechado interage com outra parte, a energia tende a dividir-se por igual, até que o sistema alcance um equilíbrio térmico (VAN WYLEN; SONNTAG; BORGNAKKE, 1998).

$$\Delta S = S_2 - S_1 \geq \int_1^2 \left(\frac{\delta Q}{T}\right) \quad (1)$$

Onde  $S_2$  e  $S_1$  são os estados de entropia final e inicial respectivamente.  $Q$  é o calor dado em *Joule* e  $T$  é a temperatura dado em escala *Kelvin*.

Enquanto a primeira lei da termodinâmica estabelece a conservação de energia em qualquer transformação, a segunda lei estabelece condições para que as transformações termodinâmicas possam ocorrer. A transformação de calor em trabalho por um processo cíclico exige a presença de duas fontes térmicas mantidas a temperaturas diferentes entre si (VAN WYLEN; SONNTAG; BORGNAKKE, 1998).

A segunda Lei da Termodinâmica foi proposta a meio do século XIX, por Clausius e Lorde Kelvin, para expressar a limitação encontrada por Carnot. Clausius escreveu "Sobre uma forma modificada do segundo teorema fundamental da teoria mecânica do calor" enquanto Kelvin foi o autor de "Sobre a teoria dinâmica do calor" (VAN WYLEN; SONNTAG; BORGNAKKE, 1998).

Num sentido geral, a segunda lei da termodinâmica afirma que as diferenças entre sistemas em contato tendem a igualar-se. As diferenças de pressão, densidade e, particularmente, as diferenças de temperatura tendem a equalizar-se. Isto significa que um sistema isolado chegará a alcançar uma

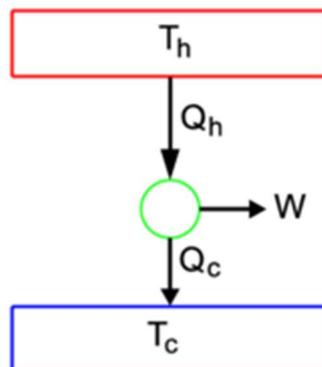
temperatura uniforme. Uma máquina térmica é aquela que provê trabalho graças à diferença de temperatura entre dois corpos. Dado que qualquer máquina termodinâmica requer uma diferença de temperatura, se deriva pois que nenhum trabalho útil pode extrair-se de um sistema isolado em equilíbrio térmico, isto é, requererá de alimentação de energia do exterior. A segunda lei se usa normalmente como a razão pela qual não se pode criar uma máquina de movimento perpétuo ou moto contínuo (VAN WYLEN; SONNTAG; BORGNAKKE, 1998).

### 3.2. MÁQUINAS TÉRMICAS

#### 3.2.1. Conceito Básico de Máquinas Térmicas

Em termodinâmica, máquinas térmicas são usualmente projetadas utilizando um modelo padrão, como o Ciclo Otto. O modelo teórico pode ser refinado e ampliado com dados reais de um motor em funcionamento, usando ferramentas como um diagrama indicador. Na Figura 1 a seguir, mostra o exemplo simplificado de como funciona uma máquina térmica:

Figura 1 - Diagrama Simplificado de Máquina Térmica



Fonte: <https://pt.wikipedia.org>, 2021.

Desde que o modelo padrão foi criado, poucas mudanças foram implementadas, podendo-se dizer que um ciclo termodinâmico é um caso ideal de um motor mecânico. Em termos gerais, quanto maior for a diferença na temperatura entre a fonte de calor e o dissipador frio, maior é a eficiência térmica potencial do ciclo. A temperatura baixa da máquina térmica está limitada a

temperatura do ambiente, ou seja, não muito inferior a 300 Kelvin. Por isso, a maioria dos esforços para melhorar as eficiências termodinâmicas das máquinas térmicas tem como foco o aumento da temperatura de origem, dentro dos limites dos materiais. A eficiência teórica máxima de uma máquina térmica é igual à diferença de temperatura entre as extremidades quentes e frias, dividida pela temperatura na extremidade quente, todas expressas em graus Kelvin, ou a temperatura absoluta. (LANGSTON, Lee.S., 2008).

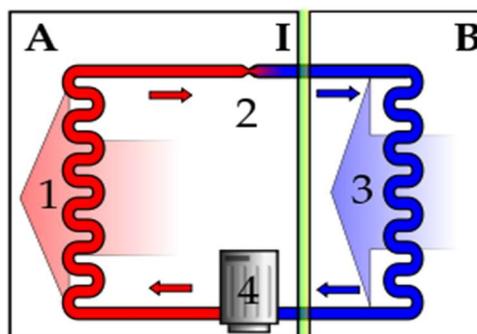
Máquinas térmicas podem ser caracterizadas pela sua potência específica, que é normalmente dada em quilowatts (KW) ou em cavalo-vapor (cv). Essa característica oferece uma aproximação da saída de pico de potência de um motor. Isto não deve ser confundido com a eficiência de combustível, uma vez que a eficiência é elevada, muitas vezes requer uma relação de ar-combustível pobre, e, portanto, menor quantidade de potência. (HALLIDAY, D., RESNICK, R., WALKER, J., 2010).

O rendimento das máquinas térmicas (equação 2) pode ser, de uma maneira geral, a razão entre o trabalho total e o trabalho (ou calor) necessário para que ela funcione, ou seja, é o que se obtém pelo que se dá de trabalho. (HALLIDAY, D., RESNICK, R., WALKER, J., 2010):

$$\eta = \frac{W_t}{Q} \quad (2)$$

Como exemplo de aplicação prática, poderíamos citar o funcionamento de um refrigerador a gás. O processo de refrigeração é um ciclo térmico, ilustrado na Figura 2:

Figura 2 - Diagrama Simplificado de um Refrigerador a gás



Fonte: <https://pt.wikipedia.org/>, 2021.

A seguir serão descritas as fases de funcionamento de um refrigerador convencional a gás:

- Fase 1: O hidrofluorcarboneto (*HFC*) é o gás atualmente utilizado em refrigeradores e ar-condicionado. É ele quem faz o ar dentro da geladeira esfriar, graças a troca de calor por condução e por convecção. Inicialmente esse gás percorre a serpentina exterior da geladeira, também chamada de radiador, para que fique à temperatura ambiente. Essa serpentina geralmente fica atrás do equipamento e auxiliará na dissipação térmica do gás.

- Fase 2: Na segunda fase do ciclo o gás passa por um pequeno capilar e em seguida entra em uma tubulação com um volume maior do que o pequeno capilar onde se encontrava. Essa mudança de volume diminui a pressão do gás e sua temperatura, deixando a tubulação gelada. Essa relação entre pressão, temperatura e volume é conhecida como equação de estado.

- Fase 3: Já resfriado o gás passa por uma outra serpentina, deixando o material mais gelado. É nessa parte que a geladeira começa a gelar: o ar troca calor com a serpentina por condução e fica mais denso que o resto do ar da geladeira. Esse ar mais pesado troca de lugar com o ar quente por convecção e o ar quente entra em contato com a serpentina, tornando-se frio também. Essa serpentina precisa estar na parte mais alta da geladeira para que o processo de convecção ocorra e a temperatura interna fique aproximadamente igual.

- Fase 4: É na quarta fase do processo que realizamos trabalho sobre o sistema, ou seja, fazemos exatamente o inverso que ocorre com as máquinas térmicas. Um motor elétrico é utilizado para comprimir o gás gelado e fazer com que ele volte a tubulação externa a uma pressão muito maior do que a anterior. Ao elevar a pressão também aumentamos a temperatura, ou seja, o gás sai mais quente do que estava dentro da geladeira. Além de estar mais quente do que antes, ele precisa estar mais quente do que a temperatura ambiente, para que o sistema dissipe perca calor. Além de elevar a temperatura, o compressor empurra o gás para que o ciclo continue. (Wikipédia, 2021).

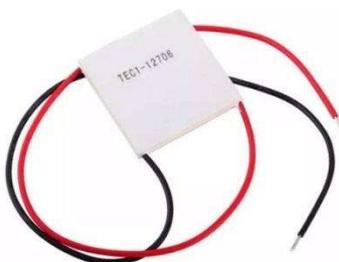
### 3.3. EFEITO PELTIER

#### 3.3.1. Conceito Básico do Efeito Peltier

Com base descrita sobre a segunda lei da termodinâmica, assim como, o funcionamento sobre máquinas térmicas e refrigeradores, o intuito é usar um dispositivo alternativo que pudesse também manter a temperatura constante no interior de uma caixa térmica, onde estaria o material crítico. Entretanto, devido à praticidade e facilidade de instalação, ao mesmo tempo que pudesse facilitar o transporte do material, foi pensado utilizar pastilhas *Peltier* ou Módulos *Peltier*.

Módulos *Peltier*, também conhecidos como pastilhas termoelétricas (Figura 3), são pequenas unidades de que utilizam tecnologia de matéria condensada para operarem como bombas de calor. Uma unidade típica tem espessura de alguns milímetros e forma quadrada (dimensões: 40x40x4 mm). Esses módulos são essencialmente um sanduíche de placas cerâmicas recheado com pequenos cubos de Bi<sub>2</sub>Te<sub>3</sub> (telureto de bismuto). (PELTIER, 2021).

Figura 3 - Pastilha Termoelétrica ou Pastilha Peltier

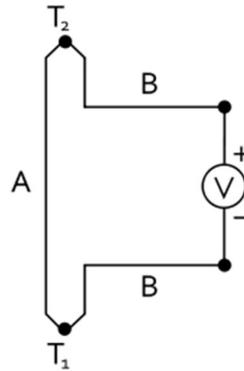


Fonte: [peltier.com.br/](http://peltier.com.br/), 2021.

Sua operação é baseada no Efeito *Peltier*, que foi descoberto em 1834. Quando uma corrente é aplicada, o calor move de um lado ao outro (onde ele deve ser removido com um dissipador). Se os pólos elétricos forem revertidos, a pastilha se tornará em um excelente aquecedor. É importante salientar que por mais tecnologicamente avançados que sejam, os módulos não “consomem” calor – por isso que se torna necessário o uso do dissipador. A Figura 4 é

mostrado um esquema simplificado ou modelo elétrico simplificado de um módulo *Peltier* de uso comercial. (PELTIER, 2021).

Figura 4 - Esquema Simplificado Efeito *Seebeck-Peltier*



Fonte: <https://pt.wikipedia.org/>, 2021.

A energia térmica dissipada/absorvida, conforme visto na equação 3, é proporcional à corrente elétrica que percorre o sistema, sendo possível assim definir o calor associado pelo efeito da seguinte forma:

$$Q_p = \pi \times I \quad (3)$$

Onde  $Q_p$  é o calor associado,  $\pi$  é o coeficiente *Peltier* e  $I$  é a corrente elétrica.

Pastilhas termoelétricas são utilizadas em aplicações pequenas de resfriamento como chips microprocessadores ou até médias como geladeiras portáteis. Atualmente, os módulos mais potentes podem transferir um máximo de 250W, tornando a tecnologia inviável para o uso em um aparelho de ar condicionado, por exemplo. As pastilhas podem ser empilhadas para se chegar a temperaturas mais baixas, embora alcançar níveis criogênicos requer processos muito complexos.

A grande vantagem de pastilhas do tipo *Peltier* são a ausência de peças móveis, gás freon, barulho e vibração; Além do tamanho reduzido, alta durabilidade e precisão. Elas são utilizadas hoje em inúmeros setores, principalmente os de bens de consumo, automotivo, industrial e militar. (PELTIER, 2021). A Figura 5 a seguir mostra a estruturação interna de uma Pastilha de *Peltier*.

Figura 5 - Esquemática interna de uma Pastilha *Peltier*



Fonte: <https://pt.wikipedia.org/>, 2021.

Conforme visto anteriormente na Figura 3, onde é mostrado uma pastilha *Peltier* comercial, no qual, possui formato retangular e achatado. Na Figura 5 é mostrado a constituição dessa pastilha. Ela é formada basicamente de duas "chapas" de material isolante (normalmente cerâmico) com uma malha de material condutor (cobre, por exemplo) na superfície interna de cada chapa. Entre as duas malhas de condutores, estão localizados diversos pares de semicondutores de tipo "n" e "p", que dão início ao efeito *Peltier*, transformando energia elétrica, proveniente de uma fonte D.C. (corrente direta ou polarizada), em energia térmica e, graças ao posicionamento e ordenação dos pares, absorvendo calor em uma chapa e dissipando calor em outra.

Os pares de semicondutores "tipo-n" e "tipo-p" possuem propriedades específicas que, se posicionados de maneira correta, permitem direcionar adequadamente o fluxo de elétrons e calor por entre seus terminais positivos e negativos. Em um semicondutor do tipo-n, o calor é absorvido próximo ao terminal negativo e rejeitado próximo ao terminal positivo, já em um semicondutor do tipo-p o processo se dá de maneira inversa, absorvendo calor próximo ao terminal positivo e rejeitando-o próximo ao terminal negativo. Dessa maneira, é possível perceber que, arranjando ambos semicondutores em pares ordenados, seus efeitos se completam, ampliando sua magnitude e continuidade.

Em uma pastilha de *Peltier*, se faz extremamente necessário utilizar dissipadores na chapa que irá fornecer calor ao meio, isso porque caso o calor gerado não consiga ser dissipado para o ambiente, será então para a própria pastilha, encaminhando o sistema a um equilíbrio térmico, reduzindo drasticamente a vida útil da pastilha. Em alguns projetos é possível encontrar

pastilhas empilhadas convergindo seus lados que dissipam calor com o lado absorvente de outra pastilha, com isso, é possível ampliar o poder de refrigeração na primeira pastilha. (PELTIER, 2021).

### 3.4. SISTEMAS EMBARCADOS

Antes de conceituar Sistemas Embarcados, é importante observar os significados de termos chaves como:

- *Hardware*: São dispositivos físicos como: discos magnéticos, mouse, impressora, monitores de vídeo, memórias, processadores ou circuitos eletrônicos cuja interligação é chamado de um sistema computacional (MACHADO; MAIA, 2007);
- *Software*: Conjunto de programas utilizado como interface entre o usuário e o hardware para que este tenha utilidade prática (MACHADO; MAIA, 2007);
- *Firmware*: Em eletrônica e computação, *firmware* é uma classe específica de *software* de computador que fornece controle de baixo nível para o *hardware* específico do dispositivo. O *firmware* pode fornecer um ambiente operacional padronizado para o *software* mais complexo do dispositivo (permitindo maior independência de *hardware*) ou, para dispositivos menos complexos, atuar como o sistema operacional completo do dispositivo, executando todas as funções de controle, monitoramento e manipulação de dados. Exemplos típicos de dispositivos que contêm firmware são sistemas embarcados. (Wikipédia, 2021);
- *Processador*: Também denominado de *Central Process Unit (CPU)*, gerencia todo o sistema computacional controlando e executando instruções presentes na memória principal através de operações básicas como: soma, subtração, comparação e movimentação de dados (MACHADO; MAIA, 2007);
- *Interrupções*: Evento externo que provoca a interrupção da execução do programa para a execução de um bloco de código chamado rotina de interrupção, pode ser decorrente de operações E/S (Entradas e Saídas) ou acontecimentos de periféricos (HIRAKAWA, 2004).

### 3.4.1. Conceito sobre Sistemas Embarcados

Um sistema embarcado (ou sistema embutido) é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. (SANTOS, G; BERTOLETI, P., 2021). Diferentemente de computadores de propósito geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos. Já que o sistema é dedicado a tarefas específicas, através de engenharia pode-se otimizar o projeto reduzindo tamanho, recursos computacionais e custo do produto. (SHIBU, K., 2009).

Em geral, tais sistemas não podem ter sua funcionalidade alterada durante o uso. Caso queira-se modificar o propósito é necessário reprogramar todo o sistema.

Sistemas embarcados são desenvolvidos para uma tarefa específica. Por questões como segurança e usabilidade, alguns inclusive possuem restrições para computação em tempo real. O *software* escrito para sistemas embarcados é muitas vezes chamado *firmware*, e armazenado em uma memória *ROM* (*Read Only Memory*) ou memória *Flash* ao invés de um disco rígido. Por vezes o sistema também é executado com recursos computacionais limitados: Sem teclado, sem tela e com pouca memória. (Wikipédia, 2021).

### 3.4.2. Interfaces do Utilizador

Sistemas embarcados podem possuir desde nenhuma interface do utilizador (dedicados somente a uma tarefa) a uma interface de utilizador completa, similar a dos sistemas operacionais *desktop*. Sistemas mais simples utilizam botões, *LED's* (diodos emissores de luz) ou telas bastante limitadas, geralmente mostrando somente números ou uma fila pequena de caracteres.

Sistemas mais complexos utilizam uma tela gráfica completa, usando tecnologias como tela tátil ou aquela em que o significado dos botões depende do contexto da tela. Computadores de mão também oferecem manípulos eletrônicos para apontar. O surgimento da *World Wide Web* forneceu aos desenvolvedores de sistemas embarcados a possibilidade de fornecer uma interface web através de uma conexão por rede. Isso evita o custo de uma tela

sofisticada, ainda que seja fornecida uma interface complexa e completa a ser acessada em outro computador. De modo geral, roteadores usam tal habilidade.

#### 3.4.3. Processamento

Em geral os sistemas embarcados possuem uma capacidade de processamento reduzida em comparação com computadores desktops. Ao invés de utilizar microprocessadores, os desenvolvedores preferem utilizar microcontroladores, pois estes já possuem diversos periféricos integrados no mesmo chip.

Outra diferença é a variedade de arquiteturas disponíveis tais como ARM, MIPS, Coldfire/68k, PowerPC, x86, PIC, 8051, Atmel AVR, Renesas H8, SH, V850, FR-V, M32R, Z80 e Z8. Isso contrasta com o mercado de computadores pessoais, limitados à somente algumas arquiteturas. (MOYER, B., 2013)

#### 3.4.4. Periféricos

Os sistemas embarcados comunicam-se com o meio externo através de periféricos. Estes periféricos podem ser combinados com o processador (como no caso dos sistemas microcontrolados) ou associados no sistema. (Wikipédia, 2021).

Entre os periféricos mais comum temos:

- Entrada de dados através de teclas (geralmente através de teclados feitos com varredura matricial);
- *LED's (Light Emitting Diode* ou Diodo Emissor de Luz);
- *Display LCD* (sendo os mais comuns os alfanuméricos);
- Interface Serial (Por exemplo RS232, I2C);
- *Universal Serial Bus (USB)*;
- TCP/IP.

