

Faculdade de Tecnologia da Zona Leste Fatec Zona Leste	8º EnGeTec 2025
8º Encontro de Gestão e Tecnologia	ISSN 2675-4479
Organizações Ágeis em Tempos de Incerteza	Dezembro de 2025

AGRO-G.E.S.F: Sistema Inteligente de Monitoramento e Prevenção de Pragas em Cultivos em Linha

Enzo Costa Paz¹

enzo.paz@etec.sp.gov.br

Felipe Vieira de Oliveira¹

felipe.oliveira849@etec.sp.gov.br

Gustavo de Souza Morais¹

gustavo.morais42@etec.sp.gov.br

Sakiri Moon Payao Cestari¹

sakiri.cestari@etec.sp.gov.br

Jeferson Roberto de Lima¹

jeferson.lima17@etec.sp.gov.br

AGRO-G.E.S.F: Intelligent System for Monitoring and Prevention of Plagues in Row Cultivations

AGRO-G.E.S.F: Sistema Inteligente para el Monitoreo y la Prevención de Plagas en Cultivos en Hileras

Palavras-chave:

Python.
Visão computacional.
Machine Learning.
Praga.

Keywords:

Python.
Computer vision.
Machine Learning.
Plague.

Palabras clave:

Python.
Visión por computadora.
Machine Learning.
Plaga.

Apresentado em:

03 dezembro, 2025

Evento:

8º EnGeTec

Local do evento:

Fatec Zona Leste

Avaliadores:

Avaliador 1
Avaliador 2



Resumo:

Este artigo se refere a um sistema embarcado para o monitoramento de possíveis sinais de pragas em plantações no formato de linha, contendo em si a documentação escrita de tal. Esse sistema contará com uma câmera, um receptor de GPS (*Global Positioning System*), bem como a utilização de uma SBC (*Single Board Computer*) para o processamento local da imagem. O dispositivo visa ajudar os pequenos agricultores que vêm sofrendo com a identificação tardia de pragas e doenças em suas plantações rasteiras, perdendo uma grande quantidade de mercadoria, e forçando o uso excessivo de agrotóxicos, que afetam a saúde do produtor, do produto e do consumidor final. Todo o projeto foi desenvolvido com base na metodologia qualitativa, onde se realiza uma análise mais detalhada e ouve depoimentos dos usuários e compara números. O projeto acredita que com o desenvolvimento do dispositivo, os pequenos agricultores poderão fazer uma identificação mais rápida, reduzindo a perda de produto, o que reduz o uso de agrotóxicos e que consequentemente melhora a saúde do pequeno agricultor, da plantação e do produto.

Abstract:

This article refers to an embedded system for monitoring potential signs of pests in crops in a line format, including written documentation. This system will include a camera, a GPS (*Global Positioning System*) receiver, and a Single Board Computer (SBC) for local image processing. The device aims to help small farmers who have been suffering from the late identification of pests and diseases in their low-growing crops, losing a large amount of produce and forcing the excessive use of pesticides, which affect the health of the producer, the product, and the end consumer. The entire project was developed based on a qualitative methodology, which involves a more detailed analysis, listening to user testimonials, and comparing data. The project believes that with the development of the device, small farmers will be able to identify pests more quickly, reducing product loss, which reduces the use of pesticides and consequently improves the health of the small farmer, the crop, and the product.

Resumen:

Este artículo describe un sistema integrado para el monitoreo continuo de posibles plagas en cultivos, incluyendo documentación escrita. El sistema consta de una cámara, un receptor GPS (*Sistema de Posicionamiento Global*) y una computadora de placa única (SBC) para el procesamiento local de imágenes. El dispositivo busca ayudar a los pequeños agricultores que han sufrido la dificultad de identificar plagas y enfermedades en sus cultivos de bajo crecimiento, lo que les ocasiona grandes pérdidas de producción y el uso excesivo de pesticidas, afectando la salud del productor, el producto y el consumidor final. El proyecto se desarrolló con una metodología cualitativa, que incluye un análisis detallado, la recopilación de testimonios de usuarios y la comparación de datos. Se espera que, con el desarrollo del dispositivo, los pequeños agricultores puedan identificar plagas con mayor rapidez, reduciendo las pérdidas de producto, el uso de pesticidas y, por consiguiente, mejorando la salud del agricultor, el cultivo y el producto.

