ISSN 2318-535X

PROTÓTIPO AUTOMATIZADO DE MONITORAMENTO DE BATIMENTOS CARDÍACOS E TEMPERATURA CORPORAL

Thiago Bariquello Cavalcante[1], Ricardo Rall[2]
[1]Tecnólogo em Análise e Desenvolvimento de Sistemas, FATEC Botucatu, thiago.cavalcante3@fatec.sp.gov.br.
[2]Professor Doutor, FATEC Botucatu, ricardo.rall@fatec.sp.gov.br.

RESUMO

O acompanhamento de sinais vitais como batimentos cardíacos e temperatura corporal é crucial na saúde, mas métodos tradicionais são muitas vezes manuais e esporádicos, limitando a eficácia das intervenções. Este trabalho teve como objetivo o desenvolvimento de um protótipo automatizado para monitorar a frequência cardíaca e temperatura corporal, utilizando tecnologias de hardware livre (ESP32, sensor MAX30102 para frequência cardíaca, sensor GY906 para temperatura sem contato) e Internet das Coisas (IoT). Os dados coletados pelo ESP32 via I2C foram transmitidos por Wi-Fi (HTTPS) para a plataforma ThingSpeak, que armazena, gera gráficos em tempo real e envia alertas por e-mail em caso de valores críticos. Um aplicativo Android simples, usando WebView, permite o acesso móvel aos dados. Os resultados demonstraram a viabilidade da solução para um monitoramento de saúde mais acessível, contínuo e personalizado.

Palavras-chave: ESP32. MAX30102. GY906. Monitoramento de Saúde. Internet das Coisas.

1 INTRODUÇÃO

A monitorização contínua e precisa de sinais vitais como batimentos cardíacos e a temperatura corporal é fundamental para a detecção precoce de agravos à saúde e para o manejo eficaz de diversas condições clínicas. No entanto, os métodos tradicionais de monitoramento frequentemente esbarram em limitações como a necessidade de intervenção manual, o caráter invasivo de alguns procedimentos e a esporadicidade das medições, o que pode retardar diagnósticos e comprometer a personalização do cuidado ao paciente. Nesse contexto, a busca por soluções tecnológicas que superem tais desafios torna-se imperativa, especialmente com o avanço da Internet das Coisas (IoT) e do hardware livre no setor da saúde (BORGES *et al.*, 2020).

A IoT tem possibilitado a conexão de dispositivos para troca e análise de informações em tempo real, permitindo monitoramento contínuo e diagnósticos mais precisos (CHEN *et al.*, 2018). Paralelamente, o conceito de hardware livre, exemplificado por plataformas como o ESP32, permite o desenvolvimento de dispositivos eletrônicos acessíveis e modificáveis, democratizando o acesso à tecnologia (STALLMAN, 2021). A combinação dessas tecnologias, utilizando sensores específicos como o MAX30102 para batimentos cardíacos e o GY906 para temperatura infravermelha, oferece um

ISSN 2318-535X

caminho promissor para criar sistemas de monitoramento de saúde eficientes e de baixo custo (ALVES *et al.*, 2020).

Diante desse cenário, o presente trabalho teve como objetivo desenvolver e validar um protótipo automatizado de baixo custo para o monitoramento contínuo dos batimentos cardíacos e da temperatura corporal, empregando tecnologias de hardware e software livres integradas a recursos de IoT, visando uma solução acessível e eficaz para acompanhamento remoto de sinais vitais.

2 MATERIAL E MÉTODOS

O desenvolvimento do protótipo envolveu a integração de componentes de hardware específicos, a programação do microcontrolador e a configuração de plataformas de software para coleta, visualização e alerta de dados.

Os principais materiais empregados foram: um microcontrolador ESP32, selecionado em razão de sua capacidade de processamento, conectividade Wi-Fi embarcada e baixo consumo de energia; o sensor de oximetria de pulso e frequência cardíaca MAX30102; e o sensor de temperatura infravermelho sem contato GY906. A escolha desses sensores fundamentou-se em critérios de precisão, acessibilidade e adequação para o monitoramento não invasivo de sinais vitais.