#### 3.4.5. Exemplos de Utilização de Sistemas Embarcados:

- Aviônicos, como sistemas de controle inercial, controle de vôo e outros sistemas integrados nas aeronaves, como sistemas de orientação de mísseis e anticolisão;
- Telefones celulares e centrais telefônicas;

- Equipamentos de redes de computadores, como roteadores, *hubs*, *switches* e *firewalls*;
- Impressoras;
- Dispositivos de armazenamento (discos rígidos e disquetes);
- Equipamentos de automação comercial;
- Em veículos automotores, controladores de tração e antibloqueio dos freios (freios *ABS*); do motor (injeção eletrônica, ignição eletrônica, recursos para redução do consumo de combustível e emissão de poluentes);
- Calculadoras;
- Eletrodomésticos, como fornos microondas, máquinas de lavar, aparelhos de TV não valvulados, *DVD players*;
- Equipamentos médicos;
- Videogames;
- PDAs;
- Urna Eletrônica;

### 3.5. SISTEMAS DE CONTROLE

Um sistema de controle pode ser definido como um conjunto de equipamentos e dispositivos que gerenciam o comportamento de máquinas ou outros sistemas físicos. Para isso, é necessário realizar a modelagem matemática da planta, seguida do projeto do controlador. Posteriormente, segue-se para a elaboração física dos sistemas. Alguns processos podem ser não controláveis ou apresentar não-linearidades de operação. Nestes casos, devem ser utilizadas técnicas de controle sofisticadas, conforme os objetivos de projeto e custos envolvidos. (Wikipédia, 2021).

Como exemplos de aplicações desses sistemas, pode-se citar: controle de ferramenta nas máquinas *CNC (Computer Numeric Control)*, estabilidade de vôo em aeronaves, controle de servomecanismos, sistemas robotizados, etc. (Wikipédia, 2021).

Há alguns tipos comuns de controladores, com muitas variações e combinações: Controle Lógico (digital), controle realimentado analógico (ou linear), controle robusto; sendo este de grande importância, pois garante um

certo padrão mínimo de desempenho para o sistema, mesmo na presença de incertezas sobre o modelo (OGATA, K. 2010). Há também a lógica *fuzzy*, que procura modelar o raciocínio humano, podendo combinar a estrutura fácil da lógica com a utilidade do mundo real dos controles realimentados.

### 3.5.1. Controles Digitais

Sistemas de controles lógicos para a indústria e equipamentos comerciais, foram historicamente implementados através da lógica de relés, projetados utilizando a lógica *ladder*. Atualmente, a maioria dos sistemas são implementados através de CLP ou microcontrolador. A notação *ladder* ainda é muito utilizada em CLP's (Controladores Lógicos Programáveis). (FERREIRA, H. C., 2020).

Controladores digitais podem utilizar chaveamento, sensores, termostatos, pressostatos, etc., e podem ligar e desligar equipamentos em diversas operações. Sistemas lógicos são utilizados em operações de sequências mecânicas em diversas aplicações. Os CLP's podem ser programados em diversas linguagens como por exemplo, *ladder*, SFC e programação estruturada. (Wikipédia, 2021).

Exemplos de aplicações incluem elevadores, máquinas de lavar e outros equipamentos com operações de liga-desliga automáticos.

Os sistemas lógicos são fáceis de projetar e podem manipular operações muito complexas. Alguns aspectos dos sistemas lógicos são projetados utilizando princípios da lógica Booleana.

### 3.5.2. Controles Analógicos

O termostato é um controlador simples de malha fechada: quando o valor do processo (PV) vai abaixo do *Set Point* (SP), a resistência é ligada. Outro exemplo pode ser um pressostato de um compressor de ar, quando a pressão de processo cai abaixo do *Set Point*, o compressor é ligado. Refrigeradores e bombas de vácuo possuem mecanismos de operação similares, porém efetuando o controle de modo proporcional, realizando assim a correção do erro. (DORF, R.; BISHOP, R. H., 2001).

Em alguns casos, sistemas simples de controle *on-off* podem ser eficazes na malha fechada, não sendo necessário a utilização de técnicas como *PID* ou

*PI*, que são mais precisas, porém mais caras de serem implementadas. Contudo, em outros processos, como no posicionamento preciso de um braço robótico, é necessário o uso de técnicas mais complexas, eliminando totalmente o erro em regime permanente. (Wikipédia, 2021).

### 3.5.3. Controle PID

O Controlador proporcional integral derivativo, controlador *PID* ou simplesmente *PID*, é uma técnica de controle de processos que une as ações derivativa, integral e proporcional, fazendo assim com que o sinal de erro seja minimizado pela ação proporcional, zerado pela ação integral e obtido com uma velocidade antecipada pela ação derivativa.

É baseado na resposta de uma malha de processo industrial a ser controlada.

Na prática os *PID* são encontrados no interior de controladores eletrônicos chamados "single-loop", muitas vezes com microprocessadores, e também através de Controladores programáveis e outros equipamentos de controle. (west-cs.com.br, 2012)

Os controladores contínuos podem controlar os processos de três formas distintas:

- Controle Proporcional (P);
- Controle Integral (PI);
- Controle Derivativo (PD);

Estes três modos de controle são também designados de ações de controle, cada uma delas reagindo de forma distinta ao erro presente nos sistemas e conseqüentemente minimizando a variação de erro. O controle proporcional ajusta a variável de controle de forma proporcional ao erro. O controle integral ajusta a variável de controle baseando-se no tempo em que o erro acontece. O controle derivativo ajusta a variável de controle tendo como base a taxa de variação do erro. A combinação destes tipos de controle forma o controlador conhecido na indústria como *PID*.

Definido pela função  $u(t)$  o sinal de saída (equação 4) para controlar o módulo refrigerador, o algoritmo do *PID* pode ser escrito por:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad (4)$$

Onde:  $K_p$  é o ganho proporcional,  $K_i$  é o ganho integral,  $K_d$  é o ganho derivativo, “e” é o erro, “t” é o tempo e  $\tau$  é o tempo de integração. Aplicando a transformada de Laplace, temos a equação 5:

$$L(s) = k_p + \frac{k_i}{s} + k_d s \quad (5)$$

Onde “s” é a frequência ou uma base de tempo.

A seguir, serão descritas um resumo sobre cada agente que compõe o algoritmo do *PID*:

### 3.5.3.1. Ação Proporcional

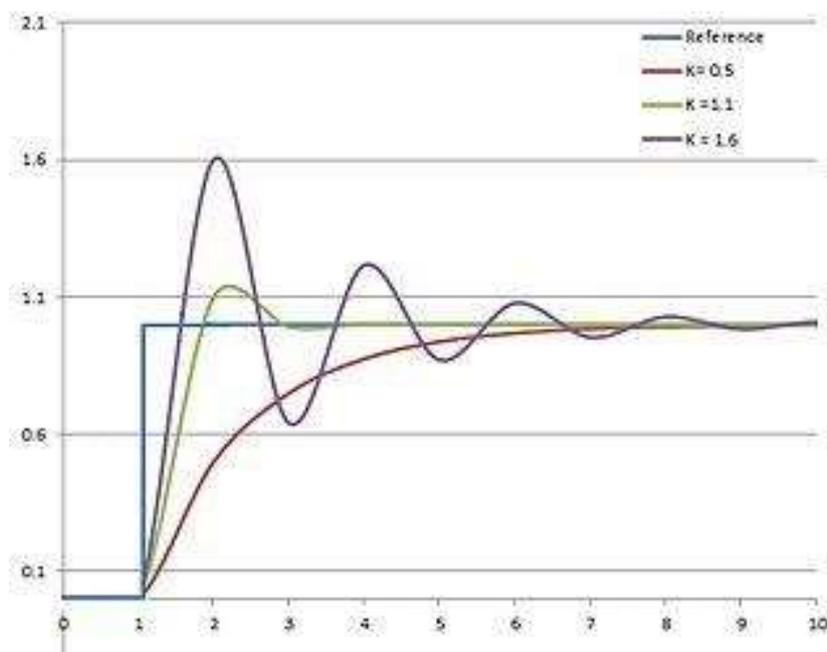
A ação proporcional produz um sinal de saída que é proporcional à amplitude do erro  $e(t)$ , sendo  $K_p$  a constante de proporcionalidade (equação 6):

$$P(\text{saida}) = K_p e(t) \quad (6)$$

Comparado com a ação liga-desliga, esse método possui a vantagem de eliminar as oscilações do sinal de saída. Para tal, o sistema permanece sempre ligado e o sinal de saída é diferente de zero. Tendo em vista que o sinal de saída é proporcional ao erro, um erro não-nulo (conhecido por erro de *off-set*) é gerado.

O valor do erro *off-set* é inversamente proporcional ao ganho  $K_p$  e pode ser compensado adicionando-se um termo ao valor de referência ou pelo controle integral. Um ganho proporcional muito alto gera um alto sinal de saída, o que pode desestabilizar o sistema. Porém, se o ganho proporcional é muito baixo, o sistema falha em aplicar a ação necessária para corrigir os distúrbios. A Figura 6 a seguir, irá mostrar respostas de saída para valores diferentes de  $K_p$ .

Figura 6 – Respostas do Controle Proporcional para diferentes Kp



Fonte: <https://pt.wikipedia.org/>, 2021.

### 3.5.3.2. Ação Integral

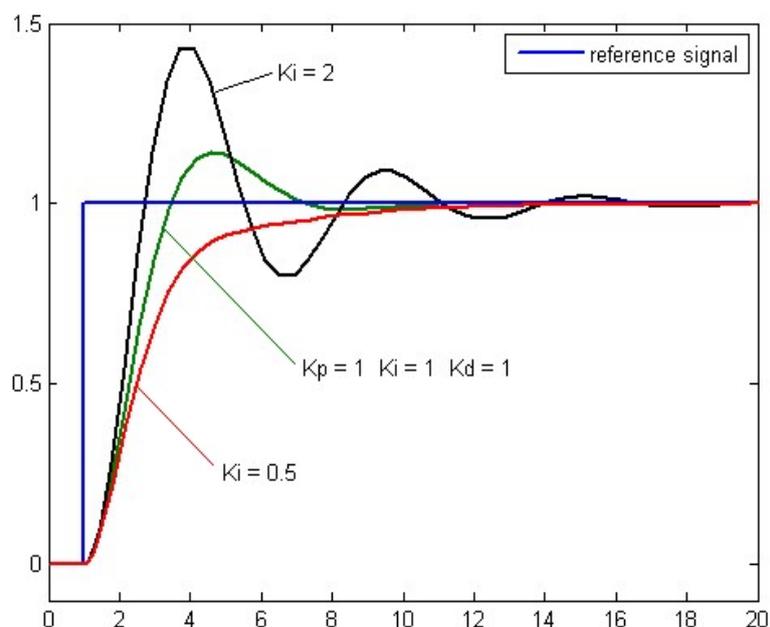
A ação integral produz um sinal de saída que é proporcional à magnitude e à duração do erro, ou seja, ao erro acumulado. Isso fornece uma alternativa para corrigir o erro de *off-set* gerado pela ação proporcional e acelera a resposta do sistema, permitindo-o chegar ao valor de referência mais rapidamente. O sinal de saída do controlador  $I$  (equação 7) pode ser descrito por:

$$I(\text{saída}) = K_i \int_0^t e(\tau) d\tau \quad (7)$$

Onde  $K_i$  é o ganho integral.

A ação integral corrige o valor da variável manipulada em intervalos regulares, chamado tempo integral. Esse tempo integral é definido como o inverso do ganho integral. Se o ganho integral é baixo, o sistema pode levar muito tempo para atingir o valor de referência. No entanto, se o ganho integral for muito alto, o sistema pode tornar-se instável. A Figura 7 a seguir, mostra a saída para diferentes valores de  $K_i$ .

Figura 7 - Respostas do Controle Integral para diferentes  $K_i$



Fonte: <https://pt.wikipedia.org/>, 2021.

### 3.5.3.3. Ação Derivativa

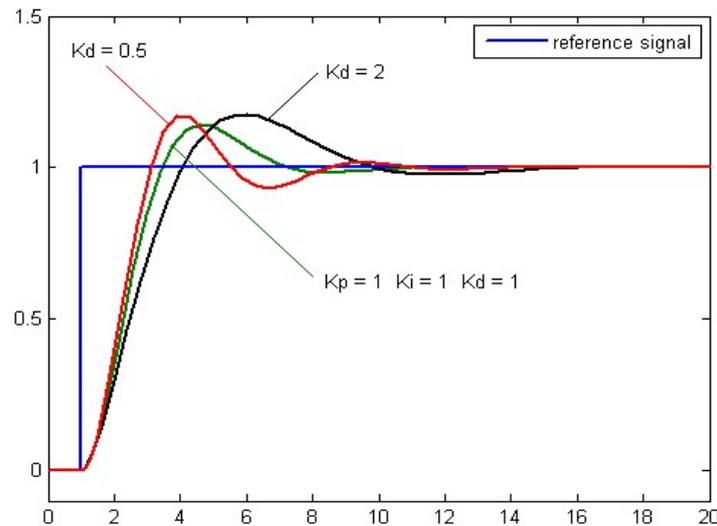
A ação derivativa produz um sinal de saída que é proporcional à velocidade de variação do erro (equação 8):

$$D(\text{saida}) = K_D \frac{de(t)}{dt} \quad (8)$$

Onde  $K_D$  é o ganho derivativo.

A ação derivativa fornece uma correção antecipada do erro, diminuindo o tempo de resposta e melhorando a estabilidade do sistema. (DORF, R.C.; BISHOP, R.H., 2018). A ação derivativa atua em intervalos regulares, chamado tempo derivativo. Esse parâmetro é inversamente proporcional à velocidade de variação da variável controlada. Isso indica que a ação derivativa não deve ser utilizada em processos nos quais o sistema deve responder rapidamente a uma perturbação, nem em processos que apresentem muito ruído no sinal de medido, pois levaria o processo à instabilidade. A Figura 8 a seguir, mostra as respostas de saída para diferentes valores de  $K_D$ .

Figura 8 - Respostas do Controle Derivativo para diferentes  $K_d$



Fonte: <https://pt.wikipedia.org/>, 2021.

Na prática os *PID* são encontrados no interior de controladores eletrônicos chamados "*single-loop*", muitas vezes com microprocessadores, e também através de software em controladores programáveis e outros equipamentos de controle.

#### 3.5.3.4. Ajuste de Parâmetros para o *PID*

O ajuste de parâmetros do controlador *PID* pode ser feito manualmente ou através de métodos de otimização como o método de *Ziegler-Nichols*. Nesse método, os ganhos  $K_I$  e  $K_D$  são primeiramente ajustados para zero. Em seguida, aumentamos o ganho proporcional  $K_P$  até que o sinal de saída começa a oscilar. Isso define um ganho crítico  $K_U$ , e um período crítico  $T_U$ . Os ganhos dos controladores *P*, *PI*, *PID* são então ajustados conforme o Quadro 1 abaixo:

Quadro 1 - Método de *Ziegler-Nichols*

Tipo de Controlador	$K_p$	$K_I$	$K_D$
P	$0,50K_U$		
PI	$0,45K_U$	$1,2K_p/T_U$	
PID	$0,60K_U$	$2K_p/T_U$	$K_pT_U/8$

Tendo em vista que o ganho proporcional do controlador  $P$  é apenas a metade do ganho crítico, esse sistema possui uma boa margem de ajuste. (DORF, R.C.; BISHOP, R.H., 2018) Devido ao efeito desestabilizante da ação integral, o ganho proporcional reduz para 0,45 do ganho crítico. Contudo, o efeito estabilizante da ação derivativa permite aumentar o ganho proporcional para 0,6 do ganho crítico no controlador  $PID$ .

### 3.6. TELEMETRIA

#### 3.6.1. Conceito de Telemetria

Telemetria é uma tecnologia que permite a medição e comunicação de informações de interesse do operador ou desenvolvedor de sistemas. A palavra é de origem grega onde “tele” significa remoto e “metron” significa medida. Sistemas que necessitam de instruções e dados enviados a eles para que sejam operados, requerem o correspondente a telemetria, ou telecomando.

É um sistema de monitoramento com diversas aplicações, muito falada nas corridas de Fórmula 1, *Dragsters* (Veículos modificados para competições de arrancadas) e qualquer outro tipo de desportos automobilístico, também é muito usada em indústrias de monitoramento, normalmente funciona via transmissão cabeada até 30 metros ou sem fio (sinal de rádio), daí o nome telemetria. Em muitos lugares tem o uso em conjunto do *Data/og* que é a função de gravar um período de tempo da leitura dos canais da telemetria. O sistema também é utilizado para recolhimento de dados meteorológicos, por exemplo.

A telemetria geralmente refere-se a comunicações sem fio, porém pode também referir-se aos dados transferidos sobre outras mídias, tais como telefone, redes de computadores ou através de um enlace óptico. No setor automotivo e de logística as informações também podem ser transmitidas via telemóvel, em interfaces próprias para transmissão de dados (*GPRS* e *EDGE* no sistema *GSM* e *1xRTT* no sistema *CDMA*). (Wikipédia, 2021).

### 3.6.2. Exemplos de Telemetria

#### 3.6.2.1. Telemetria Aplicada na Agricultura

A telemetria é utilizada na agricultura. Pois a maioria das atividades relacionadas ao cultivo de alimentos dependem das condições de solo e tempo. Portanto, estações meteorológicas sem fio representam um papel muito importante na prevenção de doenças e na correta irrigação. Estas estações retornam à estação base os principais parâmetros necessários para a tomada das melhores decisões: temperatura do ar e umidade relativa, precipitação e dados de umidade da vegetação (necessários para modelos de prognóstico de doenças), radiação solar e velocidade do vento (necessários para calcular a evapotranspiração) e, em algumas ocasiões, a umidade do solo, crucial para decisões de irrigação adequada para compreensão do progresso da água no solo e para as raízes. Devido aos micro-climas locais poderem variar significativamente, tais dados precisam vir corretamente dentro do cultivo. Estações de monitoramento normalmente transmitem de volta os dados através de rádio terrestre, apesar de sistemas de satélite serem usados eventualmente.

#### 3.6.2.2. Telemetria Aplicada para Eficiência Energética

A telemetria também é uma ótima ferramenta para eficiência energética. Nas fábricas, escritórios e residências, monitorando o uso de energia de cada seção ou equipamentos e os fenômenos decorrentes (como a temperatura) em um ponto de controle por telemetria facilita a coordenação para o uso mais eficiente da energia como no caso do Japão.

