



FELIPE LUCAS FERREIRA

**SISTEMA DE MANEJO FLORESTAL:
VISÃO COMPUTACIONAL APLICADA NA DETECÇÃO E
CARACTERIZAÇÃO DE ÁRVORES**

**LAVRAS – MG
2023**

FELIPE LUCAS FERREIRA

**SISTEMA DE MANEJO FLORESTAL: VISÃO COMPUTACIONAL APLICADA NA
DETECÇÃO E CARACTERIZAÇÃO DE ÁRVORES**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia Mecânica para obtenção do título de Bacharel.

Profa. Dra. Ana Paula Piovesan Melchiori
Orientado(a)
Profa. Dra. Joelma Rezende Durão Pereira
Coorientador(a)

DEDICATÓRIA

À Deus por me dar forças em todos os momentos em que precisei e por me dar a oportunidade de chegar ao final desta etapa da minha vida.

À minha família, em especial aos meus pais Flávio e Rosângela por todo amor, carinho e apoio sempre, aos meus avós Alaíde e Antônio, por sempre permanecerem ao meu lado, minhas irmãs Letícia e Rafaela por todo companheirismo e carinho.

À minha noiva Marina por todo amor, cuidado e apoio em todos os momentos e decisões durante esta etapa da minha vida.

Ao meu primo, Leandro, pela amizade e companheirismo que temos desde a infância.

Ao meu amigo que se tornou como um irmão para mim, Fernando (Caverna), por todos os momentos juntos.

Aos meus amigos e companheiros de curso que passaram vários anos comigo: Alan, Antônio, Arthur, Bernardo, Guilherme, Hugo, Marcos, Oscar, Paulo, Samuel e Vinícius.

Às professoras Ana Paula e Joelma, minha orientadora e coorientadora, por todo aprendizado e apoio durante os projetos desenvolvidos e por todas as dicas e conselhos.

À Zetta por apoio com o espaço de pesquisa e desenvolvimento.

RESUMO

O uso dos recursos florestais degrada cada vez mais o meio ambiente, aumentando desta forma a necessidade de utilização deste com maior eficiência. Tendo em vista a constante evolução da tecnologia e a necessidade de controle destes recursos, unir estes dois pontos torna-se uma possibilidade plausível de expansão. Uma vez que a visão computacional é uma área em constante evolução frente ao aumento do desmatamento, o trabalho apresentado trata-se do desenvolvimento de um sistema de visão computacional empregado em conjunto com banco de dados para realizar a gravação e leitura de informações e autenticação para acesso, tendo como principal função a medição do diâmetro do tronco de árvores para emprego no manejo florestal. Desta forma, o objetivo do projeto é criar um sistema que seja capaz de melhorar a eficiência da coleta de dados e ter melhor aproveitamento das florestas plantadas, fazendo isto por meio do aumento de velocidade de obtenção dos diâmetros dos troncos de árvores dado bem como a criação de dados históricos armazenados em banco de dados para construção de séries temporais para o acompanhamento do crescimento florestal. O sistema foi desenvolvido inteiramente utilizando a linguagem Python em conjunto com suas bibliotecas, sendo as principais o OpenCV e Numpy para manipulação das imagens da câmera e kivy para desenvolvimento do aplicativo, além de integração com o banco de dados do Google, o Firebase. A metodologia empregada trata-se da estruturação do sistema, o desenvolvimento de seus componentes, simulação e coleta de resultados. A partir dos dados coletados é realizada uma comparação com os métodos convencionais para verificação da eficácia do sistema.

Palavras-chave: Python. OpenCV. Kivy. Firebase. Banco de Dados.

LISTA DE FIGURAS

Figura 1 - Modelo Incremental.....	13
Figura 2 - Requisitos Funcionais.....	14
Figura 3 - Requisitos não Funcionais.....	14
Figura 4 - Logo do Aplicativo.....	17
Figura 5 - Calibração da Medida.....	21
Figura 6 - Coordenadas dos contornos.....	22
Figura 7 - Resumo estatístico - Inicialização do Aplicativo.....	25
Figura 8 - Gráfico de Barras - Tempo de Inicialização.....	26
Figura 9 - Resumo estatístico - Validação de Credenciais.....	27
Figura 10 - Gráfico de Barras - Tempo de Validação de Credenciais.....	27
Figura 11 - Resumo Estatístico - Gravação de Dados.....	28
Figura 12 - Gráfico de Barras - Tempo de Gravação de Dados.....	28
Figura 13 - Resumo Estatístico - Leitura de Dados.....	29
Figura 14 - Gráfico de Barras - Tempo de Leitura de Dados.....	30
Figura 15 - Tempos de Gravação x Leitura.....	31
Figura 16 - Tela de Salvamento de Dados.....	31
Figura 17 - Ajuste da Tabela de Dados.....	32
Figura 18 - Gráfico de Diferença.....	34
Figura 19 - Resumo Estatístico – Medidas do Aplicativo.....	34
Figura 20 - Gráfico de Dispersão - Medida Indireta.....	35

LISTA DE TABELAS

Tabela 1 - Tabela de pontuação da matriz GUT.....	12
Tabela 2 - Matriz GUT para o sistema.....	15
Tabela 3 - Comparação - Medida Direta x Medida Indireta.....	33

SUMÁRIO

1. INTRODUÇÃO	6
2. REFERENCIAL TEÓRICO	7
2.1. DESMATAMENTO	7
2.2. FLORESTAS PLANTADAS.....	7
2.3. MÉTODOS DE CONTROLE DE UMA FLORESTA PLANTADA	8
2.4. INTELIGÊNCIA ARTIFICIAL E A VISÃO COMPUTACIONAL	9
3. MATERIAL E MÉTODOS	12
4. RESULTADOS E DISCUSSÃO.....	24
CONCLUSÃO.....	36
REFERÊNCIAS.....	37
APÊNDICE A	41
APÊNDICE B.....	42
APÊNDICE C	43

1. INTRODUÇÃO

A madeira é uma fonte de matéria-prima muito utilizada desde os tempos mais remotos, e apesar dos avanços da tecnologia, o seu uso ainda é de grande importância no mercado, sendo utilizada para produção de lápis, móveis, casas, entre outros (JÚNIOR; COLODETTE, 2013). Na dinâmica global temos uma quantidade grande de produtos obtidos de recursos florestais e em contrapartida uma capacidade exploratória que no longo prazo já se apresenta insustentável para suportar as necessidades de consumo (ARAÚJO; GARCIA et al., 2017). Por consequência dessa grande exploração das florestas vê-se números elevados de remoção vegetal para fins diversos dando início ao processo de desmatamento, que, apesar de seus números aparentes apresentarem uma queda na quantidade de ocorrência ainda é um assunto delicado de ser tratado (FAO, 2020).

Esses desfalques nos trouxeram a necessidade de desenvolver alternativas que pudessem suprir a demanda por essa matéria-prima de forma menos agressiva ao ambiente. Uma delas é a formação de florestas plantadas, que nada mais é do que a plantação humana de árvores destinadas às atividades econômicas, evitando, dessa forma, a retirada de mais árvores do ecossistema (FISCHER; ZYLBERSZTAJN, 2012). Nesse método, são plantadas árvores em quantidades e espécies conhecidas e estas passam por acompanhamento durante seu crescimento, a fim de se obter informações sobre a saúde delas, tamanho ideal para corte, entre outros (SETTE JR. et al., 2010).

Para fins comerciais, o conhecimento das características estruturais das árvores garante um maior aproveitamento dos recursos disponíveis (ZENID, 2011). Sendo assim uma das características mais comumente analisadas é o diâmetro da árvore. Com esta informação é possível ter um maior controle referente ao tempo correto de tomada de decisão, volume estimado do caule da árvore, entre outros (WOO; KIM; CHOI, 2021). Desta forma, uma possível e eficiente maneira de monitoramento dessas características é o uso da visão computacional, uma vez que esta possibilita o fornecimento dessas informações de forma rápida, prática e precisa.

Visto a importância de realizar o manejo florestal de modo a garantir a melhor eficiência na utilização destes recursos, o objetivo deste trabalho é o desenvolvimento de um aplicativo para *smartphone* que funcione como uma ferramenta para auxiliar na medição do diâmetro do tronco das árvores. Deste modo, o sistema será inteiramente desenvolvido com a linguagem de programação Python em conjunto com suas bibliotecas, além de possuir integração para o armazenamento dos dados captados em um banco de dados.

2. REFERENCIAL TEÓRICO

2.1. Desmatamento

A madeira é uma matéria-prima muito utilizada desde tempos mais remotos. Seu uso apresenta uma grande variedade como produção de móveis, celulose (para produção de papel), estrutura ou decoração de imóveis, entre outros. Além do uso da madeira em si, há atividades das quais necessitam realizar a extração da vegetação, como por exemplo, a agricultura e a pecuária para ocupação do gado ou do plantio de alimentos, além de outras formas, como o próprio aumento populacional (GEIST; LAMBIN, 2002). Porém, seja para obtenção desse material ou liberação de espaço, é realizado o corte de árvores, provocando um fenômeno chamado de desmatamento.

O desmatamento pode ser definido como a retirada da vegetação de algum local, podendo ser parcial ou total (FAO, 2010), e é considerado a segunda maior causa de mudanças climáticas (FAO, 2018). Essas mudanças ocorrem porque a retirada desestabiliza o ecossistema, uma vez que a vegetação contribui para a absorção de gás carbônico e conversão em oxigênio através da fotossíntese. O desmatamento também contribui para a destruição dos abrigos dos animais, o que pode desequilibrar cadeias alimentares (BONAN, 2008).

Devido à essas graves consequências, foi necessário o desenvolvimento de medidas capazes de reduzir ou até mesmo parar a retirada da vegetação. Foram criadas então, medidas de fiscalização contra métodos de desmatamento (CASTELO, 2015), como por exemplo o “Acordo de Resultados em Defesa da Amazônia” (CNMP, 2020), desenvolvimento de materiais alternativos, como ligas de alumínio, aço carbono, entre outros, medidas de reflorestamento e também o plantio de árvores exclusivas para uso humano, conhecidas como florestas plantadas.

2.2. Florestas Plantadas

Segundo a legislação, florestas plantadas podem ser definidas como “florestas compostas por árvores provenientes de plantio para fins econômicos” (MINISTÉRIO DA AGRICULTURA, 2018). Estas alternativas para o desmatamento têm se mostrado de grande importância, não só para empresas, uma vez que há um destaque cada vez maior para a adoção de medidas sustentáveis de produção, mas também para o meio ambiente como um todo, visto que vegetações nativas deixam de ser exploradas e as árvores plantadas também contribuem para absorção do dióxido de carbono, deposição de matéria orgânica no solo, diminuição do risco de erosões, entre outros aspectos (PAZ et al., 2020).

Dentre as espécies de árvores mais plantadas encontra-se o eucalipto. A escolha dessa espécie se deve ao seu rápido crescimento, boa capacidade de adaptação à diferentes regiões, ciclo

curto, boa capacidade fotossintética, entre outros aspectos (REIS; REIS, 1997). Mas independente da espécie, pelo fato dessas árvores serem plantadas para fins econômicos, é importante que elas sejam monitoradas à fim de se certificar sobre sua integridade (controle de pragas e prevenção de incêndios, por exemplo) e também como forma de acompanhar seu crescimento até atingir dimensões ideais para a extração.

2.3. Métodos De Controle De Uma Floresta Plantada

O controle das florestas de maneira geral é de grande importância, uma vez que nos permite obter informações sobre as condições mais atuais de uma determinada área e, desta forma, obter dados em maior escala. Graças a esse controle foi observado o grande aumento de áreas desmatadas e desenvolveu-se, assim, métodos alternativos para diminuição desse tipo de atividade (BRASIL, 2023).

No caso das florestas plantadas, esse monitoramento não tem como foco o controle de áreas desmatadas, visto que estas já são destinadas ao consumo humano como forma de preservação dos ecossistemas, mas o objetivo está no controle de qualidade da matéria-prima e acompanhamento do crescimento para extração no tempo ideal, aproveitando ao máximo esse recurso (madeira). Dentre os métodos de controle disponíveis atualmente estão o método de medição manual e o de análise por imagens (WU et al., 2019)

Na medição manual, para obtenção do diâmetro da árvore, um parâmetro utilizado é o DBH (Diâmetro à Altura do Peito - *Diameter at Breast Height*), sendo que este ponto é definido na altura do peito a aproximadamente 1.3 metros (PANAGIOTIDIS; ABDOLLAHNEJAD; SLAVÍK, 2021). Porém esse método apresenta algumas desvantagens como demanda elevada de mão de obra e tempo, além disso são frequentes os erros de medição decorrentes da utilização incorreta dos equipamentos, erros de leitura, entre outros empecilhos que podem ser encontrados durante este processo (AVERY; BURKHART, 2015).

