

PROTÓTIPO DE DISPOSITIVO DIGITALIZADOR DE DIAPOSITIVOS PARA A PRESERVAÇÃO DE ACERVOS FOTOGRÁFICOS ESCOLARES

Lucas Teixeira Serafim¹, Sergio Coral²

Resumo: No contexto atual, instituições de ensino possuem em seus acervos diapositivos sem a adequada preservação, devido aos elevados custos relacionados a digitalização desse formato de fotografia. O presente artigo tem o objetivo de desenvolver um protótipo de um sistema embarcado de baixo custo para a digitalização de diapositivos escolares e o armazenamento de suas imagens em banco de dados para futuras visualizações. A metodologia adotada envolve uma pesquisa aplicada, experimental e tecnológica para o desenvolvimento do esquema com *hardware*, da aplicação em *software*, e da modelagem técnica da estrutura do sistema. Os testes iniciais indicam que a captura, envio e armazenagem acontecem em um tempo razoável, com média de 4 minutos e 57 segundos para as oito transparências comportadas. Apesar da qualidade baixa das imagens capturadas, o protótipo mostrou potencial para impactar positivamente a preservação de acervos escolares.

Palavras-chave: Sistema embarcado; Digitalização; Diapositivo; Acervo escolar.

ABSTRACT: In the current context, educational institutions have slides in their collections that are not adequately preserved due to the high costs associated with digitizing this type of photography. This article aims to develop a prototype of an embedded system for digitizing school slides and storing their images in a database for future viewing. The methodology adopted involves applied, experimental, and technological research to develop the hardware scheme, the software application, and the technical modeling of the system structure. Initial tests indicate that capture, sending, and storage occur in a reasonable time, with an average of 4 minutes and 57 seconds for the eight transparencies supported. Despite the low quality of the captured images, the prototype showed potential to positively impact the preservation of school collections.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), lucasteixeiraserafim@gmail.com

²Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), sergiocoral@unesc.net

Keywords: Embedded system; Digitization; Slide; School collection.

1 INTRODUÇÃO

Instituições de ensino com mais de algumas décadas de existência, presentes em todo o país, possuem acervos com objetos de valor histórico e cultural, como livros, documentos e fotografias analógicas, armazenados em porões, sótãos e salas expostas a intempéries e condições insalubres, adversas para a sua conservação adequada (Souza, 2013).

Acervos podem ser entendidos como conjuntos de bens materiais que compõem fundos ou coleções de uma pessoa ou entidade de natureza pública ou privada, no contexto escolar, eles se referem aos documentos, imagens e objetos que fazem parte do patrimônio e da rotina de uma instituição de ensino, como uma escola ou universidade (Viero et al., 2017).

Fotografias escolares, segundo Oliveira (2017), abrangem todas as imagens sob a posse de uma instituição educacional, e quando digitalizadas e disponibilizadas, podem auxiliar em pesquisas dedicadas ao ensino e a comunidade. Nessa linha, Fujita et al. (2004) destaca os diapositivos escolares, que podem oferecer uma abordagem ainda mais distinta na apresentação de registros históricos.

De acordo com Weidner (2013, tradução nossa), diapositivos, também chamados de transparências ou *slides*, são fotografias positivas de 35 milímetros com bases transparentes, montadas em suportes de plástico ou cartão. Bennett (2020, tradução nossa) afirma que transparências são um dos formatos mais comuns em coleções de patrimônio cultural, mas que suas exigências de manuseio e captura apresentam desafios logísticos para uma digitalização em maior escala.

Para Malta (2011), a digitalização fotográfica é uma tecnologia que vai além da preservação de bens culturais, pois permite que instituições transmitam seus acervos, que são compostos por fragmentos de experiências coletivas, na expectativa de uma democratização mais ampla dessas informações.

Entretanto, equipamentos de digitalização de diapositivos a nível de preservação são custosos e regularmente requerem profissionais qualificados, e como são comuns os projetos museológicos escolares que não dispõem de recursos orçamentários suficientes para tais investimentos, a conservação dessas fotografias é abandonada (Cruz, 2021).

Segundo Cortez et al. (2022), o MQTT (*Message Queuing Te-*

lemetry Transport) é um protocolo de comunicação que utiliza o padrão publicador/assinante e o protocolo TCP/IP, permitindo a troca de informações através de um intermediário *broker*, ele é amplamente adotado devido a sua facilidade de implementação, baixa utilização de banda e diversidade de bibliotecas compatíveis.

No entendimento de Gallassi e Martins (2013), sistemas embarcados são sistemas computacionais que, integrados a outros produtos, viabilizam ou aprimoram uma funcionalidade. Cunha (2007) destaca ainda que eles são projetados para a execução de uma tarefa específica dentro de suas limitações físicas, utilizando frequentemente de microcontroladores para o seu funcionamento.

Ademais, ferramentas computacionais para a modelagem de peças (sistemas CAD) são essenciais para o desenvolvimento rápido de novos produtos, pois permitem a criação de modelos paramétricos, o que facilita modificações, verificações de interferências, a geração de desenhos de fabricação e a prototipação (Foggiatto; Volpato; Bontorin, 2007).