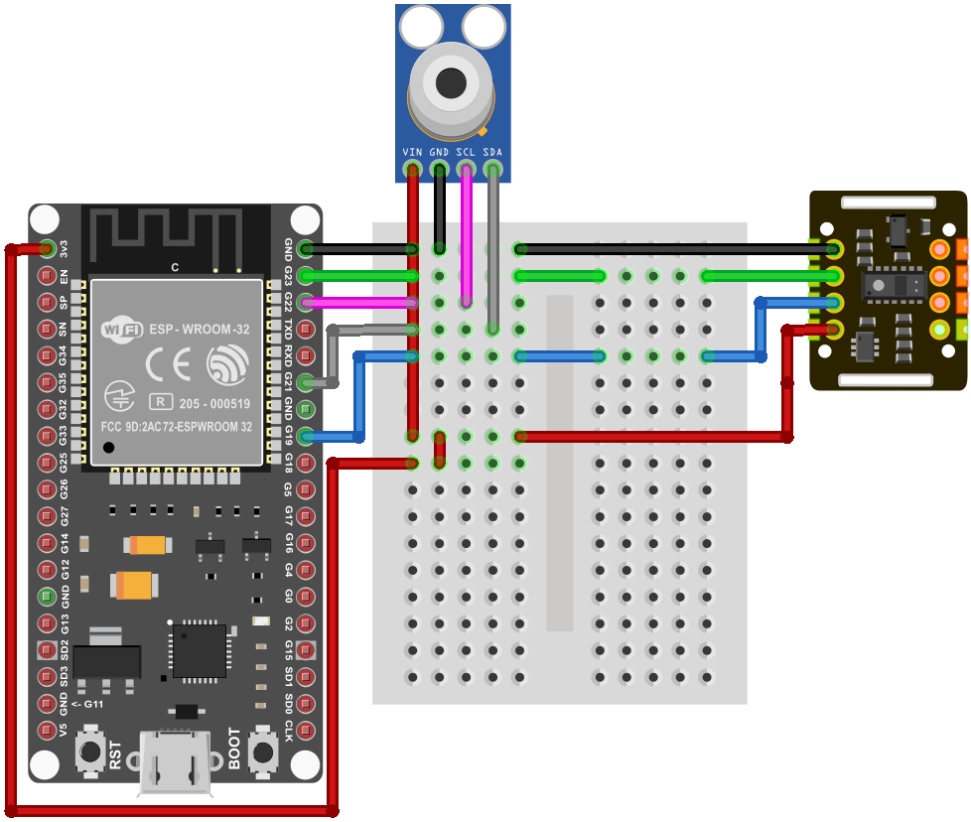
A arquitetura do sistema consistiu na conexão dos sensores ao ESP32 por meio do protocolo de comunicação I2C (Figura 1). No esquema elétrico, verifica-se que os condutores de alimentação (na cor vermelha) encontram-se conectados ao terminal 3.3 V do ESP32, o qual fornece energia aos pinos VIN dos sensores MAX30102 e GY906. O terminal GND do ESP32 (na cor preta) está interligado aos terminais GND dos sensores, garantindo o referencial comum do circuito. Em relação às linhas de comunicação, o pino D19 (DAS, na cor azul) do ESP32 foi designado para a interface SDA do sensor GY906, enquanto o pino D21 (DAS, na cor cinza) conecta-se ao terminal SDA do sensor MAX30102. Ademais, o pino D22 (SCL, na cor rosa) do ESP32 está vinculado ao terminal SCL do GY906, e o pino D23 (SCL, na cor verde) ao terminal SCL do MAX30102.

O firmware do ESP32 foi desenvolvido em linguagem C/C++, por meio do ambiente de programação Arduino IDE. O código contempla a inicialização dos sensores, a leitura periódica dos parâmetros de frequência cardíaca (via MAX30102), bem como da temperatura corporal (via GY906). Posteriormente, os dados coletados são transmitidos à plataforma de Internet das Coisas (IoT) ThingSpeak. A comunicação com

ISSN 2318-535X

essa plataforma é realizada por meio de requisições HTTP POST via conexão Wi-Fi do ESP32, utilizando a API do serviço para atualização dos campos de dados (Figura 2). Para assegurar a proteção das informações transmitidas, a comunicação foi configurada em HTTPS, em conformidade com os princípios da Lei Geral de Proteção de Dados – LGPD (BRASIL, 2018).