¹ Instituição dos autores

1. Introdução

O Sistema Inteligente de Monitoramento e Prevenção de Pragas em Cultivo de Linha é um dispositivo capaz de percorrer uma lavoura de plantas rasteiras em todos os cenários possíveis, identificando possíveis sinais de pragas e doenças nas folhas. Algumas das pragas e doenças identificáveis são: Vaquinha-Verde-Amarela, Requeima e Pinta-Preta. Ao identificar os sinais, passa as informações para o dashboard em uma aplicação desktop via cabo USB tipo C. O objetivo geral é identificar de forma precoce, em cultivos de formação em linha, possíveis sinais de doenças e pragas, e permitir que o pequeno agricultor tenha uma forma de fazer o monitoramento de sua lavoura, assim como ter uma noção do histórico das pragas e doenças em suas plantações, enquanto os objetivos mais específicos são garantir um sistema de aprendizagem do próprio dispositivo para uma melhor precisão e atuação e criar uma arquitetura que seja capaz de lidar com as situações comuns do dia a dia do pequeno produtor, como chuvas, sol quente e lugares com muitos relevos.

O principal problema é que o agricultor acaba por muitas vezes fazendo a identificação de forma tardia da doença ou da praga, o que acaba resultando em uma perda elevada do cultivo. Complementarmente, isto acaba por exigir do agricultor a aplicação de agrotóxicos de forma mais densa e agressiva, o que põe em risco tanto os trabalhadores que o aplicam, a própria plantação e o consumidor final. A praga e doença presente podem conseqüentemente gerar uma resistência maior ao agrotóxico, dificultando futuras passagens do químico e aumentando a presença dele no produto final ao consumidor.

Todas as pesquisas e o desenvolvimento do projeto se baseiam através da metodologia qualitativa, como explica Lakatos e Marconi (2017), onde se analisa informações oriundas de artigos e artigos que abordam a temática da presença de pragas e doenças dentro de lavouras, com destaque nas consequências de tal fato e como a eficácia da utilização da linguagem Python pode ajudar neste meio.

Entre 2000 e 2012, o mercado mundial de agrotóxicos cresceu em 93%, enquanto o mercado brasileiro cresceu 190%. Em 2008, o Brasil ultrapassou os Estados Unidos e assumiu como o maior utilizador de agrotóxicos no mundo, segundo CARNEIRO *et al.* (2015). O país segue com este ranking até os dias atuais, utilizando os bioquímicos de forma desenfreada.

Rigotto, Vasconcelos e Rocha (2014) sustentam que os agrotóxicos são um problema de saúde pública, visto que muita da população é exposta aos químicos em fábricas, na agricultura, no combate às endemias e outros setores, perto de plantações, mas principalmente nós, que somos os principais consumidores das lavouras. Isso evidencia que tanto a produção quanto a utilização de agrotóxicos é um perigo ao bem-estar geral da população. Não obstante, Lopes e Albuquerque (2018) afirmam que os agrotóxicos podem contaminar reservatórios de água, rios e outras fontes de água, interferindo nos organismos que os utilizam. Algumas substâncias que já são proibidas no país a muito tempo, como é o caso do Hexaclorociclohexano (HCH), ainda são detectadas nesses locais. Deste modo, pode-se considerar que a utilização de agrotóxicos não só produz problemas de saúde, como instala químicos nocivos em ecossistemas que necessitam da água, agora contaminada com estes.

Percebendo essas questões, o presente artigo apresenta um questionamento: o que é possível fazer para reduzir o uso de agrotóxicos no cultivo, conseqüentemente reduzindo seu impacto?

Com base nas pesquisas e na prototipação do projeto, o artigo visualiza que os pequenos agricultores podem começar a fazer a identificação antecipada, reduzindo a perda elevada de seus produtos, onde o uso de agrotóxicos será de forma mais moderada o que conseqüentemente irá reduzir o risco de prejudicar a saúde do pequeno agricultor, a sua própria plantação e o consumidor final.

2. Fundamentação Teórica

Na atual seção, serão apresentados os principais conceitos e tecnologias para a construção da base teórica do desenvolvimento do Agro-G.E.S.F.