Conforme a revisão de literatura, foram encontrados estudos similares, mas com diferentes abordagens. Em sua pesquisa, Tierra e Segovia (2023, tradução nossa) têm como objetivo desenvolver um protótipo digitalizador de diapositivos que capture uma imagem por vez, focando em parâmetros de captura modificáveis, e em seus resultados obtiveram estabilidade de cor e precisão de movimento satisfatórios.

Bennett (2020, tradução nossa) propõe o uso de uma mesa programável e modular para atender padrões rigorosos de qualidade de imagem, resultando em fotografias de alta resolução. Já Lima (2020) foca no desenvolvimento de um equipamento de digitalização de radiografias odontológicas de fácil manipulação e baixo custo, apresentando um manuseio facilitado e utilização funcional.

Para Sullivan e Larrimore (2018, tradução nossa), o objetivo é desenvolver um equipamento personalizado com suporte modular para a captura de diapositivos, com destaque para a definição de protocolos, validações, e fluxos de trabalho eficientes, resultando em um manuseio seguro e um pós-processamento eficiente.

Diante do exposto, o presente trabalho propõe desenvolver um protótipo de um sistema embarcado de baixo custo para a digitalização de diapositivos escolares e o armazenamento de suas imagens em banco de dados para futuras visualizações, sem a necessidade de manuseio especializado, visando simplificar e baratear o processo de preservação de acervos

escolares.

Os objetivos específicos desse estudo consistem em: idealizar e montar o circuito eletrônico; desenvolver o software para o projeto; modelar e montar a estrutura do sistema embarcado; e unir e testar a estrutura concluída com o circuito eletrônico.

Na estrutura do trabalho, o capítulo um introduz os principais conceitos abordados na pesquisa, assim como a apresentação dos seus objetivos e da sua justificativa, e a revisão de estudos similares. O segundo capítulo descreve os ambientes, as ferramentas e as etapas de desenvolvimento, tal como as metodologias adotadas.

Ademais, o capítulo três apresenta e analisa os resultados obtidos na pesquisa, como também realiza a comparação da atual pesquisa com os trabalhos correlatos. Por fim, o quarto capítulo expõem as conclusões correspondentes aos objetivos e aos resultados, evidenciando também recomendações e sugestões para trabalhos futuros.

2 MATERIAIS E MÉTODOS

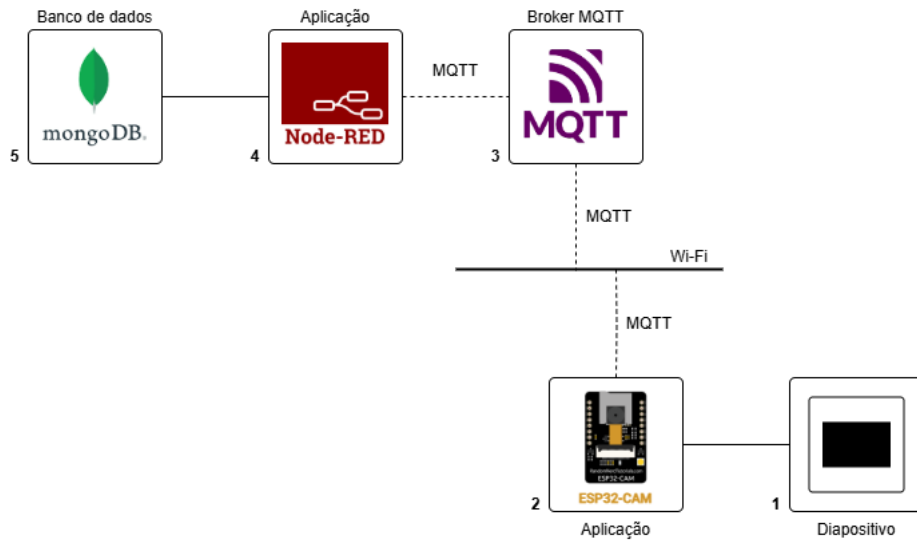
O presente estudo consiste em uma pesquisa aplicada, de natureza experimental e caráter tecnológico, na qual desenvolveu-se um sistema embarcado para capturar fotografias de oito diapositivos, quantidade escolhida devido a restrições de tamanho, e armazená-las automaticamente em um banco de dados.

De forma resumida, o funcionamento do sistema se inicia na captura da imagem do diapositivo pelo ESP32-CAM, que é responsável também por codificá-la em base64, fragmenta-la em partes (*chunks*), e enviar essas partes para o *broker* EMQX por meio do protocolo MQTT, através de uma conexão Wi-Fi.

O *broker* por sua vez atua como um intermediador, redirecionando as partes para a aplicação da plataforma de desenvolvimento online Node-RED, que está inscrita aos tópicos MQTT utilizados. Nela, os *chunks* são validados, reorganizados e reunidos, caso obtenha um erro, ou o eventual sucesso, o Node-RED o comunica para o ESP, e então envia a *string* para o Banco de Dados MongoDB para ser armazenada.

O funcionamento dessa arquitetura pode ser visualizado na Figura 1.

Figura 1 - Arquitetura do sistema.



