REVISTA OBSERVATORIO DE LA ECONOMIA LATINOAMERICANA
Curitiba, v.21, n.10, p. 15350-15373. 2023.

ISSN: 1696-8352

Desenvolvimento de uma plataforma web para sensoriamento remoto com VANT

Development of a web platform for remote sensing with UAVs

DOI: 10.55905/oelv21n10-045

Recebimento dos originais: 01/09/2023
Aceitação para publicação: 02/10/2023

Marcelo Figueiredo Terenciani
Mestre em Ciência da Computação
Instituição: Instituto Federal do Paraná, Informação, Comunicação e Informática
Endereço: Paranavaí - PR, Brasil, CEP: 87703-536
E-mail: marcelo.terenciani@ifpr.edu.br

Everton Castelão Tetila
Pós-Doutor em Inteligência Artificial
Instituição: Universidade Federal da Grande Dourados, Ciências Exatas e Tecnologia (UFGD - FACET)
Endereço: Dourados - MS, Brasil, CEP: 79825-070
E-mail: evertontetila@ufgd.edu.br

Igor Donatti Gonçalves da Silva
Graduado em Engenharia de Computação
Instituição: Universidade Federal da Grande Dourados, Ciências Exatas e Tecnologia (UFGD - FACET)
Endereço: Dourados - MS, Brasil, CEP: 79825-070
E-mail: igordonatti.id@gmail.com

Juliana Queiroz da Silva Tetila
Mestra Profissional em Matemática
Instituição: Universidade Estadual de Mato Grosso do Sul, Pró-Reitoria de Ensino (UEMS – PROE)
Endereço: Dourados - MS, Brasil, CEP: 79804-970
E-mail: ju@uems.br

Jayme Garcia Arnal Barbedo
Pós-Doutor em Engenharia Elétrica
Instituição: Embrapa Agricultura Digital
Endereço: Campinas - SP, Brasil, CEP: 13083-886
E-mail: jayme.barbedo@embrapa.br

ISSN: 1696-8352

RESUMO

As demandas atuais de eficiência econômica e ambiental da agricultura moderna indicam a necessidade de incorporar novos sistemas inteligentes de automação e manejo capazes de processar os dados coletados em campo para o monitoramento preciso das lavouras. O monitoramento viabiliza a aplicação de insumos em taxas variadas, orientando a gestão dos sistemas agrícolas, em harmonia com as questões ambientais. Além disso, possibilita aumentar a produção de alimentos no campo para atender o crescimento da população mundial. Nesse sentido, o objetivo deste trabalho foi desenvolver uma plataforma web de envio de imagens capturadas pelo VANT para um servidor de processamento. Primeiramente foi realizada uma pesquisa dos fundamentos teóricos necessários para o desenvolvimento de uma plataforma web com o intuito de apoiar o sensoriamento remoto agrícola por VANT. Na sequência, o processo de desenvolvimento do sistema foi iniciado observando as quatro atividades fundamentais para a engenharia de software: especificação, projeto e implementação, validação e evolução. Ao mesmo tempo, foi configurado um ambiente de testes para que pudessem ser executadas entregas contínuas durante o processo de desenvolvimento, utilizando a plataforma Heroku. O teste de Tukey mostrou que não há evidência de diferença significativa entre o tempo de envio ao armazenar os dados diretamente no Heroku ou em nuvem. Os resultados indicam que a plataforma web pode ser usada com sucesso para apoiar especialistas e agricultores nos campos de produção agrícola.

Palavras-chave: plataforma web, sensoriamento remoto, VANT.

ABSTRACT

The current demands for economic and environmental efficiency of modern agriculture indicate the need to incorporate new intelligent automation and management systems capable of processing the data collected in the field for the precise monitoring of crops. Monitoring makes it possible to apply inputs at varied rates, guiding the management of the agricultural systems, in harmony with environmental issues. Furthermore, it makes it possible to increase the production of food in the field to attend to the growth of the world population. In this sense, the objective of this work was to develop a web platform for sending images captured by the UAV to a processing server. First, research was carried out on the theoretical foundations necessary for the development of a web platform to support agricultural remote sensing by UAVs. Subsequently, the system development process was initiated by observing the four key activities for software engineering: specification, design and implementation, validation and evolution. At the same time, a testing environment was set up so that continuous deliveries could be made during the development process, using the Heroku platform. Tukey's test showed that there is no evidence of a significant difference between shipping time when storing data directly in Heroku or in the cloud. The results indicate that the web platform can be successfully used to support experts and farmers in the agricultural production fields.

Keywords: web platform, remote sensing, UAV.

ISSN: 1696-8352

1 INTRODUÇÃO

As demandas atuais da agricultura moderna indicam a necessidade de aumentar a produção de alimentos no campo para atender o crescimento da população mundial. Após a colheita dos cultivos de primeira safra e consolidação do plantio das culturas de segunda safra, as previsões iniciais de mais uma safra recorde no Brasil vêm se confirmando, com um volume estimado em 322,8 milhões de toneladas, crescimento de 18,4% ou 50,1 milhões de toneladas sobre a safra anterior. Tal crescimento é sustentado pelo aumento de 5,3% ou 3,9 milhões de hectares na área cultivada, totalizando 78,5 milhões de hectares e, sobretudo, pela boa performance da soja, do milho e do trigo [CONAB 2023].

Apesar desse quadro, muitas culturas agrícolas estão sujeitas à incidência de estresses bióticos (plantas invasoras, pragas e doenças) e abióticos (déficit hídrico e nutricional) que podem alterar seu crescimento e, consequentemente, a qualidade e o rendimento dos grãos. O conhecimento do dano causado por esses agentes é importante para otimizar as aplicações de defensivos agrícolas e, com isso, reduzir o alto custo de controle químico e o impacto ecológico decorrente da contaminação do ambiente [Tetila 2019].

A aplicação sistemática de defensivos caracteriza-se por um considerável desperdício de produtos químicos, mão de obra e energia. Estima-se que a maior parte do montante dos custos diretos para a produção de soja estejam concentrados nos fertilizantes (27,82%), seguidos pelos defensivos agrícolas (18,24%), operações com máquinas (9,10%), sementes (7,35%) e depreciação de máquinas e implementos (6,76%) [CONAB 2016].

O monitoramento de agentes patógenos e pragas agrícolas em fases iniciais de epidemias são aspectos fundamentais para a proteção da lavoura [Hillnhu¨tter and Mahlein 2008]. Essas informações geralmente são obtidas em inspeções regulares por amostragem para a verificação dos níveis de infestação por área e os sintomas de ataque. Assim, o controle químico ou biológico deve ser realizado somente quando o nível de ação de um problema for atingido, visto que a aplicação de defensivos na área inteira aumenta os custos de produção e contribui para o desequilíbrio ecológico. Além disso, inspeções regulares são difíceis de serem realizadas em grandes áreas.

ISSN: 1696-8352

A fim de aprimorar esse monitoramento, existe uma motivação crescente em usar Veículos Aéreos não Tripulados (VANT) na agricultura, pois eles são capazes de sobrevoar uma plantação a poucos metros de distância e capturar imagens em alta resolução. Os VANT podem ser programados para executar operações de plano de voo automático (existem modelos com autonomia superior a uma hora, como senseFly eBee X e Quantum Systems Trinity F90+) cobrindo áreas médias que podem ultrapassar 1.000 hectares por voo. Isso possibilita realizar inspeções regulares em grandes áreas e operações com maior periodicidade no acompanhamento de toda a safra. Inspeções regulares com o VANT também são imunes à presença de nuvens, o que pode encobrir, total ou parcialmente, características visuais importantes.

O trabalho realizado com o levantamento aéreo viabiliza a construção de um banco de dados da propriedade a partir da extração de informações das imagens (p. ex. histórico de produção, falhas de plantio, fertilidade do terreno, análise de biomassa, identificação de plantas infestantes, doenças e pragas) que auxilia na aplicação de insumos em taxas variadas, orientando a gestão dos sistemas agrícolas, em harmonia com as questões ambientais.

Desse modo, o uso de VANT tem sido considerado um importante instrumento para o monitoramento do cultivo de propriedades agrícolas de todos os tamanhos, indicando a necessidade de incorporar novos sistemas inteligentes de automação e manejo que sejam capazes de processar os dados coletados em campo para o monitoramento preciso das lavouras.