Já na análise por imagens o método é realizado através da visão computacional, sendo que esta trata-se de um meio de descrever de maneira computacional as percepções humanas (SZELISKI, 2022). Para obter a equivalência entre a visão computacional e visão humana é necessário realizar uma série de etapas para extrair as informações requeridas. Com isso as imagens são captadas pela forma mais conveniente para a aplicação e então ela é tratada em uma etapa de pré-processamento, e a partir deste ponto são encontradas as informações relevantes para realização da análise e compreensão, possibilitando deste modo a aplicação para diversas finalidades, como na

medicina, na indústria do entretenimento, para emprego em questões ambientais, entre outros (FORSYTH; PONCE, 2003).

Por haver estes problemas referentes ao grande consumo dos recursos florestais e aos erros comuns enfrentados nos processos manuais de medição, e visando atender ao manejo florestal aplicado às florestas plantadas, a tecnologia torna-se um ponto de grande influência no avanço de técnicas de controle, possibilitando a integração de métodos modernos para garantir a melhor eficiência e qualidade na utilização de recursos florestais.

2.4. Inteligência Artificial e a Visão Computacional

Para a compreensão da visão computacional é necessário entender o contexto de utilização dela, desta forma torna-se imprescindível conhecer as ramificações da computação que permitem a implementação de sistemas inteligentes de visão, a começar da compreensão da inteligência artificial e aprendizado de máquinas. A inteligência artificial começou a tomar forma pouco depois da segunda guerra mundial e foram estudadas abordagens distintas acerca deste tema. Para uma programação de computador alcançar um nível de processamento inteligente ele precisa agir humanamente, capacidade analisada pelo teste de Turing, proposto por Alan Turing em 1950 o qual analisa o padrão de respostas do computador por meio de perguntas que são realizadas por humanos até que estas tornem-se indistinguíveis de que estão sendo respondidas realmente pelo computador ou por uma pessoa (RUSSEL; NORVIG, 2020).

Outras abordagens para classificar uma inteligência artificial referem-se a pensar humanamente, analisada utilizando-se de testes cognitivos, devendo a inteligência do computador ter a capacidade de compreender questões de raciocínio lógico, e, por fim, agir racionalmente, sendo esta analisada pela capacidade de adaptação ao ambiente e a tomada de decisão frente a questões de escolhas lógicas a fim de alcançar um objetivo específico. Portanto, a definição da inteligência artificial é baseada na semelhança que os padrões apresentados pela máquina possuem com o comportamento humano (BUITEN, 2019).

A inteligência artificial ganhou destaque e alcançou uma abrangência que possibilita estar presente em muitas aplicações, como no reconhecimento facial, análise de imagens médicas entre outros (SONG et al., 2021). Desta forma, a inteligência artificial é utilizada como uma poderosa ferramenta de análise e extração de padrões, sendo, portanto, empregada na evolução e desenvolvimento de campos de estudos complexos, como neurociência, análise de dados, biomedicina entre outras aplicações (SHOEIBI et al., 2021). Ainda sobre sua utilização na área

médica, diversas técnicas referentes à aprendizagem de máquina (*machine learning*) e aprendizado profundo (*deep learning*) foram empregadas durante o período em que o mundo parou devido à pandemia.

Assim, técnicas de análise de dados em massa em tempo real foram empregadas para realizar um mapeamento de zonas de maiores taxas de contaminação, para verificar andamento de aplicação de vacinas entre outros. Foram empregadas também técnicas de visão computacional para realizar algoritmos preditivos para análise de condicionamento pulmonar de pacientes acometidos pelo vírus, bem como para análise das condições cerebrais após a recuperação para identificação de padrões de sequelas (BULLOCK et al., 2020). Dentre as ferramentas disponíveis para trabalhar com inteligência artificial, o *Machine Learning* possui uma ampla gama de aplicações e diversas formas de aprendizagem, sendo a utilização de capturas de imagens e vídeos em tempo real ou bancos de imagens algumas das formas de treinar o algoritmo para detecção de padrões.

O termo *machine learning* remete ao aprendizado de máquina, sendo este realizado com base na experiência na qual é submetido um algoritmo, desta forma, este tema trata-se de um subgrupo de pesquisa da inteligência artificial que é composta da união de campos como estatística, teoria da informação, ciência cognitiva, entre outros (MITCHELL, 1951), e todo o treinamento destes algoritmos é feito por meio de detecção de padrões para permitir a previsão da tendência baseada na experiência destas máquinas (LECUN; BENGIO; HINTON, 2015).

Utilizando algoritmos de *Machine Learning* é possível condicionar uma máquina a treinamentos que são realizados utilizando conjuntos de dados e a partir disto o algoritmo é capaz de adaptar-se para a tomada de decisão com base nas informações recebidas (BUITEN, 2019). Algumas aplicações bastante difundidas das técnicas de aprendizado de máquina são para aplicação em extração de dados de redes sociais, principalmente para utilização em estratégias de mercado, e outra aplicação a qual tem ganhado espaço é na neurociência na busca de obter modelos preditivos de doenças através de padrões que se repetem em determinadas doenças (SHOEIBI et al., 2021).

A compreensão destes conceitos é fundamental para desenvolver algoritmos de visão computacional, sendo esta última um campo multidisciplinar que envolve um dos sentidos mais complexos do ser humano, a visão, e abrange áreas de neurociência, computação, entre outros. Com isso a visão computacional trata de estudar o mundo da forma como os humanos veem para extrair informações visuais, assemelhando ao nível de compreensão do cérebro humano e em alguns aspectos tendo desempenho melhor (KRISHNA, 2017).

A visão computacional surgiu na década de 1960 e trata-se da detecção e processamento de

imagens captadas por câmeras e processadas por computadores possuindo como objetivo reproduzir o processamento realizado pelo cérebro humano (KHAN et al., 2018). Desta forma a visão computacional pode ser definida como uma ciência que busca extrair informações de imagens para algum objetivo ou também tratada como um meio de transformar a informação visual captada em alguma ação (KRISHNA, 2017).

A visão computacional pode ser desenvolvida em muitas linguagens de programação conhecidas, sendo o Python uma das mais populares escolhas por conta das diversas bibliotecas disponíveis para tratar de todas as etapas de desenvolvimento de scripts de visão computacional. Com isso, com o Python é possível fazer desde a manipulação das imagens, de forma a extrair as informações mais relevantes destas, até a mineração de dados, sendo capaz de tratar de grandes volumes de informações com detalhes precisos, alta velocidade e um grande nível de detalhamento por meio de análises cruzadas (SOLEM, 2012).

Como um dos meios de programação de inteligência artificial mais utilizados atualmente, o Python vem sendo muito difundido para estas aplicações, sendo ela uma linguagem de programação orientada a objetos de alto nível, seu lançamento ocorreu no começo da década de 1990 na Holanda. Foi criada por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI) e está em constante atualização até os dias de hoje, sendo que ela se trata de uma linguagem de fácil compreensão por sua proximidade com as expressões humanas e possui aplicações com diversas finalidades. Assim sendo, o Python consegue se mostrar uma linguagem sofisticada para tratar questões computacionais complexas por meio de uma sintaxe de fácil compreensão e rápido aprendizado (PYTHON, 2023a).

Com base nisto, o desenvolvimento de software nada mais é do que um projeto baseado em uma série de requisitos que ao serem agrupados formam um sistema capaz de executar as ações que foram propostas para ele. Uma vez que o nível de complexidade dos incrementos adicionados aos sistemas são definidos por uma série de requisitos, é importante realizar um método de seleção das principais características que este deve possuir, permitindo com isto a entrega de partes funcionais do mesmo por meio de ferramentas de gerenciamento de projetos, como a matriz GUT. A matriz GUT trata-se de uma ferramenta para priorização de problemas, permitindo a tomada de decisão no meio gerencial (KEPNER; TREGOE, 1981), assim as prioridades são classificadas segundo as categorias de gravidade, urgência e tendência, onde para cada uma destas as notas são dadas em uma escala que vai de 1 a 5 e então multiplicadas entre elas para obtenção da nota GUT, e a partir desta nota a priorização é analisada em ordem decrescente, sendo a de maior valor a principal prioridade levantada (FÁVERI; SILVA, 2016). A nota destinada a cada uma das categorias segue uma classificação que representa as seguintes prioridades da Tabela 1:

Tabela 1 - Tabela de pontuação da matriz GUT.

NOTA	GRAVIDADE	URGÊNCIA	TENDÊNCIA
1	dano mínimo	longuíssimo prazo (dois ou mais meses)	desaparece
2	dano leve	longo prazo (um mês)	reduz-se ligeiramente
3	dano regular	prazo médio (uma quinzena)	permanece
4	grande dano	curto prazo (uma semana)	aumenta
5	dano gravíssimo.	imediatamente (está ocorrendo).	piora muito.

Fonte: Adaptado de Meirelles (2001)

Sabendo da vasta possibilidade de utilização do Python, o desenvolvimento de sistemas inteligentes utilizando seus recursos é uma escolha interessante do ponto de vista de abrangência e poder de processamento computacional. Desta forma, para que seja possível o desenvolvimento de qualquer sistema de visão computacional é necessário haver alguns requisitos principais que permitam qualquer tomada de ação, para isto é preciso uma fonte de requisição de imagens, sendo estas obtidas por meio de um banco de dados ou por câmeras para então serem realizados o processamento e tratamento delas. Considerando a diferença de interpretação de imagens por computadores e pelos seres humanos, são necessárias transformações para que as informações sejam obtidas de maneira adequada, dessa forma diversos filtros podem ser aplicados, sendo realizadas transformações pelas quais são extraídas das imagens informações como contornos, pontos focais, entre outros (WANG et al., 2021).

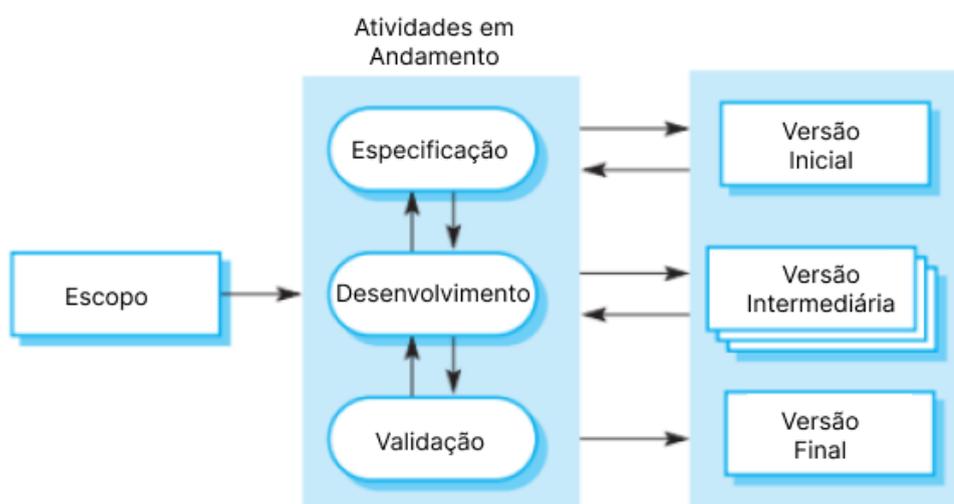
3. MATERIAL E MÉTODOS

O projeto trata-se do desenvolvimento de um aplicativo feito com a linguagem de programação Python que fornece uma solução para auxiliar no manejo florestal medindo o diâmetro do tronco de árvores, sendo utilizado para isto as bibliotecas Kivy e KivyMD que permitem desenvolver um código e rodar em multiplataformas, como sistemas Android e IOS. O desenvolvimento do projeto e a simulação foram realizadas utilizando um computador de mesa com um processador Intel Core i5-4590 com GPU integrada Intel HD Graphics 4600, memória RAM total de 16GB, durante o período de ABRIL de 2023 a JULHO de 2023, nas dependências da Zetta – Agência UFLA de Inovação em Geotecnologias e Sistemas Inteligentes no Agronegócio, Av. Norte UFLA – Aqueça Sol, Lavras-MG, seguindo as seguintes etapas: Montagem do escopo do sistema, montagem do padrão visual, programação do aplicativo e testes funcionais.

O desenvolvimento do *software* deu-se seguindo os conceitos clássicos de engenharia de

software. Desta forma, seguiu-se o fluxo de ciclo de vida com uma estrutura de modelo incremental, sendo este representado na Figura 1. O modelo incremental permite a entrega de pequenas partes do software de maneira que estas sejam viáveis para utilização, apesar de não estarem em sua versão final, garantindo a evolução deste com a implementação de melhorias constantes (SOMMERVILLE, 2020).

Figura 1 - Modelo Incremental.