Fonte: Elaborado pelo autor.

Devido a abrangência de diferentes áreas, como eletrônica, redes de computadores, desenvolvimento de *software* e modelagem paramétrica assistida por computador, o projeto passou por diversas etapas, podendo ser dividido entre a montagem do *hardware*, a elaboração da aplicação e a modelagem do sistema.

2.1 MONTAGEM DO *HARDWARE*

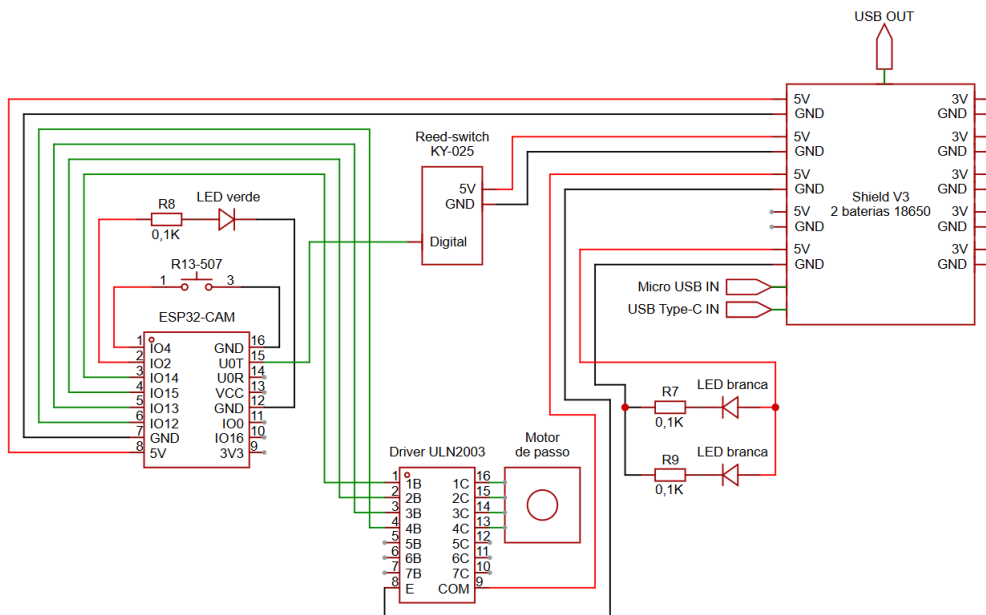
O hardware do projeto é fundamental para o funcionamento de todo o sistema, ele é constituído de componentes escolhidos a partir de seus funcionamentos individuais, de como interagem entre si e com a aplicação, assim como dos seus valores monetários e da facilidade de acesso em plataformas de venda nacionais.

2.1.1 Montagem do circuito eletrônico

O circuito eletrônico é composto por um *Shield V3*, que junto de duas baterias de modelo 18650 fornece tensão regulada de 5V DC e corrente de até 2,2A, um motor de passo 28BYJ-48 com um *driver* ULN2003, um ESP32-CAM com um programador ESP32-CAM-MB para facilitar o desenvolvimento, um módulo interruptor magnético *Reed-switch* KY-025, dois LEDs brancos de 5mm e um LED verde de 5mm, um resistor 100R 1/4W e um botão R13-507 do tipo SPST normalmente aberto.

A disposição e conexão dos componentes pode ser observada no esquema eletrônico conforme a Figura 2.

Figura 2 - Diagrama esquemático.



Fonte: Elaborado pelo autor.

O *Shield V3* possibilita o carregamento das baterias por meio de suas entradas Micro USB ou USB Tipo C, e em conjunto a elas alimenta o microcontrolador, o motor de passo e o seu driver, os LEDs, e o sensor magnético com suas tensões de 5V já reguladas.

Após a ativação do circuito por um botão integrado ao *Shield*, o ESP32-CAM é o principal responsável pelo seu controle. Depois de ligado, ele aguarda o sinal recebido quando o *push button* R13-507 é pressionado, a partir do qual ele inicia o movimento do motor de passo conforme o sinal digital do sensor KY-025, e a captura e envio das fotos.

Ambos os LEDs de cor branca são ligados ao *Shield* com a devida resistência, e utilizados de iluminação de fundo para o diapositivo, o LED verde por outro lado é conectado ao ESP, também com um resistor, para indicar ao usuário se o sistema está capturando e enviando imagens ou se finalizou e está aguardando novas transparências.

2.2 ELABORAÇÃO DA APLICAÇÃO

O estágio de desenvolvimento da aplicação pode ser dividido entre o *software* do microcontrolador, o fluxo na plataforma Node-RED, e o armazenamento no banco de dados.

2.2.1 Software do microcontrolador

O sistema foi implementado para o ESP32-CAM na plataforma Arduino IDE na sua versão 2.3.5, devido a sua fácil usabilidade, e com a linguagem de programação C++, a padrão da IDE. O seu objetivo é capturar e transmitir sequencialmente as partes das imagens em base64 via MQTT. A sua estrutura é modular, separada em funções específicas para fácil manutenção e robustez em relação a falhas.