Nesse sentido, o objetivo deste trabalho foi desenvolver uma plataforma web de envio de imagens capturadas pelo VANT para um servidor de processamento. Estamos considerando como estudo de caso o processamento das seguintes tecnologias de sensoriamento remoto: ortomosaico, Modelo Digital de Superfície (MDS) e Modelo Digital de Terreno (MDT). Consideramos também as análises agronômicas baseadas em curvas de nível, índices de vegetação, contagem de plantas, presença de daninhas e modelo 3D. A abordagem proposta utiliza cinco etapas para processar os dados coletados em campo na forma de mapas de recomendação: coleta de imagens, transferência de

ISSN: 1696-8352

imagens, processamento e análise agronômica, produção de relatório e mapa de recomendação de aplicação de insumos.

Este artigo está organizado da seguinte forma. A Seção 2 oferece a revisão da literatura. A abordagem proposta para criar a plataforma de processamento de imagem com VANT está descrita na Seção 3. A Seção 4 relata os materiais e métodos adotados nesta pesquisa. A Seção 5 mostra os resultados experimentais, seguidos por uma discussão. Finalmente, conclusões e trabalhos futuros são apresentados na Seção 6.

2 REVISÃO DA LITERATURA

Esta seção apresenta artigos publicados entre 2015 e 2021, pesquisados em revisão sistemática com os termos "plataforma web" e "sensoriamento remoto". Além disso, as palavras-chave também foram traduzidas para o inglês para a realização de buscas em bases internacionais, gerando "remote sensing" e "web-based".

Em [Ferreira et al. 2015] os autores introduziram uma arquitetura de uma plataforma distribuída e de código aberto para interpretação automática de imagens baseada em conhecimento, denominada InterIMAGE Cloud Plataform. Essa plataforma foi projetada para distribuir o processamento e ser capaz de trabalhar com grandes volumes de dados. Entretanto, não contempla a interoperabilidade entre a coleta de imagens e o envio para o processamento.

Tambe´m em 2015, [Bayma-Silva et al. 2015] realizaram um estudo cujo objetivo foi desenvolver metodologias para identificação e monitoramento de níveis de degradação em pastagens dos biomas Amazônia, Cerrado e Mata Atlântica. Nessa plataforma, usuários podiam adicionar dados, ou usar os que estiverem disponíveis, para construir um WebGIS personalizado em forma de mapas. Essa plataforma permitia ao gerente de uma fazenda de pesquisa ou propriedade particular visualizar online as bases de informações geoespaciais da área em questão, auxiliando-o no gerenciamento do uso da terra.

Entre os trabalhos encontrados, [Yamamoto et al. 2019] apresentou detalhes do desenvolvimento de uma plataforma SIG, web e mobile de inventário florestal utilizando princípios de sensoriamento remoto, computação aplicada e geoprocessamento para

ISSN: 1696-8352

auxílio da concepção, planejamento e execução da coleta de campo do inventário florestal. Entretanto, não contemplou a coleta de imagens e não envolveu o processamento das imagens para identificação automatizada de problemas florestais, então não poderia ser adaptada para o escopo deste trabalho.

Abordando aspectos referentes à coleta de imagens, [Cerbaro et al. 2015] estudou com detalhes o desenvolvimento de um VANT. Apesar de apresentar um estudo relevante no que tange a coleta de imagens no campo, esse estudo limitou-se a apresentação da metodologia de construção do VANT, não objetivando o processamento das imagens coletadas.

Para analisar dados de campo na forma de imagens, [Ruiz Guzman et al. 2015] objetivou a construção de uma plataforma web de código aberto para análise de imagens usando Open Source Computer Vision (OpenCV). Na sequência seriam gerados índices vegetativos para prever o desempenho da planta em condições reais. Este trabalho poderia atuar em conjunto com o presente estudo, facilitando a transferência dos dados coletados no campo para a plataforma de análise.

Na área de classificação de imagens, [Lozano Silva et al. 2015] propuseram uma metodologia para a geração de mapas temáticos semiautomáticos para sensoriamento remoto a partir de dados de imagens. A metodologia é integrada a um servidor de mapeamento acoplado a uma interface de supervisão em HyperText Markup Language (HTML) que oferece suporte à navegação interativa e também ao treinamento e ajuste do modelo. Percebe-se que esse trabalho também não aborda aspectos da interoperabilidade entre a coleta de imagens e os softwares de processamento.

No estudo de [Hou et al. 2019] foi desenvolvida uma plataforma baseada na web de acesso aberto chamada de V-RSIR que facilita a participação de voluntários na geração de novos conjuntos de dados de referência para métodos de recuperação de imagens de sensoriamento remoto (RSIR). Contudo, o estudo não focou na coleta das imagens. No estudo, os autores recrutaram 32 voluntários para rotular e recortar imagens de sensoriamento remoto usando a ferramenta.

Um framework web foi criado por [Li et al. 2020] para integração de dados e modelos que permite a extensão dos sistemas envolvidos no sensoriamento remoto. O

ISSN: 1696-8352

objetivo desse trabalho foi construir um sistema unificado para permitir o gerenciamento de dados heterogêneos de múltiplas fontes e a integração de modelos de domínios diferentes. No projeto, pesquisadores de várias disciplinas, por exemplo, estudos ambientais, ecologia e biologia, propõem seus próprios produtos de informação espacial e os algoritmos correspondentes. O sistema serve como uma plataforma única para todo o ciclo de vida desses produtos, incluindo indexação de dados de origem, pré-processamento de dados, cálculo de produtos e compartilhamento de produtos.

Por fim, uma plataforma web de análise de áreas verdes usando imagens de satélite Moderate-resolution Imaging Spectroradiometer (MODIS) foi proposta em [Kalpoma et al. 2020]. Normalized Difference Vegetation Index (NDVI) é o índice de vegetação mais comumente usado em sensoriamento remoto para identificar áreas com vegetação (biomassa) e corpos d'água. Esse estudo teve como meta criar um banco de dados sistêmico para análise de áreas verde da capital de Bangladesh, Dhaka.

Na literatura há vários trabalhos sobre desenvolvimento de plataformas web para sensoriamento remoto. No entanto, não encontramos trabalhos que abordem o envio de imagens coletadas com o VANT para análise e produção de relatório técnico. Neste aspecto, destaca-se a relevância da plataforma desenvolvida para a área de sensoriamento remoto utilizando VANT.

3 ABORDAGEM PROPOSTA

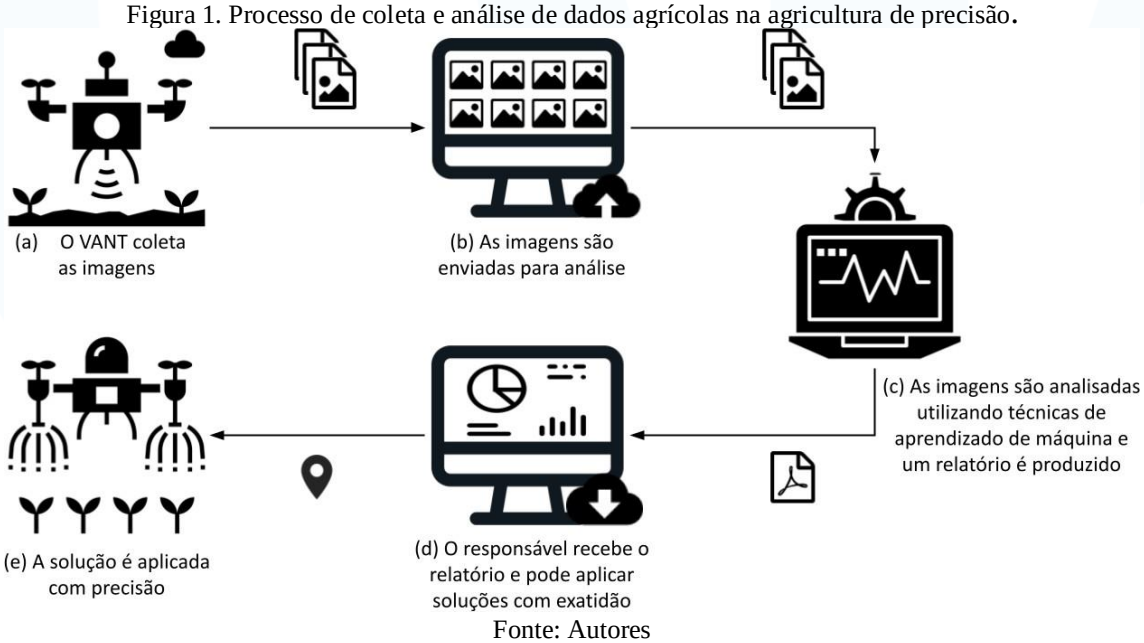
Há uma grande necessidade de fornecer soluções que otimizem o uso dos recursos na produção. É exatamente esse o objetivo da agricultura de precisão. Uma das tecnologias utilizadas são os VANT, com eles é possível coletar dados do campo, produzir relatórios técnicos detalhados e aplicar corretivos de forma precisa, sem desperdício. Entretanto, muitas vezes existe uma barreira logística entre a coleta de dados no campo com os VANT e a análise das imagens em empresas de geoprocessamento. Nessa situação, o operador do VANT precisa levar as imagens coletadas até o local onde a análise será realizada, utilizando uma mídia de armazenamento ou enviando por e-mail, o que dificulta a gestão dessas informações.