#### 3.6.2.3. Telemetria Aplicada na Logística

A telemetria é uma grande aliada dos transportadores, pois as informações relativas a condução do veículo, são transmitidas à base de monitoramento. Com isso é possível identificar e corrigir hábitos dos motoristas, identificar situações que podem expor o veículo, a carga e o próprio condutor a riscos e otimizar o desempenho de cada veículo. As principais informações oferecidas pela telemetria são: Hodômetro (distância percorrida), velocidade em pista seca ou molhada, tempo de acionamento da embreagem (pé na embreagem), freadas bruscas, acelerações bruscas, tempo de uso do veículo

parado, em marcha lenta e em movimento, rotações do motor, combustível, temperatura do motor, falha na pressão de óleo e uso do freio motor para veículos de carga. Através da Telemetria também é possível gerar relatórios de uso do veículo com identificação do motorista, jornada de trabalho, tempo de condução na faixa de rotação econômica, etc. A quantidade de informações depende do veículo (marca, ano, modelo, etc) em que será instalado o sistema. As informações podem ser enviadas juntamente com a localização via sistema de rastreamento.

Algumas fornecedoras de água, energia elétrica ou gás, possuem contadores eletrônicos que enviam a informação do consumo para uma central ou visor individual que gere toda a informação recolhida, podendo assim ser efetuada a respectiva e correta faturação temporal, normalmente mensal. O contador de água, eletricidade, ou gás, é mecânico ou eletrônico, ambos transmite através de um protocolo específico o consumo medido usando uma certa cadência temporal, ou a pedido. Essa informação enviada pelo contador eletrônico é enviada para um concentrador central ou individual, que reencaminha essa informação para um servidor do fornecedor do serviço, normalmente através de redes de sistemas cabeada ou sem fios (*GPRS*). O fornecedor do serviço pode assim efetuar uma faturação correta para o período temporal não sendo necessária a leitura presencial dentro da unidade habitacional ou local que necessita a medição por parte de um técnico, nem efetuar estimativas para obter valores para faturação.

#### 3.6.2.4. Telemetria Aplicado em *Softwares*

Alguns *softwares*, tal como *Office* (a partir da versão 365) e *Windows* (de fábrica a partir da versão 10, mas opcionalmente disponível para versões 7, 8 e 8.1) utilizam de telemetria para enviar dados ao fabricante sobre a forma como o usuário final lida com a interface gráfica do *software*, bem como outros dados pessoais do usuário.

#### 3.6.2.5. Telemetria Aplicada na Internet das Coisas (*IoT*)

A internet avança para conectar coisas diversas, trazendo inteligência e reinventando dispositivos. A telemetria é a base do conceito de Internet das Coisas. Dispositivos como termostatos inteligentes, iluminação automatizada,

sistemas de som e audio-visual já são exemplos de coisas conectadas que têm seu funcionamento baseado em princípios da telemetria. (Wikipédia, 2021).

#### 3.6.2.6. Função da Telemetria

Monitorar canais analógicos e digitais em tempo real ou via *DataLog*.

#### 3.6.2.7. *DataLog*

Função de gravar um período de tempo da leitura da telemetria, o tempo é variado com o espaço de memória disponível para gravar.

### 3.7. GENERAL PACKET RADIO SERVICE (*GPRS*)

#### 3.7.1. Conceito Sobre *GPRS*

O *General Packet Radio Service (GPRS)* é uma tecnologia que aumenta as taxas de transferência de dados nas redes *GSM (Global System for Mobile Communications)* existentes. Esta permite o transporte de dados por pacotes (Comutação por pacotes). Sendo assim, o *GPRS* oferece uma taxa de transferência de dados muito mais elevada que as taxas de transferência das tecnologias anteriores, que usavam comutação por circuito, que eram em torno de 12kbps. Já o *GPRS*, em situações ideais, pode ultrapassar a marca dos 170kbps. No entanto na prática, essa taxa está em torno dos 40kbps. (Wikipédia, 2021).

Diferente das tecnologias de comutação de circuitos, que é um modo no qual uma conexão (ou circuito) é estabelecida do ponto de origem da transferência de dados ao destino e os recursos da rede são dedicados por toda a duração da chamada (ou até que o usuário interrompa a conexão), no *GPRS* o serviço é "sempre ativo", ou seja, ele é um modo no qual os recursos somente são atribuídos a um usuário quando for necessário enviar ou receber dados. Esta técnica permite que vários usuários compartilhem os mesmos recursos, aumentando assim a capacidade da rede e permitindo uma gerência razoavelmente eficiente dos recursos. Isto permite às operadoras *GPRS* disponibilizar acesso à Internet móvel em alta velocidade e a um custo razoável, pois a cobrança é feita pela quantidade de pacotes de dados transmitidos e não pelo tempo de conexão à rede. (Wikipédia, 2021).

### 3.7.2. Principais vantagens da Utilização do *GPRS*

- Utilização de voz e dados simultaneamente no mesmo canal;
- Ampla cobertura em todas as unidades;
- Acesso imediato e permanente para dados. Para se conectar à rede utilizando *GSM* são necessários de 15 a 30 segundos, sendo que esse tempo é consumido a cada reconexão. Com o *GPRS*, uma vez estabelecida a conexão, a mesma estará permanentemente ativa;
- Aumento significativo na velocidade de transmissão de dados (através da rede *GSM* é possível alcançar uma velocidade máxima de 9,6 kbps; com o *GPRS* a velocidade varia de 40 kbps até 144 kbps);
- Utilização de protocolos *X.25* e *IP* amplamente divulgados;
- Possibilidade de utilização de várias operadoras de telefonia e modelos diferentes de celular-modems, havendo assim uma maior flexibilidade e independência em relação ao mercado;
- Redução de custos. Com o *GSM* a tarifação é efetuada por tempo de conexão. Com o *GPRS*, a tarifação é efetuada com base na quantidade de dados transmitidos;

Os celular-modems utilizados em *GPRS* podem ser classificados em três tipos diferentes, conforme abaixo:

- Classe A: Para uso simultâneo de voz e dados;
- Classe B: Para uso de voz e dados, porém não simultâneo;
- Classe C: Para uso apenas de dados.

### 3.7.3. Disponibilidade imediata

Os sinais *GPRS* facilita conexões instantâneas pois a informação pode ser enviada ou recebida imediatamente conforme a necessidade do usuário. Não há necessidade de conexões dial-up através de modems. Algumas vezes, diz-se que os usuários de *GPRS* estão “sempre conectados”. Disponibilidade imediata é uma das vantagens de *GPRS* (e *SMS*) quando comparado com *CSD*. A alta disponibilidade imediata é uma característica muito importante para aplicações críticas como autorização remota de lançamento em cartões de

crédito, quando é inaceitável que o cliente seja mantido em estado de espera por mais de 30 segundos além do necessário, por exemplo.

#### 3.7.4. Acesso ao serviço GPRS

Para usar *GPRS*, os usuários precisam especificamente de:

- Um telefone móvel ou terminal que suporte *GPRS*;
- Uma assinatura em uma rede de telefonia móvel que suporte *GPRS*;
- Ter o uso de *GPRS* habilitado. Acesso automático ao *GPRS* pode ser permitido por algumas operadoras; outras poderão requerer uma opção específica de adesão;
- Conhecimento de como enviar e receber informações através do *GPRS* usando seu aparelho telefônico, incluindo configurações de hardware e software, o que cria a necessidade de um serviço de atendimento ao cliente;
- Um destino para enviar ou um local de onde receber informações através do *GPRS*. Enquanto que com *SMS* esse destino ou origem era frequentemente outro telefone móvel, com *GPRS* é mais provável que se pareça com um endereço Internet, já que *GPRS* foi projetado para tornar o acesso à Internet totalmente disponível aos usuários móveis desde o início. Desde a disponibilidade do serviço, os usuários do *GPRS* podem acessar qualquer página da Web ou outras aplicações Internet - fornecendo uma massa crítica inicial de uso.

Tendo visto as características principais do *GPRS* do ponto de vista do usuário, vejamos quais são essas características do ponto de vista de uma operadora da rede.

#### 3.7.5. Características principais da rede GPRS

Com o *GPRS*, a informação é dividida em “pacotes” relacionados entre si antes de ser transmitida e remontada no destinatário. A comutação de pacotes é semelhante a um jogo de quebra-cabeças (puzzle) - a imagem que o quebra-cabeças representa é dividida em pequenas peças pelo fabricante e colocada em um saco plástico. Durante o transporte do quebra-cabeças entre a fábrica e o comprador, as peças são misturadas. Quando o comprador do jogo retira as

peças da embalagem ele as remonta, formando a imagem original. Todas as peças são relacionadas entre si e se encaixam, mas a forma como são transportadas e remontadas varia.

A Internet é um outro exemplo de rede de dados baseada em comutação de pacotes, o mais famoso de muitos tipos de rede. As características principais de uma rede GPRS são:

- Pacotes de redes;
- São semelhantes a um jogo de quebra-cabeças.

### 3.7.6. Compatibilidade com a Internet

A princípio, o *GPRS* permite uma funcionalidade completa no que se refere a Internet Móvel por disponibilizar interoperabilidade entre a Internet existente e as novas redes *GPRS*. Qualquer serviço atualmente utilizado na Internet - FTP, navegação na *WEB*, *chat*, *email*, *telnet* - estará disponível através da rede móvel com o *GPRS*. Na verdade, muitas operadoras estão considerando a oportunidade de usar o *GPRS* como forma de ajudar a se tornarem Provedores de Serviço Internet.

A *World Wide Web* está se tornando a primeira escolha das pessoas que desejam acessar a Internet para entretenimento e coleta de informações, a intranet para acessar informações da companhia e conexão com colegas de trabalho e a extranet para acessar clientes e fornecedores. Tudo isso deriva da *World Wide Web* com o intuito de conectar comunidades com interesses diversos.

Há uma tendência em se armazenar informações localmente por meio de pacotes de software específicos e acessar essas informações remotamente, via Internet. Quando você quer verificar sua programação de tarefas ou contatos, ao invés de usar algo semelhante ao software "Act!", você pode ir a um site semelhante a um portal na Internet. Assim, a navegação na Web é uma aplicação importante para *GPRS*. Como os protocolos em uso são os mesmos, as redes *GPRS* podem ser encaradas como sub-redes da Internet e os telefones *GPRS* compatíveis podem ser vistos como nós móveis dessa rede. Isso significa que cada terminal *GPRS* pode potencialmente ter seu próprio endereço *IP* e ser endereçável por ele.

### 3.7.7. Suporte a *TDMA* e *GSM*

Seria importante notar que o *GPRS* não é um serviço projetado para ser utilizado exclusivamente em redes móveis baseadas no padrão *GSM*. O padrão *IS-136 TDMA (Time Division Multiple Access)*, popular nas Américas do Norte e do Sul, também suporta *GPRS*. Essa aceitação permite seguir um caminho evolutivo em direção às redes móveis de terceira geração, conforme acordado em 1999 pelas associações da indústria que suportam esses dois tipos de redes.

## 3.8. GLOBAL POSITIONING SYSTEM (*GPS*)

### 3.8.1. Conceito Sobre *GPS*

O Sistema de Posicionamento Global (*GPS*) é um sistema de navegação baseado em satélite composto de pelo menos 24 satélites. O *GPS* funciona em qualquer condição climática, em qualquer lugar do mundo, 24 horas por dia, sem taxas de assinatura ou taxas de configuração. O Departamento de Defesa dos Estados Unidos (*USDOD*) originalmente colocou os satélites em órbita para uso militar, mas eles foram disponibilizados para uso civil na década de 1980. (garmim.com, 2021).

### 3.8.2. Funcionamento do *GPS*

Os satélites *GPS* giram em torno da Terra duas vezes por dia em uma órbita precisa. Cada satélite transmite um sinal único e parâmetros orbitais que permitem que os dispositivos *GPS* decodifiquem e calculem a localização precisa do satélite. Os receptores *GPS* usam essas informações e trilateração para calcular a localização exata de um usuário. Essencialmente, o receptor *GPS* mede a distância de cada satélite pela quantidade de tempo que leva para receber um sinal transmitido. Com medições de distância de mais alguns satélites, o receptor pode determinar a posição de um usuário e exibi-la eletronicamente para medir sua rota de corrida, mapear um campo de golfe, encontrar o caminho de casa ou aventura em qualquer lugar. (garmim.com, 2021).

Para calcular sua posição 2D (2 dimensões: latitude e longitude) e o movimento da trilha, um receptor *GPS* deve ser travado no sinal de pelo menos

3 satélites. Com 4 ou mais satélites à vista, o receptor pode determinar sua posição 3D (latitude, longitude e altitude). Geralmente, um receptor *GPS* rastreia 8 ou mais satélites, mas isso depende da hora do dia e de onde você está na Terra. Alguns dispositivos podem fazer tudo isso no seu pulso.

Assim que sua posição for determinada, a unidade *GPS* pode calcular outras informações, como:

- Velocidade
- Consequência
- Acompanhar
- Distância da viagem
- Distância até o destino
- Hora do nascer e pôr do sol
- E mais

### 3.8.3. Precisão do GPS

Os receptores *GPS* de hoje são extremamente precisos, graças ao design multicanal paralelo. Muitos modelos receptores são rápidos para travar em satélites quando são ligados pela primeira vez. Eles mantêm um bloqueio de rastreamento em densa cobertura de árvores ou em ambientes urbanos com edifícios altos. Certos fatores atmosféricos e outras fontes de erro podem afetar a precisão dos receptores *GPS*.

Alguns modelos de receptores *GPS* têm, normalmente, uma precisão de 10 metros. A precisão é ainda melhor na água. Alguma precisão do receptor *GPS* é melhorada com *WAAS* (*Wide Area Augmentation System*). Esta capacidade pode melhorar a precisão para mais de 3 metros, fornecendo correções para a atmosfera. Nenhum equipamento adicional ou taxas são necessárias para aproveitar as vantagens dos satélites *WAAS*.

Os usuários também podem obter melhor precisão com o *GPS* diferencial (*DGPS*), que corrige as distâncias do *GPS* em uma média de 1 a 3 metros. A Guarda Costeira dos EUA, por exemplo, opera o serviço de correção *DGPS* mais comum, consistindo em uma rede de torres que recebem sinais de *GPS* e transmitem um sinal corrigido por transmissores de farol. Para obter o sinal

corrigido, os usuários devem ter um receptor de farol diferencial e uma antena de farol, além de seu *GPS*. (garmim.com, 2021).

#### 3.8.4. Sistema de Satélite *GPS*

Os 31 satélites que atualmente constituem o segmento espacial do *GPS* estão orbitando a Terra a cerca de 12.000 milhas acima de nós. Esses satélites estão em constante movimento, fazendo duas órbitas completas em menos de 24 horas. Eles viajam a velocidades de aproximadamente 7.000 milhas por hora. Pequenos impulsadores de foguetes mantêm cada satélite voando no caminho correto. Aqui estão alguns outros fatos interessantes sobre os satélites *GPS*:

- O nome oficial do *USDOD* para *GPS* é *NAVSTAR*;
- O primeiro satélite *GPS* foi lançado em 1978;
- Uma constelação completa de 24 satélites foi alcançada em 1994;
- Cada satélite é construído para durar cerca de 10 anos. Substituições estão constantemente sendo construídas e lançadas em órbita;
- Um satélite *GPS* pesa aproximadamente 2.000 libras e tem cerca de 5 metros de largura com os painéis solares estendidos;
- Os satélites *GPS* são movidos a energia solar, mas têm baterias de reserva a bordo, no caso de um eclipse solar;
- A potência do transmissor é de apenas 50 Watts ou menos.

#### 3.8.5. Sinal *GPS*

Os satélites *GPS* transmitem pelo menos 2 sinais de rádio de baixa potência. Os sinais viajam pela linha de visão, o que significa que eles passarão por nuvens, vidro e plástico, mas não passarão pela maioria dos objetos sólidos, como edifícios e montanhas. No entanto, os receptores modernos são mais sensíveis e geralmente podem rastrear através das casas.

Um sinal *GPS* contém 3 tipos diferentes de informação:

- O código pseudo-aleatório é um código de identificação que identifica qual satélite está transmitindo informações. Você pode ver de quais satélites está recebendo sinais na página de satélite do seu dispositivo;

- Os dados das efemérides são necessários para determinar a posição de um satélite e fornecem informações importantes sobre a saúde de um satélite, data e hora atuais;
- Os dados do Almanaque informam ao receptor *GPS* onde cada satélite *GPS* deve estar a qualquer hora do dia e mostram as informações orbitais desse satélite e de todos os outros satélites do sistema.

### 3.8.6. Fatores de erros do *GPS*

Os fatores que podem afetar o sinal e a precisão do *GPS* incluem o seguinte:

- Atrasos na ionosfera e na troposfera: Os sinais de satélite tornam-se mais lentos à medida que passam pela atmosfera. O sistema *GPS* usa um modelo embutido para corrigir parcialmente este tipo de erro;
- Sinais multipercurso: O sinal *GPS* pode refletir objetos como edifícios altos ou grandes superfícies rochosas antes de chegar ao receptor, o que aumentará o tempo de viagem do sinal e causará erros;
- Erros do relógio do receptor: O relógio embutido de um receptor pode ter pequenos erros de tempo porque é menos preciso do que os relógios atômicos dos satélites *GPS*.
- Erros orbitais: A localização informada do satélite pode não ser precisa.
- Número de satélites visíveis: Quanto mais satélites um receptor *GPS* puder "ver", melhor será a precisão. Quando um sinal é bloqueado, você pode obter erros de posição ou possivelmente nenhuma leitura de posição. As unidades de *GPS* normalmente não funcionam debaixo d'água ou no subsolo, mas novos receptores de alta sensibilidade são capazes de rastrear alguns sinais quando dentro de edifícios ou sob a cobertura de árvores;
- Geometria/Sombreamento do satélite: Os sinais do satélite são mais eficazes quando os satélites estão localizados em ângulos amplos em relação uns aos outros, em vez de em uma linha ou agrupamento compacto;

- Disponibilidade seletiva: O Departamento de Defesa dos Estados Unidos uma vez aplicou a Disponibilidade Seletiva (SA) aos satélites, tornando os sinais menos precisos para evitar que os "inimigos" usem sinais *GPS* de alta precisão. O governo desligou o SA em Maio de 2000, o que melhorou a precisão dos receptores *GPS* civis. (garmim.com, 2021).

### 3.9. PÁGINA WEB E PHP

#### 3.9.1. Conceito de Página WEB

Uma página de rede, (SAWAYA, M. R. 2002. p.22) página de *web* ou página *web* (em inglês: *web page* ou *webpage*) é uma coleção específica de informações fornecidas por um *site* e exibidas a um usuário em um navegador *web*. Um site geralmente consiste em muitas páginas web ligadas de maneira coerente. O nome "página web" é uma metáfora de páginas de papel encadernadas em um livro.

O elemento principal de uma página web é um ou mais arquivos de texto escritos na linguagem *HTML (Hypertext Markup Language)*. Muitas páginas web também usam código *JavaScript* para comportamento dinâmico e código *CSS (Cascading Style Sheets)* para semântica de apresentação. Imagens, vídeos e outros arquivos multimídia também são frequentemente incorporados em páginas web.