Para auxiliar no desenvolvimento da aplicação foram utilizadas seis bibliotecas, são elas: `esp32cam` de Junxiao Shi na versão 0.0.20-250112, `PubSubClient` de Nick O'Leary na versão 2.8, e `ArduinoJson` de Benoit Blanchon na versão 7.4.1. Além dessas utilizou-se ainda as bibliotecas `WiFi.h` e `base64.h`, incluídas com o core ESP32, e `Stepper.h`, disponibilizada pela própria IDE.

A biblioteca `esp32cam` facilitou o uso da câmera do microcontrolador, a `PubSubClient` proveu um cliente para publicações e assinaturas MQTT, e utilizou-se a `ArduinoJson` para deserializar as respostas JSON recebidas. Para estabelecer a conexão a uma rede Wi-Fi foi utilizada a biblioteca `WiFi.h`, já a conversão da imagem capturada em uma *string* Base64 foi feita pela `base64.h`, por fim, a `Stepper.h` foi implementada para controlar o motor de passo via o driver.

A estrutura do código pode ser compreendida em duas partes: a primeira sendo a configuração de parâmetros e funcionalidades do projeto, e a segunda a execução cíclica da lógica implementada, a cada vez que o botão R13-507 é pressionado.

Após a definição das configurações, que acontece logo quando o usuário liga o *Shield*, a função `setup` fica encarregada de configurar os pinos físicos, inicializar a câmera integrada ao ESP, configurar o Wi-Fi por meio da função `setup_wifi`, baseando-se no SSID e senha previamente declarados, estabelecer o cliente MQTT com o endereço do *broker* EMQX, definir a função de retorno `callback` para mensagens recebidas do Node-RED, e acender o LED de indicação.

A segunda etapa da lógica do programa possui o loop como sua principal função, nela se encontra um laço de repetição *for*, que é iterado oito vezes para os oito diapositivos que o sistema comporta, mas que só é iniciado quando a função `aguardando_botao` recebe o sinal do *push button*.

Dentro do laço de repetição, o fluxo começa designando o piscamento do LED, então ele checa se a última imagem foi enviada com sucesso por meio de uma variável booleana, caso ela não tenha sido, é

tentada a reconexão ao *broker* por meio da função `reconectar_mqtt`, e é reiniciada a lógica do *for-loop*. A variável é inicializada como verdadeira para que o fluxo funcione corretamente na primeira imagem capturada.

Caso a última imagem tenha sido enviada com sucesso, o motor é movido conforme a função `mover_motor`, que o desloca até o sinal digital do sensor magnético ser recebido, isso é feito para que o motor pare sempre na posição necessária para que a transparência esteja centralizada na foto.

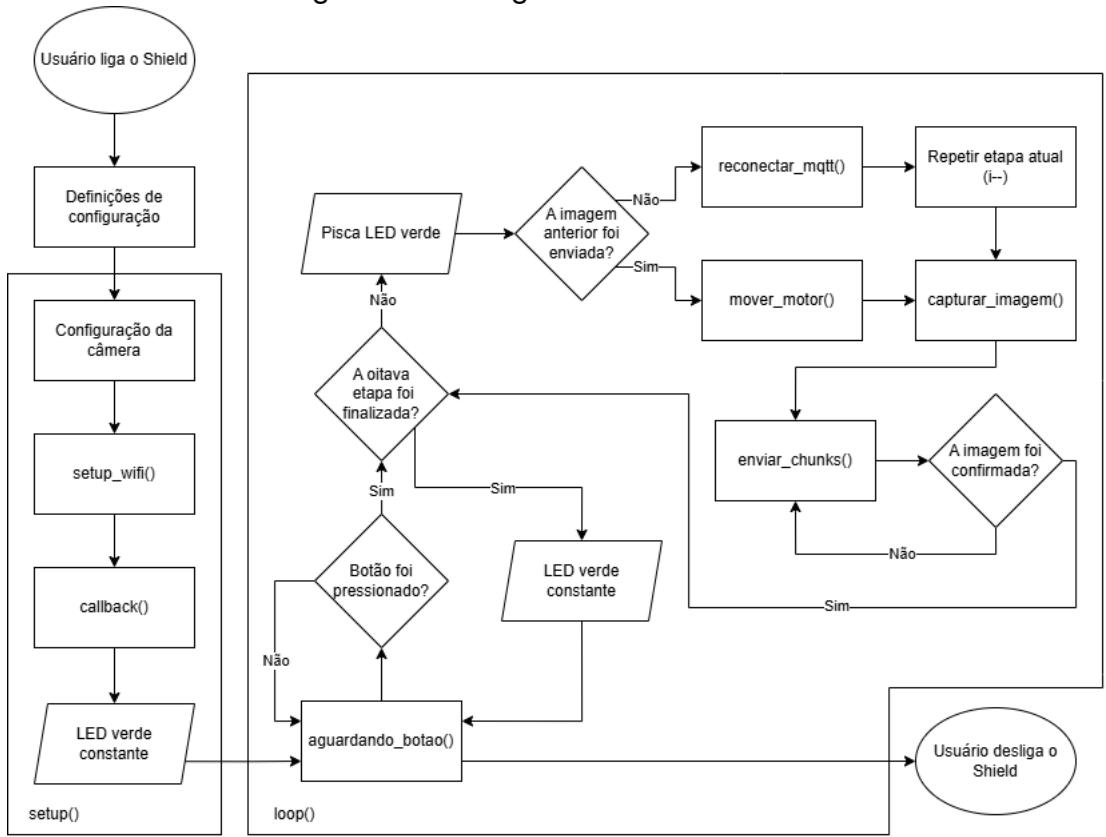
Após o movimento do motor, a função `capturar_imagem` é chamada, ela é encarregada de capturar a fotografia da transparência e codificá-la em uma *string* de Base64, pela sua compatibilidade com o protocolo e facilidade de manipulação, retornando-a por meio de referência para o *loop* junto de seu ID único. Caso ocorra uma falha na captura da imagem, a iteração é interrompida e reiniciada.