ISSN: 1696-8352

Em resumo, o processo de coleta e análise de dados agrícolas na agricultura de precisão por meio de VANT (Figura 1), se inicia com a coleta das imagens no campo (Figura 1a). Na sequência, essas imagens são enviadas para análise por meio de uma plataforma web, mobile, ou até mesmo utilizando um dispositivo de armazenamento, como um pendrive (Figura 1b). Essas imagens são recebidas pelo provedor de serviço responsável por analisar as imagens utilizando as técnicas de aprendizado de máquina e produzir um relatório detalhado (Figura 1c).

Após essa etapa, o responsável pela lavoura recebe o relatório produzido (Figura 1d) e, a partir dele, pode tomar a decisão mais adequada para a sua realidade. Uma das possibilidades é a aplicação localizada de defensivos agrícolas em áreas problemáticas, isto é, com incidência de pragas, doenças e plantas daninhas (Figura 1e). Dessa forma, os custos operacionais podem ser minimizados e a produtividade da lavoura pode aumentar.

Figura 1. Processo de coleta e análise de dados agrícolas na agricultura de precisão.

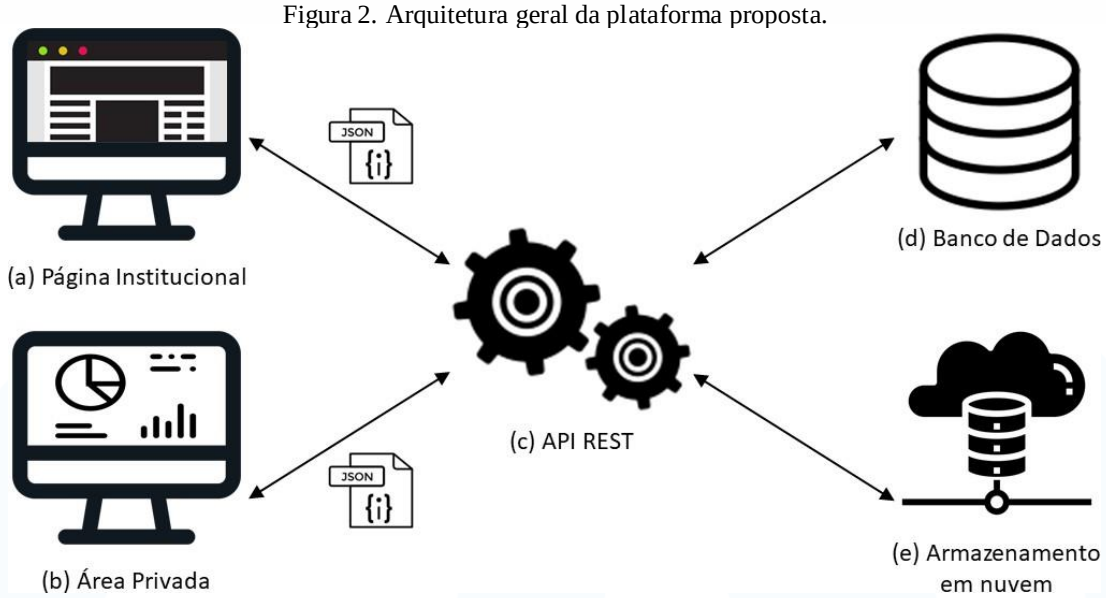


Fonte: Autores

O foco deste trabalho é na segunda etapa do processo (Figura 1b), ou seja, desenvolver uma plataforma que possibilite o envio das imagens coletadas com VANT no campo para análise e produção do relatório técnico. Para solucionar esse problema, o projeto foi organizado de acordo com a arquitetura geral detalhada na Figura 2.

ISSN: 1696-8352

Figura 2. Arquitetura geral da plataforma proposta.



Fonte: Autores

A plataforma proposta possui uma página institucional pública (Figura 2a), logo não é necessário estar autenticado para acessar. Nessa página é possível visualizar os serviços oferecidos, os valores cobrados, entrar em contato com o prestador de serviço, autocadastrar-se para acessar a área privada (Figura 2b). Na área privada, o usuário autenticado pode contratar serviços, alterar seu perfil, solicitar serviços enviando as imagens coletadas no campo, acompanhar as solicitações de serviço e fazer download do relatório técnico produzido.

A área privada possui diferentes níveis de acesso. Além das funcionalidades supracitadas, um usuário com privilégios de administrador pode alterar as informações da página institucional e gerenciar os serviços oferecidos, enquanto usuários podem acompanhar as solicitações de serviço e transferir o relatório final. Nessa área também são oferecidos relatórios gerenciais para os administradores terem maior controle dos serviços prestados.

A página institucional e a área privada compartilham as informações por meio de uma Interface de Programação de Aplicação (API, acrônimo da expressão em inglês Application Programming Interface) com banco de dados MySQL [Oracle 2023]. A transferência dos dados entre os módulos é feita utilizando JavaScript Object Notation

ISSN: 1696-8352

(JSON), tornando-os completamente independentes. Portanto, as manutenções e evoluções da plataforma não afetarão o projeto como um todo.

Para prover uma maior escalabilidade e controle dos custos de armazenamento, optou-se pela utilização de plataformas de armazenamento em nuvem (Figura 2e). Além da redução de custos, essa abordagem garante segurança no acesso às informações e backups das informações armazenadas, bem como caracteriza-se por sua alta disponibilidade. Para o armazenamento em nuvem foi utilizado a funcionalidade de armazenamento de blobs da Azure [Microsoft 2023]. A Tabela 1 detalha os requisitos funcionais das áreas citadas na Figura 2.

Tabela 1. Requisitos funcionais.