Figura 1. Esquema elétrico.



Fonte: Autoria própria (2025).

A plataforma ThingSpeak foi empregada para o armazenamento histórico dos dados, a visualização em gráficos dinâmicos em tempo real e a definição de alertas. Nessa etapa, foi estabelecida uma regra de notificação automática, via correio eletrônico, em situações em que os valores de batimento cardíaco ou temperatura corporal ultrapassassem os limiares previamente configurados.

Complementarmente, desenvolveu-se uma aplicação móvel para o sistema Android, utilizando o componente WebView (Figura 3). Essa aplicação possibilita a exibição das visualizações públicas dos gráficos gerados pelo ThingSpeak, permitindo o acompanhamento remoto e simplificado das medições em dispositivos móveis.

JORNADA CIENTÍFICA E TECNOLÓGICA
ISSN 2318-535X

Figura 2. Trecho do Código do ESP32.

```
  // --- Envia os dados para o ThingSpeak a cada 20 segundos ---
  if (millis() - lastThingSpeakUpdate >= 20000) {
    // Escreve a média de batimentos no Campo 2 do canal
    int y = ThingSpeak.writeField(myChannelNumber, 2, beatAvg, myWriteAPIKey);
    if (y == 200) { // Código 200 significa sucesso na requisição HTTPS
      Serial.println("Canal atualizado com sucesso. Batimento cardíaco enviado.");
    } else {
      Serial.println("Problema ao atualizar o canal. Código de erro HTTPS " + String(y));
    }

    delay(15000); // Pausa de 15 segundos exigida pela API do ThingSpeak no plano gratuito

    // Escreve a temperatura no Campo 1 do canal
    int x = ThingSpeak.writeField(myChannelNumber, 1, temperatura, myWriteAPIKey);
    if (x == 200) {
      Serial.println("Canal atualizado com sucesso. Temperatura enviada.");
    } else {
      Serial.println("Problema ao atualizar o canal. Código de erro HTTPS " + String(x));
    }

    lastThingSpeakUpdate = millis(); // Atualiza o tempo do último envio
  }
}
```

Fonte: Autoria própria (2025).

Figura 3. Trecho do código App Android.

```
// Declara a classe da tela principal do aplicativo.
public class MainActivity extends AppCompatActivity {

    // Declara quatro variáveis para os componentes WebView.
    // Cada WebView funcionará como um "mini-navegador" para exibir um gráfico ou widget.
    4 usages
    WebView webView;  // Para o gráfico de Temperatura
    4 usages
    WebView webView1; // Para o widget de Temperatura
    4 usages
    WebView webView2; // Para o gráfico de Batimento Cardíaco
    4 usages
    WebView webView3; // Para o widget de Batimento Cardíaco

    // O método onCreate() é chamado quando a atividade (tela) é criada pela primeira vez.
    // É aqui que toda a configuração inicial acontece.
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Associa esta classe Java ao seu arquivo de layout XML (activity_main.xml),
        // que define a aparência visual da tela.
        setContentView(R.layout.activity_main);

        // --- Configuração das WebViews ---

        // 1. Linka as variáveis Java com os componentes do layout XML através de seus IDs.
        webView = findViewById(R.id.webView);
        webView1 = findViewById(R.id.webView1);
        webView2 = findViewById(R.id.webView2);
        webView3 = findViewById(R.id.webView3);

        // 2. Habilita a execução de JavaScript em cada WebView.
        // Isso é ESSENCIAL para que os gráficos dinâmicos do ThingSpeak funcionem corretamente.
        webView.getSettings().setJavaScriptEnabled(true);
        webView1.getSettings().setJavaScriptEnabled(true);
        webView2.getSettings().setJavaScriptEnabled(true);
        webView3.getSettings().setJavaScriptEnabled(true);
```

ISSN 2318-535X

Fonte: Autoria própria (2025).