Fonte: Adaptado de Sommerville (2020)

Para o desenvolvimento do software foi necessário levar em consideração diversas características e requerimentos, tanto do ponto de vista computacional quanto do usuário, para isso foram levantados os requisitos funcionais e os requisitos não funcionais. Os requisitos funcionais podem ser definidos por “como” o software deve funcionar, assim nestas definições são estabelecidas as funcionalidades que o software terá (SOMMERVILLE, 2020). Já os requisitos não funcionais estão relacionados às propriedades do sistema, portanto são estabelecidas as características de qualidade, segurança de acesso, desempenho do sistema, entre outras (PRESSMAN; MAXIM, 2021). Os requisitos funcionais levantados estão representados na figura 2.

Figura 2 - Requisitos Funcionais.

GERAL	
<ul style="list-style-type: none"> [geral] O sistema deve ter 4 telas a princípio; 	
<p>TELA01</p> <ul style="list-style-type: none"> [tela01] Ao abrir a tela deve aparecer uma caixa de mensagem com informações sobre como realizar a medição; [tela01] O frame de captura de imagem deve conter linhas de divisão de tela para centralização da posição da imagem; [tela01] O frame capturado deve ser tratado para obtenção dos contornos da árvore; [tela01] A partir dos contornos obtidos deve ser realizado a medida dos pixels na posição vertical; [tela01] Os pixels contabilizados devem ser transformados para unidade de medida "mm" (cálculo); [tela01] A medida do diâmetro que foi obtido deve ser exibida na posição da imagem de onde foi feita a medição; [tela01] após a captura e os cálculos da medida terão mais dois botões: salvar e refazer; [tela01] Se selecionar salvar: salva os dados coletados da imagem (data/hora/número da árvore /diâmetro da árvore); [tela01] Os dados serão salvos em um banco de dados online. [tela01] O número da árvore deve ser obtido como uma entrada de informação do usuário; [tela01] Se selecionar refazer: a imagem capturada desaparece e a câmera é novamente iniciada; [tela01] A tela01 terá um botão de voltar que leva até o menu 	<p>MENU</p> <ul style="list-style-type: none"> O menu terá dois botões (TELA01 e TELA02)
<p>TELA02</p> <ul style="list-style-type: none"> [tela02] Tela de consulta de informações com exibição de tela de login; [tela02] Para acessar dados antigos é necessário ter credenciais cadastradas; [tela02] Os dados inseridos são comparados com as credenciais salvas; [tela02] Esta tela terá um botão de voltar que retorna a tela menu; [tela02] Se NOK: mensagem de erro e inserir novamente os dados; [tela02] Se OK: Login é realizado e então uma nova tela aparece; 	<p>TELA03</p> <ul style="list-style-type: none"> [tela03] A tela de exibição de dados é exibida com informações referentes às árvores já cadastradas; [tela03] É possível somente visualizar os dados da tabela; [tela03] É possível realizar filtro por data e por nome (número da árvore); [tela03] Os dados da tabela são possíveis de serem salvos no formato PDF e planilha.

Fonte: Do Autor (2023)

Seguindo o levantamento dos requisitos funcionais, na figura 3 estão os requisitos não funcionais.

Figura 3 - Requisitos não Funcionais.

GERAL	
<ul style="list-style-type: none"> [geral] O sistema terá uma tela inicial de carregamento com um máximo de 8 segundos de espera; [geral] O menu deve ter botões que se destacam em relação ao fundo para melhor visualização; 	
<p>TELA01</p> <ul style="list-style-type: none"> [tela01] A tela 01 referente ao acesso à câmera deverá ser carregada em menos de 2 segundos; [tela01] As linhas de divisão de enquadramento da imagem deverá ter uma cor que se destaca na imagem para garantir a visualização; [tela01] Ao clicar no botão de scan a imagem com as informações deve ser exibida em um tempo máximo de 5 segundos; [tela01] Ao clicar no botão de refazer a câmera deve voltar à sua aparência inicial, permitindo refazer o processo de captura de imagem; [tela01] Ao clicar no botão de salvar terá uma mensagem de sucesso e a tela retornará ao seu visual inicial; [tela01] Ao clicar no botão voltar a tela de menu inicial deverá ser carregada em menos de 2 segundos; 	
<p>TELA02</p> <ul style="list-style-type: none"> [tela02] Ao clicar no botão consultar a tela de login será exibida em menos de 2 segundos; [tela02] Ao preencher os dados de login, o campo de senha deve estar formatado no padrão "password" escondendo a visualização da senha; [tela02] Ao clicar no botão login o encaminhamento para a próxima tela deve ser realizado em menos de 2 segundos; 	<p>TELA03</p> <ul style="list-style-type: none"> [tela03] A tela 03 deverá exibir os dados em ordem cronológica descendente; [tela03] Ao clicar nos filtros a organização dos dados devem ser finalizadas em 20 menos de 2 segundos; [tela03] Ao clicar no botão de retornar à tela inicial o carregamento deve ser feito em menos de 2 segundos;

Fonte: Do Autor (2023)

Com base no modelo incremental, para permitir a entrega de incrementos, para o primeiro protótipo funcional da aplicação desenvolvida foi feito uma matriz GUT com base nos requisitos funcionais para compor as características iniciais do sistema, sendo ela utilizada para a definição dos requisitos funcionais que foram necessários para a realização do primeiro protótipo. Os requisitos não funcionais que foram levantados serão utilizados de parâmetros iniciais, sendo que após os testes preliminares as características levantadas serão comparadas com os dados obtidos a partir de testes de segurança, performance entre outros. Com isso os requisitos não funcionais serão ajustados para garantir um funcionamento realista, principalmente em questões relacionadas ao tempo de carregamento de mudanças de tela e de requisição à serviços externos. Para o desenvolvimento do sistema, os 3 parâmetros utilizados para avaliação de prioridade serão tratados com a gravidade medindo o grau de importância da implementação do requisito funcional, a urgência levando em consideração a importância de implementação do requisito nesta etapa e a tendência o quanto o requisito tende a diminuir a experiência do usuário, com isso temos na Tabela 2 as notas.

Tabela 2 - Matriz GUT para o sistema.

<i>ITEM</i>	<i>DESCRIÇÃO</i>	<i>G</i>	<i>U</i>	<i>T</i>	<i>GXUXT</i>
01	O sistema deve ter 4 telas a princípio	5	5	5	125
02	[menu] O menu terá dois botões [tela01, tela02]	5	5	5	125
03	[tela01] Ao abrir a tela deve aparecer uma caixa de mensagem com informações sobre como realizar a medição	2	2	3	12
04	[tela01] O frame de captura de imagem deve conter linhas de divisão de tela para centralização da posição da imagem	3	5	5	75
05	[tela01] O frame capturado deve ser tratado para obtenção dos contornos da árvore	5	5	5	125
06	[tela01] A partir dos contornos obtidos deve ser realizado a medida dos pixels na posição vertical	5	5	5	125
07	[tela01] Os pixels contabilizados devem ser transformados para unidade de medida “mm” (cálculo)	5	5	5	125
08	[tela01] A medida do diâmetro que foi obtido deve ser exibida na posição da imagem de onde foi feita a medição	4	3	5	60
09	[tela01] após a captura e os cálculos da medida terão mais dois botões: salvar e refazer	4	5	5	100
10	[tela01] Se salvar: salva os dados coletados da imagem (data/hora/número da árvore /diâmetro da árvore)	5	3	5	75
11	[tela01] Os dados serão salvos em um banco de dados online	4	3	5	60
12	[tela01] O número da árvore deve ser obtido como uma entrada de informação do usuário	2	3	2	12
13	[tela01] Se refazer: a imagem capturada desaparece e a	4	5	5	100

	câmera é novamente iniciada				
14	[tela01] A tela01 terá um botão de voltar que leva até o menu	5	5	5	125
15	[tela02] Tela de consulta de informações com exibição de tela de login	5	5	5	125
16	[tela02] Para acessar dados antigos é necessário ter credenciais cadastradas	3	4	3	36
17	[tela02] Os dados inseridos são comparados com as credenciais salvas	3	4	3	36
18	[tela02] Esta tela terá um botão de voltar que retorna a tela menu	5	5	5	125
19	[tela02] Se NOK: mensagem de erro e inserir novamente os dados	3	3	2	18
20	[tela02] Se OK: Login é realizado e então uma nova tela aparece	5	5	5	125
21	[tela03] A tela de exibição de dados é exibida com informações referentes às árvores já cadastradas	5	3	2	30
22	[tela03] É possível somente visualizar os dados da tabela	5	3	2	30
23	[tela03] É possível realizar filtro por data e por nome (número da árvore)	1	1	1	1
24	[tela03] Os dados da tabela são possíveis de serem salvos no formato PDF e planilha.	1	1	1	1

Fonte: Do Autor (2023)

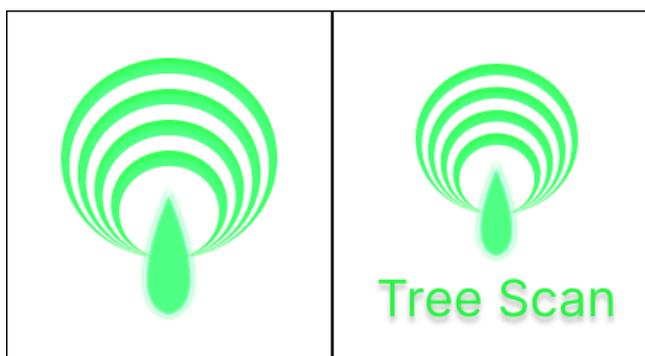
Com base nas notas obtidas tabela 2, os requisitos funcionais que obtiveram nota GxUxT acima de 50 foram implementadas no protótipo 01, sendo limitado neste valor por escolha própria do autor. Uma vez definido os requisitos necessários para o funcionamento do software, foi realizada a modelagem de processos, sendo esta necessária para que a compreensão da interação usuário e software, bem como os eventos provenientes destas ações sejam mais bem compreendidos. A modelagem e mapeamento de processos foi realizada utilizando a notação BPMN (Notação e Modelagem de Processos de Negócio - *Business Process Model and Notion*), esta notação é comumente utilizada para modelagem de negócios e trata-se de uma forma de representação gráfica para apresentação dos aspectos e informações de negócio (SILVER, 2011; OMG, 2011). A notação BPMN que representa o software desenvolvido está no Apêndice A. Nesta modelagem de negócio referente ao funcionamento do software o modelo proposto apresenta uma visão macro com a interação entre usuário, sistema e requisição externa.

O BPMN apresentado no Apêndice A mostra a ligação entre as ações operantes do sistema de modo que se tem ao meio do quadro a representação do usuário, podendo este, assim que o menu inicial for carregado, escolher entre duas opções: *scan* e *consult*. O botão *scan* leva até uma tela que permite obter uma captura de imagem pelo smartphone e então escolher entre salvar a foto e refazer o processo. Se a escolha selecionada for salvar a foto, uma requisição é feita para que as informações

da imagem capturada sejam salvas no banco de dados, por outro lado, se a opção escolhida for a de refazer, então a câmera é liberada para obter uma nova captura ou retornar ao menu inicial. Caso, no menu inicial, o botão *consult* seja pressionado, então uma tela de login é aberta para que o usuário possa inserir as credenciais, sendo estas verificadas em uma requisição externa. Deste modo, existem duas possibilidades resultantes da ação de login: credencial encontrada ou credencial não encontrada, sendo que a segunda opção não libera o login e é necessário reinserir as credenciais, e a primeira opção leva a uma nova tela que permite a exibição dos dados históricos que foram armazenados.

Após estabelecido o funcionamento global do sistema, foi realizada a definição do *front-end* do aplicativo. O padrão de cores foi definido pensando na funcionalidade final do aplicativo, a obtenção de características do diâmetro do tronco de árvores, assim a paleta de cores foi composta por tons de preto, cinza e verde, tendo os seguintes códigos no padrão hexadecimal: #1B1A1A, #2A2A2B e # 27F442. A definição da logo do aplicativo foi feita com a união de conceitos envolvidos com o aplicativo, desta forma ela remete a uma ilustração de uma árvore com as folhas desenhadas de modo a ter um modelo gráfico que representa as ondas de propagação, como exemplo as ondas sonoras ou as perturbações que ocorrem quando um objeto cai na água, assim a logo foi vetorizada e a cor que a representa é o verde, dando destaque para ela nas telas de fundo que são de cor escura. Com os componentes definidos, o resultado final é apresentado na Figura 4.

Figura 4 - Logo do Aplicativo.



Fonte: Do Autor (2023)

O desenvolvimento do sistema deu-se em 3 partes principais, sendo elas a construção do *front-end*, no qual foi projetada a interface de acesso do usuário, a outra segmentação é a do *back-end* com ela dividida em duas partes, uma delas para o tratamento das imagens e outra para a integração lógica do sistema.