Cada página da web é identificada por um *URL (Uniform Resource Locator)* distinto. Quando o usuário insere um *URL* no navegador, os elementos dessa página são baixados dos servidores web. O navegador transforma todos os elementos em uma representação visual interativa no dispositivo do usuário. (TALIGARSIEL, 2012).

Se o usuário clicar ou tocar em um link para outra página, o navegador repetirá esse processo para exibir a nova página, que pode fazer parte do site atual ou de outro.

Da perspectiva da implementação de sites do lado do servidor, existem dois tipos de páginas web: Estática e Dinâmica. As páginas estáticas são recuperadas do sistema de arquivos do servidor web sem nenhuma modificação, enquanto as páginas dinâmicas devem ser criadas pelo servidor web em tempo

real, normalmente utilizando um banco de dados para preencher um template web, antes de serem enviadas ao navegador do usuário. (pcmag.com, 2021).

### 3.9.2. Conceito do *PHP*

O *PHP* (*Hypertext Pre-processor*) é uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do *HTML*.

Em vez de muitos comandos para mostrar *HTML* (como acontece com a linguagem C ou Perl), as páginas em *PHP* contêm *HTML* em código mesclado que faz "alguma coisa". O código *PHP* é delimitado pelas instruções de processamento (tags) de início e fim "<?php" e "?>" que permitem que você entre e saia do "modo *PHP*". (php.net, 2021).

### 3.9.3. Diferenças entre *PHP* e *JavaScript*

O que distingue o *PHP* de algo como o *JavaScript* no ponto de vista de um usuário comum, é que o código é executado no servidor, gerando o *HTML* que é então enviado para o navegador. O navegador recebe os resultados da execução desse *script*, mas não sabe qual era o código fonte. O desenvolvedor *web* pode inclusive configurar seu servidor para processar todos os seus arquivos *HTML* com o *PHP*, e então não há como os usuários dizerem o que foi feito para execução do projeto. (php.net, 2021).

### 3.9.4. Vantagens de Utilizar *PHP*

A vantagem de utilizar o *PHP* é que ele é extremamente simples para um iniciante, mas oferece muitos recursos avançados para um programador profissional. Apesar do desenvolvimento do *PHP* ser focado nos *scripts* do lado do servidor, é possível aproveitar muitos recursos com esta ferramenta.

O *PHP* é focado principalmente nos *scripts* do lado do servidor, portanto, é possível fazer qualquer coisa que outro programa *CGI* pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber *cookies*. Existem três áreas principais onde os *scripts PHP* são usados:

- *Scripts no lado do servidor (server-side)*: Este é o mais tradicional e principal campo de atuação do *PHP*. São necessárias três coisas para

isto funcionar: O interpretador do *PHP* (*CGI* ou módulo do servidor), um servidor web e um navegador web. Você precisa rodar o servidor web conectado a uma instalação do *PHP*. Você pode acessar os resultados de seu programa *PHP* com um navegador web, visualizando a página *PHP* através do servidor web. Tudo isso pode rodar numa máquina pessoal apenas experimentando programar com o *PHP*.

- Scripts de linha de comando: É possível fazer um script *PHP* para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador *PHP*. Esse tipo de uso é ideal para script executados usando o cron (*Unix, Linux*) ou o Agendador de Tarefas (no *Windows*). Esses scripts podem ser usados também para rotinas de processamento de texto simples.
- Escrever aplicações desktop: O *PHP* provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas é possível utilizar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o *PHP-GTK* para escrever programas assim. Você também tem a possibilidade de escrever aplicações multi-plataformas desse jeito. O *PHP-GTK* é uma extensão do *PHP*, não disponibilizada na distribuição oficial. (php.net, 2021).

### 3.9.5. Ambiente de Utilização do *PHP*

O *PHP* pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, macOS, RISC OS e provavelmente outros. O *PHP* também tem suporte à maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. E isso inclui qualquer servidor web que possa utilizar o binário FastCGI do *PHP*, como o lighttpd e o nginx. O *PHP* trabalha tanto como módulo quanto como um processador CGI.

Com o *PHP*, portanto, é possível escolher qualquer sistema operacional e servidor web. Além disso, é possível escolher entre utilizar programação estruturada ou programação orientada a objeto (*OOP*), ou ainda uma mistura das duas.

Com *PHP*, o desenvolvedor não está limitado a gerar somente *HTML*. As habilidades do *PHP* incluem geração de imagens, arquivos *PDF* e até animações *Flash* (utilizando *libswf* e *Ming*) criados dinamicamente, on the fly. Como também, o desenvolvedor poderá facilmente criar qualquer texto, como *XHTML* e outros arquivos *XML*. O *PHP* pode gerar esses arquivos e salvá-los no sistema de arquivos, em vez de mostrá-los em tela, formando um cache no lado do servidor para seu conteúdo dinâmico. (php.net, 2021).

Uma das características mais fortes e mais significativas do *PHP* é seu suporte a uma ampla variedade de banco de dados. Escrever uma página web consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de banco de dados (por exemplo, *mysql*), ou usando uma camada de abstração como o *PDO* ou conectar a qualquer banco de dados que suporte o padrão "*Open Database Connection*" usando a extensão *ODBC*. Outros bancos de dados podem utilizar *cURL* ou *sockets*, como o *CouchDB*.

O *PHP* também tem suporte para comunicação com outros serviços utilizando protocolos como *LDAP*, *IMAP*, *SNMP*, *NNTP*, *POP3*, *HTTP*, *COM* (no Windows) e incontáveis outros. Você também pode abrir *sockets* de rede e interagir diretamente usando qualquer outro protocolo. O *PHP* também suporta o intercâmbio de dados complexos *WDDX*, utilizado em virtualmente todas as linguagens de programação para web. Falando de comunicação, o *PHP* implementa a instanciação de objetos Java e os utiliza transparentemente como objetos *PHP*. (php.net, 2021).

O *PHP* tem recursos úteis para processamento de texto, incluindo expressões regulares compatíveis com *Perl* (*PCRE*), e muitas outras extensões e ferramentas para analisar e acessar documentos *XML*. O *PHP* padroniza todas as extensões *XML* a partir da base sólida da *libxml2*, além de estender o conjunto de recursos adicionando suporte a *SimpleXML*, *XMLReader* e *XMLWriter*.

E existem muitas outras extensões interessantes, que são categorizadas tanto alfabeticamente quanto por categoria. E existem também as extensões *PECL* adicionais que podem, ou não, estar documentadas dentro do próprio manual do *PHP*. (php.net, 2021).

### 3.10. BANCO DE DADOS E SQL

### 3.10.1. Conceito Banco de Dados

Um banco de dados é uma coleção organizada de informações ou dados estruturados, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (*DBMS*). Juntos, os dados e o *DBMS*, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (*SQL*) para escrever e consultar dados (ORACLE, 2021).

### 3.10.2. Conceito Sobre SQL

O *SQL* é uma linguagem de programação usada por quase todos os bancos de dados relacionais para consultar, manipular e definir dados e fornecer controle de acesso. O *SQL* foi desenvolvido pela primeira vez na *IBM* nos anos 1970, com a Oracle como principal contribuinte, o que levou à implementação do padrão *SQL ANSI*; o *SQL* estimulou muitas extensões de empresas como *IBM*, *Oracle* e *Microsoft*. Embora o *SQL* ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer.

Os bancos de dados evoluíram muito desde a sua criação no início dos anos 1960. Bancos de dados de navegação, como o banco de dados hierárquico (que se baseava em um modelo de árvore e permitia apenas um relacionamento um-para-muitos), e o banco de dados de rede (um modelo mais flexível que permitia múltiplos relacionamentos) eram os sistemas originais usados para armazenar e manipular dados (ORACLE, 2021).

Embora simples, esses primeiros sistemas eram inflexíveis. Nos anos 1980, bancos de dados relacionais tornaram-se populares, seguidos por bancos de dados orientados a objetos na década de 1990. Mais recentemente, bancos de dados *NoSQL* surgiram como uma resposta ao crescimento da internet e à necessidade de maior velocidade e processamento de dados não estruturados.

Hoje, bancos de dados na nuvem e bancos de dados autônomos estão abrindo novos caminhos quando se trata de como os dados são coletados, armazenados, gerenciados e utilizados (ORACLE, 2021).

### 3.10.3. Tipos de Banco de Dados

Existem muitos tipos diferentes de bancos de dados. O melhor banco de dados para uma organização específica depende de como a organização pretende usar os dados.

- Bancos de dados relacionais: Os bancos de dados relacionais se tornaram dominantes na década de 1980. Os itens em um banco de dados relacional são organizados como um conjunto de tabelas com colunas e linhas. A tecnologia de banco de dados relacional fornece a maneira mais eficiente e flexível de acessar informações estruturadas.
- Bancos de dados orientados a objetos: As informações em um banco de dados orientado a objetos são representadas na forma de objetos, como na programação orientada a objetos.
- Bancos de dados distribuídos: Um banco de dados distribuído consiste em dois ou mais arquivos localizados em sites diferentes. O banco de dados pode ser armazenado em vários computadores, localizados no mesmo local físico ou espalhados por diferentes redes.
- Data warehouses: Um repositório central de dados, um data warehouse é um tipo de banco de dados projetado especificamente para consultas e análises rápidas.
- Bancos de Dados *NoSQL*: Um *NoSQL*, ou banco de dados não relacional, permite que dados não estruturados e semiestruturados sejam armazenados e manipulados (em contraste com um banco de dados relacional, que define como todos os dados inseridos no banco de dados devem ser compostos). Os bancos de dados *NoSQL* se tornaram populares à medida que os aplicativos *Web* se tornaram mais comuns e mais complexos.
- Bancos de dados gráficos: Um banco de dados gráfico armazena dados em termos de entidades e os relacionamentos entre entidades.

- Bancos de dados *OLTP*: Um banco de dados *OLTP* é um banco de dados rápido e analítico projetado para um grande número de transações realizadas por vários usuários.

Esses são apenas alguns dos vários tipos de bancos de dados em uso atualmente. Outros bancos de dados menos comuns são adaptados para funções científicas, financeiras ou outras muito específicas. Além dos diferentes tipos de banco de dados, as mudanças nas abordagens de desenvolvimento de tecnologia e os avanços dramáticos, como a nuvem e a automação, estão impulsionando os bancos de dados em direções totalmente novas. Alguns dos mais recentes bancos de dados incluem:

- Bancos de dados de código aberto: Um sistema de banco de dados de código aberto é aquele cujo código-fonte é código aberto; esses bancos de dados podem ser bancos de dados *SQL* ou *NoSQL*.
- Bancos de dados em nuvem: Um banco de dados em nuvem é uma coleção de dados, estruturados ou não estruturados, que residem em uma plataforma de computação em nuvem privada, pública ou híbrida. Existem dois tipos de modelos de banco de dados em nuvem: tradicional e banco de dados como um serviço (*DBaaS*). Com o *DBaaS*, as tarefas administrativas e a manutenção são executadas por um provedor de serviços.
- Banco de dados multimodelo: Bancos de dados multimodelo combinam diferentes tipos de modelos de banco de dados em um back-end único e integrado. Isso significa que eles podem acomodar vários tipos de dados.
- Banco de dados de documentos/JSON: Projetado para armazenar, recuperar e gerenciar informações orientadas a documentos, os bancos de dados de documentos são uma maneira moderna de armazenar dados no formato JSON, em vez de linhas e colunas.
- Bancos de dados autônomos: Os bancos de dados independentes mais novos e inovadores (também conhecidos como bancos de dados autônomos) são baseados em nuvem e usam *machine learning* para automatizar o ajuste de banco de dados, segurança, *backups*,

atualizações e outras tarefas de gerenciamento de rotina tradicionalmente executadas por administradores de banco de dados.

#### 3.10.4. Software Banco de Dados

O software de banco de dados é usado para criar, editar e manter arquivos e registros de banco de dados, facilitando a criação de arquivos e registros, entrada de dados, edição, atualização e relatórios de dados. O software também processa armazenamento de dados, backup e relatórios, controle multiacesso e segurança. A segurança forte do banco de dados é especialmente importante hoje, porque o roubo de dados se torna mais frequente. O software de banco de dados às vezes também é conhecido como "sistema de gerenciamento de banco de dados (*DBMS*)".

O *software* de banco de dados simplifica o gerenciamento de dados, permitindo que os usuários armazenem dados em um formulário estruturado e depois os acessem. Ele normalmente tem uma interface gráfica para ajudar a criar e gerenciar os dados e, em alguns casos, os usuários podem construir os próprios bancos de dados usando o *software* do banco de dados.

#### 3.10.5. Gerenciamento de Banco de Dados

Um banco de dados normalmente requer um programa abrangente de banco de dados, conhecido como sistema de gerenciamento de banco de dados (*DBMS*). Um *DBMS* serve como uma interface entre o banco de dados e seus usuários finais ou programas, permitindo que os usuários recuperem, atualizem e gerenciem como as informações são organizadas e otimizadas. Um *DBMS* também facilita a supervisão e o controle de bancos de dados, permitindo uma variedade de operações administrativas, como monitoramento de desempenho, ajuste e backup e recuperação.

#### 3.10.6. Exemplos de Banco de Dados

Alguns exemplos de softwares de bancos de dados populares ou *DBMSs* incluem *MySQL*, *Microsoft Access*, *Microsoft SQL Server*, *FileMaker Pro*, *Oracle Database* e *dBASE*.

### 3.10.7. Banco de Dados *MySQL*

O *MySQL* é um sistema de gerenciamento de banco de dados relacional de código aberto baseado em *SQL*. Ele foi projetado e otimizado para aplicativos da web e pode ser executado em qualquer plataforma. Como surgiram requisitos novos e diferentes com a internet, o *MySQL* tornou-se a plataforma preferida para desenvolvedores da web e aplicativos baseados na web. Por ter sido projetado para processar milhões de consultas e milhares de transações, o *MySQL* é uma escolha popular para empresas de comércio eletrônico que precisam gerenciar várias transferências de dinheiro. A flexibilidade sob demanda é o principal recurso do *MySQL*.

O *MySQL* é o *DBMS* por trás de alguns dos principais sites e aplicativos baseados na web do mundo, incluindo Airbnb, Uber, LinkedIn, Facebook, Twitter e YouTube.

### 3.11. USO DE *API*'s

#### 3.11.1. Conceito de *API*:

*API* é um conjunto de definições e protocolos usados no desenvolvimento e na integração de software de aplicações. *API* é um acrônimo em inglês que significa interface de programação de aplicações.

Uma *API* permite que sua solução ou serviço se comunique com outros produtos e serviços sem precisar saber como eles foram implementados. Isso simplifica o desenvolvimento de aplicações, gerando economia de tempo e dinheiro. Ao desenvolver novas ferramentas e soluções (ou ao gerenciar aquelas já existentes), as *API*'s oferecem a flexibilidade necessária para simplificar o design, a administração e o uso, além de fornecer oportunidades de inovação.

As *API*'s costumam ser vistas como contratos, com documentações que representam um acordo entre as partes interessadas. Se uma dessas partes enviar uma solicitação remota estruturada de uma forma específica, isso determinará como o *software* da outra parte responderá (REDHAT, 2021).

### 3.11.2. Vantagens de utilizar *API's*

As *APIs* simplificam a forma como os desenvolvedores integram novos componentes de aplicações a uma arquitetura preexistente. Por isso, elas ajudam na colaboração entre as empresas e as equipes de TI. Muitas vezes, as necessidades empresariais mudam rapidamente para responder aos mercados digitais em transformação. Nesse ambiente, novos concorrentes podem redefinir o setor inteiro com uma nova aplicação. Para manter a competitividade, é importante oferecer suporte à implantação e desenvolvimento rápidos de serviços inovadores. O desenvolvimento de aplicações nativas em *cloud* é uma forma conhecida de aumentar a velocidade de criação. Ele depende de uma arquitetura de aplicações de micro-serviços conectada por meio de uma *API*.

As *API's* são uma maneira simplificada de conectar a própria infraestrutura por meio do desenvolvimento de aplicações nativas em nuvem. No entanto, elas também possibilitam o compartilhamento de dados com clientes e outros usuários externos. As *API's* públicas agregam valor comercial porque simplificam e ampliam a forma como você se conecta aos parceiros, além de possivelmente monetizar seus dados. Um exemplo famoso é a *API* do *Google Maps* (REDHAT, 2021).

Por exemplo, imagine uma empresa distribuidora de livros. Essa distribuidora de livros poderia oferecer aos clientes uma aplicação em que os atendentes de uma livraria pudessem verificar a disponibilidade de um título diretamente com a distribuidora. Essa aplicação poderia ser cara de desenvolver, limitada pela plataforma e exigiria longos períodos de desenvolvimento e manutenção contínua.

Como alternativa, a distribuidora, poderia fornecer uma *API* para verificar a disponibilidade no estoque. Essa abordagem proporciona vários benefícios, incluindo:

- O acesso aos dados por meio de uma *API* ajuda os clientes a consolidarem informações sobre seu inventário em um único local;
- A distribuidora de livros pode fazer alterações nos sistemas internos sem causar impacto nos clientes, contanto que o comportamento da *API* não mude;

- Com uma *API* disponibilizada publicamente, os desenvolvedores que trabalham para a distribuidora de livros, os vendedores ou terceiros poderiam desenvolver uma aplicação para ajudar os clientes a encontrar os livros que procuram. Isso poderia resultar no aumento das vendas ou outras oportunidades de negócios.

Para resumir, com as *API's*, você libera o acesso aos seus recursos sem abrir mão da segurança e do controle. O desenvolvedor irá determinar como isso será feito e quem terá acesso. A segurança das *API's* depende de um bom gerenciamento. É possível conectar *API's*, bem como criar aplicações que fazem uso dos dados ou funcionalidades disponibilizadas por elas, usando uma plataforma de integração distribuída que ligue todos os elementos, incluindo sistemas legados e dispositivos de Internet das Coisas (*IoT*).

Existem três abordagens para políticas de lançamento de *API's*:

- *API Privada*: A *API* é usada apenas internamente. Isso oferece às empresas um maior controle;
- *API de Parceiros*: A *API* é compartilhada com parceiros de negócios específicos. Isso pode fornecer fluxos de receita adicionais sem comprometer a qualidade;
- *API Pública*: A *API* é disponibilizada para todos. Terceiros podem desenvolver aplicações que interajam com a sua *API* e isso pode se tornar uma fonte de inovação.

### 3.11.3. Inovação com *API*

A exposição das suas *API's* aos parceiros ou ao público pode:

- Criar novos canais de receita ou ampliar os existentes;
- Expandir o alcance da sua marca;
- Facilitar a inovação aberta ou aumentar a eficiência por meio da colaboração e de desenvolvimento externos.