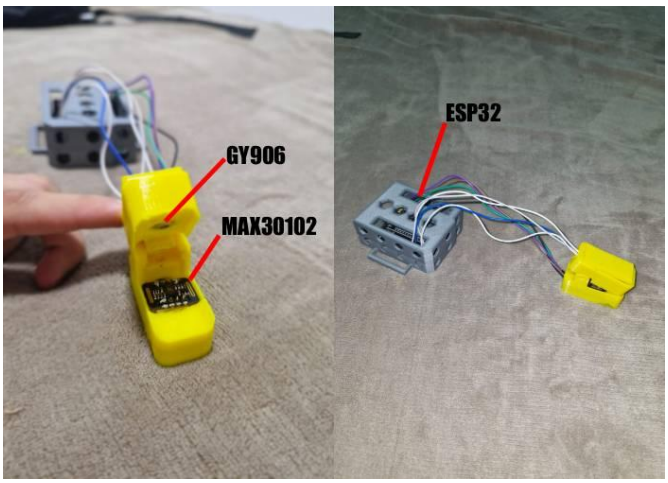
3 RESULTADOS E DISCUSSÕES

O protótipo desenvolvido demonstrou sucesso na integração dos componentes e na funcionalidade proposta. O sistema foi capaz de coletar dados de frequência cardíaca e temperatura corporal de forma contínua e transmiti-los eficazmente para a plataforma ThingSpeak via Wi-Fi.

Os sensores MAX30102 e GY906 apresentaram leituras consistentes em testes preliminares, alinhadas com as expectativas para dispositivos desta categoria, embora uma validação formal contra equipamentos médicos padrão não tenha sido objeto deste trabalho inicial. Ainda assim, os resultados indicam que o protótipo apresenta potencial para aplicação em cenários de monitoramento não clínico e acadêmico.

A Figura 4 apresenta a disposição física do sistema, evidenciando o microcontrolador ESP32, os sensores GY906 e MAX30102, bem como a case projetada e confeccionada por impressão 3D. A estrutura impressa contempla uma capa de proteção para o ESP32 e uma adaptação semelhante a um oxímetro de dedo, destinada a posicionar os sensores de forma estável e ergonômica durante as medições.

Figura 4. Protótipo e case impresso em 3D.

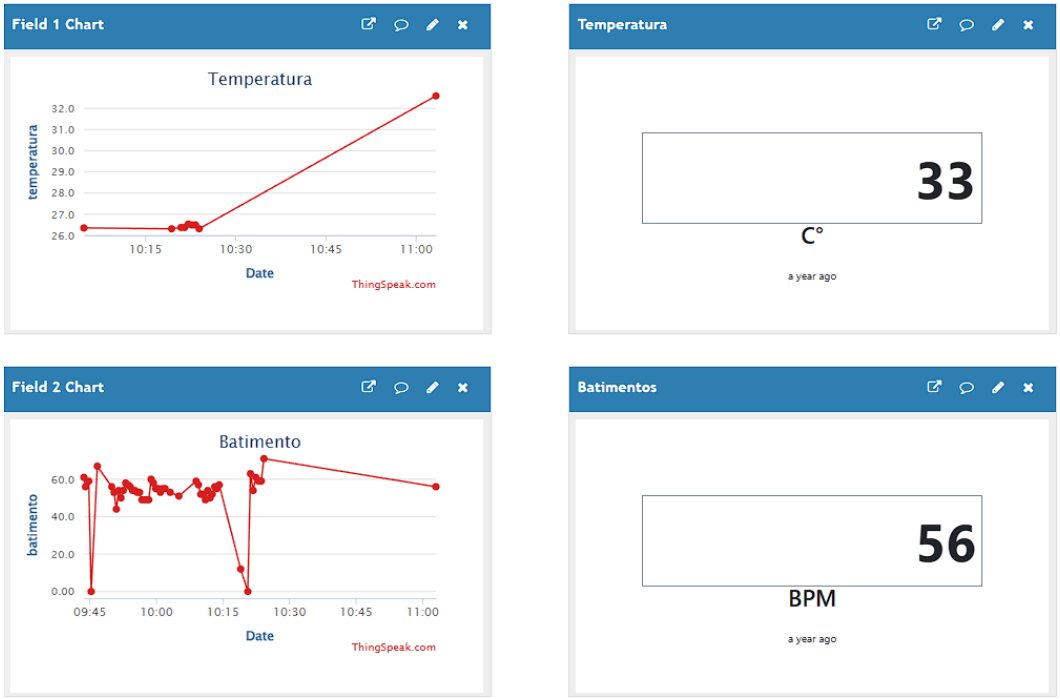


Fonte: Autoria própria (2025).