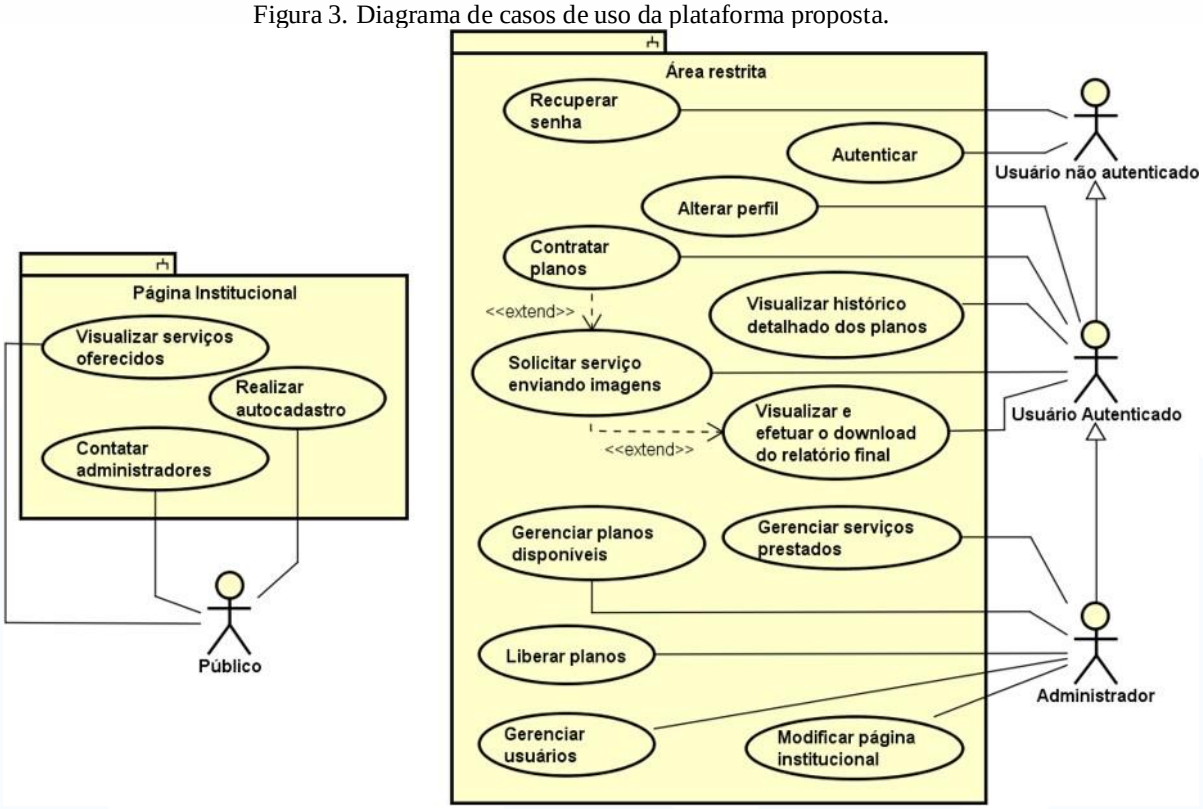
Identificação	**Descrição**
RF01	A plataforma deve permitir a visualização dos planos e serviços prestados pelo provedor de serviços para os usuários não autenticados.
RF02	A plataforma deve permitir que novos usuários realizem autocadastro.
RF03	A plataforma deve permitir que usuários não autenticados enviem mensagem de contato.
RF04	A plataforma deve permitir que usuários recuperem sua senha informando o e-mail cadastrado.
RF05	A plataforma deve permitir que os usuários autenticados contratem um plano. A contratação só pode acontecer se o usuário não possuir outro plano vigente.
RF06	A plataforma deve permitir que os usuários autenticados alterem suas informações cadastrais.
RF07	A plataforma deve permitir que os usuários autenticados alterem sua senha.
RF08	A plataforma deve permitir que os usuários autenticados solicitem serviços por meio do envio de imagens. Para solicitar um serviço o usuário deve ter contratado um plano.
RF09	A plataforma deve permitir que usuários autenticados visualizem e realizem o *download* do relatório técnico produzido em suas solicitações.
RF10	A plataforma deve permitir que usuários autenticados visualizem o histórico detalhado dos planos contratados.
RF11	A plataforma deve permitir que usuários administradores gerenciem os planos disponíveis.
RF12	A plataforma deve permitir que usuários administradores gerenciem os serviços prestados.
RF13	A plataforma deve permitir que usuários administradores modifiquem informações estáticas da página institucional.
RF14	A plataforma deve permitir que usuários administradores liberem os planos para usuários que os contratarem.
RF15	A plataforma deve permitir que usuários administradores gerenciem usuários cadastrados.

Fonte: Autores

A Figura 3 traz o Diagrama de Casos de Uso, da Linguagem de Modelagem Unificada (UML) [OMG 2023]. Esse diagrama possibilita uma melhor compreensa˜o de como os requisitos esta˜o relacionados.

ISSN: 1696-8352

Figura 3. Diagrama de casos de uso da plataforma proposta.



Fonte: Autores

4 MATERIAIS E MÉTODOS

Esta seção apresenta os materiais e o método de desenvolvimento do projeto.

4.1 MATERIAIS

A plataforma foi separada em diferentes módulos, sendo eles: Página Institucional (PI), Área Privada (AP) e API. Isso possibilita que futuras manutenções e extensões não tenham impacto direto no restante do sistema. Além disso, essa arquitetura permite o desenvolvimento de aplicações mobile sem impactar nos módulos já existentes. Por isso, várias tecnologias/frameworks foram utilizadas. A Tabela 2 traz a descrição dos materiais utilizados no desenvolvimento do projeto.

ISSN: 1696-8352

Tabela 2. Materiais utilizados no desenvolvimento do projeto.

Material	Versão	Descrição	PI	AP	API
@azure/storage-blob	12.5.0	Integração com serviços da Azure Cloud.			X
axios	0.21.1	Cliente HTTP para fazer requisições.	X	X	
busboy	0.3.1	Utilizado para envio de grandes arquivos por stream.			X
core-js	3.12.1	Biblioteca modular padrão para JavaScript.			X
cors	2.8.5	Pacote para fornecer conexão com o Express que pode ser usado para habilitar o CORS.			X
dotenv	8.2.0	Facilita a leitura das variáveis de configuração do ambiente.			X
express	14.17.1	Framework para desenvolvimento de aplicações JavaScript com o Node.js	X	X	X
generate-password	1.6.0	Utilizada para geração de senhas temporárias, no caso de recuperação de senha.			X
Git/Github	2021	Ferramenta utilizada para versionamento e disponibilização do código fonte desenvolvido.	X	X	X
Heroku	2021	Ferramenta utiliza para implantação de uma instância do sistema em ambiente de testes.	X	X	X
http-status-codes	2.1.4	Padronização dos códigos de retorno das requisições HTTP.			X
jsonfile	6.1.0	Biblioteca utilizada leitura e salvamento de arquivos de configuração no formato JSON.			X
jsonwebtoken	8.5.1	Framework utilizado para autenticação e autorização.			X
knex	0.95.1	Criador de consultas SQL para Postgres, MSSQL, MySQL, MariaDB, SQLite3, Oracle e Amazon Redshift.			X
md5	2.3.0	Função JavaScript para criptografia com o algoritmo MD5.	X	X	
multer	1.4.2	Utilizado para lidar com multipart/form-data, que é usado principalmente para fazer upload de arquivos.			X
mysql2	2.2.5	Cliente MySQL para Node.js com foco na performace.			X
node.js	14.15.0	Ambiente de execução JavaScript no servidor.	X	X	
nodemailer	6.5.0	Módulo que utilizado para o envio de e-mails.			X
rimraf	3.0.2	Utilizado para excluir arquivos do servidor.			X
v-currency-field	3.1.1	Biblioteca de máscara para valores monetários.		X	
v-mask	2.2.4	Biblioteca de máscara para campos do tipo telefone, cpf, e-mail.	X	X	
vue	2.6.11	Framework Progressivo JavaScript utilizado para construção de páginas.	X	X	
vue-router	3.2.0	Utilizado para navegar entre as rotas/páginas.	X	X	
vue-video	0.1.7	Módulo utilizado para exibir o vídeo em páginas.	X	X	
vuetify	2.4.0	Biblioteca de interface gráfica baseada nos componentes do Material UI.	X	X	
vuex	3.4.0	Módulo para controle de estado de variáveis.	X	X	

Fonte: Autores

A plataforma foi implementada na linguagem JavaScript, utilizando o framework Node.js [OpenJS Foundation 2023], com banco de dados MySQL [Oracle 2023]. A comunidade Node.js é extensa e por isso existem diversos pacotes disponíveis por meio do gerenciador de pacotes Node Package Manager (NPM) [npm, Inc. 2023]. Portanto, todas as tecnologias utilizadas estão disponíveis no NPM.

Utilizamos o tempo de envio de arquivos em solicitações de serviço como métrica para avaliar o desempenho da plataforma considerando duas estratégias de armazenamento: local e em nuvem. Calculamos a média entre os tempos de envio e

ISSN: 1696-8352

usamos o teste de hipóteses ANOVA1 para determinar se existe diferença significativa no desempenho entre as estratégias. Registramos os valor-*p* encontrado e estabelecemos o nível de significância em 5%. Fizemos a realização dos testes estatísticos no software RStudio [POSIT 2023], versão 1.0.136, usando R na versão 4.1.0.