A estruturação do *front-end* do aplicativo com base nos requisitos funcionais selecionados

foi realizado como um modelo de teste, sendo este modelo projetado inicialmente para realização de experimentos para ajustes de funcionalidade, sendo esta utilizada como um guia para a montagem do aplicativo. A estrutura *front-end* do aplicativo, disponível no Apêndice B, representa as ligações entre as telas, sendo uma tela inicial para realização do login, havendo um botão que faz a passagem para a tela de inserção de credenciais, e a partir desta, caso as credenciais estejam corretas, tem-se a tela de menu, nesta tela são disponibilizados dois botões. Um dos botões é o “*scan*”, que leva até a tela que abre a câmera e a partir desta, ao pressionar o botão de captura, a mensagem informativa é apresentada para a escolha de “salvar” e “refazer”. Retornando à tela de menu, o outro botão é o de “*consult*”, este botão permite o acesso às informações salvas das leituras de diâmetro das árvores que foram salvas no banco de dados.

Para este protótipo, o modelo para verificação do diâmetro das árvores deu-se por técnicas de manipulação de imagens realizadas por ferramentas das bibliotecas disponíveis em Python, o OpenCV e Numpy. Para realizar o processamento da imagem, o dado de entrada vem por meio da captura realizada pela câmera, então esta imagem é convertida para uma escala de cinza, sendo esta transformação realizada utilizando uma função da biblioteca OpenCV.

Uma vez que o padrão de cores presentes nas imagens captadas pela câmera não foram relevantes para a realização da análise, a conversão para a escala de cinza foi realizada para utilização das funções de obtenção dos contornos da imagem. A conversão da imagem do padrão BGR (*Blue, Green e Red* – Azul, Verde e Vermelho, respectivamente) para a escala de cinza acontece por meio de manipulação de pixels, sendo esta manipulação realizada com uma operação matemática de média a qual é feita levando em consideração cada componente que compõe a imagem (BURGER; BURGE, 2016).

A fim de obter características mais bem definidas para a detecção dos contornos que fazem o limite do tronco da árvore para que seja possível realizar a medição do diâmetro da mesma, a imagem passa por um processo de limiarização, este processo trata-se da segmentação da imagem por meio da divisão dos pixels em preto e branco. Para realizar esta segmentação a imagem convertida para a escala de cinza tem seus pixels divididos entre as cores branco e preto por meio de um valor de limiar, ou seja, pixels com seu valor abaixo do limite definido são atribuídos a uma categoria e os que estão acima deste valor limiar à categoria oposta, criando com isso um padrão de imagem binária (GONZALEZ; WOODS; EDDINS, 2020).

Para a extração dos contornos da imagem foi utilizada uma função do OpenCV que permite realizar este tratamento, na qual foi utilizada como parâmetro de limiar o valor 70, este valor divide a classificação dos pixels em preto e branco a depender do valor de sua intensidade ser abaixo ou acima dele, além disso o valor 255 indica o máximo de intensidade que o pixel deve ter após o

tratamento, desta forma definindo este pixel máximo como a cor branca, e por fim o tipo de análise é definido pela combinação da limiarização invertida (pixels brancos estão abaixo do limiar e os pixels pretos acima do limiar) com o parâmetro otsu, sendo que este calcula o melhor valor de limiar necessário para a análise da imagem com base no padrão exposto na imagem convertida para a escala de cinza. Este tipo de definição do valor limiar foi selecionado para melhor adaptação do algoritmo de captação e interpretação da imagem pois espera-se que a utilização do aplicativo aconteça em ambientes abertos, portanto não existe um controle de luminosidade e este parâmetro torna-se ideal para esta condição uma vez que o ajuste de parâmetro é calculado com base nos componentes da imagem obtida.

Após feita a limiarização, foram realizadas as manipulações para encontrar os contornos existentes na imagem capturada, sendo este método utilizado para conseguir identificar os contornos que delimitam o tronco da árvore. Para possibilitar encontrar os contornos com melhor precisão foi aplicado um filtro gaussiano na imagem, sendo que este filtro utiliza manipulações matemáticas do pixel em questão com os pixels vizinhos ao analisado para que seja possível suavizar a transição entre eles, reduzindo os ruídos presentes na imagem (GONZALEZ; WOODS; EDDINS, 2020).

Com este filtro é realizada a redução de ruídos por meio da suavização da imagem, sendo delimitada a região na qual o filtro gaussiano utiliza para realizar as manipulações matemáticas com os pixels vizinhos e a comparação pelo desvio padrão deste dados coletados no filtro, sendo este calculado com base na janela de aplicação (OPENCV, 2023d).

Com a aplicação do filtro gaussiano a detecção das bordas é realizada por meio de um algoritmo de detecção de bordas Canny implementado dentro das funcionalidades da biblioteca OpenCV. Este algoritmo faz a detecção das bordas por meio do mapeamento das variações de intensidade dos pixels na imagem, sendo estas mudanças abruptas observadas de maneira geral em regiões com a existência de bordas, a identificação das bordas por este algoritmo é feita por meio de uma série de passos que auxiliam a validação da intensidade de pixels, sendo que estas etapas consistem da suavização da imagem, cálculo dos gradientes, supressão não máxima, limiarização e por fim a conexão de bordas (YOUSAF et. al, 2018).

O algoritmo de Canny foi implementado com os parâmetros blur, que representa a variável na qual foi armazenada o filtro gaussiano aplicado na imagem, além de definidos as delimitações para a classificação dos pixels, sendo que os valores abaixo do limite mínimo são definidos como não sendo bordas e os pixels acima do limite máximo são classificados como bordas, por outro lado os valores que estão entre estas limitações são pixels que possuem um perfil intermediário para classificação sendo estes definidos de acordo com fatores de supressão de pixels e conexão com as borda e por fim a aplicação do filtro sobel, sendo ele utilizado para uma das etapas necessárias para

o algoritmo, o cálculo de gradientes, no qual este é realizado por manipulações matemáticas de derivação parcial para obtenção dos gradientes pontuais da imagem (GONZALEZ; WOODS, 2008; CANNY, 1986; OPENCV, 2023b).

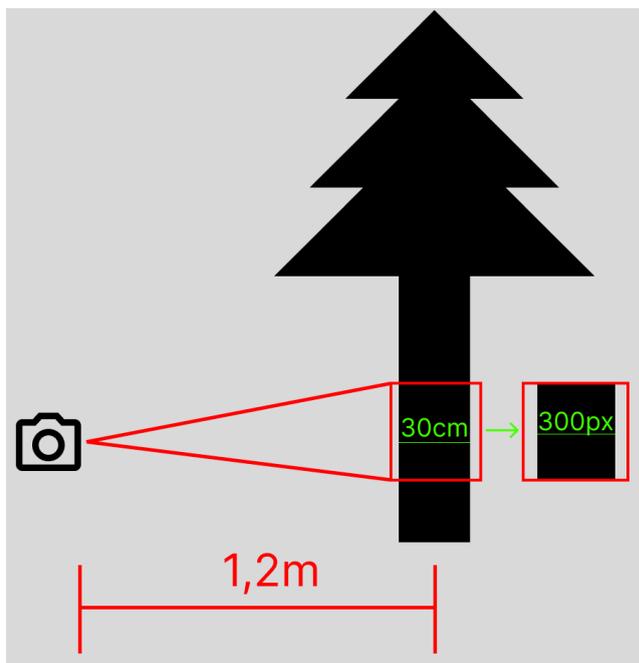
Após a detecção das bordas algumas falhas de pixels podem ter sido deixadas entre os contornos identificados, uma maneira de melhorar a conexão destes é fazer um processo de dilatação na imagem limiarizada resultante do processo anterior, desta forma qualquer lacuna existente nestas regiões de interesse é preenchida por meio da expansão de seus pixels (GONZALEZ; WOODS, 2008). Os parâmetros utilizados são a imagem de entrada, armazenada na variável canny a qual possui as informações dos pixels detectados nas bordas da imagem no padrão binário, a região que delimita a zona de expansão dos pixels, e por fim o número de iterações de dilatação, neste caso definida como 0 indicando que a dilatação será aplicada uma única vez na imagem (OPENCV, 2023e).

Os processos de tratamento de imagem realizados auxiliam para a extração dos contornos, desta forma com as bordas identificadas e tratadas para possuir uma melhor visualização é realizada uma detecção dos contornos, sendo esta operação realizada para possibilitar a segmentação e características de forma das imagens analisadas, desta forma esta operação é realizada conforme a

Para a obtenção dos contornos a função da biblioteca OpenCV necessita de uma imagem de entrada, a qual foi utilizada uma cópia da dilatação feita no processo anterior, o outro parâmetro trata-se do modo pelo qual os contornos foram recuperados da imagem, sendo para este caso utilizado o modo de extração dos contornos mais externos identificados, e por fim o método de aproximação utilizado, definindo o armazenamento de todos os pontos do contorno, fornecendo com isso o mapeamento de todo o perímetro do objeto analisado (OPENCV, 2023a; OPENCV, 2023c).

Os tratamentos realizados foram necessários para a extração dos contornos encontrados na imagem para que, a partir deles, seja obtido o diâmetro do tronco da árvore. Sendo a imagem captada composta por pixels e os contornos dos objetos detectados armazenados, o modo de cálculo do diâmetro dos troncos foi definido baseado em imagens capturadas a uma distância da câmera até a árvore de 1.2 metros e foi realizada a calibração da correspondência de cada pixel em unidade de medida de espaço, para este trabalho dada em “cm” (centímetro). Desta forma, a calibração da medida foi feita como no exemplo da Figura 5.

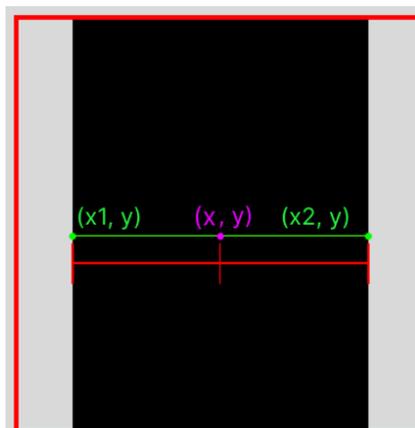
Figura 5 - Calibração da Medida.



Fonte: Do Autor (2023)

Conforme este exemplo, a calibração foi feita medindo um objeto base com medidas controladas colocado a uma distância de 1.2 metros do ponto de captura, sendo esta distância necessária também para realizar as medições com o aplicativo, e a partir desta imagem foi contabilizada o número de pixels que corresponde a medida do objeto real, através desta relação foi possível identificar o resultado de saída do aplicativo com as medidas reais do objeto. Após a calibração, o modo selecionado para identificação da área de leitura do diâmetro da árvore foi feito detectando o primeiro contorno a esquerda e a direita de um ponto fixado no centro da tela e então calculado a distância entre estes dois pontos, assim esta distância, que é retornada em pixels, é convertida para a unidade de medida “cm” (centímetro) baseado na calibração feita anteriormente. O posicionamento dos pontos utilizados para o cálculo do diâmetro está representado na Figura 6, assim, utilizando as coordenadas cartesianas de cada um dos dois pontos (contorno esquerdo e contorno direito), localizados a uma mesma altura y que corresponde ao meio do frame detectado, a distância de x_1 até x_2 corresponde ao diâmetro da árvore.

Figura 6 - Coordenadas dos contornos.



Fonte: Do Autor (2023)

Tendo realizadas as definições referentes ao funcionamento do sistema, à definição base do *front-end* e o desenvolvimento da lógica de tratamento e análise da imagem foi realizada a construção do aplicativo. Para a construção deste foi utilizado o framework de desenvolvimento Kivy, o qual trata-se de uma biblioteca *open-source* que permite criar interfaces multiplataformas (KIVY, 2023). Além do *framework* Kivy foi utilizado também a biblioteca de pacotes de widgets KivyMD, que possui recursos gráficos e com visuais comuns para o desenvolvimento de *software* e implementação simplificada de recursos (KIVYMD, 2023). Por meio destes *frameworks* foi possível criar todos os recursos da interface do usuário, como botões, campos de texto, tabelas de visualização entre outros.

Outro ponto importante para o desenvolvimento da *back-end* do aplicativo foi a integração dele com um banco de dados, o qual foi implementado a conexão com o banco de dados em tempo real do Google, o Firebase. Esta trata-se de uma plataforma de desenvolvimento que possui diversas ferramentas que auxiliam a criação e integração de sistemas, possuindo pacotes básicos de forma gratuita, além de permitir o gerenciamento destes sistemas sem a necessidade de servidores. O banco de dados do Firebase é um NoSQL (*Not Only Structured Query Language* – Não Apenas Linguagem de Consulta Estruturada) (FIREBASE, 2023). Os bancos de dados do tipo NoSQL são um tipo de armazenamento não estruturado, desta forma os dados são armazenados de forma dinâmica e flexível, permitindo um desempenho elevado no tratamento de dados em grande escala e sem a necessidade de uma estrutura inicial de armazenamento pré-definida (SADALAGE; FOWLER, 2012). Portanto, utilizando os recursos disponibilizados por esta plataforma foi criada a estrutura de conexão e armazenamento de dados, para isso foram utilizados conceitos de gerenciamento de banco de dados conhecidos pelo acrônimo CRUD (*Create, Read, Update e Delete*, que são respectivamente Criar, Ler, Atualizar e Deletar), sendo estas as principais funções realizadas na

operação de banco de dados (GARCIA-MOLINA; ULLMAN; WIDOM, 2008).