A integração com a plataforma ThingSpeak foi bem-sucedida, permitindo o armazenamento e a visualização dos dados em gráficos atualizados em tempo real (Figura 5). Além disso, a configuração de alertas por e-mail (Figura 6) para valores críticos funcionou conforme esperado, adicionando uma camada de segurança ao processo de monitoramento.

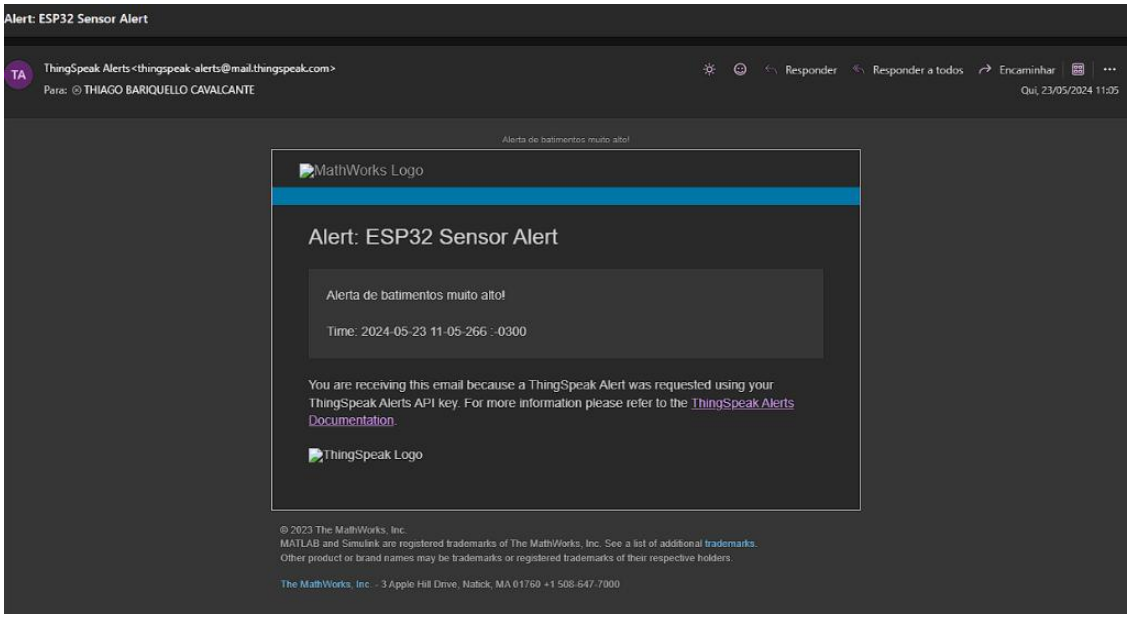
ISSN 2318-535X

Figura 5. Gráficos ThingSpeak.



Fonte: Autoria própria (2025).

Figura 6. Exemplo de alerta por e-mail pelo ThingSpeak.