4.2 MÉTODO DE DESENVOLVIMENTO

O processo de desenvolvimento do sistema foi iniciado observando as quatro atividades fundamentais para a engenharia de software: especificação, projeto e implementação, validação e evolução [Sommerville 2019]. Primeiro foi realizado um levantamento das principais funcionalidades do sistema (requisitos funcionais e não funcionais). Esse levantamento baseou-se em um protótipo de sistema já existente. Além disso, um colaborador do projeto auxiliou no levantamento das funcionalidades inexistentes no protótipo.

A partir da visão geral dos requisitos foi definida a arquitetura geral do sistema e também a definição do banco de dados, dando início à implementação do software. Baseando-se nas metodologias ágeis, foi configurado um ambiente de testes para que pudessem ser executadas entregas contínuas durante o processo de desenvolvimento, utilizando a plataforma Heroku [Salesforce 2023]. Além disso, houve controle de versões durante todo o desenvolvimento por meio do Github [GitHub 2023].

Com base nas entregas contínuas foi possível coletar feedbacks continuamente e, com isso, tornar a solução mais robusta. Paralelamente ao desenvolvimento do sistema, foram escritas orientações sobre a implantação em um ambiente real, facilitando essa tarefa para os futuros usuários. Essas orientações estão disponíveis nos repositórios onde o código foi disponibilizado. Como a solução foi disponibilizada no Github [Terenciani 2023], as evoluções da plataforma podem ser realizadas por qualquer pessoa interessada neste projeto.

[1] ANOVA, ou análise de variância, é uma técnica estatística que permite verificar se existe uma diferença significativa entre as médias de populações. A análise de variância é utilizada quando se quer decidir se as diferenças amostrais observadas são reais (causadas por diferenças significativas nas populações observadas) ou casuais (decorrentes da mera variabilidade amostral). Portanto, essa análise parte do pressuposto que o acaso só produz pequenos desvios, sendo as grandes diferenças geradas por causas reais

ISSN: 1696-8352

Após o término da funcionalidade de envio de imagens, realizamos uma bateria de testes de envio de arquivos de vários tamanhos para verificar o comportamento da aplicação em diferentes cenários.

5 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados e a discussão. Para uma melhor análise dos resultados, a Seção 5.1 apresenta os detalhes do ambiente computacional onde os testes foram executados. A Seção 5.2 mostra os resultados experimentais, seguidos por uma discussão.

5.1 CONFIGURAÇÃO DO AMBIENTE DE TESTES

Os testes buscaram registrar o tempo de envio em relação ao tamanho e quantidade de arquivos. Dessa forma, buscou-se realizá-los em configurações próximas às utilizadas pelos usuários. Assim sendo, os testes foram realizados em uma máquina local e no ambiente Heroku.

A máquina local possui 8 Gigabytes (GB) de memória, processador i5-3230M, com 223 GB Solid State Drives (SSD) para armazenamento interno. O ambiente no Heroku possui 512 Megabytes (MB) de memória, 64 GB de armazenamento interno, com uma Unidade Central de Processamento (CPU) compartilhada. Ambos os testes foram realizados em rede sem fio, com uma velocidade média de 70 Megabits por segundo (Mbps). Para o armazenamento em nuvem foi utilizada a funcionalidade de armazenamento de blobs da Azure.

Para fins de comparação entre diferentes arquiteturas, foram analisados envios com origem da máquina local armazenando localmente contra armazenamento em nuvem; e envios originários da interface web hospedada no Heroku com armazenamento local contra armazenamento em nuvem. As quatro combinações possíveis foram avaliadas durante os testes.

A execução dos testes foi separada em duas etapas. Inicialmente, analisou-se o envio de um arquivo compactado de tamanhos 0,212 GB, 0,425 GB, 0,850 GB, 1,699 GB e 3,399 GB. Em cada arquivo compactado havia, respectivamente, 50, 100, 200, 400 e

ISSN: 1696-8352

800 imagens. Foram enviadas 5 solicitações para cada tamanho de arquivo com a finalidade de obter a média entre elas.

No segundo cenário de testes foram enviadas várias imagens em solicitações de serviço de diferentes tamanhos, sendo eles, 0,217 GB, 0,434 GB, 0,867 GB, 1,800 GB e 3,500 GB. Seguindo a mesma lógica, em cada solicitação de serviço haviam, respectivamente, 50, 100, 200, 400 e 800 imagens. Também foram enviadas 5 solicitações para cada tamanho de arquivo com a finalidade de obter a média entre elas.

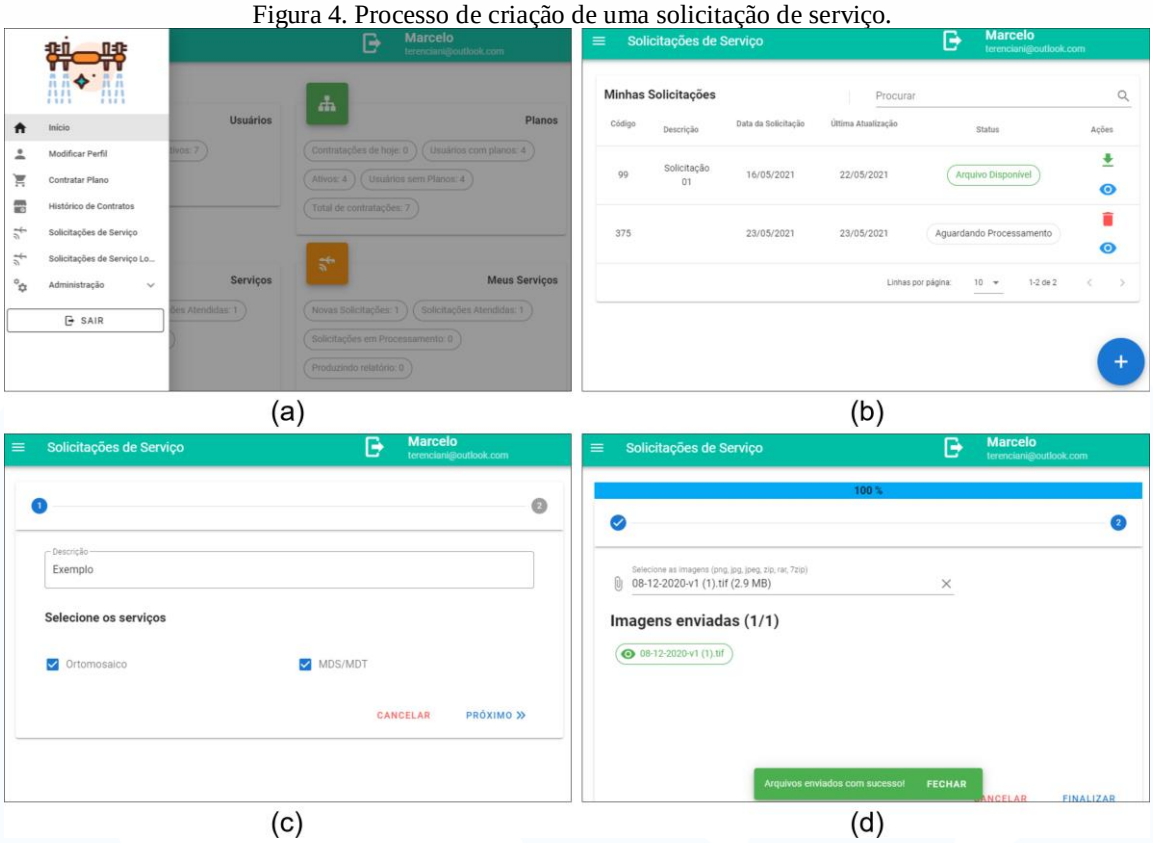
Dessa forma, considerando o envio de um arquivo compactado foram enviadas 25 solicitações da máquina local com armazenamento local (localhost_localhost), 25 solicitações da máquina local com armazenamento em nuvem (localhost_cloud), 25 solicitações do Heroku com armazenamento local (heroku_local) e 25 solicitações do Heroku com armazenamento em nuvem (heroku_local). O mesmo se aplicou para vários arquivos, totalizando 200 solicitações.

5.2 ANÁLISE DOS RESULTADOS

A Figura 4 apresenta os screenshots referentes ao processo de criação de uma solicitação de serviço.

ISSN: 1696-8352

Figura 4. Processo de criação de uma solicitação de serviço.