Portanto os principais recursos utilizados para esta etapa do desenvolvimento foram o de leitura e escrita, sendo estes utilizados para realização da verificação de credenciais, salvamento de dados de leitura das árvores e leitura dos dados temporais salvos. Para a confirmação das credenciais foram realizadas conexões com o banco de dados do google utilizando a biblioteca requests, a qual permite fazer a conexão de requisição com recursos externos, ou seja, através desta biblioteca são realizados os pedidos necessários e estes são ligados às respostas esperadas para eles.

Os dados armazenados nesta plataforma do Google são estruturados como objetos JSON (*JavaScript Object Notation* - Notação de objeto em JavaScript), sendo este uma forma de armazenamento de dados que acontece por meio de uma chave e seu respectivo valor, podendo este valor ser armazenado em diversos formatos, como texto, número inteiro, número decimal entre outros. O arquivo JSON não sofre com incompatibilidade de sistema, uma vez que ele independe de qualquer plataforma, ele pode ser lido e interpretado de forma rápida e eficaz, tornando-se um formato comum de armazenamento de dados e troca de informações entre sistemas (JSON, 2023). Para utilização deste formato de dados dentro do Python é fornecida uma biblioteca chamada json, a qual permite fazer a codificação e a decodificação da estrutura de dados de leitura ou escrita (PYTHON, 2023b).

Para o acesso ao aplicativo foram cadastrados previamente dois usuários padrão para acesso e suas respectivas senhas, sendo estes os seguintes:

1. Usuário_01: admin; Senha: admin123;
2. Usuário_02: scantree; Senha: scantree123.

O acesso aos recursos do aplicativo é liberado somente para estes dois usuários, desta forma qualquer outra credencial diferente desta não libera a passagem da tela inicial para as telas seguintes, portanto somente estes cadastros foram liberados para fazer o salvamento e leitura de dados.

Dentro da guia “scan”, os métodos de gerenciamento de banco de dados também são empregados para realizar o armazenamento dos dados da análise das características das árvores, assim é realizada a conexão com o Firebase e dentro dele é feita a escrita dos dados neste banco, sendo esta conexão feita por meio da biblioteca requests e a codificação dos dados realizada por meio da biblioteca json, a qual codifica as informações para que seja enviada ao banco de dados no formato requerido. Por fim, outra requisição, do tipo leitura, é realizada dentro da guia “consult” para que os dados temporais das leituras sejam acessados e visualizados em formato de tabela nesta tela.

Assim, para realizar a validação do desempenho foram analisados os dados coletados de duas formas, uma primeira coleta realizada por meio da medição direta, utilizando um barbante passando ao redor da circunferência do tronco das árvores na altura do peito e então obtidas as medidas de diâmetro sendo igual a circunferência dividida pelo valor de pi, com isso foram coletados dados de 10 árvores. Uma segunda análise foi feita de maneira indireta por meio do sistema desenvolvido, sendo para este caso coletados 15 amostras de medida para cada uma das 10 árvores, as quais são as mesmas coletadas pelo modo direto, com isso os dados foram validados por meio de estatística descritiva contendo informações de média, desvio padrão, mínimo e máximo e os quartis. Além destas análises, por meio das 15 coletas de medida do diâmetro de cada uma das 10 árvores será possível verificar possíveis erros referentes à imprecisão da distância da câmera até a árvore e erros provenientes da luminosidade a qual a câmera estará exposta, causando variações nas medidas obtidas.

4. RESULTADOS E DISCUSSÃO

A segurança do aplicativo, baseada na possibilidade de acesso aos dados definida por meio das credenciais demonstrou ser efetiva, liberando o acesso às funcionalidades do sistema somente aos usuários que foram previamente cadastrados, sendo que este meio de cadastro nesta etapa do desenvolvimento do sistema foi realizada de forma manual inseridas diretamente dentro do ambiente do Firebase, portanto, existindo a segurança para acesso ao aplicativo e também para o acesso ao banco de dados dentro do ambiente do Google assegurado pelos fatores de segurança dele. Desta forma, a segurança do aplicativo se dá pela validação de correspondência exata das credenciais, sendo que ao inserir as informações incorretas, a mensagem de erro é exibida, sendo que estas credenciais são invalidadas em caso de o nome de usuário inserido, a senha ou ambos estiverem divergentes dos dados cadastrados. Outro fator de verificação para a questão de segurança de acesso é referente aos caracteres, sendo feita a invalidação dos caracteres inseridos caso haja diferença entre as letras maiúsculas e minúsculas, garantindo desta forma que a senha necessária para acessar todo o conteúdo do sistema deve ser 100% compatível, além de possuir todos os níveis de segurança oferecidos pelos serviços do Google.

Além da segurança da infraestrutura do aplicativo e dos serviços do Google, na tela de login foi inserido um botão a direita do campo *password*, o qual permite exibir ou ocultar a exibição da senha enquanto esta é digitada, fornecendo com isso mais um fator de segurança relacionado à visualização da senha de acesso.

Durante os testes do aplicativo foram coletados dados de tempo de resposta, sendo estes

dados de desempenho podendo ser observados nas análises abaixo, nas quais foram coletados dados de execução de ações do programa, para isto foi utilizada a biblioteca do Python chamada *time*, sendo que esta função foi inserida nos pontos de interesse dentro do *script*, para que, desta forma, o tempo entre a chamada da função e sua finalização fosse calculada. Os dados de desempenho que foram obtidos estão relacionados ao desempenho de inicialização do sistema, chamada para funções específicas e requisição a serviços externos, neste caso acesso ao banco de dados para leitura, gravação e autenticação de usuário. Ao todo foram coletadas 20 tomadas de tempo para cada análise de desempenho, sendo este número de análises definida pelo autor.

Todos os testes foram realizados em duas máquinas diferentes para simulação do funcionamento do aplicativo, estas com características distintas uma da outra, sendo a primeira máquina (rotulada de MAQ_01) referente ao computador utilizado para o desenvolvimento do sistema e já identificado no tópico de materiais e métodos, e a segunda máquina (rotulada de MAQ_02), um computador com sistema operacional Windows 11, processador Intel Celeron 5100, 24GB de memória ram e sem placa de vídeo.

A partir dos dados coletados foi realizada uma análise estatística referente aos seus valores, sendo estes dispostos no resumo estatístico da Figura 7:

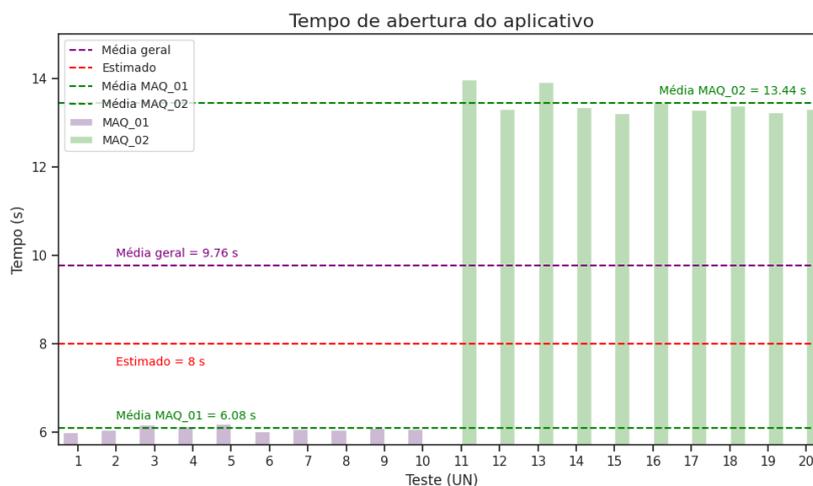
Figura 7 - Resumo estatístico - Inicialização do Aplicativo.

Resumo Estatístico							
count	mean	std	min	25%	50%	75%	max
20.0	9.761	3.783	5.981	6.062	9.7	13.314	13.977

Fonte: Do autor (2023)

A partir do resumo estatístico e por meio do gráfico da figura 8, pode-se verificar que a média de tempo, em segundos, para a inicialização do aplicativo ficou com um valor de 9.761 segundos com um desvio padrão de 3.783, sendo que este valor mostrou uma variação consideravelmente alta em relação ao tempo de inicialização e seu resultado final foi cerca de 22% maior do que o valor estipulado nos requisitos não funcionais.

Figura 8 - Gráfico de Barras - Tempo de Inicialização.



A partir do gráfico da figura 8, nota-se uma grande diferença no tempo de inicialização do sistema na simulação feita na máquina 1 em comparação com a máquina 2, sendo que todas as tomadas de tempo realizadas na MAQ_01 estiveram abaixo do tempo estipulado nos requisitos não funcionais, 8 segundos, sendo que para este caso a média de tempos de inicialização ficou em 6.08 segundos, apresetando um resultado menor do que os observados para a máquina 2, a qual teve todas as suas tomadas de tempo acima do valor estipulado, fornecendo uma média de 13.44 segundos. A razão pela qual os valores observados divergiram tanto entre si está relacionada às características de cada máquina utilizada na simulação, sendo que aquela com maior capacidade de processamento demonstra valores menores. Por conta desta diferença de tempo entre a MAQ_01 e MAQ_02 serem grandes, e além disso, os dados para o segundo caso estão muito afastados do valor estimado, foi observado o aumento do valor da média geral.

Após a análise do tempo de inicialização, outro ponto do sistema que foi verificado o desempenho de carregamento foi o sistema de login, sendo que este depende de recursos externos do aplicativo para acesso ao banco de dados. O tempo de processamento de validação das credenciais foi obtido contabilizando o tempo de realizar a requisição ao Firebase e retornar uma resposta. Baseado nos dados coletados verifica-se que o tempo de resposta corresponde a uma parcela muito pequena se comparada aos tempos obtidos para inicialização, sendo esta informação verificada na figura 9.

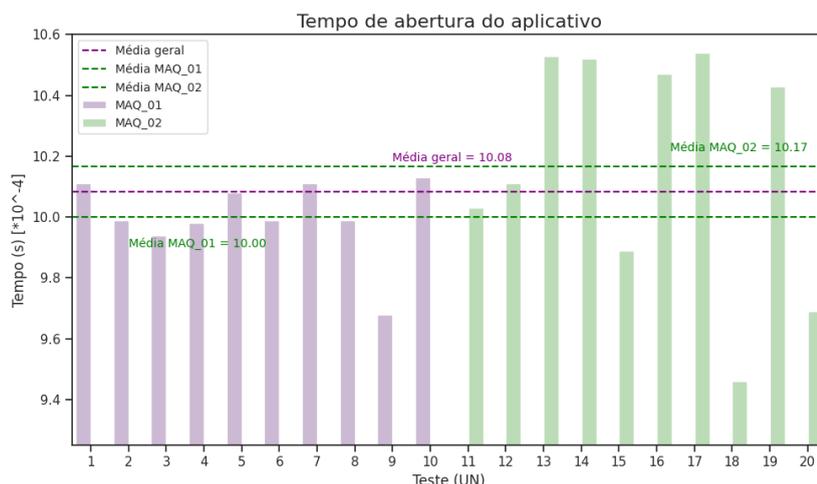
Figura 9 - Resumo estatístico - Validação de Credenciais.

count	mean	std	min	25%	50%	75%	max
20.0	0.001	0.0	0.0009	0.001	0.001	0.001	0.0011

Fonte: Do Autor (2023)

Os dados da figura 9 mostram uma média, arredondada em 3 casas decimais, de 1 milésimo de segundo com valores de mínimo e máximo iguais a 0.0009 segundos e 0.0011 segundos, respectivamente. Na Figura 10 é possível verificar o comportamento destes dados para cada teste realizado.

Figura 10 - Gráfico de Barras - Tempo de Validação de Credenciais.



Fonte: Do Autor (2023)

Conforme observado na figura 10, os tempos de resposta para a requisição ao Firebase ficaram muito próximos entre a máquina 1 e a máquina 2, porém, ao contrário do que ocorreu no tempo de inicialização do aplicativo, neste caso o fator relevante para estes resultados trata-se da conexão com a internet, sendo os valores de tempo obtidos de aproximadamente 1 milésimo de segundo. Analisando o gráfico apresentado, nota-se que o tempo médio de conexão para os testes na máquina 1 foi menor do que para a máquina 2, apesar disto a máquina 2 apresentou os resultados de maior oscilação, contendo o menor valor entre os analisados e também os maiores. Estes dados mostram uma oscilação relacionada à conexão de internet da máquina 2 maior do que na 1, porém estes resultados para o desempenho do correto funcionamento do aplicativo são de baixa relevância para o desempenho geral do sistema uma vez que seus valores são muito abaixo dos tempos definidos nos requisitos não funcionais.