A próxima porção do laço é composta sobretudo por uma estrutura de repetição *while*, que permanece ativa enquanto a variável booleana `imagemConfirmada` não for verdadeira. Nela o programa chama a função `enviar_chunks`, que divide a *string* Base64 inteira ultrapassa o limite de envio permitido pela biblioteca em uma única mensagem MQTT. As partes são então publicadas uma a uma no tópico definido.

Ainda na estrutura *while*, após o envio das partes, o código aguarda indiretamente a resposta do Node-RED por meio de uma variável global atualizada via `callback`. Caso seja recebido 'sucesso', as variáveis de controle são atualizadas, a etapa é finalizada e a próxima se inicia, caso seja recebido 'erro' o *loop* é repetido, reenviando os *chunks* da imagem.

Ao finalizar as oito iterações o LED volta a ficar fixo e o sistema volta a aguardar o pressionamento do botão ou o desligamento do circuito. A Figura 3 representa visualmente o fluxo explicado anteriormente.

Figura 3 - Fluxograma do Software.



Fonte: Elaborado pelo autor.

2.2.2 Fluxo do Node-RED

A plataforma Node-RED foi escolhida para a segunda parte da aplicação devido a sua proeminência no meio de prototipagem e sistemas embarcados, especialmente em relação a utilização do protocolo MQTT. A sua integração simples com o banco de dados MongoDB e a vasta quantidade de materiais de estudo disponíveis na Internet também foram pontos positivos.

O seu fluxo pode ser dividido em duas seções para um melhor entendimento: o recebimento da imagem, composta pelos nós "Receber Chunks" e "Armazenar Chunks com Índice", e a montagem e armazenamento da imagem, composta pelos nós "Receber Done", "Salvar imagem", "Validar, Salvar ou Rejeitar", "Armazenar", e "Responder ESP".

O processo se inicia com a primeira seção citada, no nó "Receber Chunks", ele é do tipo entrada MQTT, inscrito no tópico em que o ESP publica as partes individuais das imagens, e está ligado ao nó "Armazenar Chunks com Índice". Esse por sua vez é do tipo função, responsável por

extrair o índice das partes a partir do final do tópico e armazená-las em *imageChunks*, uma variável do contexto de fluxo.

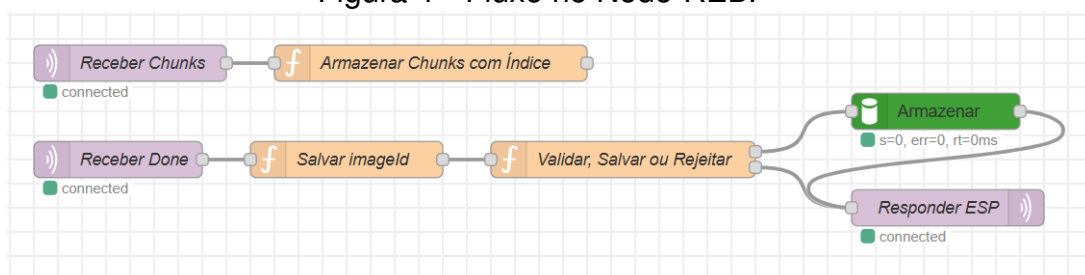
A segunda seção prossegue com o nó "Receber Done", também do tipo entrada MQTT, que aguarda o ESP enviar uma mensagem com o ID da imagem e o total de partes dela. Dele o fluxo segue para "Salvar *imagemId*", do tipo função, que salva o ID da imagem e a quantidade de *chunks* esperados no contexto da aplicação, enviando-os adiante para validação.

A validação dos dados recebidos é realizada no nó "Validar, Salvar ou Rejeitar", do tipo função. Ele possui duas saídas: uma que vai para o nó "Armazenar", apenas quando a imagem está completa, e a outra que vai direto para "Responder ESP", para sinalizar caso tenha ocorrido sucesso ou erro.

A função no nó de validação recupera todos os pedaços da imagem que foram armazenados anteriormente, os ordena com base nos seus índices, e verifica se a quantidade de *chunks* recebido é igual ao número esperado. Se for ela os une novamente, limpa os dados para a próxima imagem, cria a mensagem de confirmação para o ESP, prepara os dados que serão enviados ao MongoDB: hora, ID, e a string, e envia os dados para as suas saídas. Se a imagem estiver incompleta, o código limpa os dados, cria a mensagem de erro com o ID da imagem e o respectivo status, e a envia pela segunda saída.