Fonte: Autores

Após realizar a autenticação, a página inicial é exibida com informações úteis, no menu de navegação lateral há a opção "Solicitações de Serviço" (Figura 4a). Na sequência a lista das solicitações de serviço do usuário é exibida (Figura 4b). Nessa área é possível selecionar a opção de criação de uma nova solicitação. Dentro da área de criação de solicitação, o usuário deve selecionar ao menos um serviço e pode inserir uma descrição (Figura 4c). No próximo passo, as imagens são selecionadas, enviadas, e a mensagem de confirmação é exibida. Por fim, o usuário pode cancelar ou finalizar a solicitação (Figura 4d).

A funcionalidade apresentada na Figura 4 foi testada por meio da criação de solicitações de serviço, conforme descrito na seção anterior. A Tabela 3 mostra os tempos de execução (em minutos) considerando o local de hospedagem da aplicação e a forma de armazenamento.

ISSN: 1696-8352

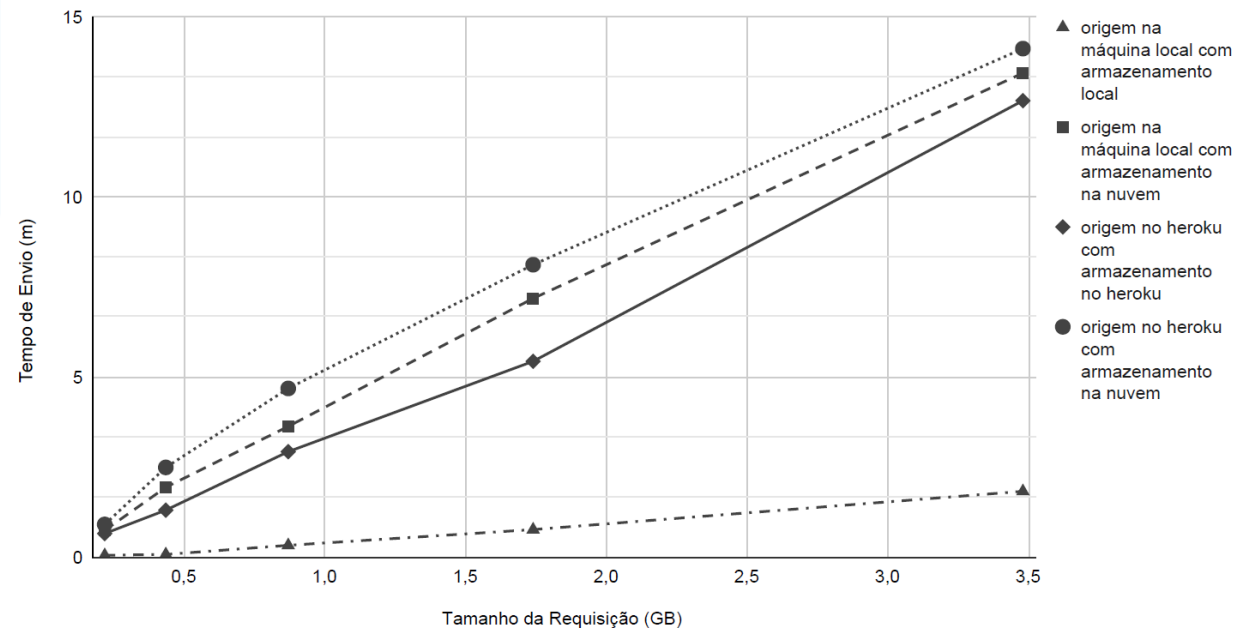
Tabela 3. Tempo de envio de um arquivo compactado com diferentes tamanhos.

localhost_localhost		localhost_cloud		heroku_heroku		heroku_cloud	
Tamanho (GB)	Tempo (m)	Tamanho (GB)	Tempo (m)	Tamanho (GB)	Tempo (m)	Tamanho (GB)	Tempo (m)
0,212	0,042	0,212	0,733	0,212	0,646	0,212	0,899
0,425	0,066	0,425	1,931	0,425	1,296	0,425	2,484
0,850	0,319	0,850	3,622	0,850	2,921	0,850	4,677
1,699	0,756	1,699	7,179	1,699	5,431	1,699	8,115
3,399	1,820	3,399	13,438	3,399	12,673	3,399	14,119

Fonte: Autores

A Figura 5 mostra os resultados do tempo de envio, considerando os valores da Tabela 3. Quando a aplicação está em ambiente local com armazenamento local, o tempo de envio é significativamente menor que o obtido com a hospedagem no Heroku. Entretanto, quando o armazenamento é feito em nuvem, o tempo de envio é similar ao tempo da aplicação hospedada no Heroku. Além disso, os tempos de envio para o armazenamento nesse provedor e em nuvem são similares.

Figura 5. Relação entre tempo de envio e tamanho da solicitação considerando o envio de um arquivo compactado.



Fonte: Autores

A Tabela 4 mostra os tempos de envio considerando o local de hospedagem da aplicação e a forma de armazenamento para o segundo cenário.

ISSN: 1696-8352

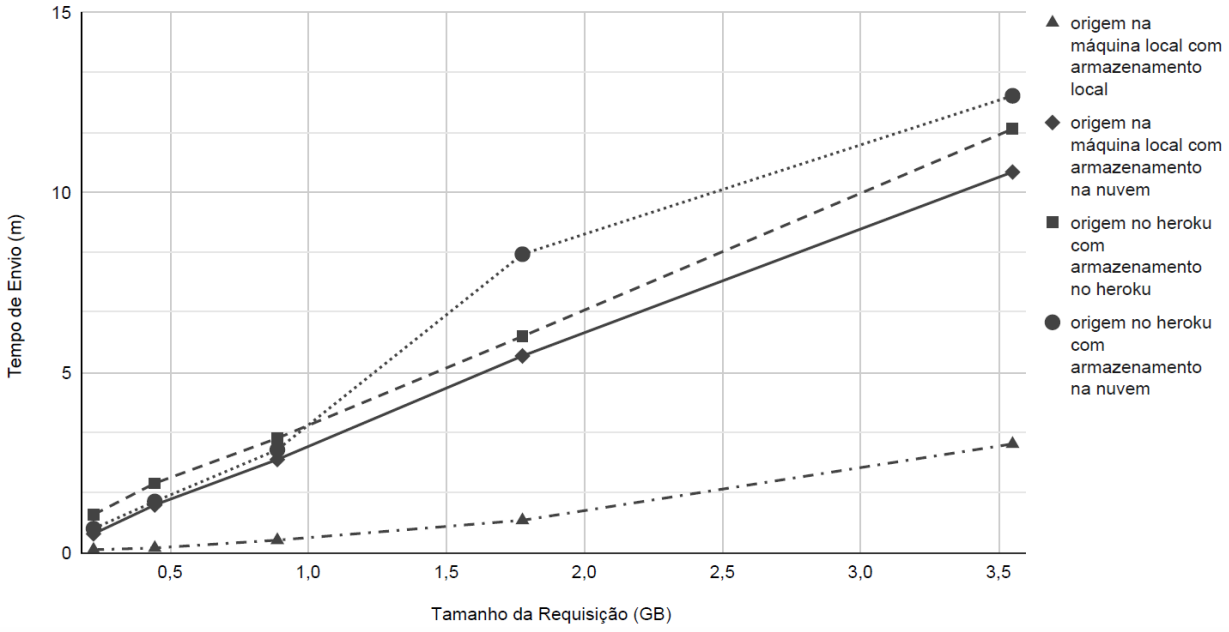
Tabela 4. Tempo de envio de vários arquivos com diferentes tamanhos.

localhost_localhost		localhost_cloud		heroku_heroku		heroku_cloud	
Tamanho (GB)	**Tempo (m)**	**Tamanho (GB)**	**Tempo (m)**	**Tamanho (GB)**	**Tempo (m)**	**Tamanho (GB)**	**Tempo (m)**
0,217	0,073	0,217	0,514	0,217	1,047	0,217	0,660
0,434	0,122	0,434	1,316	0,434	1,918	0,434	1,413
0,867	0,339	0,867	2,578	0,867	3,170	0,867	2,850
1,800	0,891	1,800	5,455	1,800	6,000	1,800	8,282
3,500	3,012	3,500	10,564	3,500	11,760	3,500	12,681

Fonte: Autores

Considerando os valores apresentados nas Tabelas 3 e 4 é possível observar que ao enviar um arquivo compactado de 3,399 (GB) da máquina local para o armazenamento em nuvem tem-se um tempo de 13,438 minutos, enquanto 800 imagens (3,500 (GB)) foram enviadas em 10,564 minutos. O mesmo acontece para as outras estratégias de armazenamento: o tempo de envio de várias imagens é menor do que o tempo de um arquivo compactado. A Figura 6 mostra o gráfico considerando os valores da Tabela 4. Nele podemos observar um comportamento semelhante ao gráfico apresentado na Figura 5.