As análises de desempenho anteriores são relacionadas à etapa de início e acesso ao sistema, a partir deste ponto foram analisadas as velocidades de conexão dos dados captados pela câmera no banco de dados, assim serão validados o desempenho de leitura e gravação de informação. Iniciando pela tela *scan*, na qual as imagens são captadas e os dados enviados para armazenamento de dados, desta forma as aquisições do tempo de gravação foram realizadas a partir do momento em que o sistema inicia a tentativa de conexão com o banco de dados até obter uma resposta da requisição feita, assim, os dados coletados estão identificados no resumo estatístico da figura 11.

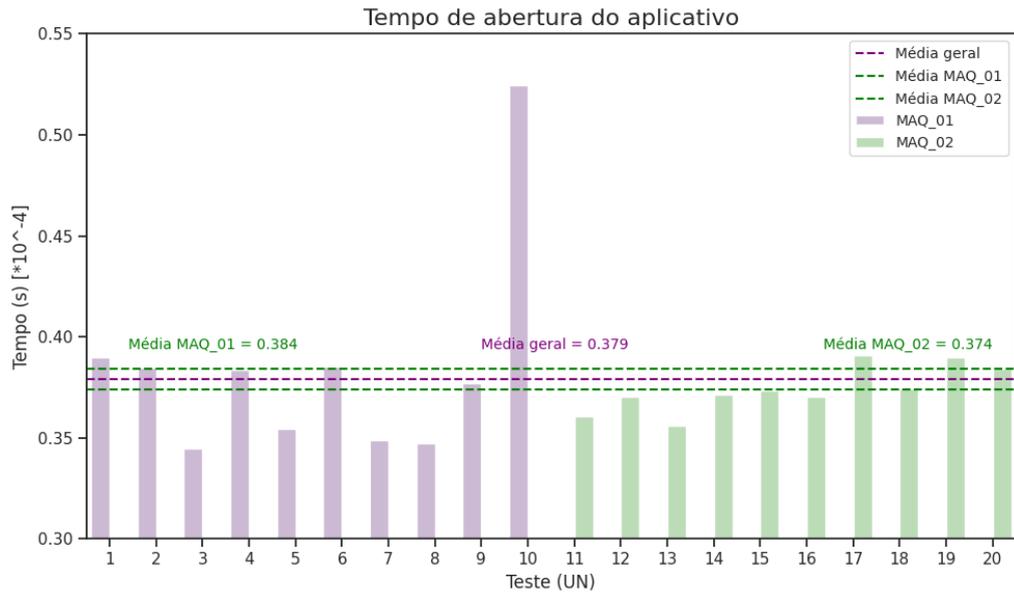
Figura 11 - Resumo Estatístico - Gravação de Dados.

Resumo Estatístico							
count	mean	std	min	25%	50%	75%	max
20.0	0.3791	0.0373	0.345	0.3594	0.3738	0.3848	0.5243

Fonte: Do Autor (2023)

Deste resumo estatístico pode-se observar que a média dos dados coletados foi no valor de 0.3791 segundos com um desvio padrão com valor de aproximadamente a décima parte da média, com isso a variação dos dados coletados não mostrou-se muito elevada e permaneceram próximos a média. O comportamento dos dados para cada teste referente a cada uma das duas máquinas está representado no gráfico de barras, na Figura 12.

Figura 12 - Gráfico de Barras - Tempo de Gravação de Dados.



Fonte: Do Autor (2023)

O gráfico da figura 12 possibilita verificar que a média de cada máquina esteve bem próxima da média geral dos dados, porém para este teste a máquina 2 obteve os menores tempos de resposta para acesso ao banco de dados. Neste conjunto de dados é verificado um valor entre aqueles da máquina 1 que esteve muito maior do que o restante dos dados, sendo este valor o responsável por aumentar a média dos dados da primeira máquina.

A partir dos gráficos da figura 12 conclui-se que para a gravação dos dados as tomadas de tempo comportam-se com menor variância, e que o tempo de acesso ao banco de dados depende majoritariamente da conexão com a internet, sendo que a parcela de contribuição do hardware da máquina influencia somente na função estabelecida dentro do algoritmo, como esta função possui baixa complexidade, a capacidade de processamento torna-se um fator secundário para contabilizar o desempenho desta etapa de funcionamento do sistema.

Por fim, a última etapa de desempenho testada foi a de leitura de dados, ou seja, da etapa de consulta ao Firebase. A partir dos dados coletados foram montados os gráficos para visualização do comportamento de tomada de tempo para cada máquina, sendo o resumo estatístico disposto na Figura 13.

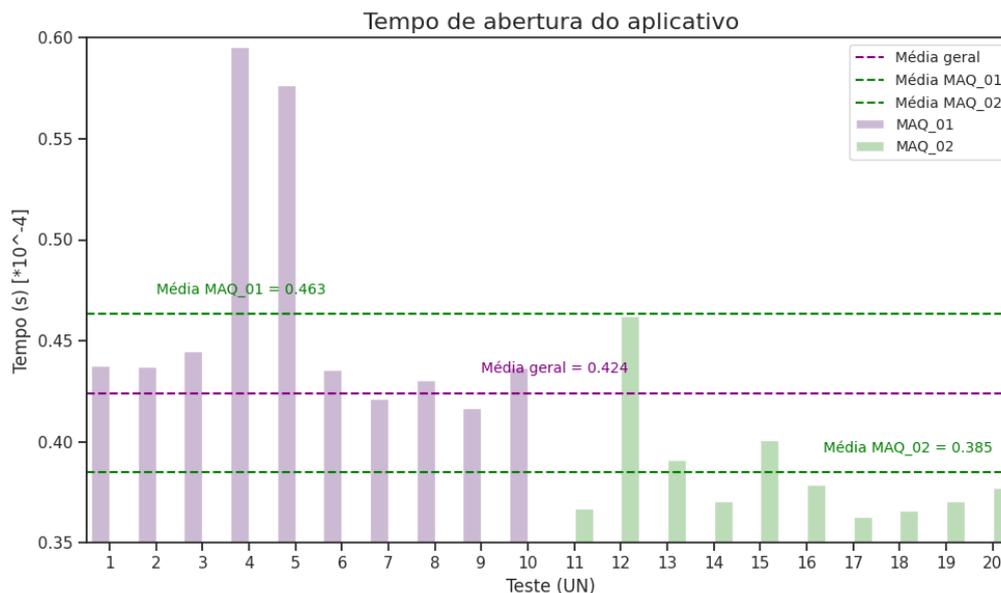
Figura 13 - Resumo Estatístico - Leitura de Dados.

Resumo Estatístico							
count	mean	std	min	25%	50%	75%	max
20.0	0.4241	0.0638	0.3628	0.3755	0.4191	0.4375	0.5957

Fonte: Do Autor (2023)

Com o resumo estatístico da figura 13, nota-se que a média de tempo para leitura dos dados foi maior do que o tempo para a gravação dos dados, sendo observado também nesta etapa um aumento do desvio padrão em relação ao tempo de gravação. O comportamento dos dados em relação a média para cada teste estão na Figura 14.

Figura 14 - Gráfico de Barras - Tempo de Leitura de Dados.

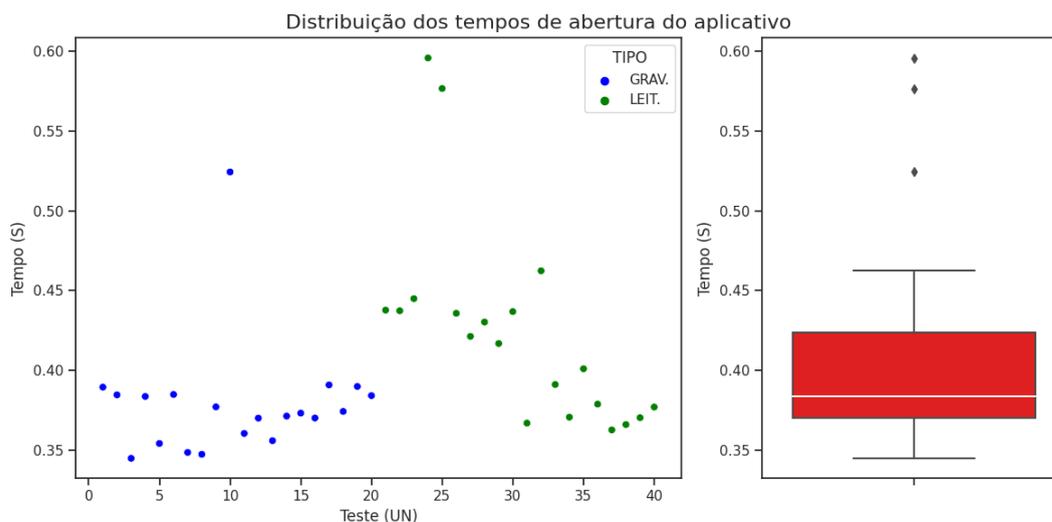


Fonte: Do Autor (2023)

Para este caso de análise de desempenho, os dados para a máquina 1 apresentaram valores bem acima dos obtidos para a máquina 2, sendo a média geral encontrada considerando ambas as máquinas é de 0.424 segundos. Além de obtidos valores maiores para os testes na máquina 1 estes ainda apresentaram dois picos de leitura.

Se comparando os resultados de desempenho para os dois últimos casos, é possível concluir que o tempo de gravação tende a ficar com valores mais baixos se comparados ao tempo de leitura, isto ocorre, pois, para realização da gravação é necessário tratar de um único dado pontual para convertê-lo e enviar ao Firebase, por outro lado, para realização da leitura é necessário percorrer toda a lista de dados existente para então convertê-la para visualização na tabela do aplicativo. Ainda em se tratando da diferença de tempo, conforme o tamanho da base de dados salvas no Firebase aumentar, o tempo de leitura tende a aumentar também por necessitar percorrer uma quantidade de termos maior. Uma comparação entre os dados de leitura e gravação pode ser visualizada na Figura 15, sendo possível verificar a variação maior de tomada de tempos de leitura, estes ocasionados pelo período de conexão necessário maior, fazendo com que o sistema tenha maior chance de enfrentar oscilações no período de espera da finalização da requisição.

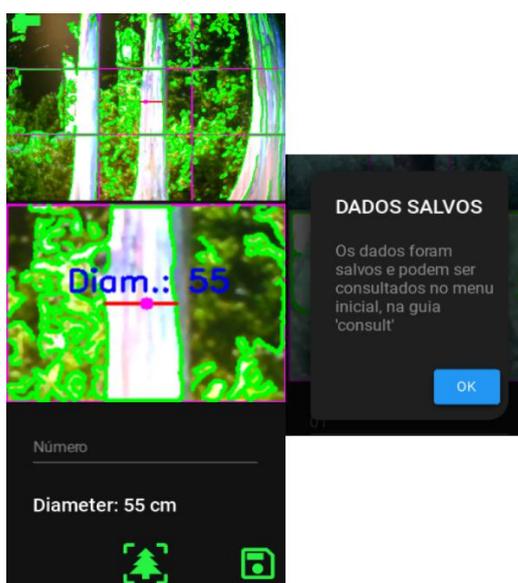
Figura 15 - Tempos de Gravação x Leitura.



Fonte: Do Autor (2023)

Em relação à usabilidade do sistema, o modelo seguido, disponível no Apêndice B, foi utilizado de base para a construção do aplicativo, porém ao longo das simulações alguns ajustes em questão de usabilidade foram modificados a fim de melhorar a experiência do usuário, sendo que os resultados finais após as modificações podem ser observados no apêndice C. Conforme observado neste apêndice, alguns alterações foram feitas para aumentar, principalmente, a cadência de utilização das funções do aplicativo e a visibilidade das informações, como a Figura 16, a qual a estrutura de captura foi modificada.

Figura 16 - Tela de Salvamento de Dados.