Voltando para o exemplo da empresa distribuidora livros: Suponhamos que um dos parceiros da empresa desenvolva uma aplicação para ajudar as pessoas a encontrar livros nas prateleiras de livrarias. Essa experiência

aprimorada atrai mais consumidores para livraria (que é cliente da distribuidora) e amplia o canal de receita existente.

Talvez um terceiro use uma *API* pública para desenvolver uma aplicação que permita que as pessoas comprem livros diretamente da distribuidora em vez de em uma livraria. Isso abre um novo canal de receita para a distribuidora de livros.

O compartilhamento de *API's* (com parceiros selecionados ou aberta a todos) pode ter efeitos positivos. Cada parceria amplia o reconhecimento da sua marca para além dos esforços de marketing da sua empresa. Abrir a tecnologia para todos, como fazem as *API's* públicas, estimula os desenvolvedores a criar um ecossistema de aplicações com base na sua *API*. Quanto mais pessoas usam sua tecnologia, maior é a sua chance de gerar novos negócios.

Tornar a tecnologia pública pode gerar resultados novos e inesperados. Às vezes, esses resultados revolucionam setores inteiros. No caso da nossa empresa distribuidora de livros, novas organizações (um serviço de empréstimos de livros, por exemplo) podem mudar fundamentalmente a maneira de fazer negócios. Com as *API's* públicas e em parceria, você pode se beneficiar dos esforços criativos de uma comunidade muito maior do que a sua equipe interna de desenvolvedores. Novas ideias podem surgir de qualquer lugar. Por disso, as empresas precisam estar cientes das mudanças no mercado e prontas para agir. As *API's* podem ajudar (REDHAT, 2021).

#### 3.11.4. *API's* remotas

As *API's* remotas foram projetadas para interagir por meio de uma rede de comunicações. Quando falamos “remota”, queremos dizer que os recursos manipulados pela *API* estão em algum lugar fora do computador que faz a solicitação. Como a rede de comunicações mais usada é a Internet, a maioria das *API's* são projetadas com base em padrões da web. Nem todas as *API's* remotas são web, mas é justo afirmar que, em geral, as *API's web* são remotas.

As *API's web* normalmente usam o protocolo *HTTP* para mensagens de solicitação e fornecem uma definição da estrutura das mensagens de resposta.

Essas mensagens de resposta geralmente têm o formato de arquivo *XML* ou *JSON*. Tanto *XML* quanto *JSON* são formatos de preferência porque

apresentam os dados de forma simplificada, o que facilita a manipulação por outras aplicações (REDHAT, 2021).

### 3.11.5. *API's da Google*

Basicamente são um conjunto de interfaces de programação de aplicativos desenvolvidas pelo Google, que permitem a comunicação com as aplicações da empresa e sua integração a outros serviços, como:

- Pesquisa;
- Gmail;
- Tradutor;
- *Google Maps*.

Aplicativos de terceiros podem usar essas *API's* para aproveitar ou ampliar suas funcionalidades. Por meio das *API's* do *Google Maps*, por exemplo, é possível incorporar um mapa em um site, mostrando onde fica uma empresa e oferecendo a possibilidade de traçar rotas até o local.

### 3.11.6. *API do Google Maps*

Essas *APIs* fazem parte da *Google Maps Platform*, e estão divididas entre seus três produtos: *Maps*, *Routes* e *Places*. (cloud.google.com, 2021).

- *Maps*: Permite aos usuários visualizar o mundo real, por meio de mapas estáticos ou interativos, que podem ser personalizados e incorporados a sites ou aplicativos. A seguir, serão apresentadas as *API's* disponíveis no *Maps*, e suas funcionalidades:
  - *SDK do Maps para Android*: Permite adicionar mapas com base nos dados do *Google Maps* ao seu aplicativo para *Android*;
  - *SDK do Maps para iOS*: Permite adicionar mapas com base nos dados do *Google Maps* ao seu aplicativo para *iOS*;
  - *API JavaScript do Maps*: Permite adicionar mapas interativos e personalizá-los com seu próprio conteúdo e imagens para exibição em páginas da web e dispositivos móveis;

- *API Static do Maps*: Permite incorporar uma imagem estática do *Google Maps* ao seu site, sem a necessidade de usar *JavaScript* ou qualquer carregamento dinâmico de página;
  - *API Street View*: Permite incorporar um panorama ou uma miniatura estática (não interativa) do *Street View* à sua página da Web, sem o uso de *JavaScript*;
  - *URL's do Maps*: Permite criar uma *URL* universal entre plataformas, que possibilita abrir o *Google Maps* e realizar pesquisas, tendo acesso a rotas, mapas e imagens panorâmicas.
- *Routes*: Ajuda os usuários encontrarem o melhor trajeto até seu destino, com a sugestão de rotas e atualizações de trânsito em tempo real. Além disso também permite criar itinerários para até 25 pontos de referência, ideal para um planejamento de rotas de entrega mais eficiente. A seguir, serão apresentadas as *API's* disponíveis no *Routes*, e as funções de cada uma delas:
    - *API Directions*: Permite fornecer rotas de transporte público, bicicleta, carro e a pé;
    - *API Distance Matrix*: Fornece a distância e o tempo de viagem entre os pontos de origem e de destino de uma rota recomendada;
    - *API Roads*: Possibilita criar itinerários precisos, determinando o trajeto a ser percorrido, além de fornecer dados a respeito dessas estradas, como o limite de velocidade, por exemplo.
  - *Places*: Ajuda os usuários a conhecer e explorar o mundo ao seu redor. Com dados de milhões de lugares, possibilita encontrar locais específicos, usando números de telefone, endereços ou mesmo nomes de estabelecimentos. É uma ótima forma de facilitar com que os consumidores encontrem uma empresa, tendo acesso a informações detalhadas sobre o seu negócio, como nome do local, endereço, avaliações e dados de contato. A seguir, serão apresentadas as *API's* disponíveis no *Places*, e suas funções:
    - *API Places*: Exibe informações atualizadas sobre milhões de locais usando solicitações HTTP. Os locais são definidos dentro desta

API como estabelecimentos, localizações geográficas ou pontos de interesse;

- SDK do Places para Android: Permite criar aplicativos para Android com reconhecimento de localização que respondam contextualmente às empresas locais e a outros lugares próximos ao dispositivo;
- SDK do Places para iOS: Permite criar aplicativos para iOS com reconhecimento de localização que respondam contextualmente às empresas locais e a outros lugares próximos ao dispositivo;
- Places Library, API JavaScript do Maps: Permitem que o aplicativo pesquise lugares, como estabelecimentos, locais geográficos ou pontos de interesse, contidos em uma área definida, como dentro dos limites de um mapa ou em torno de determinado ponto;
- API Geocoding: Permite converter endereços em coordenadas geográficas ou vice-versa;
- API Geolocation: Permite encontrar a localização de um dispositivo com base em dados fornecidos por torres de celular e pontos de Wi-Fi;
- API Time Zone: Permite ver fuso horário de uma coordenada específica de latitude e longitude;
- API Elevation: Fornece dados de elevação de qualquer ponto do mundo, incluindo locais de profundidade, como o fundo do oceano (que retornam valores negativos). (MAPLINK.GLOBAL, 2021).

## 4. METODOLOGIA

O *hardware* do sistema é composto basicamente por três partes: Unidade de Refrigeração, Unidade de Controle e Unidade de Rastreamento. Em seguida serão abordados elementos de firmware (Unidade Controle e Unidade de Rastreamento) e do sistema *WEB*, assim como, o processo de armazenamento das informações em um banco de dados.

### 4.1. UNIDADE DE REFRIGERAÇÃO

A Unidade de Refrigeração será responsável por manter a temperatura no interior da caixa térmica com maior precisão e rapidez, além do próprio isolamento térmico da caixa. A temperatura no interior da caixa térmica será controlada pela unidade central de controle.

#### 4.1.1. Caixa Térmica

A Caixa Térmica é um dos utensílios mais versáteis e úteis que um usuário pode ter. Elas podem ser usadas para fins profissionais como transporte de medicamentos ou outros materiais que necessitam de manutenção da temperatura e também para fins pessoais e de lazer como manter bebidas geladas durante uma viagem ou passeio.

Para este projeto, foi escolhido para a confecção do protótipo uma caixa térmica da Soprano com volume interno de 12 litros (Figura 9). Esta caixa irá acomodar o material a ser transportado.

Figura 9 - Caixa térmica simples da Soprano 12L



Fonte: [www.soprano.com.br](http://www.soprano.com.br), 2021.

A Figura 10 a seguir é mostrado um modelo de caixa térmica comercial com medidor de temperatura instalado.

Figura 10 - Caixa térmica com medidor de temperatura comercial



Fonte: [www.lazershop.com.br](http://www.lazershop.com.br), 2021.

A desvantagem desse modelo acima, é que não há um sistema ativo de refrigeração no qual poderia a temperatura de resfriamento por mais tempo, assim como, uma forma que possa manter a temperatura com precisão.

Como proposta deste trabalho, o modelo terá um sistema ativo de refrigeração, assim como um *IHM* (Interface Homem-Máquina), que é integrado à unidade central de controle. Essa interface será responsável pelo monitoramento das variáveis e eventos, assim como, permitirá efetuar ajustes dos parâmetros para que o transporte da cadeia de frios seja efetuado adequadamente conforme o tipo de material a ser transportado.

#### 4.1.2. Pastilha de *Peltier*

Será responsável pela estabilidade térmica no interior da caixa. Este módulo quando acionado irá resfriar com auxílio de dissipadores térmicos o material presente no interior da caixa.

#### 4.1.3. Dissipador de Calor e Ventoinha

Será responsável por dissipar o calor do módulo *Peltier* (Figura 11). A ventoinha (Figura 12) irá proporcionar maior circulação de ar no dissipador de calor instalado na face quente da pastilha de *Peltier*, e assim, fazer com que o sistema permaneça funcional e com temperatura estável conforme o Sistema de Refrigeração permanecer energizado por longo período de tempo.

Figura 11 - Dissipador de Calor  
(Alumínio).



Fonte: [blog.novaeletronica.com.br](http://blog.novaeletronica.com.br),  
2021.

Figura 12 - Exaustor para auxiliar a  
refrigeração da Pastilha *Peltier*

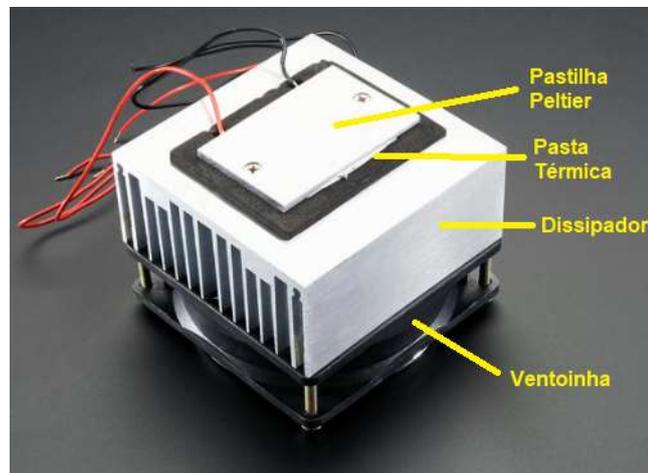


Fonte: [www.proesi.com.br](http://www.proesi.com.br), 2021.

#### 4.1.4. Montagem do Protótipo – Unidade de Refrigeração

A Figura 13 a seguir, mostra o conjunto de um sistema de refrigeração usando pastilha de *Peltier*. O conjunto consiste na própria pastilha de *Peltier*, dissipador de alumínio e Ventoinha. Devido a superfície irregular do dissipador de alumínio e da própria placa cerâmica que compõe a pastilha de *Peltier*, foi utilizado pasta térmica (da Implastec) para auxiliar na condução de calor entre as superfícies que estão em contato.

Figura 13 - Conjunto Refrigeração.



Fonte: Próprio autor.

Basicamente, esta etapa de montagem, se deve em efetuar as devidas alterações na caixa térmica comercial, e assim, através de recortes no plástico (camada interior e camada exterior), foi possível instalar a Pastilha de *Peltier* com o dissipador de alumínio conforme mostrado nas Figuras 14 e 15.a seguir:

Figura 14 - Parte interna - Instalação Pastilha Peltier



Fonte: Próprio Autor.

Figura 15 - Parte externa - Instalação Dissipador e Ventoinha



Fonte: Próprio Autor

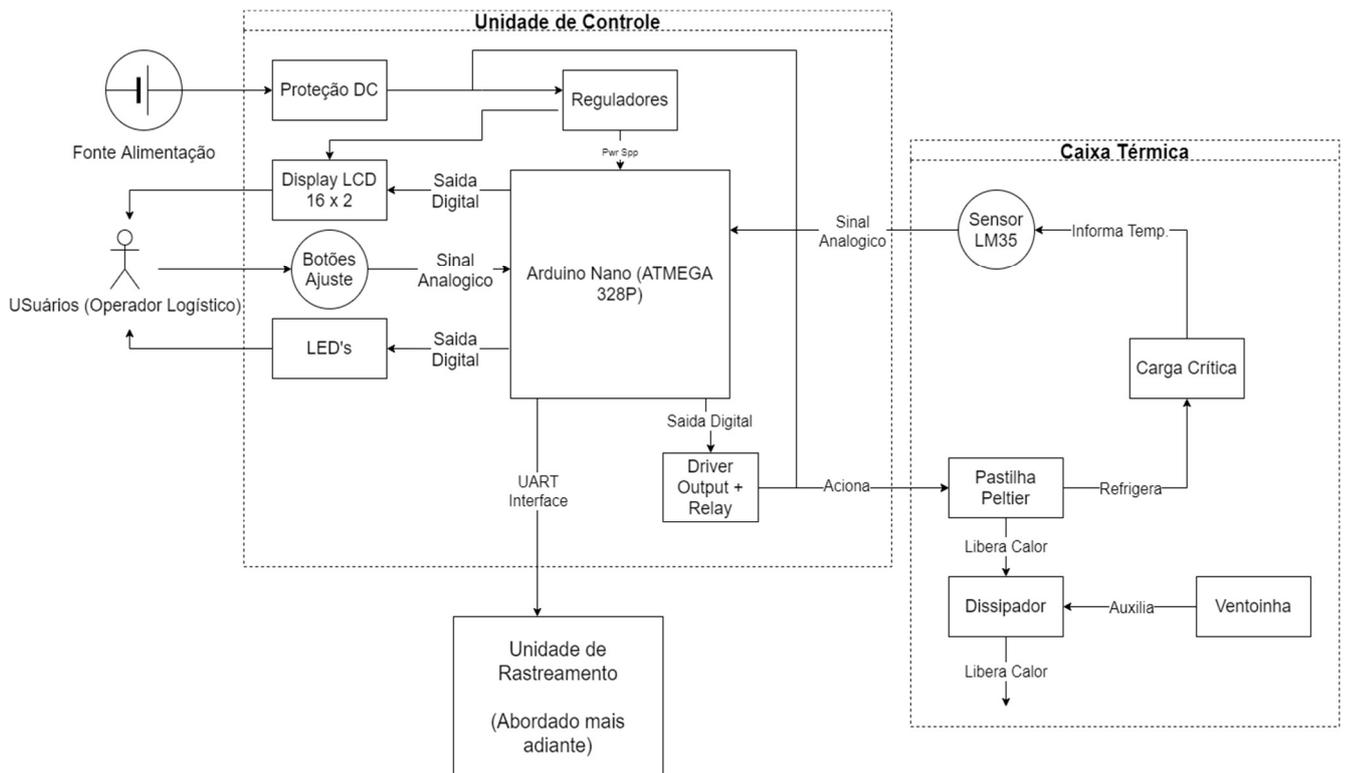
#### 4.2. Unidade Central de Controle

A Unidade Central de Controle será responsável por interpretar o sinal analógico do sensor de temperatura, definir e armazenar na memória parâmetros de ajustes, além de efetuar controle da temperatura, através de operações matemáticas, transportar de informações em tempo real, além de permitir exportar essas informações para a Unidade de Rastreamento.

Este sistema, a priori, será acoplado a uma caixa térmica, no qual haverá uma sonda instalada no interior da parte refrigerada no interior da caixa térmica e em contato com a carga crítica, para que possa obter o correto monitoramento da temperatura.

Então, para facilitar a montagem, foi preferido compor o sistema de processamento e as interfaces de entrada e saída com o usuário em um único acondicionamento. E por fim, este dispositivo de controle será instalado na posição frontal da caixa térmica. A Figura 16 a seguir, mostra o diagrama de contexto do sistema de controle e de refrigeração:

Figura 16 - Diagrama de Contexto - Unidade Controle



Fonte: Próprio autor.

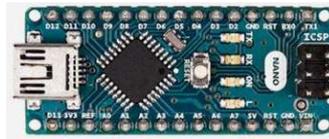
Essas Interfaces de Entrada e Saída com o Usuário, será o IHM (Interface Homem-Máquina) e esta parte conterà o display, no qual irá mostrar as informações de temperatura e ajustes de parâmetros; Os *LEDs* indicadores de alerta e os botões (*Menu*, *Up*, *Down* e *ESC*) para efetuar ajustes, conforme mostrado na Figura 17:



#### 4.2.2. Microcontrolador ATMEGA328P

É o centro de processamento da Unidade de Controle, no qual, será responsável em converter o sinal elétrico dos sensores em informações, gerar alertas e controlar o acionamento do módulo *Peltier*. A Figura 19 mostra o módulo do Arduino Nano que possui o microcontrolador ATMEGA 328p versão SMD.

Figura 19 - Arduino Nano com Microcontrolador ATMEGA328p



Fonte: <https://store.arduino.cc/usa/arduino-nano>, 2021.

#### 4.2.3. Display LCD

O display será responsável pela interface direta com todos os usuários. Pois na tela do LCD (Figura 20) mostrará os dados em tempo real (temperatura no interior da caixa, temperaturas máximas e mínimas registradas, tensão da bateria ou fonte de alimentação externa) e o status do equipamento (operacional ou não além de mensagens de alertas críticos).

Figura 20 - Display LCD alfanumérico 16 caracteres e 2 linhas.



Fonte: [www.proesi.com.br](http://www.proesi.com.br), 2021.

#### 4.2.4. Alertas Visuais e Sonoros

São indicadores luminosos (e/ou sonoros) que mostram as condições de operação do equipamento e notificações importantes. Os indicadores luminosos (Figura 21) podem ser feitos com *LED's* montados na placa e direcionados com guias de onda (*light pipe*). O principal objetivo do indicador luminoso é alertar que a temperatura máxima e/ou mínima foi atingida. Opcionalmente, poderá ser inserido um alerta sonoro (Figura 22) através de um *Buzzer*.