Figura 6. Relação entre tempo de envio e tamanho da solicitação considerando o envio de múltiplos arquivos.



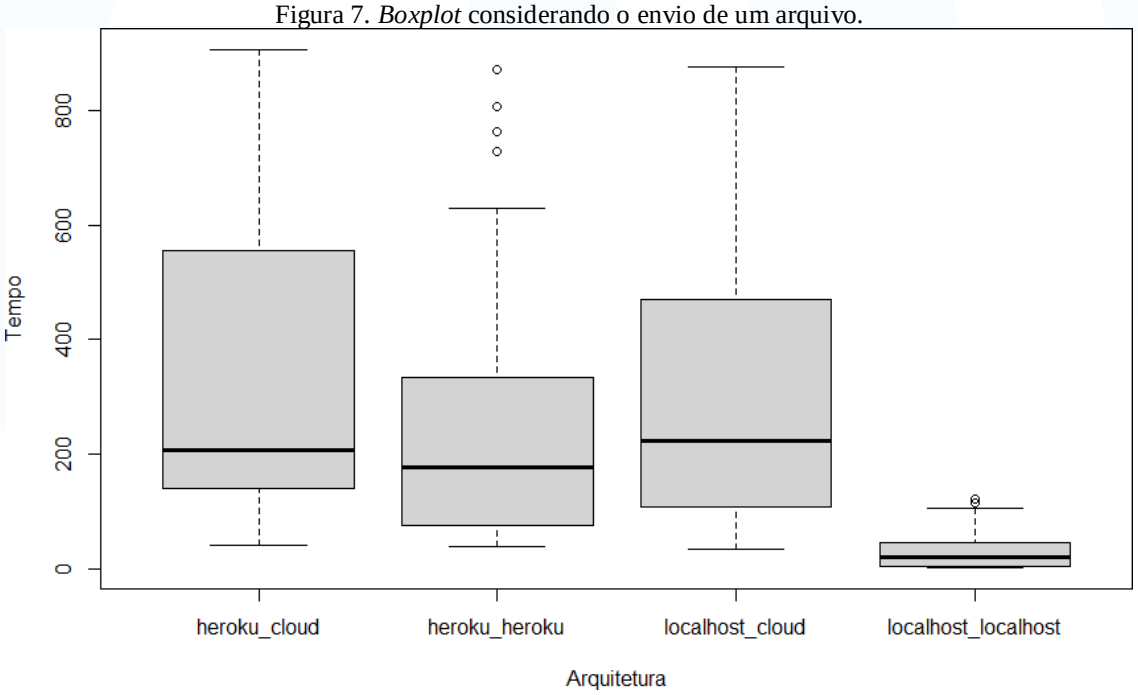
Fonte: Autores

ISSN: 1696-8352

Para cada estratégia de armazenamento foi calculado o desempenho médio do tempo de envio para cada requisição contendo um arquivo ou vários arquivos de diferentes tamanhos. As estratégias de armazenamento usadas nos experimentos foram avaliadas estatisticamente em relação ao desempenho com o teste de hipóteses ANOVA.

Nos experimentos, relatou-se os valores-p encontrados para o tempo de envio e usou-se um nível de significância de 5% para descartar a hipótese nula. Os valores-p foram analisados com o pós-teste de Tukey para comparação das variáveis entre os grupos. Também foi utilizado o diagrama de caixa (boxplot) para analisar e comparar a dispersão das variáveis observadas por meio de quartis entre os diferentes grupos de dados2. Esses gra´ficos podem ser observados nas Figuras 7 e 8.

Figura 7. *Boxplot* considerando o envio de um arquivo.

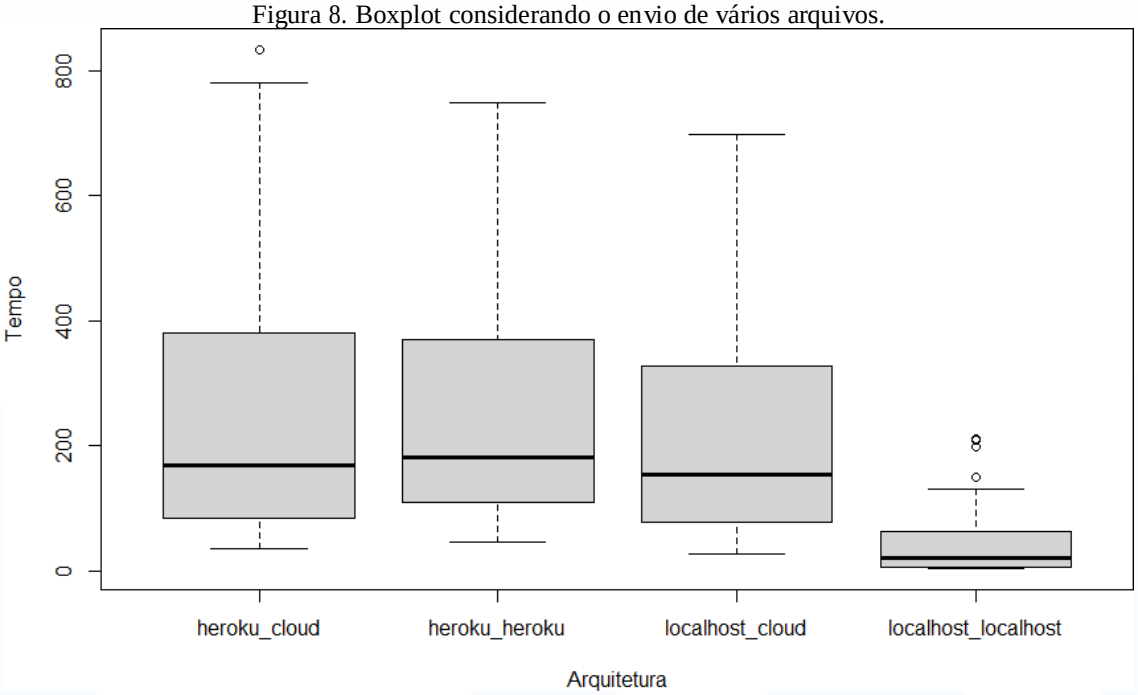


Fonte: Autores

[2] Os espaços entre as diferentes partes da caixa (onde estão localizados 50% dos valores mais prováveis) indicam o grau de dispersão com a mediana em destaque, a obliquidade (medida da assimetria de uma distribuição) nos dados e os *outliers* (valores extremos).

ISSN: 1696-8352

Figura 8. Boxplot considerando o envio de vários arquivos.



Fonte: Autores

Os resultados do teste ANOVA indicam que existe evidência de diferença significativa no desempenho médio entre as estratégias de armazenamento, usando como métrica o tempo de envio de requisição com um arquivo (valor-p = $3.18 \times e^{-05}$) e o tempo de envio de requisição com vários arquivos (valor-p = $4.03 \times e^{-04}$) e um nível de significância de 5%.

Para comparar toda e qualquer diferença entre duas estratégias de treinamento realizamos um pós-teste com o teste de Tukey. Os resultados indicam que existe evidência de diferença significativa entre as estratégias de armazenamento localhost_localhost e heroku_heroku, localhost_localhost e localhost_cloud, localhost_localhost e heroku_cloud. No entanto, não existe evidência de diferença significativa entre heroku_heroku e heroku_cloud, localhost_cloud e heroku_cloud, localhost_cloud e heroku_heroku. A Tabela 5 mostra as comparações de pares de estratégias de armazenamento com os valores-p correspondentes.

ISSN: 1696-8352

Tabela 5. Comparações de pares de estratégias de armazenamento com os valores-p correspondentes.