A Figura 4 ilustra o fluxo descrito.

Figura 4 - Fluxo no Node-RED.



Fonte: Elaborado pelo autor.

2.2.3 Armazenamento no banco de dados

O armazenamento das imagens é a última fase da aplicação, após o Node-RED validar e aceitar os *chunks* reunidos, e enviá-los como uma *string* de Base64 para o banco de dados MongoDB, esse fica encarregado de salvar as imagens para futuras visualizações.

O MongoDB foi escolhido como o banco de dados para a aplicação pela sua integração nativa com a plataforma Node-RED, por lidar bem com arquivos Base64, e por sua estrutura flexível, útil para protótipos que podem evoluir com a necessidade de novos campos. Ele foi utilizado na nuvem por meio da plataforma MongoDB Atlas, devido a sua facilidade de configuração e uso.

Cada registro salvo corresponde a um diapositivo e possui quatro campos: um id gerado automaticamente pelo banco, "*timestamp*" do tipo *Date*, que informa quando o dado foi salvo, "*imageld*", do tipo *String*, que é o identificador da imagem utilizado na aplicação, e "*imageBase64*", também do tipo *String*, que possui o conteúdo da imagem codificado.

Para a visualização da imagem final é necessária a utilização de um conversor que a decodifique, com diversas opções disponíveis gratuitamente. É possível visualizar o exemplo de uma imagem capturada, armazenada, e convertida na Figura 5.

Figura 5 - Imagem armazenada.



Fonte: Elaborado pelo autor.

2.3 MODELAGEM DO SISTEMA

A modelagem do sistema embarcado foi a última etapa desenvolvida, ela foi compreendida primeiramente da criação de um protótipo de papelão para a obtenção de uma melhor ideia tridimensional do produto, seguida da composição dos desenhos técnicos das peças, assim como a disposição 3D das mesmas, e por fim do envio das peças para serem cortadas a laser em MDF.

Para a criação dos desenhos técnicos e da modelagem 3D das

peças utilizou-se o *software* modelador CAD FreeCAD, devido a características como o seu código aberto, a sua arquitetura modular com diversas funcionalidades e soluções, e a sua comunidade ativa com vários tutoriais e documentações para iniciantes.

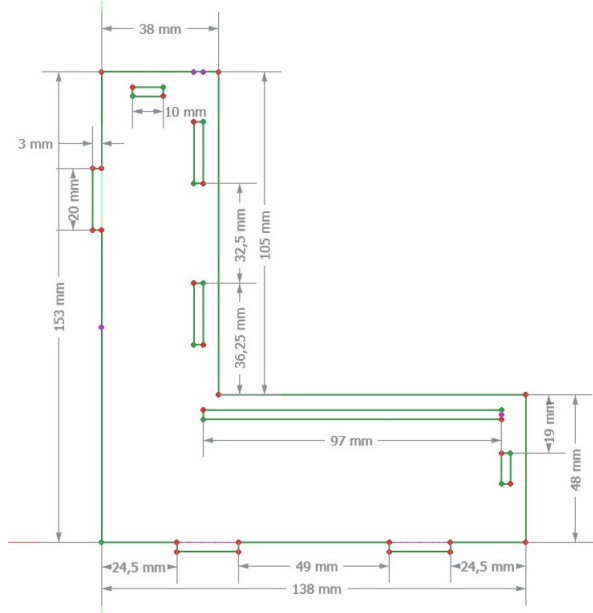
Os desenhos técnicos das peças foram produzidos conforme as orientações e recomendações propostas na metodologia do artigo redigido por Foggiatto, Volpato e Bontorin (2007), destacando-se o planejamento, a utilização de geometrias simples e restrições geométricas, e a prevenção de relações desnecessárias.

O sistema embarcado utiliza um total de trinta peças, das quais dezenove são únicas, desenhadas considerando os seus propósitos específicos no sistema e a necessidade de se encaixarem perfeitamente entre si. Para isso foram criadas variáveis globais para a espessura da madeira, as duas larguras de aberturas de encaixe, e a distância padrão entre tais aberturas e as bordas das peças.

As dimensões das partes foram definidas a partir de medidas já existentes de componentes do circuito, como o tamanho do *Shield*, do ESP32-CAM e do motor de passo, assim como dos diapositivos e da distância necessária entre eles e a câmera do microcontrolador para a captura das imagens.

Na Figura 6 é possível visualizar o desenho com medidas de referências de uma das peças utilizadas no sistema embarcado.

Figura 6 - Desenho técnico CAD.