Figura 21 - LED como indicador visual.    Figura 22 - Buzzer como indicador sonoro.



Fonte: [www.proesi.com.br](http://www.proesi.com.br), 2021.



Fonte: [www.proesi.com.br](http://www.proesi.com.br), 2021.

#### 4.2.5.    Botões de Acionamento

Os botões de acionamento (Figura 23) serão responsáveis em realizar alguns ajustes da Unidade de Controle. Haverá quatro botões de fácil acesso para que os usuários possam ajustar o *Set Point* da temperatura, liga e desliga do *Buzzer*, alternâncias de escala *Celsius* (°C) ou *Fahrenheit* (°F);

Figura 23 - Micro Chave tipo táctil



Fonte: [www.proesi.com.br](http://www.proesi.com.br)

Apesar que este dispositivo poderia ser acionado por meio de sinais digitais, mas cada chave necessitaria uma porta digital disponível do Arduino Nano. Uma forma de economizar portas, seria fazer um *array* de chaves, combinando com divisores resistivos e assim, com apenas um único sinal (analógico) poderíamos instalar os 4 botões para ajustes.

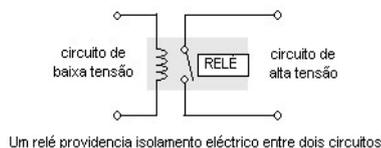
#### 4.2.6.    Relé de Acionamento

O relé de acionamento ou relé eletromecânico nesse caso, será responsável por auxiliar o microcontrolador acionar a pastilha de *Peltier*. Como a tensão de trabalho desta pastilha é superior que o microcontrolador é capaz de suportar, assim como, a corrente drenada pela pastilha é bastante elevada para a porta de saída do microcontrolador, é recomendável a utilização de um elemento intermediário, no qual escolhemos o uso do relé.

O relé também permite isolamento elétrico, aumentando a segurança (diminuindo o risco de choque elétrico para utilização de dispositivos que

trabalhem com alta tensão). A Figura 24 abaixo mostra a esquematização interna de um relé eletromecânico e a Figura 25 um modelo comercial da Metaltex:

Figura 24 - Esquematização interna de um Relé eletromecânico



Fonte: [www.profelectro.info](http://www.profelectro.info), 2021.

Figura 25 - Relé Metaltex para acionar a pastilha Peltier



Fonte: [proesi.com.br](http://proesi.com.br), 2021.

#### 4.2.7. Reguladores de Tensão

Serão responsáveis por converter níveis de tensão para que possa alimentar periféricos que não trabalham com tensões mais elevadas de uma bateria ou fonte de alimentação externa.

#### 4.2.8. Fonte de Alimentação

A fonte de alimentação será responsável pela alimentação de todo o sistema de refrigeração e controle. Como o sistema será portátil e auxiliar no transporte de materiais biológicos ou bioquímicos críticos, a ideia é que o dispositivo seja alimentado também por uma bateria de uso veicular. Garantindo assim, o funcionamento do dispositivo em casos de viagens prolongadas.

#### 4.2.9. Desenvolvimento do *Firmware* – Unidade de Controle

A programação deste dispositivo foi feita em linguagem C, de forma estruturada de modo que cumprisse as premissas a seguir:

- Efetuar leitura das portas analógicas (sensor de temperatura, tensão de entrada de alimentação e botões de ajustes);
- Efetuar operações matemáticas para que pudesse conversões de sinais em resultados de medição;
- Salvar parâmetros de ajustes na memória *EEPROM* (Como *SetPoint*, Temperatura em °C ou °F, acionamento do *backlight* do display LCD e do *Buzzer*);
- Salvar valores máximos e mínimos registrados durante a medição;

- Expor informações de parâmetros e ajustes através do display *LCD* alfanumérico;
- Acionar a saída (Pastilha *Peltier*) seguindo o algoritmo do *PID*;
- Transpor informações pela serial, à medida que recebe caracteres (Cada caractere específico retornará a informação de uma respectiva variável que está sendo processada).

A Figura 26 a seguir, mostra um exemplo parcial da programação feita no ambiente próprio do Arduino:

Figura 26 - Exemplo de Programação feita no ambiente do Arduino

```

ThermalControl_revINITIAL_20180923 | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda

ThermalControl_revINITIAL_20180923

boolean mode_t()
{
  lcd.clear();
  lcd.print("  UP - Temp C  ");
  lcd.setCursor(0, 1);
  lcd.print("  DW - Temp F  ");

  loop_cond_6 = true;

  while (loop_cond_6)
  {
    switch (entrada())
    {
      case 2: EEPROM.write(0x05, 1); //Temp graus C;
              lcd.clear();
              lcd.print("Temp. Celsius ");
              lcd.setCursor(0, 1);
              lcd.print("Sucessfull SAVE ");
              delay(displ_m_delay);
              lcd.clear();
              loop_cond_6 = false;
              break;

      case 3: EEPROM.write(0x05, 0); //Temp graus F;
              lcd.clear();
              lcd.print("Temp. Fahrenheit");
              lcd.setCursor(0, 1);
              lcd.print("Sucessfull SAVE ");}
    }
  }
}

```

364 Arduino Nano, ATmega328P em COM8

Fonte: Próprio autor.

#### 4.2.10. Montagem do Protótipo – Unidade de Controle

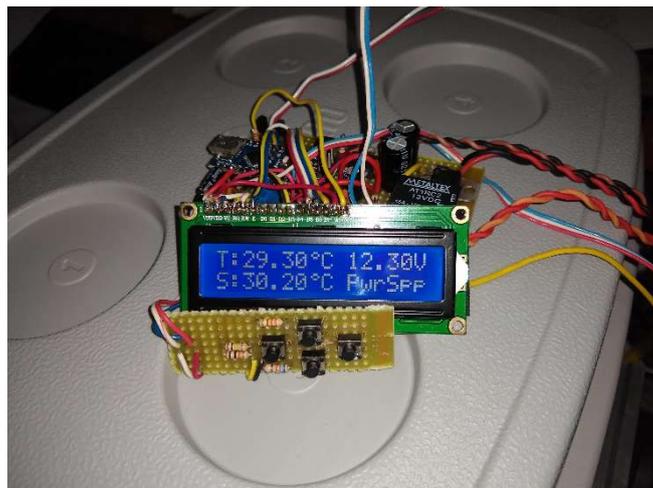
Também visando menor tempo para executar a montagem e menor custo para o protótipo, a Unidade de Controle foi montada em uma placa de circuito impresso padrão (Figura 27). E a caixa plástica irá proteger o circuito da umidade gerada pelo próprio sistema de refrigeração. A Figura 28 mostra o dispositivo em funcionamento.

Figura 27 - Montagem da Unidade Central em Placa Padrão



Fonte: Próprio Autor.

Figura 28 - Testes Iniciais com a Unidade de Controle



Fonte: Próprio Autor.

Com a integração entre a Unidade Central e a sonda (sensor de temperatura LM35), foi constatado o funcionamento pleno com 1 litro de água desmineralizada em diferentes níveis de temperatura. E com o auxílio de um termômetro digital Minipa MT320A e um multímetro Icel MD-6120 com termopar. O erro de leitura foi de apenas +/- 0.5°C, sendo assim, não foi necessária compensação para calibração.

A Figura 29 a seguir, mostra a integração da Unidade de Refrigeração com a Unidade de Controle e em funcionamento:

Figura 29 - Montagem Integração Unidade de Refrigeração e Controle



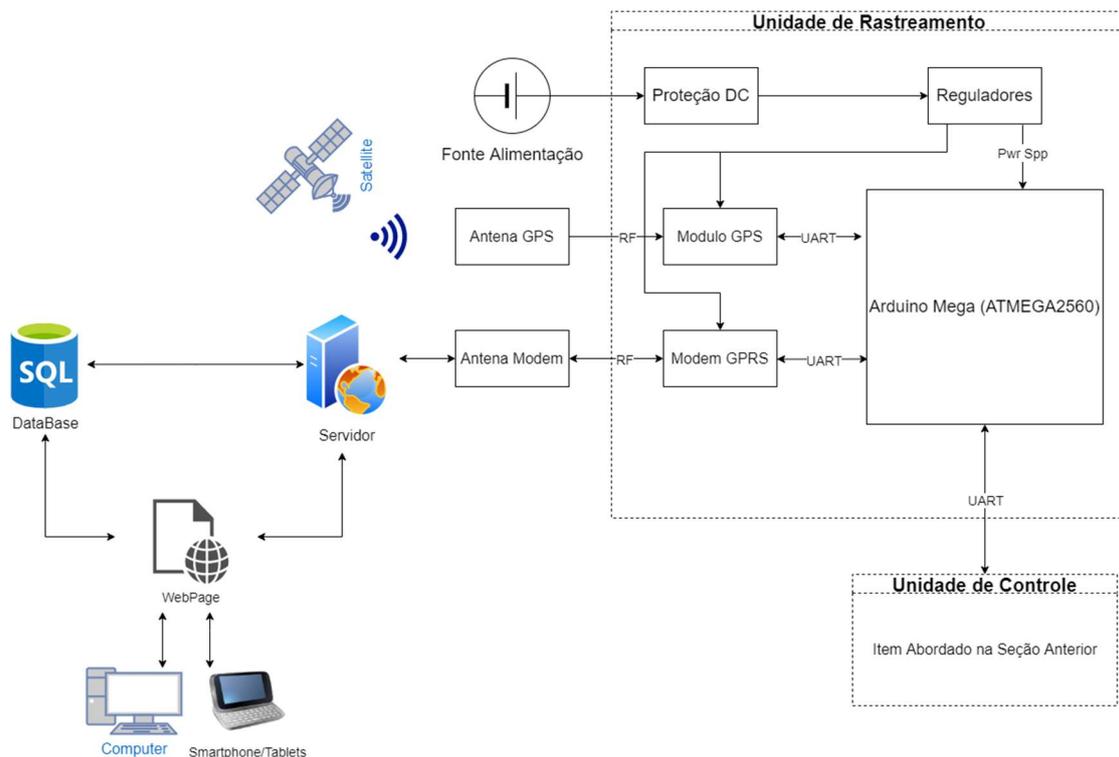
Fonte: Próprio Autor

#### 4.3. UNIDADE DE RASTREAMENTO

A Unidade de Rastreamento será responsável por coletar informações da Unidade de Controle (Temperatura no interior da Caixa Térmica, *SetPoint* de Temperatura, Temperaturas máximas e mínimas registradas, *Status* de alarmes), como também irá coletar informações de Geolocalização do módulo *GPS* (latitude, longitude e velocidade de deslocamento), esta unidade também será responsável em estabelecer comunicação *GPRS*, através de um modem e assim, irá estabelecer comunicação com um servidor *WEB* para que as informações possam ser armazenadas e mostradas em tempo real.

A Figura 30 a seguir, mostra um diagrama de contexto da Unidade de Rastreamento. Assim será mais fácil de identificar os elementos descritos aqui:

Figura 30 - Diagrama de Contexto - Unidade de Rastreamento

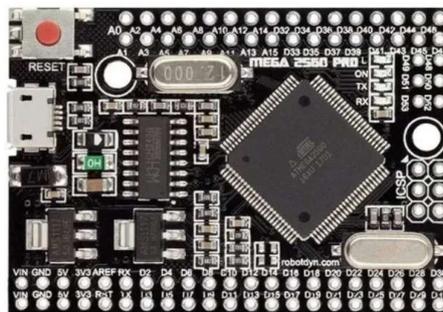


Fonte: Próprio autor.

#### 4.3.1. Microcontrolador ATMEGA2560

É o centro de processamento da Unidade de Rastreamento. Este dispositivo será responsável em estabelecer comunicação entre a Unidade de Controle e o sistema *WEB*, além de estabelecer comunicação com o modem e o módulo *GPS*. A Figura 31 a seguir, mostra o módulo do Arduino Mega PRO Mini que possui o microcontrolador ATMEGA 2560.

Figura 31 - Arduino PRO Mini com microcontrolador ATMEGA2560



Fonte: arducore.com.br, 2021.

O principal motivo de escolha deste dispositivo se deu pelo fato de ter uma ótima capacidade de processamento e armazenamento com um custo

baixo. Além da facilidade de implementação e expansão e também por possuir quatro interfaces de comunicação *UART*.

Esta versão também possui um clock externo que pode operar a 16MHz, além de possuir 70 portas de entrada e saídas digitais, 16 entradas analógicas, 14 portas *PWM* (*Pulse With Modulation*) e canais de comunicação *SPI* e *I2C*. Por ser um modelo simples, a interface de gravação e *debug*, é feito por meio de um conversor de sinais *USB/TTL* com o circuito integrado CH340.

#### 4.3.2. Módulo GPS

O modelo de dispositivo *GPS* escolhido para confeccionar o protótipo da Unidade de Rastreamento, é o uBlox Neo 6m. A Figura 32 a seguir, mostra um modelo comercial já soldado em uma placa de circuito impresso, vendido em forma de kit didático, com conexão própria para antena, além de um regulador de tensão para que possa ser alimentado corretamente. A interface de comunicação com o microcontrolador é feito via *UART*.

Figura 32 - Módulo GPS uBlox Neo 6m com antena



Fonte: <https://www.baudaeletronica.com.br/>, 2021.

Com este periférico, será possível determinar o posicionamento global do dispositivo (latitude e longitude), data e hora, além da velocidade de deslocamento.

#### 4.3.3. Modem *GPRS*

O Modem SIM800L (Figura 33) é utilizado para comunicação via dados *GSM/GPRS*, necessita de um *chip* (*simcard*) de operadora de telefonia móvel para comunicação. O módulo pode ter suas ações controladas por diversos tipos de microcontroladores, como o Arduino por exemplo. A Figura 34 mostra a antena compatível para utilizar com este modem.

Figura 33 - Modem SIM800L



Fonte: [www.byteflop.com.br](http://www.byteflop.com.br)

Figura 34 - Antena para modem GSM/GPRS

3dBi



Fonte: <https://www.autocorerobotica.com.br/>

As principais características desse modem são:

- Compatibilidade com redes Quad-band 850/900/1800/1900MHz;
- Conecta-se em qualquer rede GSM global com qualquer SIM 2G;
- Faz e recebe chamadas de voz usando um fone de ouvido ou um alto-falante externo 8 Ohms e microfone do tipo eletreto;
- Enviar e receber mensagens SMS;
- Enviar e receber dados GPRS (TCP/IP, HTTP, etc.);
- Digitalizar e receber transmissões de rádio FM;
- Levar a campainha e a porta de controle do motor de vibração;
- Atende rede 2G;
- Interface com comandos atômicos (comandos AT) com a detecção de "auto de transmissão";

#### 4.3.4. Desenvolvimento do *Firmware* – Unidade Rastreamento

Como o microcontrolador escolhido é o ATMEGA 2560, utilizado no Arduino Mega, por questão de praticidade, o desenvolvimento da programação que compõe o controle do sistema também foi desenvolvido sob o compilador de programa do próprio Arduino.

Além da facilidade de programação, essa interface é totalmente gratuita, além de permitir com facilidade implementações de novas funcionalidades, desde que respeite as capacidades de processamento e memória do microcontrolador escolhido.

A programação deste dispositivo também foi feita em linguagem C, de forma estruturada de modo que cumprisse as premissas a seguir:

- Efetuar inicialização e requisições do Modem GPRS;

- Receber parâmetros (latitude, longitude velocidade, data e hora) do módulo *GPS*;
- Requisitar ao Módulo de Controle (que contém o Arduino Nano) as variáveis das informações coletadas da caixa térmica, além dos parâmetros de ajustes;
- Encapsular as mensagens (gerando *URL*) de todos os periféricos e enviar informações de variáveis para uma página *WEB* (Esta página em específico, irá salvar os parâmetros em um banco de dados);

A Figura 35 a seguir, mostra um exemplo do *LOG* da unidade rastreamento requisitando informações da unidade de controle e do módulo *GPS* (as tarjas amarelas são para suprimir algumas informações pessoais):

Figura 35 - *LOG* da Unidade de Rastreamento

```

COM3
Latitude= [redacted]
Longitude= [redacted]
Velocidade (Km/h)= 0
Temperature (°C)= 25.88
Set Temp. (°C)= 18.10
Max. Temp. (°C)= 28.32
Min. Temp. (°C)= 25.39
Voltage Input Power= 12.30
Buzzer Status Menu= 1
Backlight Status Menu= 1
http://thermalcontrolweb.000webhostapp.com/gpsdata.php?lat=[redacted]&lng=[redacted]&spd=0
AT+CFUN=1
OK
AT+CGATT=1
OK
AT+SAPBR=3,1,"Contype","GPRS"
OK
AT+SAPBR=3,1,"AFN","g.claro.com.br"
OK
AT+SAPBR=1,1
ERROR

AT+HTTPIINIT
ERROR

AT+HTTTPARA="CID",1
OK
hermalcontrolweb.000webhostapp.com/gpsdata.php?lat=[redacted]&lng=[redacted]&spd=0"
OK
AT+HTTIPACTION=0
OK

AT+HTTIPIERM
ERROR

AT+CIPSHUT

```

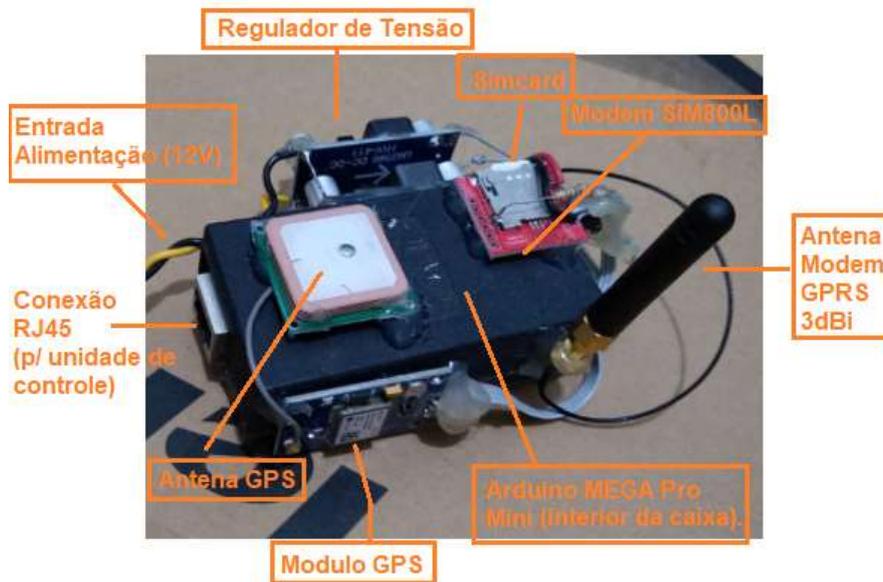
Auto-rolagem  Show timestamp  Nova-linha  115200 velocidade  Deleta a saída

Fonte: Próprio autor.