Estratégia de armazenamento	**Valor-*p***	
	1 arquivo	***n* arquivos**
localhost_localhost e heroku_heroku	0.0048881	0.0012074
localhost_localhost e localhost_cloud	0.0004873	0.0110695
localhost_localhost e heroku_cloud	0.0000537	0.0016212
heroku_heroku e heroku_cloud	0.5916529	0.9997730
localhost_cloud e heroku_cloud	0.9372773	0.9317475
localhost_cloud e heroku_heroku	0.9058857	0.9022424

Fonte: Autores

6 CONCLUSÃO

Neste artigo propomos uma nova plataforma web para envio de imagens capturadas por VANT para um servidor de processamento. As funcionalidades desenvolvidas foram disponibilizadas em um ambiente de testes, onde puderam ser testadas e analisadas computacionalmente. Durante a execução dos testes o envio de imagens para o armazenamento em nuvem apresentou desempenho equivalente ao armazenamento no Heroku.

Entre as quatro estratégias avaliadas, localhost_localhost obteve os melhores resultados. Apesar disso, quase sempre o cliente acessa a aplicação de maneira remota. Resultados com o teste de Tukey mostraram que não há evidência de diferença significativa entre armazenar os dados diretamente no Heroku ou em nuvem. A aplicação está utilizando armazenamento em nuvem devido à escalabilidade – não há limite de armazenamento para as imagens enviadas nesta arquitetura, basta alterar o plano de acordo com a necessidade da aplicação. Por outro lado, Heroku (assim como outros serviços de hospedagem de sites) possuem espaço limitado, podendo ocasionar sobrecarga de armazenamento de dados.

Apesar de ser possível enviar imagens para o servidor, é necessário implementar restrições aos usuários para essa solução ser efetivamente utilizada em um ambiente real. Na versão atual é possível enviar várias requisições com tamanhos distintos, podendo sobrecarregar o servidor e gerar custos no armazenamento em nuvem. Além disso, faz-se necessária a realização de testes de sobrecarga para analisar o comportamento da aplicação em cenários específicos.

ISSN: 1696-8352

A solução desenvolvida encontra-se disponível no Github, dessa forma pode ser melhorada e atualizada. Como trabalhos futuros sugere-se o aprimoramento das regras de negócio da funcionalidade de gestão de planos para não permitir o envio ilimitado de imagens. Na sequência, é possível implantá-la em um ambiente real e realizar testes comparativos com os resultados obtidos neste trabalho.

AGRADECIMENTOS

Agradecemos ao Centro Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Processo: 423256/2021-1, à Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia do estado de Mato Grosso do Sul (FUNDECT) – Processo: 83/007.898/2023 TO: 11/2023 SIAFEM: 32834, à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Processo: 2023/03870-8 e à Embrapa Agricultura Digital/CNPTIA/EMBRAPA – Processo: 2022/09319-9.

ISSN: 1696-8352

REFERÊNCIAS

Bayma-Silva, G., Nogueira, S. F., Drucker, D. P., Siqueira, G. R., Fernandes, R. M., Custo´dio, D. d. O., and Alvarenga, S. V. A. (2015). Plataforma web para sistemas de informac¸a˜o geoespacial (sig): aplicac¸o˜es no projeto geodegrade. In XVII Simpo´sio Brasileiro de Sensoriamento Remoto, Joa˜o Pessoa.

Cerbaro, V. A., Fornari, M., Pavan, W., Fernandes, J., and Cehchetti, N. P. (2015). Plataforma de baixo custo para coleta de imagens ndvi. In XX Congresso Brasileiro de Agroinforma´tica, Ponta Grossa.

CONAB (2016). Compeˆndio de estudos conab: Evoluc¸a˜o dos custos de produc¸a˜o de soja no Brasil, volume 2. Companhia Nacional de Abastecimento, Brasília, DF. 1-22 p. ISSN 2448-3710. Disponível em: https://www.conab.gov.br/institucional/publicacoes/compendio-de-estudos-da-conab. Último acesso: 23-09-2023.

CONAB (2023). Acompanhamento da safra brasileira gra˜os, v.10 - Safra 2022/23, volume 10 of 12º Levantamento. Companhia Nacional de Abastecimento, Brasília, DF. 1-109 p. ISSN 2318-6852. Disponível em: https://www.conab.gov.br/info-agro/safras/graos/boletim-da-safra-de-graos. U´ltimo acesso: 22-09-2023.

Ferreira, R. d. S., Oliveira, D. A. B., Happ, P. N., da Costa, G. A. O. P., Feitosa, R. Q., and Bentes, C. (2015). InterIMAGE Cloud Platform: Em direc¸a˜o a` arquitetura de uma plataforma distribu´ıda e de co´digo aberto para a interpretac¸a˜o automa´tica de imagens baseada em conhecimento. In XVII Simpo´sio Brasileiro de Sensoriamento Remoto.

GitHub (2023). Github: Let's build from here. Disponível em: https://github. com/. Último acesso: 23-09-2023.

Hillnhu¨tter, C. and Mahlein, A.-K. (2008). Early detection and localisation of sugar beet diseases: new approaches. Gesunde Pflanzen - GESUNDE PFLANZ, 60:143–149.

Hou, D., Miao, Z., Xing, H., and Wu, H. (2019). V-rsir: An open access web-based image annotation tool for remote sensing image retrieval. IEEE Access, 7:83852–83862.

Kalpoma, K. A., Leman, M., Islam, M. T., Poddar, S., and Ahmed, J. (2020). Development of greenness analysis tool using remote sensing satellite images. In IEEE International Geoscience and Remote Sensing Symposium, pages 4299–4302.

Li, H., Zhang, Z., and Tang, P. (2020). A web-based remote sensing data processing and production system with the unified integration of multi-disciplinary data and models. IEEE Access, 8:162961–162972.

Lozano Silva, J., Aginako Bengoa, N., Quartulli, M., Olaizola, I. G., and Zulueta, E. (2015). Web-based supervised thematic mapping. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(5):2165–2176.

ISSN: 1696-8352

Microsoft (2023). Armazenamento de blobs do azure. Disponível em: https://azure.microsoft.com/pt-br/services/storage/blobs/. Último acesso: 23-09-2023.

Milone, G. (2003). Estat´ıstica geral e aplicada. Thomson Pioneira, Sa˜o Paulo.

npm, Inc. (2023). Node package manager (npm). Disponível em: https://www.npmjs.com/. Último acesso: 23-09-2023.

OMG (2023). Unified modeling language - uml. Disponível em: https://www.uml. org/. Último acesso: 23-09-2023.

OpenJS Foundation (2023). Node.js. Disponível em: https://nodejs.org. Último acesso: 23-09-2023.

Oracle (2023). Mysql database. Disponível em: https://www.mysql.com/. Último acesso: 23-09-2023.

POSIT (2023). Download rstudio — the popular open-source ide from posit. Disponível em: https://posit.co/products/open-source/rstudio/. Último acesso: 23-09-2023.

Ruiz Guzman, H. A., Gonza´lez-Navarro, F. F., Selvaraj-Gomez, M., Valencia, M., and Delgado, A. (2015). Field phenomics: A web based image analysis platform using open source tools. In 2015 International Conference on Computational Science and Computational Intelligence (CSCI), pages 851–852.

Salesforce (2023). Cloud application platform — heroku. Disponível em: https://www.heroku.com/. Último acesso: 23-09-2023.

Sommerville, I. (2019). Engenharia de Software. 10ª Edic¸a˜o. Pearson Education-BR. Terenciani, M. F. (2023). Reposito´rios do projeto no github. https://github.com/terenciani/{admin-hectaremaps,front-hectaremaps,api-hectaremaps}.

Tetila, E. C. (2019). Detecc¸a˜o e classificac¸a˜o de doenc¸as e pragas da soja usando imagens de ve´ıculos ae´reos na˜o tripulados e te´cnicas de visa˜o computacional. PhD thesis, Universidade Cato´lica Dom Bosco, Campo Grande, MS. Orientac¸a˜o do Prof. Hemerson Pistori.

Yamamoto, M. K., Berbert, M. L. D. G., Gasparoto, E. A. G., Shinzato, E. T., and Ju´nior, M. A. (2019). Desenvolvimento de uma aplicac¸a˜o sig web mobile para o planejamento e a coleta de dados de inventa´rio florestal. In XIX Simpo´sio Brasileiro