Faculdade de Tecnologia da Zona Leste Fatec Zona Leste	8º EnGeTec 2025
8º Encontro de Gestão e Tecnologia	ISSN 2675-4479
Organizações Ágeis em Tempos de Incerteza	Dezembro de 2025

2.1. Pragas nas plantações

Pragas, conforme estabelece a Embrapa (2024) são qualquer espécie, raça ou biotipo de um animal ou agente patogênico que venha a causar algum tipo de dano a plantas ou culturas. Em auxílio, Senar (2017) complementa que pragas são usualmente caracterizadas como ácaros, cochonilhas e nematoides, enquanto fungos, vírus e bactérias são usualmente coligados a doenças. Complementarmente, Martins, Fontes e Fornazier (2013 apud FERRÃO et al., 2017) informam que as pragas podem ocorrer de forma esporádica ou sistemática. Aquelas que ocorrem de forma esporádica e causam danos significativos, são conhecidas como pragas secundárias, já aquelas que ocorrem de forma sistemática toda vez que se implementa a cultura na lavoura e causam danos quantitativos/qualitativos, são conhecidas como pragas-chaves.

Barros *et al.* (2019), apontam que pragas e doenças, em casos como da cultura de soja, podem causar uma perda de até 30% de produção agrícola nacional, caso não haja o devido controle.

A agricultura, os modos de produção agrícola, a situação cultural e até social, estão inseridas em diferentes contextos de forma histórica, política e econômica, de forma que, Nogueira, Szwarchald, Damacena (2019) mostram que isso tende a afetar a vida do residente rural, de forma que, a combinação dessas variáveis tende a gerar níveis de exposição maior a tais agricultores à agrotóxicos.

2.2. Internet das Coisas (IoT)

Oliveira (2017) afirma que a internet das coisas (IoT) é um conceito que está se tornando cada vez mais presente na vida cotidiana. Nas últimas duas décadas, a IoT tem auxiliado a vivência humana em diversas áreas por meio da automação de tarefas, ou em objetos como crachás eletrônicos e veículos. A IoT consiste em qualquer objeto capaz de manipular dados em uma rede, visando auxiliar e simplificar ações complexas (MAGRANI, 2018).

2.3. Raspberry Pi 4

Conforme Dobbin (2022) indica, o Raspberry pi é um computador de placa única (SBC), montado em uma placa de circuito impresso (PCB). Se caracteriza por poder realizar as mesmas funções de um computador doméstico comum, como acessar a internet, pesquisar sites e assistir vídeos, tendo somente o adendo de levar tempo levemente maior para realizar tais funções devido a seu tamanho.

Como apontam Santos, Lopes e Dias (2024), o Raspberry pi 4 apresenta um nível de processamento mediano devido a seu processador de maior capacidade em atividades multitarefas.

Em complemento Lauxen, Lovatto e Rosa (2023) afirmam que o Raspberry pi 4 pode ser alocado dentro de um sistema como o cérebro do projeto, de forma que permita fazer a captura de imagens e controle de elementos ligados a ele.

2.4. Python

Segundo Luiz Eduardo Borges (2014), a linguagem Python foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), tendo como foco, o auxílio para profissionais como físicos e engenheiros.

Em conformidade a Paiva et al. (2020), Python é uma linguagem com características interessantes e de simples aprendizado. Inicialmente, possuía a meta inicial de ser uma linguagem de código mais enxuto e menos verboso. Em respaldo à está programação simples, Python permite o uso de diversos módulos de extensão, que fornecem ao Python, uma forte presença e poder.

Consoante a Menezes (2014), Python vem se tornando uma forte linguagem de programação em diversas áreas da computação, como inteligência artificial, banco de dados, biotecnologia e até mesmo em aplicações web. Em suporte, Python é uma linguagem conhecida por ter “batteries included”, ou seja, possui baterias inclusas, uma linguagem de programação pronta para ser usada. Em auxílio a tudo

isto, Python é uma linguagem de programação para se obter resultados em pouco tempo, devido a sua simples complexidade e alta densidade de módulos para importação.

O exemplo de código abaixo, é um código em Python que serve para fazer a soma de dois números, estes quais são inseridos pelo próprio usuário previamente dentro do próprio código. Abaixo, segue uma breve explicação sobre o código e seu funcionamento:

Linha 1 a 3: Cria uma “Docstring” (tipo de string), no qual no contexto atual, serve como uma forma de criar comentários, neste caso, o Python irá passar por estas linhas e irá ignorá-las;

Linha 5: Cria a função “soma”, está qual serve para modularizar o código e o deixar mais prático, permitindo que o programador não precise ficar repetindo as mesmas linhas de código, e, chamando a função somente quando necessária. Nesta aplicação, a função soma recebe dois parâmetros, dois itens, e os salva em variáveis (variáveis são formas de salvar valores específicos dentro da memória do computador) denominadas como “x” e “y”;