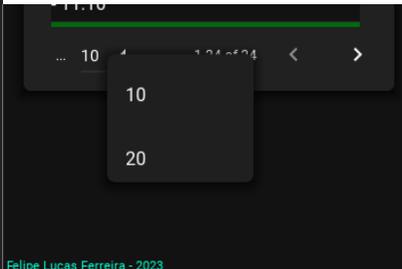
Fonte: Do Autor (2023)

De acordo com a Figura 16, pode-se observar que o modo de operação de captura e salvamento dos dados foram alterados, desta forma, ao capturar a imagem, pressionando o botão inferior central da tela, a imagem logo abaixo, referente a um recorte da área central da imagem acima, com a aplicação de uma ampliação é exibida, juntamente com o diâmetro encontrado referente à linha horizontal na altura do ponto magenta central. Nesta adaptação, o aumento de velocidade de operação do aplicativo ao refazer a captura é dado pelo fato de não ser exibida uma caixa de mensagem com uma indagação se o usuário quer refazer ou salvar a análise, mas este processo foi modificado para simplesmente pressionar novamente o botão de captura para refazer o cálculo e reexibir a imagem. Já o botão de salvar fica logo à direita do botão de captura, logo, ao inserir a numeração da árvore analisada e obtido o diâmetro requerido, a caixa de diálogo é exibida somente após salvar os dados caso a conexão com o banco de dados tenha sido finalizada corretamente. Portanto, a usabilidade de captura da tela de análise de imagem foi implementada de modo a fornecer um modo de operação simplificado para o usuário.

Ainda relacionado à usabilidade do aplicativo, a tela *consult* permitiu a visualização dos dados salvos de maneira organizada, sendo que ela possui a possibilidade de exibição de quantidades variadas de dados por tela, além de permitir a função de arrastar os dados na tela para visualizar o restante, sendo que estes dados podem ser exibidos de 10 em 10 valores com acréscimo de 10 por janela conforme mais dados são anexados, como pode ser observado na Figura 17. Além disto, a tabela configurada de maneira a permitir a troca de páginas, desta maneira os dados podem ser separados por quantidades delimitadas e separadas por páginas de visualização, sendo estas definidas pela quantidade de dados salvos dentro do banco de dados do Firebase. A taxa de atualização dos dados é realizada instantaneamente, sendo que assim que os dados são salvos na tela *scan*, ao entrar na guia *consult* os dados salvos podem ser visualizados na tabela, sem a necessidade de espera para exibição destas informações.

Figura 17 - Ajuste da Tabela de Dados.

Data	Number	Diameter
2023_06_21 - 11:08	01	14.449218
2023_06_21 - 11:10	01	14.571669
2023_06_21 - 11:10	01	14.204316
2023_06_21 - 11:10	01	14.326767
2023_06_21 - 11:10	01	14.204316
2023_06_21 - 11:10	01	14.326767



Fonte: Do Autor (2023)

Para as medidas coletadas de diâmetro das árvores, foram obtidas medidas de 10 árvores de maneira direta e 10 de maneira indireta, sendo que para a coleta de maneira indireta foram feitas 15 capturas para verificação dos valores da medida direta em relação à indireta, desta forma o total de dados analisados é igual a 150 medições indiretas, sendo este valor definido pelo autor a fim de experimentação da qualidade e precisão do sistema. Os valores do diâmetro das árvores estão dispostos na tabela 3.

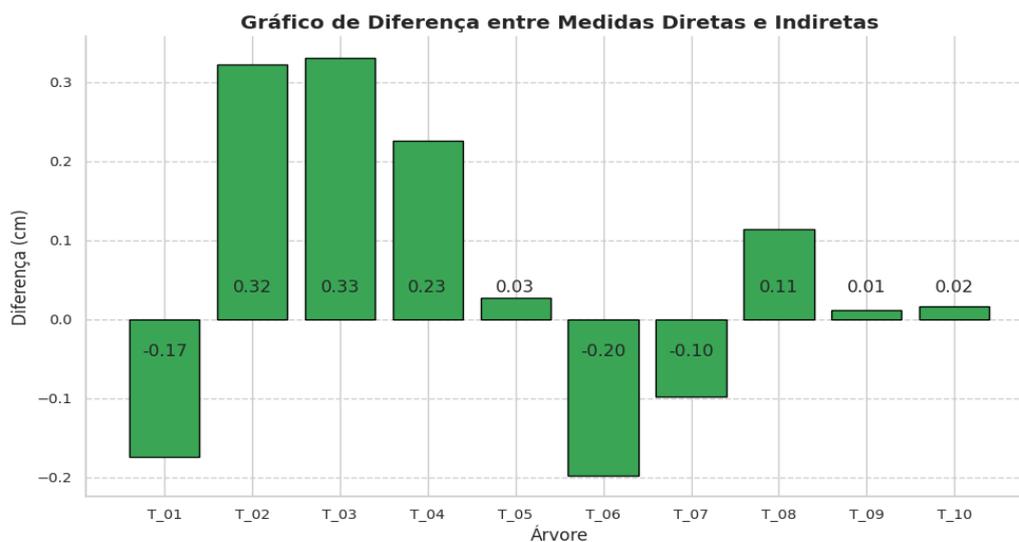
Tabela 3 - Comparação - Medida Direta x Medida Indireta

ÁRVORE	DIÂMETRO - MEDIDA DIRETA (cm)	DIÂMETRO - MEDIDA INDIRETA (cm)
T_01	14,005635	14,180000
T_02	21,963382	21,641000
T_03	23,523101	23,192000
T_04	16,870424	16,645000
T_05	23,236622	23,209000
T_06	24,096059	24,294000
T_07	20,212678	20,311000
T_08	23,395777	23,282000
T_09	21,326762	21,315000
T_10	21,167607	21,151000

Fonte: Do Autor (2023)

A tabela 3 mostra os valores da medida direta em comparação com as medidas indiretas do diâmetro do tronco da árvore captadas a aproximadamente 1,3m do chão, assim estes dados mostram a proximidade da obtenção desta medida pelos 2 métodos, sendo esta diferença observada na figura 18, a qual mostra que as maiores diferenças de medição observada foram no valor de 0,33cm, sendo que este número representa aproximadamente 1,4% de divergência entre os dois métodos de aquisição de dados.

Figura 18 - Gráfico de Diferença



Fonte: Do Autor (2023)

Uma vez observado isto, é possível verificar o comportamento dos dados coletados para as medidas indiretas, sendo estas representadas na figura 19, na qual é possível verificar que a série de dados de medida realizada para a árvore 10 obteve o menor valor de desvio padrão, sendo este valor igual a 0,182 e representa um valor baixo equivalente a aproximadamente 0,86% da média encontrada para esta análise. Por outro lado, para a árvore de número 4 o valor observado foi de 0,706 e este é o maior entre todas as análises, sendo assim este desvio padrão representa cerca de 4,24% da média encontrada.

Figura 19 - Resumo Estatístico – Medidas do Aplicativo

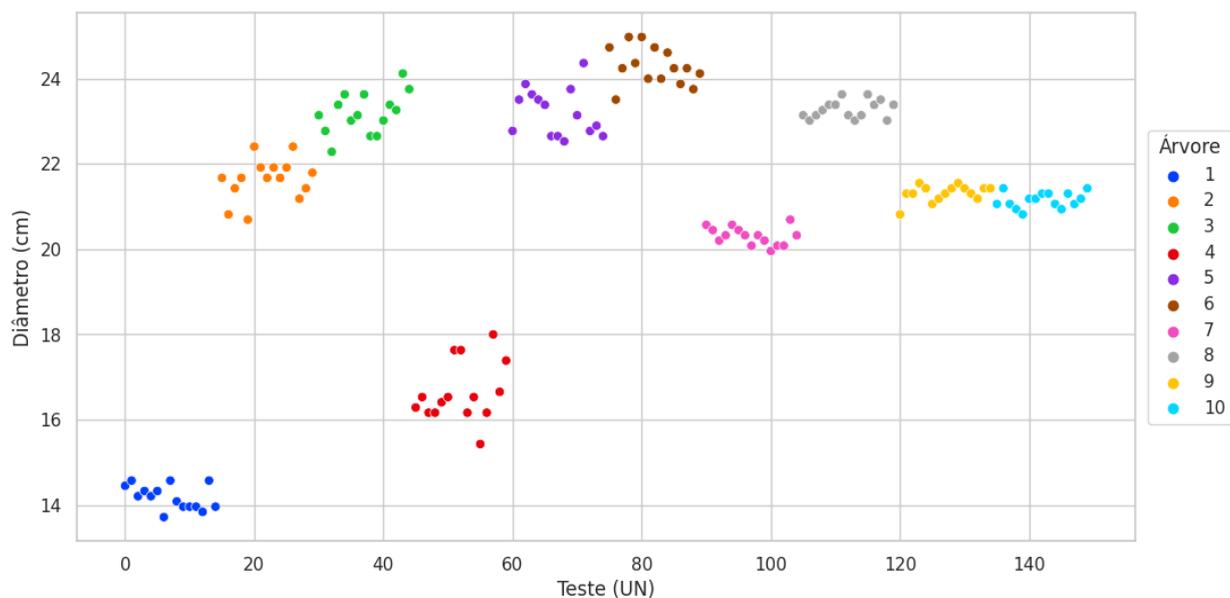
Resumos Estatísticos por Número

	count	mean	std	min	25%	50%	75%	max
Número 1	15.0	14.18	0.282	13.715	13.959	14.204	14.388	14.572
Número 2	15.0	21.641	0.486	20.694	21.429	21.674	21.919	22.409
Número 3	15.0	23.192	0.485	22.286	22.898	23.143	23.511	24.123
Número 4	15.0	16.645	0.706	15.429	16.164	16.531	17.021	18.0
Número 5	15.0	23.209	0.557	22.531	22.715	23.143	23.572	24.368
Número 6	15.0	24.294	0.441	23.511	24.0	24.245	24.674	24.98
Número 7	15.0	20.311	0.211	19.96	20.143	20.327	20.449	20.694
Número 8	15.0	23.282	0.211	23.021	23.143	23.266	23.388	23.633
Número 9	15.0	21.315	0.193	20.817	21.245	21.306	21.429	21.551
Número 10	15.0	21.151	0.182	20.817	21.062	21.184	21.306	21.429

Fonte: Do Autor (2023)

Tendo em vista estes pequenos valores de diferença, os dados de leitura utilizando o aplicativo apresentou variações em seus resultados para cada medida coletada, sendo as dispersões destas representadas no gráfico de dispersão da Figura 20.

Figura 20 - Gráfico de Dispersão - Medida Indireta.



Fonte: Do Autor (2023)

De acordo com a figura 20 nota-se que algumas árvores apresentaram uma dispersão na medida do diâmetro maiores do que outras, como é o caso das árvores 2, 3, 4 e 5 que apresentaram as maiores variações de medida dentre os valores obtidos. Estas variações da coleta de dados são encontradas por questões referentes à imagem capturada pela câmera, sendo que estas possuem a tendência de enfrentar problemas de sombra e luz, indo ao encontro ao que foi abordado por Ferreira e Melchiori (2023), sendo que a incidência de luz sob a área de foco da câmera interfere na interpretação das zonas de interesse do objeto. No caso das árvores, as variações de exposição da luz a qual a câmera foi exposta em ambiente aberto causou pequenas interferências no resultado final do aplicativo, gerando algumas divergências para a análise, que podem ter sido agravadas pelo posicionamento da distância da câmera até o tronco da árvore que não teve um controle rigoroso desta medida, o que apesar destes agravantes, os dados apresentaram resultados próximos aos obtidos pelas medidas de maneira direta, o que demonstra um bom desempenho do sistema, uma vez que a aquisição de medida por meio da medição direta também está exposta a imprecisões de medida, podendo estas serem decorrentes do ponto em que é medido a árvore, do equipamento de medida, entre outros.

Um ponto de atenção para obtenção destas medidas, conforme citado, está ligado à distância correta para medição do diâmetro da árvore, como observado por Celes et al. (2019), sendo que a precisão das medidas do diâmetro das árvores depende da distância controlada entre a câmera e a árvore a ser analisada, desta forma recursos tecnológicos complementares que garantam o posicionamento correto em relação ao ponto a ser analisado torna-se um complemento que garante uma maior confiabilidade dos dados.

CONCLUSÃO

A utilização de tecnologias acessíveis e equipamentos de uso comum, o manejo florestal pode alcançar melhores resultados. O uso da medição indireta para a aquisição dos dados de diâmetro do tronco da árvore permite maior velocidade na obtenção destes valores e o armazenamento destas informações em bancos de dados em tempo real aumenta a confiabilidade destes bem como permite gerar relatórios com dados históricos de maneira facilitada. Os fatores de segurança adotados para o uso destas tecnologias permitem o funcionamento completo do sistema sem riscos aparentes de vazamento ou perda de informação importante. Alguns pontos de atenção referentes às imprecisões de medidas devem ser levados em consideração ao realizar a análise, no entanto a utilização de recursos complementares garante a maior precisão na coleta destas medidas. Com isso, um sistema equipado com tecnologias de visão computacional garante uma maior cadência na coleta de dados e possibilita a redução da equipe necessária para o mapeamento florestal sem perder a qualidade e realizando isto em menores tempos, tornando-se com isso um sistema viável.