#### 4.3.5. Montagem do Protótipo – Unidade de Rastreamento

Conforme mostrado no item anterior, este dispositivo também foi construído visando menor tempo para executar a montagem e menor custo para o protótipo. A Unidade de Rastreamento foi montada em forma modularizada (Figura 36). E a caixa plástica irá proteger os circuitos contra choques mecânicos.

Figura 36 - Protótipo da Unidade de Rastreamento



Fonte: Próprio autor.

#### 4.4. Sistema *WEB* para este dispositivo

Conforme a ideia da construção dos *hardwares*, o Sistema *WEB* foi pensado para que fosse de rápido implementação e que entregasse resultados.

A primeira coisa que precisou ser feita, foi estabelecer quais parâmetros deveriam ser mostrados na página. E isso precisou que fosse feito juntamente com a programação da Unidade de Rastreamento.

Uma vez que a Unidade de Rastreamento requisita as informações da Unidade de Controle e do Módulo *GPS*, o sistema irá preparar o modem para transmitir essas informações para o servidor. Devido a limitação do próprio Modem no envio das mensagens, foi decidido que fosse feito através de 3 pacotes de mensagens. Então, foi criado três arquivos (páginas *WEB* em *PHP*)

para que cada página possa receber pacotes de mensagens. A Figura 37 a seguir mostra uma dessas páginas, por exemplo:

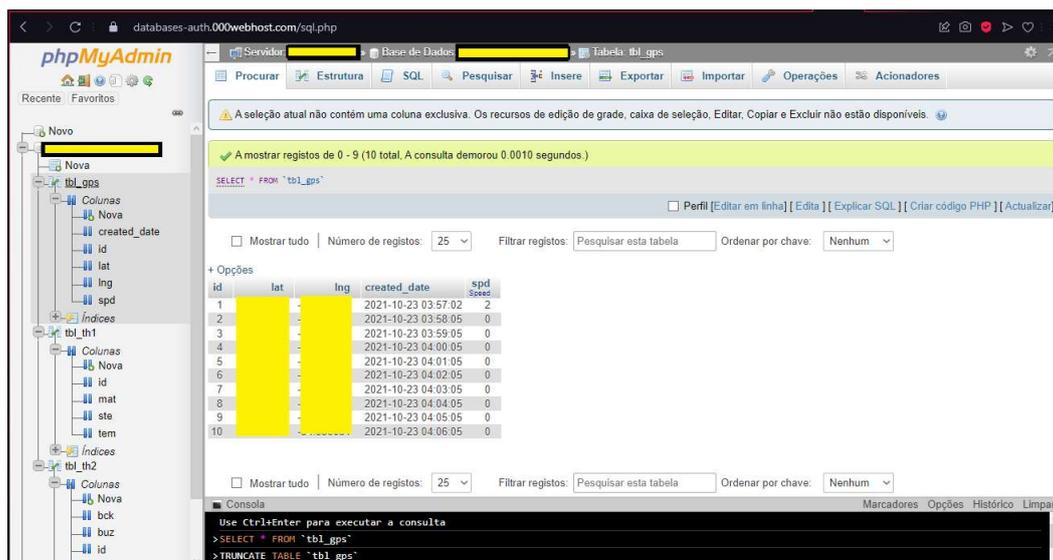
Figura 37 - Página *WEB* que receberá pacote de mensagens

```
× gpsdata.php
public_html > gpsdata.php
1 <?php
2
3     require 'config.php';
4
5     $lat = $_GET['lat'];
6     $lng = $_GET['lng'];
7     $spd = $_GET['spd'];
8
9     echo $lat;
10    echo "<br>";
11    echo $lng;
12    echo "<br>";
13    echo $spd;
14    echo "<br>";
15
16
17    $sql = "INSERT INTO tbl_gps(lat,lng,created_date,spd)
18    VALUES('".$lat."','".$lng."','".$date("Y-m-d H:i:s")."', '".$spd."')";
19
20    if($db->query($sql) === FALSE)
21        { echo "Error: " . $sql . "<br>" . $db->error; }
22
23    echo "<br>";
24    echo $db->insert_id;
```

Fonte: Próprio autor.

Então, cada página *WEB* irá escrever tabelas previamente configuradas no banco de dados. A Figura 38 a seguir mostra um exemplo como é recebido as informações no banco de dados (as tarjas amarelas são para suprimir algumas informações pessoais):

Figura 38 - Tela do Banco de Dados



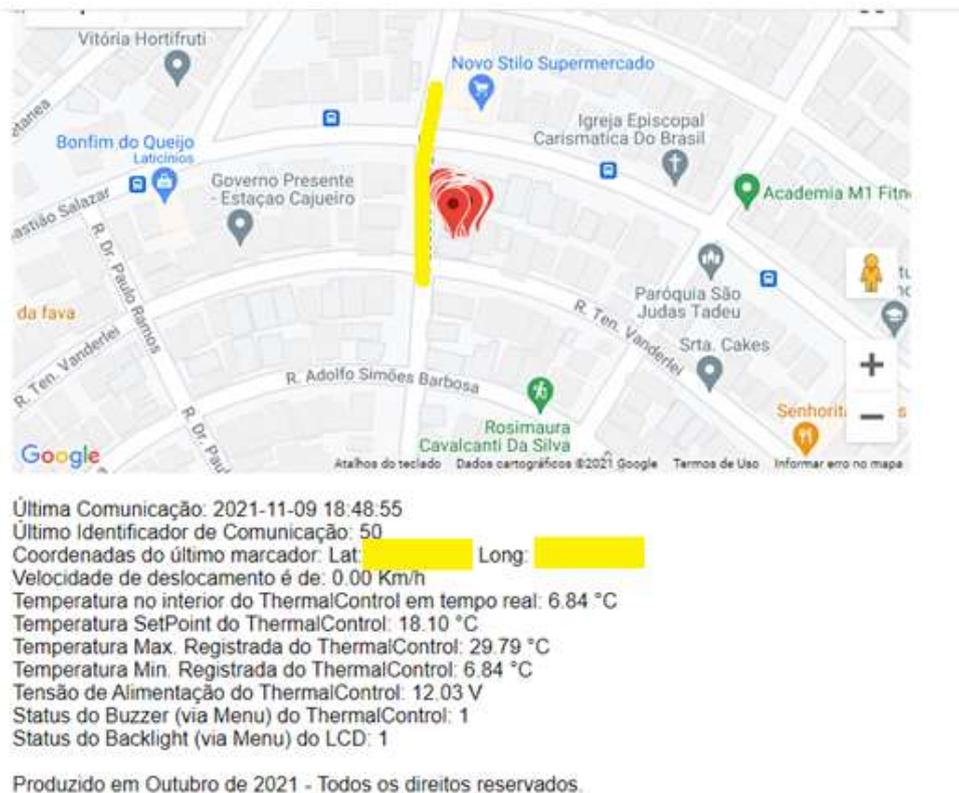
Fonte: Próprio autor.

Cada tabela tem o seu identificador e gera informações de data e hora do sistema.

- Tabela `tbl_gps`: irá receber as informações de variáveis de latitude, longitude e velocidade de deslocamento do dispositivo;
- Tabela `tbl_th1`: irá receber as informações de variáveis de temperatura, setpoint e temperatura máxima registrada;
- Tabela `tbl_th2`: irá receber as informações de variáveis de temperatura mínima registrada, tensão de alimentação e os status de *backlight* do display *LCD* e *buzzer*, configurados pelo Menu de ajustes da Unidade de Controle.

Já na página inicial (*index.php*), basicamente, será a requisição das informações armazenadas no banco de dados. A Figura 39 a seguir, mostra um exemplo real da requisição dessas informações do banco de dados:

Figura 39 - Exemplo real da página inicial



Fonte: Próprio autor.

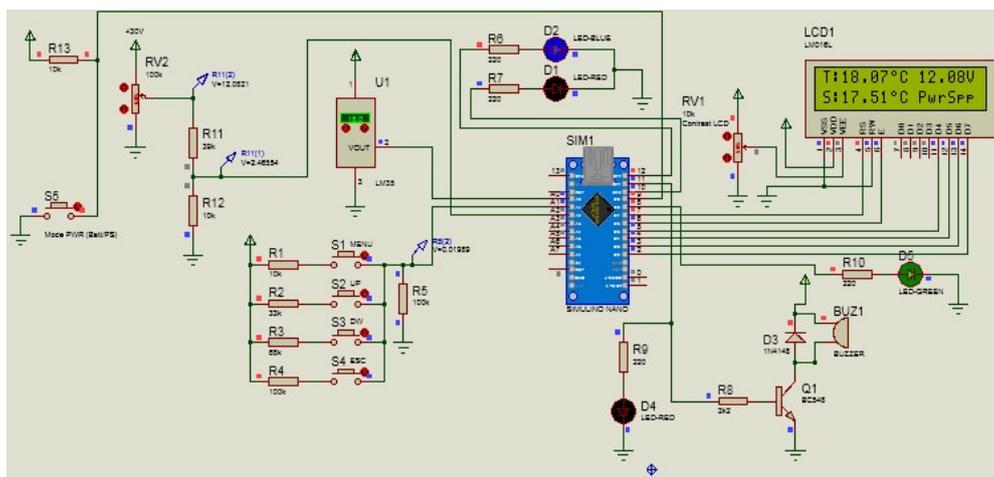
Para utilizar o *API* do *Google Maps*, foi necessário efetuar cadastro como desenvolvedor. E esse serviço não é gratuito.

## 5. RESULTADOS E DISCUSSÃO

Visando menor tempo de execução e desenvolvimento de *hardware* e *software*, a priori, foi decidido que os primeiros testes fossem executados em um ambiente de simulação para a unidade de controle para que posteriormente convergir para a montagem da unidade de refrigeração do sistema.

Dessa forma, foi possível observar e testar com um simulador as funções de conversões dos sinais elétricos. A priori, foi utilizado o programa *Proteus*, no qual é possível editar o diagrama esquemático, simular o funcionamento de alguns componentes e até mesmo confeccionar placa de circuito impresso. Mas a exploração deste programa ficou restrita apenas na simulação do sistema de controle, conforme mostra a Figura 40 abaixo:

Figura 40 - Simulação da Unidade Central usando o programa *Proteus*.



Fonte: Próprio autor.

Nesse ambiente foi comprovado o funcionamento do sensor de temperatura, no qual, não foi necessário um processo de calibração direta, já que o sinal do sensor LM35 é linear ( $10\text{mV}/^\circ\text{C}$ ) e bastante preciso.

Foi possível também comprovar o funcionamento do algoritmo de ajustes de parâmetros, assim como, o armazenamento das informações mesmo após o desligamento do dispositivo. E também os testar os eventos que irão habilitar os alertas visuais e sonoros nas condições que foram descritos anteriormente na seção 4.2.4. Além, dos elementos secundários, como atuação das chaves e display *LCD*.

Os testes a seguir são relatos obtidos após a montagem do protótipo.

Foram necessários alguns ajustes finais no sinal analógico proveniente do teclado de ajustes no protótipo da Unidade de Controle do sistema,

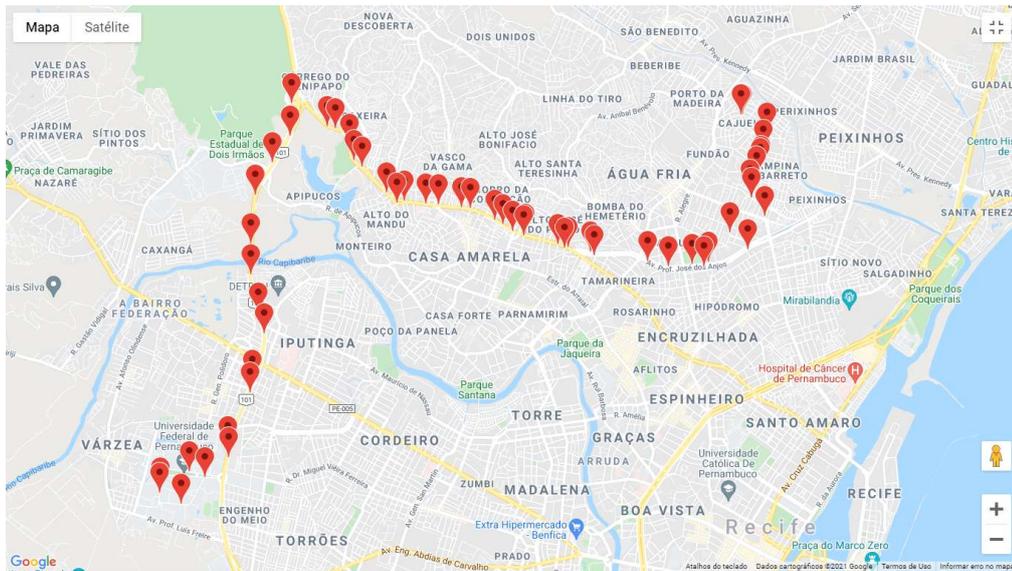
A Figura 29, da seção 4.2.10 deste material, mostra o funcionamento dos dispositivos (Unidade de Refrigeração e Unidade de Controle) sendo testado em bancada. Nesta condição, sem carga fria no interior da caixa térmica, o tempo de estabilização da temperatura vai depender da temperatura escolhida no *SetPoint* (na configuração do sistema na unidade de controle). Quanto menor a temperatura, maior será o tempo de estabilização. Durante os testes, o tempo mínimo foi de 40 minutos para *SetPoint* de 18°C e 2 horas para *Set Point* de 10°C. E a corrente média total drenada pela fonte foi de 2,60 Amperes. (A temperatura ambiente estava em 26,5°C).

O escopo deste trabalho foi apenas de validar a ideia proposta, não visando estritamente a busca pela melhor eficiência. Mas é possível aumentar a eficiência de refrigeração do sistema associando em paralelo as pastilhas *Peltier*. Entretanto, o consumo de energia irá aumentar proporcionalmente. Ou então, utilizar um sistema de refrigeração com compressores de refrigeradores, que são projetados para este fim.

Na seção 4.4 deste material, foi abordado sobre o Sistema *WEB* para este dispositivo. As requisições e as respostas das informações foram dadas de formas diretas, ou seja, sem necessidade de conversões extras.

A Figura 41 a seguir, mostra exemplo de trajeto mostrado pelo Sistema *WEB* (da minha residência até a Cidade Universitária e depois o caminho de volta). Os marcadores indicados no mapa abaixo são reais e foram gerados automaticamente pelo sistema *WEB* com *API* do *Google Maps*.

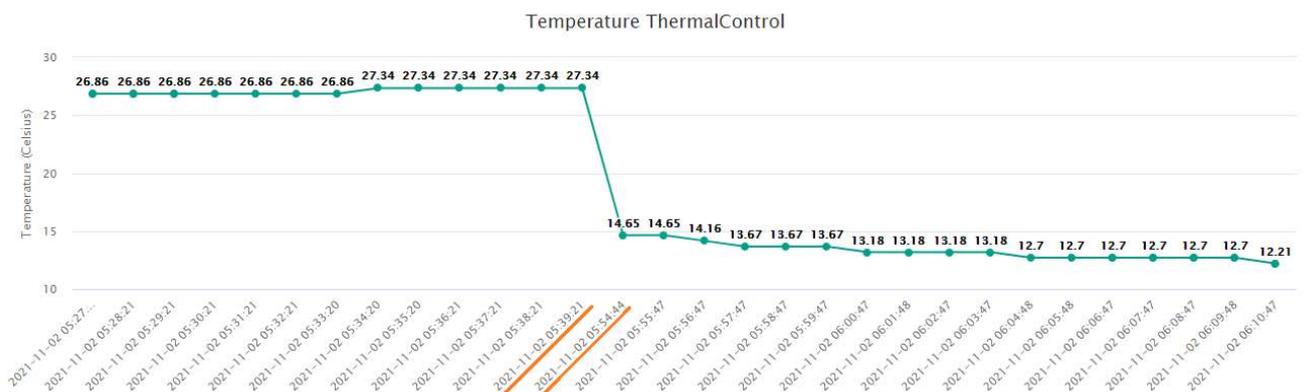
Figura 41 - Exemplo de Trajeto sendo mostrado no Sistema WEB



Fonte: Próprio autor.

Para ter uma função similar a um *DataLog* ou *DataLogger*, a Figura 42 abaixo mostra um gráfico com informações reais da temperatura no interior da caixa térmica (também gerada por uma página WEB, criado para este projeto). A priori tentei usar uma API do *Google Chart*. Mas tive dificuldades na integração dos dados, e por fim, terminei utilizando uma API da “code.highcharts.com”.

Figura 42 - Primeiras medições de Temperatura - Sistema WEB

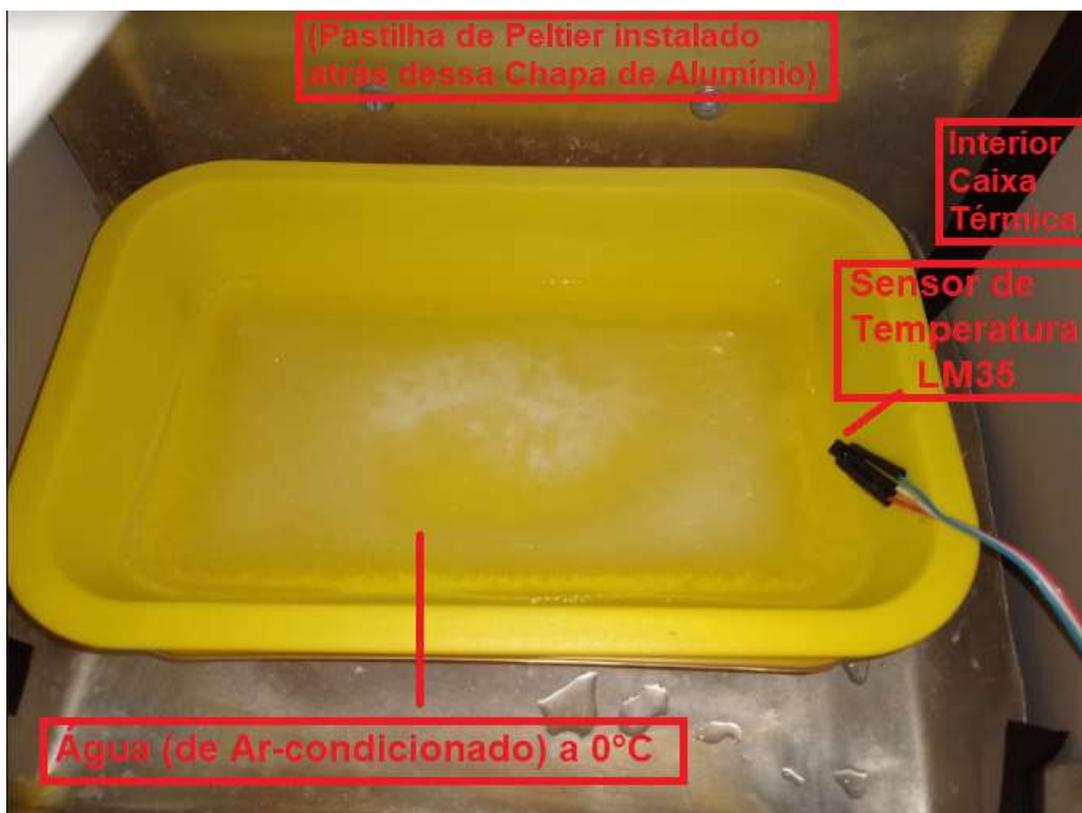


Fonte: Próprio autor.