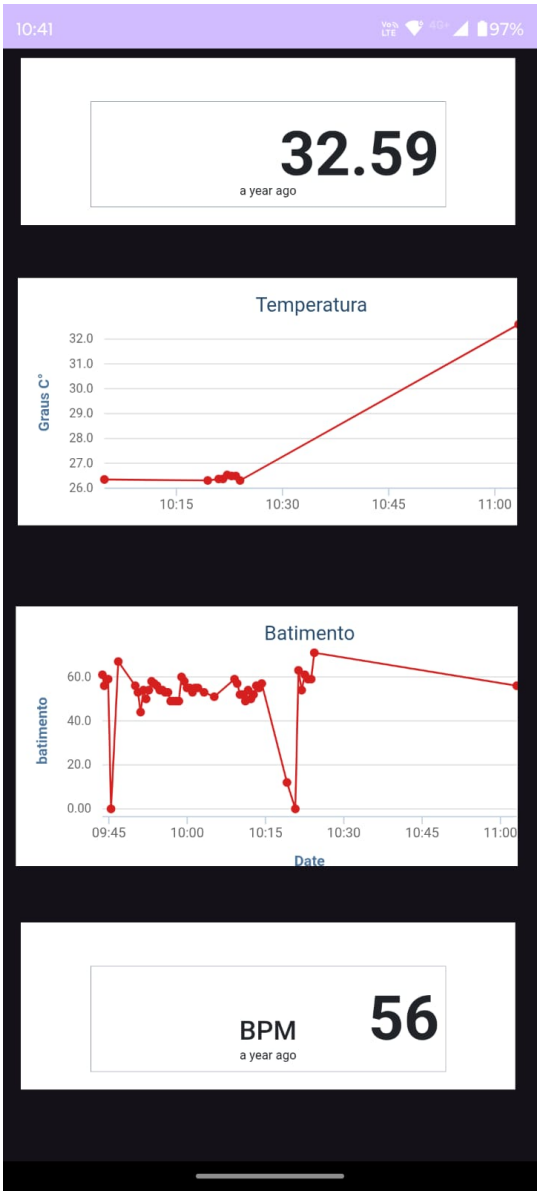


Fonte: Autoria própria (2025).

ISSN 2318-535X

Por fim, a aplicação Android, embora simples, cumpriu o objetivo de fornecer acesso móvel aos gráficos do ThingSpeak (Figura 7), demonstrando a viabilidade de um acompanhamento remoto prático e acessível.

Figura 7. Exemplo da interface do aplicativo Android exibindo dados.



Fonte: Autoria própria (2025).

A discussão dos resultados aponta para o potencial desta abordagem de baixo custo para aplicações em saúde, como monitoramento domiciliar de pacientes crônicos ou triagem em unidades de saúde com recursos limitados. A utilização de hardware livre e plataformas IoT abertas contribui para a acessibilidade e replicabilidade da solução. No entanto, desafios persistem, principalmente relacionados à validação clínica rigorosa, à

ISSN 2318-535X

usabilidade por diferentes perfis de usuários e à garantia robusta da privacidade e segurança dos dados em conformidade com a LGPD (BRASIL, 2018), que exigiria consentimento informado, políticas claras de dados e segurança aprimorada para uma implementação real.

4 CONCLUSÕES

Este trabalho demonstrou a viabilidade técnica do desenvolvimento de um protótipo automatizado de baixo custo para monitoramento de batimentos cardíacos e temperatura corporal, utilizando ESP32, sensores MAX30102 e GY906, e a plataforma IoT ThingSpeak. O sistema atingiu o objetivo proposto, coletando e disponibilizando dados vitais remotamente através de gráficos e alertas. A solução apresenta potencial para aplicações em saúde, promovendo um monitoramento mais acessível e contínuo. Trabalhos futuros devem focar na validação clínica, aprimoramento da usabilidade, otimização energética e robustez da segurança dos dados para viabilizar sua aplicação prática.

5 REFERÊNCIAS

ALVES, A. L. *et al.* Aplicações em redes de sensores na área da saúde e gerenciamento de dados médicos: tecnologias em ascensão. In: ALBUQUERQUE, C. V. N. *et al.* (org.). **Redes de sensores sem fio: conceitos e aplicações**. Rio de Janeiro: SBC, 2020. p. 67-96. Disponível em: https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/47/213/434-1?inline=1. Acesso em: 6 set. 2023.

BORGES, M. *et al.* Advances in sensors and wireless communications for healthcare and well-being applications: a survey of the state-of-the-art and future trends. ***IEEE Communications Surveys & Tutorials,*** v. 22, n. 4, p. 2435-2481, 2020. Disponível em: https://ieeexplore.ieee.org/document/8948364. Acesso em: 2 nov. 2023.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). **Diário Oficial da União,** Brasília, DF, 15 ago. 2018. Seção 1, p. 59. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 1 nov. 2023.

CHEN, M. *et al. Narrow Band Internet of Things (NB-IoT) for medical care applications in typical regions of China and Japan: a comparative study of the technology and standards of NB-IoT for e-healthcare systems in China and Japan.* ***IEEE Access,*** v. 6, p. 14909-14919, 2018. Disponível em: https://ieeexplore.ieee.org/document/8316978. Acesso em: 10 out. 2023.

STALLMAN, R. M. Hardware livre e designs de hardware livre. **Projeto GNU - Free Software Foundation**, 2021. Disponível em: https://www.gnu.org/philosophy/free-hardware-designs.pt-br.html. Acesso em: 30 set. 2023.