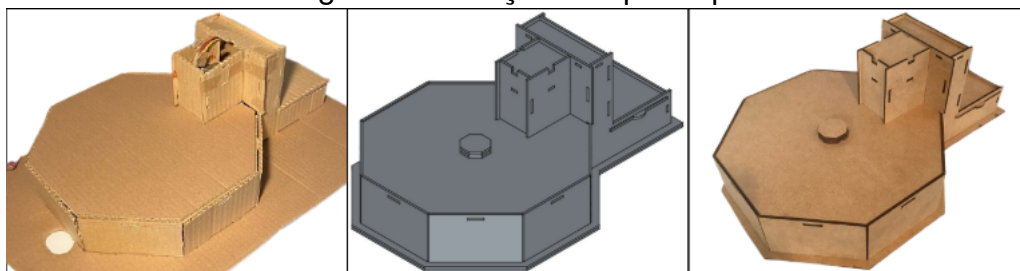


Fonte: Elaborado pelo autor.

Apesar de não ter sido necessário para o corte a laser, a disposição das peças em formato 3D foi essencial para a verificação dos seus encaixes ao longo do desenvolvimento dos desenhos técnicos, assim como a comparação com o protótipo de papelão já existente.

Na Figura 7 é possível verificar lado a lado o protótipo de papelão, o modelo 3D, e o protótipo final do sistema embarcado, e na Figura 8 pode-se observar o protótipo aberto com os seus componentes internos.

Figura 7 - Iterações do protótipo.



Fonte: Elaborado pelo autor.

Figura 8 - Protótipo aberto.



Fonte: Elaborado pelo autor.

3 DISCUSSÃO E RESULTADOS

O sistema embarcado se demonstrou capaz de armazenar as oito fotografias capturadas pelo ESP32-CAM, com o uso do protocolo MQTT sendo essencial para os seus envios do *hardware* para a plataforma Node-RED, que por sua vez facilitou a validação e união das partes para o armazenamento no banco de dados.

Destaca-se o tempo necessário para a captura e armazenamento de todos os diapositivos, que se deu principalmente pela quantidade de *chunks* por imagem, com uma média de 14 nos testes realizados, e pelo tempo de espera colocado entre o envio de cada *chunk* de uma imagem, definido a partir de testes como 2 segundos. Esse tempo foi necessário pois o servidor MQTT não suporta o envio de muitas mensagens repetidamente.

Na Tabela 1 são apresentados os tempos levados do movimento do motor até o armazenamento de cada diapositivo, medidos em três testes diferentes realizados em uma rede Wi-Fi com 90.63 Mbps de *download* e 42.58 Mbps de *upload*.

Tabela 1 – Tempo para o armazenamento de cada diapositivo.

| Diapositivo | Teste 1 | Teste 2 | Teste 3 |
|-------------|-----------|----------|----------|
| Primeiro | 01min 01s | 36s | 37s |
| Segundo | 37s | 36s | 37s |
| Terceiro | 37s | 34s | 37s |
| Quarto | 36s | 37s | 37s |
| Quinto | 36s | 37s | 36s |
| Sexto | 36s | 36s | 36s |
| Sétimo | 37s | 36s | 36s |
| Oitavo | 35s | 36s | 34s |
| Total | 5min 15s | 4min 48s | 4min 50s |

Fonte: Elaborado pelo autor.

É válido destacar o tempo que o primeiro diapositivo do primeiro teste levou para ser armazenado, de um minuto e um segundo, significativamente maior que o tempo dos outros envios. Isso se deu pois o Node-RED não recebeu todos os *chunks* enviados pelo microcontrolador, e por isso requisitou o reenvio.

Outro ponto importante a salientar é a baixa qualidade das imagens capturadas. Apesar da câmera OV2640 acoplada ao ESP oferecer uma resolução de até dois megapixels, que já não é considerada alta, as imagens capturadas têm apenas 0,3 megapixels (resolução de 640 por 480 pixels), uma densidade de 53 pixels por polegada (ppi), e uma qualidade de compressão JPEG de 40/100.

Essa discrepância entre a resolução máxima que a câmera oferece e a resolução utilizada nas imagens registradas se dá pelas configurações de qualidade adotadas no código. Devido a utilização do MQTT, a tentativa do envio de imagens de maior resolução aumenta a quantidade de partes a serem enviadas, assim como a ocorrência da perda de da-

dos e conseqüentemente o total de reenvios necessários. Por esse motivo optou-se por reduzir a resolução para garantir uma maior estabilidade na transmissão.

A estrutura do sistema embarcado atendeu as necessidades básicas do projeto, mas na etapa de testes sobressaíram-se pontos que não foram pensados previamente, como a altura da câmara que abriga o microcontrolador, que não fornece espaço suficiente para os fios conectados a ele, assim como o local onde se encontra o *Shield*, que não provê entradas para o seu carregamento e ligação, obrigando o usuário a removê-lo quando as baterias precisam ser carregadas.

A pesquisa conduzida por Tierra e Segovia (2023, tradução nossa) focou no desenvolvimento de um dispositivo digitalizador que captura a fotografia de um diapositivo por vez, a partir de uma câmara fotográfica desacoplada, com uma abordagem que visa a modificação de diversos parâmetros específicos da captura. Enquanto a atual pesquisa se concentra na facilidade de digitalização de diversas transparências por indivíduos leigos na área da fotografia, sem a necessidade de configurações manuais.