Este gráfico acima mostra o monitoramento da temperatura, inicialmente do ambiente externo, e logo em seguida, foi inserido uma caixa plástica com gelo (água desmineralizada a temperatura 0°C) no interior da caixa térmica (o sensor não estava em contato com a carga durante este teste inicial).

A Figura 43 a seguir, mostra o interior da caixa térmica, com água desmineralizada (drenada de ar-condicionado) em um recipiente plástico. Neste teste, o sensor não estava em contato com a carga fria. A ideia desse teste é validar a leitura do sensor e observar as medições no sistema *WEB*. Tanto que a pastilha de *Peltier* foi desligada nesse teste).

Figura 43 - Teste de validação do Sensor e Leituras pelo Sistema *WEB*

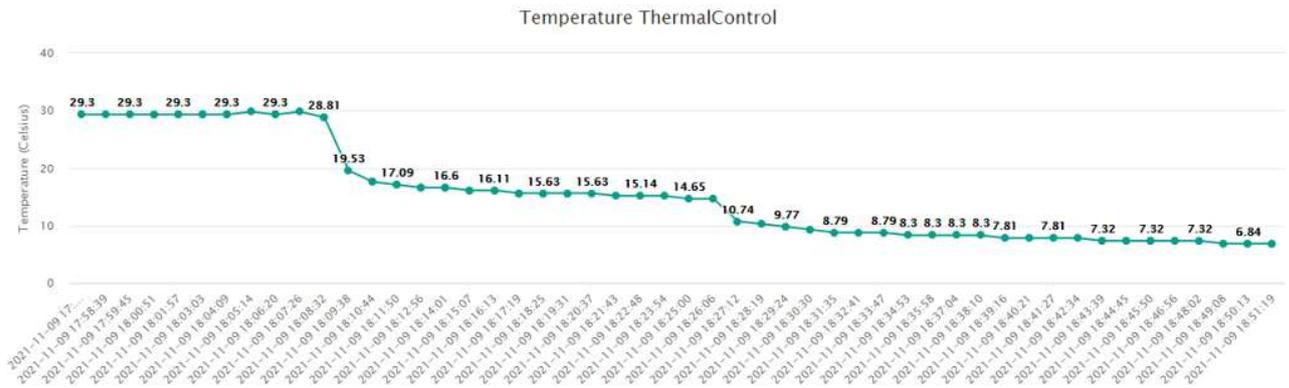


Fonte: Próprio autor.

Então, voltando para a Figura 44, vale salientar que o sistema *WEB* atualizava e mostrava as informações a cada 1 minuto. O sistema de rastreamento subia as informações para o sistema *WEB*. Ao desligar o dispositivo no dia 02/11/2021 as 05:39 da manhã e depois religar as 05:54, o sistema foi capaz de processar e atualizar as informações de imediato. O envio das informações permaneceu para que haja atualização a cada 1 minuto. Todas as informações são guardadas no banco de dados, embora que na tela, para este teste, mostrava apenas as últimas 30 medições de temperatura.

Já na Figura 44 a seguir, houve alteração no sistema *WEB* para que pegasse as 50 últimas medições, e a carga fria sendo inserida sem desligar o dispositivo. Dessa vez, o sensor de temperatura estava encostado a carga fria.

Figura 44 - Segunda etapa de testes com Carga Fria.



Fonte: Próprio autor.

As condições desse teste ocorreram da seguinte forma: Até as 17:58, o sistema de refrigeração estava desligado e sem carga fria e a temperatura no interior da caixa era a mesma da temperatura ambiente (29,3°C). Entre 18:08 e 18:09, com a unidade de refrigeração energizada e a carga fria inserida no interior da caixa com a tampa fechada, a temperatura caiu de 28,81°C para 19,53°C.

Outro marco significativo foi as 18:26 (17 minutos depois) onde a temperatura no interior da caixa caiu de 14,65°C para 10,74°C no intervalo de 1 minuto. No final deste conjunto de medições, as 18:51, a temperatura caiu para apenas 6,54°C.

Nesse teste, o *SetPoint* até foi configurado para 10,0°C. O desligamento do sistema de refrigeração ocorreu normalmente. Como a carga fria era um bloco de gelo, a temperatura continuou a cair, entretanto, os alarmes de não conformidade entraram em ação. Sendo assim, foi possível observar todos os parâmetros registrados, armazenar as informações em um banco de dados e visualizá-los com facilidade pela página *WEB*.

O sítio real deste projeto, para ter acesso às informações geradas em tempo real, poderá ser visto aqui: <https://thermalcontrolweb.online/>

Este sítio requereu a compra do domínio e a hospedagem foi feita no *Hostinger*.

## 6. CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentado uma solução onde os dispositivos do sistema de refrigeração, composto por uma pastilha *Peltier* e uma caixa térmica comum, foram utilizados em um aparato experimental simples para fazer a estabilização térmica, onde é necessária para o transporte adequado de uma carga crítica de uma cadeia de frios.

Assim como, foi demonstrado o funcionamento do dispositivo de controle de temperatura do sistema, e também, os dispositivos utilizados no sistema de rastreamento, ambos formaram uma excelente ferramenta no auxílio na cadeia de frios. Todos os dispositivos físicos foram obtidos facilmente em mercado nacional, permitindo assim qualquer estudante ou entusiasta (não restringindo apenas os estudantes de engenharia ou profissionais) possam vivenciar diferentes aspectos da montagem laboratorial.

O Operador logístico da cadeia de frios será responsável pelo gerenciamento da operação do transporte e armazenamento de bens de alto valor cuja estabilidade térmica deve ser cuidadosamente controlada. O Transportador atuará na operação de transporte e armazenamento dessas cargas. Ambos, poderão ter acesso aos dados providos pelo sistema de telemetria. Mas o operador logístico será o responsável por verificar se o sistema supre suas necessidades. Como também, o operador assegurará que a utilização do sistema será sustentável, realizável e factível.

Neste trabalho é apresentado um experimento didático, construído com materiais de baixo custo que permite medidas precisas de temperatura e assim, com menor erro possível de controle da mesma. Foi obtido estabilidade menor que 1% na potência de refrigeração, corrigindo dessa forma a deriva térmica da leitura. Assim como, o dispositivo de rastreamento agregado ao sistema de controle foi capaz de informar o posicionamento global, velocidade de deslocamento e os parâmetros no interior da caixa térmica com precisão. Assim como, os eventos inesperados poderão ser acompanhados facilmente pelo operador logístico de cargas críticas, agilizando assim, medidas mais rápidas e eficientes para contornar os problemas que acerbam no transporte.

## 7. REFERÊNCIAS

SANCHES, Cristina. Os desafios da cadeia de frios na indústria farmacêutica. **Labnetwork**, 2015. Disponível em: <<http://www.labnetwork.com.br/especiais/destaque/os-desafios-da-cadeia-de-frio-na-industria-farmaceutica/>>. Acessado em: 12 de out. de 2021.

ANVISA. Transplantes: Anvisa publica normas para transporte de órgãos, 2015. Disponível em: <<https://www.gov.br/anvisa/pt-br/assuntos/noticias-anvisa/anos-anteriores/transplantes-anvisa-publica-normas-para-transporte-de-orgaos>>. Acessado em: 12 de out. de 2021.

ANDRIOLI, Lívia Meneghel. Transporte de Órgãos para Transplante., **Semana Acadêmica**, 2015. Disponível em: <[semanaacademica.org.br/system/files/artigos/transporte\\_de\\_orgaos\\_para\\_transplantes\\_0.pdf](http://semanaacademica.org.br/system/files/artigos/transporte_de_orgaos_para_transplantes_0.pdf)>. Acessado em 12 de out. de 2021.

RVIMOLA. Cadeia Fria e os Desafios para a Indústria Farmacêutica. **RV Imola**, 2019. Disponível em: <[rvimola.com.br/blog/cadeia-fria-e-os-desafios-para-a-industria-farmaceutica](http://rvimola.com.br/blog/cadeia-fria-e-os-desafios-para-a-industria-farmaceutica)>. Acessado em: 12 de out. de 2021.

VACINAS. Desafios da Distribuição e Armazenamento de Vacinas no Brasil. **Vacinas**, 2021. Disponível em: <[vacinas.com.br/blog/desafios-da-distribuicao-e-armazenamento-de-vacinas-no-brasil](http://vacinas.com.br/blog/desafios-da-distribuicao-e-armazenamento-de-vacinas-no-brasil)>. Acessado em: 12 de out. 2021.

SEGUNDA lei da termodinâmica. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Segunda\\_lei\\_da\\_termodinamica](https://pt.wikipedia.org/wiki/Segunda_lei_da_termodinamica)>. Acessado em: 14 out. 2021.

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. **Fundamentos da Física**. Vol. 2. 7.ed. São Paulo: Editora LTC, 2006.

VAN WYLEN, G.J.; SONNTAG, R.E.; BORGNAKKE, C. **Fundamentos da Termodinâmica**. 5. ed. São Paulo: Editora Edgard Blucher Ltda., 1998.

LANGSTON, Lee S. Efficiency by the Numbers. **Mechanical Engineering Magazine**. Website, Web Exclusives, 2008. p.1-3

EFEITO Peltier. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Efeito\\_Peltier](https://pt.wikipedia.org/wiki/Efeito_Peltier)>. Acessado em: 12 de out. 2021.

PELTIER. Teoria sobre as Pastilhas Peltier. **Peltier**, 2021 Disponível em: <<http://www.peltier.com.br/>>. Acesso em: 14 de out. 2021.

HEBEILTD. Datasheet Módulo Peltier. **Hebeiltd**, 2021. Disponível em: <<http://www.hebeiltd.com.cn/peltier.datasheet/TEC1-12706.pdf>>. Acessado em: 14 de out. de 2021.

LEI de Joule. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Lei\\_de\\_Joule](https://pt.wikipedia.org/wiki/Lei_de_Joule)>. Acessado em: 12 de out. 2021.

ALMEIDA, R. M.; MORAES, C. H.; SERAPHIM, T. F. **Programação de Sistemas Embarcados - Desenvolvendo Software para Microcontroladores em Linguagem C**. 1.ed. São Paulo: Editora LTC, 2016.

MACHADO, Luiz Paulo; MAIA, Francis Berenger. **Arquitetura de Sistemas Operacionais**, 4.ed. São Paulo: Editora LTC, 2016.

HIRAKAWA, A. R. Interrupções - Laboratório de Microprocessadores - PCS, USP, 2004. Disponível em: <[http://sites.poli.usp.br/d/pcs2529/index\\_arquivos/2529e052005.pdf](http://sites.poli.usp.br/d/pcs2529/index_arquivos/2529e052005.pdf)>. Acessado em: 18 de out. de 2021.

TANENBAUM, A. S.; FILHO, M. **Sistemas Operacionais Modernos**. 3. ed. [S.I.]: Prentice-Hall, 1995.

LIMA, Thiago. AGC - O primeiro grande Sistema Embarcado. **Embarcados**, 2014. Disponível em: <<https://www.embarcados.com.br/agc-primeiro-grande-sistema-embarcado/>>. Acessado em: 13 de out. de 2021.

SANTOS, Geise; BERTOLETI, Pedro. Prototipação: Plataformas de Hardware e Software para Sistemas Embarcados. **Embarcados**, 2021. Disponível em: <<https://www.embarcados.com.br/prototipacao-plataformas-de-hardware-e-software-para-sistemas-embarcados/>>. Acessado em 13 de out de 2021.

OLIVEIRA, A. S.; ANDRADE, F. S. **Sistemas embarcados: Hardware e Firmware na Prática**. 1.ed. Editora Érica, 2009.

MOYER, Bryon. **Real World Multicore Embedded Systems**, 1.ed. Editora. Newnes, 2013.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5.ed. Editora Pearson, 2010.

SISTEMA de Controle. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Sistema\\_de\\_controle](https://pt.wikipedia.org/wiki/Sistema_de_controle)>. Acessado em: 12 de out. 2021.

TEXAS INSTRUMENTS. Datasheet do Sensor LM35. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acessado em: 14 de out. 2021.

PROFELECTRO. Relé Electromagnético. **Profelectro**, 2012. Disponível em: <<http://www.profelectro.info/rele-electromagneticoelectromecanico-principio-de-funcionamento>>. Acessado em: 14 de out. de 2021.

MONK, Simon. **Programação com Arduino - Começando com Sketches**. 2.ed. Editora Bookman, 2017.

SILVEIRA, Cristiano Bertulucci. Controle PID de Forma Simples e Descomplicada. **Citisystems**, 2016. Disponível em: <<https://www.citisystems.com.br/controle-pid>>. Acessado em 16 de out. de 2021.

CONTROLADOR proporcional integral e derivativo. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Controlador\\_proporcional\\_integral\\_derivativo](https://pt.wikipedia.org/wiki/Controlador_proporcional_integral_derivativo)>. Acessado em: 16 de out. 2021.

GONÇALVES, D. W.; SILVA, M. J. N.; VALENTE, T. Construindo um Climatizador de Ar. **Projeto Garagem**, 2015. Disponível em: <<https://projetogaragem.wordpress.com/2015/10/23/construindo-um-climatizador-de-ar-portatil>>. Acessado em 16 de out. de 2021.

FERREIRA, Henrique C. Controle Digital. **Universidade de Brasília**, 2020. Disponível em: <<http://www2.ene.unb.br/henrique/CDig/introdu%C3%A7%C3%A3o.pdf>>. Acessado em 17 de out. de 2021.

DORF, Richard C; BISHOP, Robert H. **Sistemas de Controle Modernos**. 8.ed. Editora LTC, 2018.

WEST control solutions. O que é Controle PID? **West-cs**, 2012. Disponível em: <<http://www.west-cs.com.br/blog/o-que-e-controle-pid>>. Acessado em: 17 de out. de 2021.

CONTROL Tutorials for Matlab & Simulink. **Introduction: PID Controller Design**. Disponível em: <<https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID>>. Acessado em: 17 de out de 2021.

FINCO, Nina. Telemetria Veicular: O que é?. **Cobli.co**, 2019. Disponível em: <<https://www.cobli.co/blog/telemetria-veicular>>. Acessado em: 16 de out. de 2021.

TELEMETRIA. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <<https://pt.wikipedia.org/wiki/Telemetria>>. Acessado em: 16 de out. 2021.

EMBRAPA. GPS – Global Positioning System. **Embrapa**, 2020. Disponível em: <<https://www.embrapa.br/satelites-de-monitoramento/missoes/gps>>. Acessado em: 16 de out. de 2021.

GPS. Space Segment. **gps.gov**, 2021. Disponível em: <<https://www.gps.gov/systems/gps/space>>. Acessado em: 17 de out. de 2021.

GOGONI, Ronaldo. O que é GPS? **Tecnoblog**, 2019. Disponível em: <<https://tecnoblog.net/312677/o-que-e-gps>>. Acessado em 18 de out. de 2021.

GARMIN. What is GPS? **Garmin**, 2021. Disponível em: <<https://www.garmin.com/pt-BR/aboutgps/>>. Acessado em 18 de out. de 2021.

LEONE, Leonello. Guia para desenvolver o seu 1º site dinâmico utilizando PHP. **Becode**, 2016. Disponível em: <<https://becode.com.br/site-dinamico-com-php>>. Acessado em 19 de out. de 2021.

PHP. Manual do PHP. **Php**, 2021. Disponível em: <[https://www.php.net/manual/pt\\_BR/](https://www.php.net/manual/pt_BR/)>. Acessado em 19 de out de 2021.

PHP. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2021. Disponível em: <<https://pt.wikipedia.org/wiki/PHP>>. Acessado em: 22 de out. 2021.

SAWAYA, M. R. **Dicionário de Informática & Internet**. 1. ed. Editora NBL, 2002.

GARSIEL, Tali. Behind the Scenes of Modern Web Browsers. **Taligarsiel**, 2021. Disponível em: <<http://taligarsiel.com/Projects/howbrowserswork1.htm>>. Acessado em: 22 de out. de 2021.

PCMAG. Broswe Encyclopedia. **Pcmag**, 2021. Disponível em: <<https://www.pcmag.com/encyclopedia/term/dynamic-web-page/>>. Acessado em 22 de out. de 2021.

ORACLE. What is Database? **Oracle**, 2021. Disponível em: <<https://www.oracle.com/database/what-is-database/>>. Acessado em 23 de out. de 2021.

PHPUNIT. Testando Banco de Dados. **Readthedocs**, 2018. Disponível em: <[https://phpunit.readthedocs.io/pt\\_BR/latest/database.html](https://phpunit.readthedocs.io/pt_BR/latest/database.html)>. Acessado em: 23 de out. de 2021.

CLARES, Rafael. Gerando Gráficos com PHP. **BlogClares**, 2021. Disponível em: <<https://blog.clares.com.br/gerar-graficos-com-php/>>. Acessado em: 24 de out. de 2021.

REDHAT. Interface de Programação de Aplicações - O que é API?. **Redhat**, 2021. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acessado em 1 de nov de 2021.

GOOGLE cloud. API do Google Cloud. **Google**, 2021. Disponível em: <<https://cloud.google.com/apis/docs/overview>>. Acessado em 2 de nov. de 2021.

MEDEIROS, Larissa. Google Maps Platform: Novo recurso de estilização de Mapas. **Maplink**, 2021. Disponível em: <<https://maplink.global/blog/estilizacao-mapas/>>. Acessado em 2 de nov. de 2021.

POMILIO, J. A. Dimensionamento de Sistemas de Dissipação de Calor para Dispositivos Semicondutores de Potência. **Unicamp**, 2009. Disponível em: <<https://www.dsce.fee.unicamp.br/~antenor/pdffiles/eltpot/cap11.pdf>>. Acessado em 07 de nov. de 2021.