REFERÊNCIAS

- ARAÚJO, V. A.; GARCIA, J. N., et al. Importância da madeira de florestas plantadas para a indústria de manufaturados. **Brazilian Journal of Forestry Research**. Piracicaba, 2017.
- AVERY, T. E.; BURKHART, H. E. **Forest Measurements**. 5 ed. Waveland Press, 2015.
- BONAN, G. B. Forests and climate change: forcings, feedbacks, and the climate benefits of forests. **Science**, v. 320, n. 5882, p. 1444-1449, 2008.
- BRASIL. Ministério do Meio Ambiente. **Controle e prevenção do desmatamento e dos incêndios florestais**. Disponível em: <<https://www.gov.br/mma/pt-br/assuntos/servicosambientais/control-de-desmatamento-e-incendios-florestais>> . Acesso em: 15 jun. 2023.
- BUITEN, M. C. Towards Intelligent Regulation of Artificial Intelligence. **European Journal of Risk Regulation**. Cambridge University Press, 2019.
- BULLOCK et al. Mapping the Landscape of Artificial Intelligence Applications against COVID19. **Journal of Artificial Intelligence Research**, 2020.
- BURGER, W.; BURGE, M. J. **Digital Image Processing: An Algorithmic Introduction Using Java**. Springer, 2016.
- CANNY, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, p. 679-698, 1986.
- CASTELO, T. B. Legislação florestal brasileira e políticas do governo de combate ao desmatamento na amazônia legal. **Ambiente & Sociedade**. São Paulo, 2015.
- CELES et al. Digital Approach for Measuring Tree Diameters in the Amazon Forest. **Floresta e Ambiente**, 2019.
- CONSELHO NACIONAL DO MINISTÉRIO PÚBLICO (CNMP). **Acordo de Resultados em Defesa da Amazônia**. Disponível em <https://www.cnmp.mp.br/portal/images/noticias/2020/julho/Acordo_de_Resultados_em_Defesa_da_Amazonia_06.08.pdf>. Acesso em: 15 jun. 2023.
- FÁVERI, R.; SILVA, A. Método GUT aplicado à gestão de risco de desastres: uma ferramenta de auxílio para hierarquização de riscos. **Revista Ordem Pública**, v. 9, n. 1, p. 93-107, 2016.
- FERREIRA, F. L.; MELCHIORI, A. P. P. Sistema De Visão Aplicado Na Indústria Lean: A Inteligência Artificial Acessível. **Estudos Interdisciplinares em Engenharia Mecânica**, 2023.
- FIREBASE. **Deixe seu app o melhor possível**. Disponível em: <<https://firebase.google.com/>>. Acesso em: 13 jun. 2023.
- FISCHER, A.; ZYLBERSZTAJN, D. O fomento florestal como alternativa de suprimento de matéria-prima na indústria brasileira de celulose. **Revista Eletrônica de Administração**, v. 18, n. 2, p. 494–520, 2012.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS (FAO). **Global Forest Resources Assessment: Key findings**, 2020.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS (FAO). **Global Forest Resources Assessment: Key findings**, 2018.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. **FAO Forestry Paper 163: Global Forest Resources Assessment 2010 - Main report**. Rome: FAO, 2010.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. 1^a ed. Upper Saddle River, NJ: Prentice Hall, 2003.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. **Database Systems: The Complete Book**. 2 ed. Pearson, 2008.

GEIST, H. J.; LAMBIN, E. F. Proximate Causes and Underlying Driving Forces of Tropical Deforestation: Tropical forests are disappearing as the result of many pressures, both local and regional, acting in various combinations in different geographical locations. **BioScience**, v. 52, p. 143-150, 2002.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3 ed. Pearson, 2008.

GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. **Digital Image Processing Using MATLAB**. 3 ed. Gatesmark, 2020.

JÚNIOR, D. L.; COLODETTE, J. L. Importância e versatilidade da madeira de eucalipto para a indústria de base florestal. **Brazilian Journal of Forestry Research**. Viçosa, 2013.

JSON. **Introducing JSON**. Disponível em: <<https://www.json.org/json-en.html>>. Acesso em: 13 jun. 2023.

KEPNER, C. H.; TREGOE, B. B. **O administrador racional: Uma Abordagem Sistemática à Solução de Problemas e Tomada de Decisões**. São Paulo: Atlas, 1981.

KHAN, S. et al. **A Guide to Convolutional Neural Networks for Computer Vision**. Morgan & Claypool, 2018.

KIVY. **Welcome to Kivy**. Disponível em: <<https://kivy.org/doc/stable/>>. Acesso em: 8 jun. 2023.

KIVYMD. **Welcome to KivyMD's documentation!** Disponível em: <<https://kivymd.readthedocs.io/en/latest/index.html>>. Acesso em: 8 jun. 2023.

KRISHNA, R. **Computer vision: foundations and applications**. Stanford University, 2017.

MEIRELES, M. **Ferramentas administrativas para identificar, observar e analisar problemas: organizações com foco no cliente**. 2 ed. São Paulo: Arte & Ciência Villipress, 2001.

MINISTÉRIO DA AGRICULTURA. **Plano Nacional de Desenvolvimento de florestas plantadas**. 2018. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/camaras-setoriais-tematicas/documentos/camaras-setoriais/florestas-plantadas/2018/copy_of_40aro/ap-salomaoplantarflorestas.pdf>. Acesso em: 7 jun. 2023.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1951.

OMG. **Business Process Model and Notation (BPMN)**. V 2.0. Needham: Object Management Group, 2011.

OPENCV. **Contours: Getting Started**. Disponível em: <https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html>. Acesso em: 8 jun. 2023a.

OPENCV. **Feature Detection: Canny**. Disponível em: <https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html#canny>. Acesso em: 7 jun. 2023b.

OPENCV. **Finding contours in your image**. Disponível em: <https://docs.opencv.org/3.4/df/d0d/tutorial_find_contours.html>. Acesso em: 8 jun. 2023c.

OPENCV. **Image Filtering: GaussianBlur**. Disponível em: <<https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html>>. Acesso em: 7 jun. 2023d.

OPENCV. **Morphological Transformations**. Disponível em: <https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html>. Acesso em: 7 jun. 2023e.

PANAGIOTIDIS, D.; ABDOLLAHNEJAD, A.; SLAVÍK, M. Assessment of Stem Volume on Plots Using Terrestrial Laser Scanner: A Precision Forestry Application. **Sensors**, 2021.

PAZ, L. R. S. et al. Análise Ambiental da Expansão de Florestas Plantadas no Extremo Norte do Tocantins. **Anuário do Instituto de Geociências**, v. 43, 2020.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: uma abordagem profissional**. 9 Ed. Porto Alegre: AMGH, 2021.

PYTHON. **História e Licença**. Disponível em: <<https://docs.python.org/pt-br/3/license.html>>. Acesso em: 13 jun. 2023a.

PYTHON. **Json: JSON encoder and decoder**. Disponível em: <<https://docs.python.org/3/library/json.html>>. Acesso em: 13 jun. 2023b.

REIS, G. G.; REIS, M. G. Fisiologia da brotação de eucalipto com ênfase nas suas relações hídricas. **Série Técnica Ipef**. V. 11, n. 30, p. 9-22, 1997.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4 ed. Pearson Education, 2020.

SADALAGE, P.; FOWLER, M. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. Addison-Wesley Professional, 2012.

Hill. ex. Maiden e relação com as variáveis climáticas e fertilização mineral. **Revista Árvore**, v. 34, n. 6, p. 979–990, 2010.

SETTE JR, C. R. et al. Crescimento em diâmetro do tronco das árvores de *Eucalyptus grandis* W. Hill. ex. Maiden e relação com as variáveis climáticas e fertilização mineral. **Revista Árvore**, v. 34, n. 6, p. 979–990, 2010.

SHOEIBI, A. et al. Applications of deep learning techniques for automated multiple sclerosis detection using magnetic resonance imaging: A review. **Computers in Biology and Medicine**, v. 136, 2021.

SILVER, B. **BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0**. Cody-Cassidy Press, 2011.

SOLEM, J. E. **Programming Computer Vision with Python**. Ed. O'Reilly Media, 2012.

SOMMERVILLE, I. **Software Engineering**. 10th ed. Boston, MA: Pearson Education, 2016.

SONG, J. et al. A subgraph-based knowledge reasoning method for collective fraud detection in E-commerce. **Neurocomputing**. v. 461, 2021.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. 2 ed. Springer, 2022.

WANG et al. A multi-focus image fusion framework based on multi-scale sparse representation in gradient domain. **Signal Processing**, 2021.

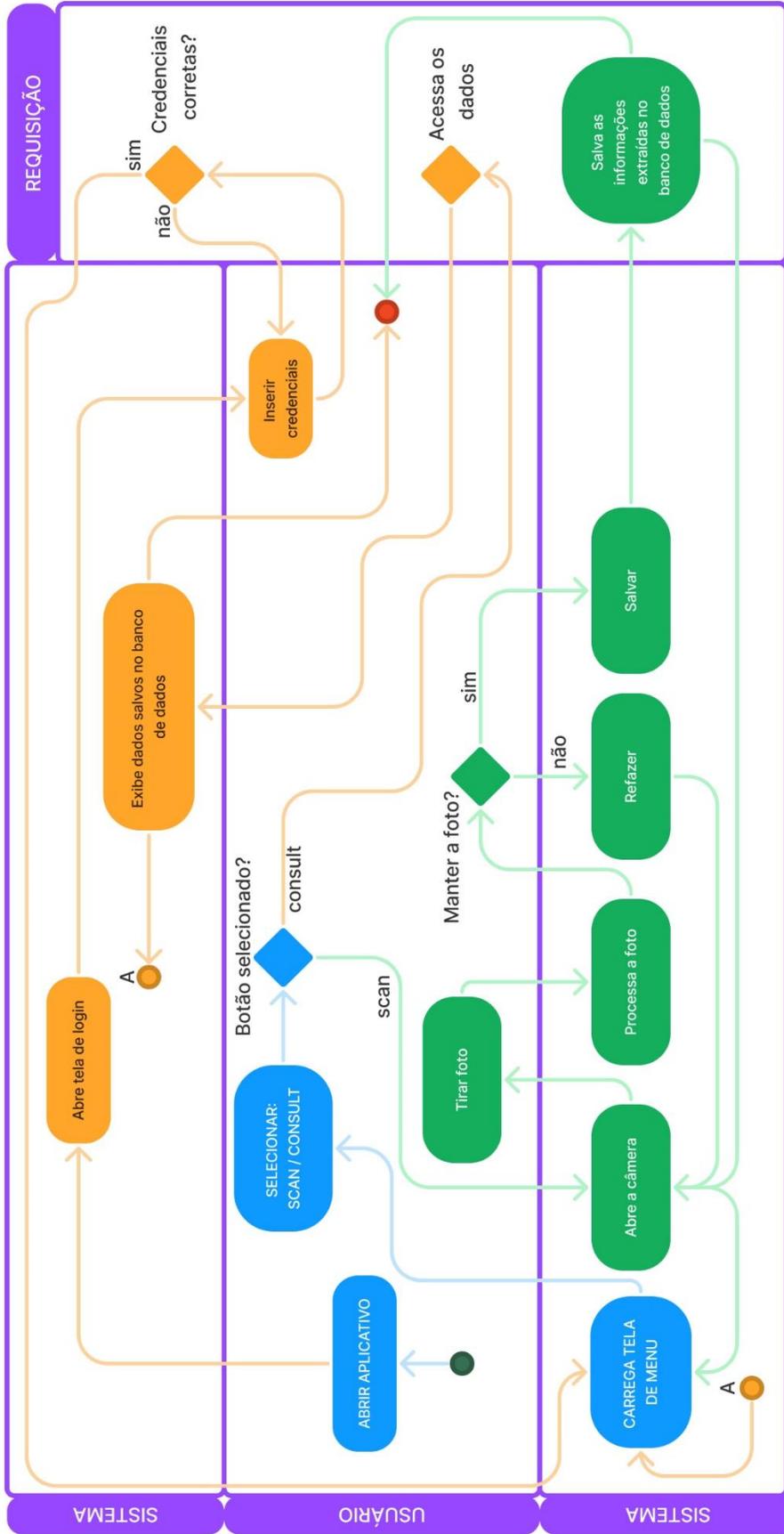
WOO, H.; KIM, I.; CHOI, B. Computer Vision Techniques in Forest Inventory Assessment: Improving Accuracy of Tree Diameter Measurement Using Smartphone Camera and Photogrammetry. **Sensors and Materials**, 2021.

WU, X. et al. Passive measurement method of tree diameter at breast height using a smartphone. **Computers and Electronics in Agriculture**, v. 163, 2019.

YOUSAF, R. M. et al. **A Comparative Study of Various Edge Detection Methods**. 14th International Conference on Computational Intelligence and Security (CIS), p. 96-99. Hangzhou, China, 2018.

ZENID, G. J. Madeira na construção civil. **Instituto de Pesquisas Tecnológicas do Estado de São Paulo**. São Paulo, 2011.

APÊNDICE A



APÊNDICE C