Bennett (2020, tradução nossa) analisou a utilização de uma mesa programável e modular, com performance profissional e sem limitações de custo, focada em projetos arquivísticos museológicos. Em contraste, a presente pesquisa se direciona a instituições educacionais sem recursos financeiros significativos, com a possibilidade de armazenamento de imagens com menor qualidade.

Lima (2020) desenvolveu um aparelho para a captura de radiografias, que utiliza um smartphone Moto g7 Play como dispositivo digitalizador, focando na fácil manipulação e baixo custo. Apesar dessa pesquisa compartilhar do enfoque no baixo custo e fácil manuseio, o armazenamento automático em um sistema de banco de dados na nuvem as diferencia, já que a utilização de um celular limita o funcionamento do sistema para a estrutura de câmeras e tamanho de dispositivos específicos.

A pesquisa de Sullivan e Larrimore (2018, tradução nossa) explora também a modularidade na captura de diapositivos, mas com um foco expandido no pós-processamento das imagens, nos seus fluxos de trabalho, e na implementação de protocolos para a validação de metadados. Em comparação, a atual pesquisa possui um interesse maior na captura e armazenamento confiável das fotografias, com dados simplificados.

4 CONCLUSÃO

Esse trabalho foi composto pelo desenvolvimento de um protótipo de um sistema embarcado para a digitalização e armazenamento de diapositivos escolares via Internet, que possibilite a instituições de ensino capturar e salvar imagens das transparências de seus acervos de forma fácil, simples e econômica.

O *hardware* empregado atendeu as necessidades do sistema, funcionando corretamente nos testes realizados. Entretanto, devido a utilização do protocolo MQTT, não se pôde aproveitar a resolução máxima que a câmera integrada oferece, ou utilizar um módulo de maior resolução, fazendo com que as imagens capturadas não apresentassem qualidade suficiente para uma real conservação.

A aplicação também atendeu aos requisitos do protótipo, mas com ressalvas, a dependência da comunicação no protocolo MQTT significa que o tempo necessário para o envio de cada imagem pode variar conforme a velocidade da rede, um ponto que pode ser impeditivo para operações em locais com sinal fraco.

Apesar dos testes terem mostrados pontos positivos, para a comercialização ou utilização em instituições educacionais, devem ser consideradas outras formas de envio para o armazenamento das fotografias, melhorias no hardware, como a integração de uma câmera de maior resolução, e a aplicação de uma antena para a ampliação do alcance e estabilidade da conexão Wi-Fi.

Com base nos conhecimentos adquiridos, bem como nos resultados obtidos, se propõe para trabalhos futuros: estudar outros métodos para o envio das imagens pela Internet; adicionar uma câmera de melhor resolução; incorporar uma antena externa ao ESP; integrar um *display* permitindo uma maior interação com o sistema; e refinar a estrutura do sistema embarcado.

REFERÊNCIAS

BENNETT, M. J. **Automating 35mm Photographic Film Digitization: XY Table Capture System Design and Assessment**. 2020.

CORTEZ, H. B. de C. et al. **Arquitetura IoT para monitoramento de planta industrial: uma comparação dos protocolos MQTT e ESP-Now**. 2022.

CRUZ, F. H. S. d. **Ações de digitalização em acervos documentais em escolas públicas: uma revisão bibliográfica**. [S.l.]: Brasil, 2021.

- CUNHA, A. F. **O que são sistemas embarcados.** 2007. 1–6 p.
- FOGGIATTO, J. A.; VOLPATO, N.; BONTORIN, A. C. B. **Recomendações para modelagem em sistemas CAD-3D.** 2007. 91 p.
- FUJITA, M. S. L. et al. **A memória acadêmica em imagens fotográficas: representação documentária e digitalização de fotografias.** Seminário Nacional de Bibliotecas Universitárias, v. 13, p. 2004.
- GALLASSI, T. T.; MARTINS, L. E. G. **Um estudo exploratório sobre sistemas operacionais embarcados.** [S.l.]: 004, 2013.
- LIMA, A. d. P. **Desenvolvimento de um digitalizador de radiografias periapicais de baixo custo.** 2020.
- MALTA, A. O. L. **Coleção francisco rodrigues—a digitalização de fotografias dos séculos xi e xx para o portal domínio público, uma estratégia de preservação e difusão da memória.** 2011.
- OLIVEIRA, M. A. M. de. **Escola Maria Teresa Vilanova Castilhos—escola polivalente: acervos fotográficos e história da educação.** 2017. 323–336 p.
- SOUZA, R. F. d. **Preservação do patrimônio escolar no Brasil: notas para um debate.** 2013.
- SULLIVAN, B.; LARRIMORE, W. **Digitizing and Managing 35mm Mounted Slides: The Flip Side.** 2018. 10–20 p.
- TIERRA, M. S. T.; SEGOVIA, F. E. I. **Diseño y construcción de un prototipo digitalizador de imágenes registradas en material fotosensible.** [S.l.]: Escuela Superior Politécnica de Chimborazo, 2023.
- VIERO, T. V. et al. **Acervos escolares: possibilidades e aproximações entre a Arquivologia e a Museologia.** 2017.
- WEIDNER, T. **Fading Out: The End of 35 mm Slide Transparencies.** 2013. 160 p.