Figura 1 – Exemplo de código em Python para soma

```
1  """
2  Retorno de valores das funções (return)
3  """
4
5  def soma(x,y):
6
7      resultadosoma = x+y
8      return resultadosoma
9
10
11 resultadochamada = soma(1,2)
12 print(resultadochamada)
```

Fonte: Autores, 2025

Linha 7 a 8: Na linha 7, é criada uma variável para receber o valor da soma das duas variáveis declaradas inicialmente, logo em seguida na linha 8, o valor dessa variável é devolvido ao meio que lhe chamou;

Linha 11 a 12: É criada uma variável chamada “resultadochamada”, à qual chama a função soma, enviando como parâmetros os números 1 e 2, ao mesmo tempo que recebe o valor do retorno dado pela função. Na linha 12, por meio da função “print()”, é exibido ao usuário por meio do prompt de comando integrado da IDE, o valor da variável “resultadochamada”.

Figura 2 - Resultado do código em Python

```
0 resultado da soma é:3
G: / Python > 0.164s
```

Fonte: Autores, 2025

Como consequência da execução do código antes ilustrado, na imagem acima há o retorno do código Python, no caso, foram passados o número um e dois, e o retorno da soma de tais números foi o número três.

Faculdade de Tecnologia da Zona Leste Fatec Zona Leste	8º EnGeTec 2025
8º Encontro de Gestão e Tecnologia	ISSN 2675-4479
Organizações Ágeis em Tempos de Incerteza	Dezembro de 2025

2.5. Kotlin

O Kotlin teve sua origem em 2011 com o primeiro anúncio sobre sua existência, contudo sua primeira versão foi lançada em 2016, conforme Resende (2018), visando ser uma linguagem de programação mais pragmática e com uma sintaxe muito mais simples de aprender.

O Kotlin traz consigo uma vantagem sobressalente em relação ao Java que é a característica de ser null-safe, no caso como mostra Toledo (2019), o compilador é capaz de identificar possíveis objetos nulos antes da execução do código, o que evita erros na hora da execução do código.

Em 2019, como mostra o site oficial Android Developers (2025), com uma ação da Google, o Kotlin se tornou oficialmente a linguagem preferencial para o desenvolvimento de aplicações mobile.

2.6. Machine Learning

O aprendizado de máquina, ou mais comumente conhecido como “Machine Learning”, é, segundo Paixão *et al.* (2022), uma subárea da ciência da computação que trabalha com a união de técnica matemáticas e estatísticas aplicadas à algoritmos computacionais. Parafraseando Escovedo e Koshiyama (2020), o conceito de Machine Learning se foca na descoberta de padrões por meio dessa união de técnicas, independentemente do grau de utilidade de tais descobertas, ademais, a utilização de tais padrões para a automatização de determinadas tarefas, que antes, se poderiam se mostrar como difíceis para serem realizadas por humanos.

Quando se trata do aprendizado de máquina, se prova por meio de Ludemir (2021), que é necessário um grande volume de dados para que o algoritmo possa aprender de forma automática. Em auxílio, a acurácia de um modelo treinado pode variar conforme a quantidade de dados fornecidos ao modelo, juntamente a qualidade de tais dados, como diz Tsunoda *et al.* (2020). Em acréscimo, destaca-se o fato de que os modelos treinados têm se destacado dentro do cenário da agricultura de precisão, estes quais são sustentados por grandes volumes de dados e poder de processamento, como também, são amplamente utilizados para a detecção de doenças e ervas daninhas em culturas, previsão de produtividade e gestão da saúde do solo, conforme demonstra Santos, Beko e Leithardt (2022).

2.7. Visão Computacional

Segundo Barelli (2018), em 1982, Ballard e Brown definiram a visão computacional em sua obra chamada Computer Vision como a ciência que estuda e desenvolve meios para permitir que computadores possam por meio de sensores, enxergar e extrair informações do que enxergam do mundo. Conforme estabelece Silva (2020), a visão computacional é comumente associada a coleta, análise e processamento dos dados visuais por meio de computadores para diversos objetivos, indo desde a identificação de rostos até a identificação de objetos.

2.8. UML

Criado em 1995 com a unificação de três linguagem de modelagem, a Unified Modeling Language, traduzindo para o português, Linguagem de Modelagem Unificada ou UML é uma linguagem teórica para a modelagem de softwares que seguem o modelo de orientado objeto. (Guedes, 2009).

Segundo os autores Booch, Rumbaugh e Jacobson (2012), a UML é utilizada em softwares de estrutura complexa para determinar os requisitos, processos e funções de um sistema de nível simples ou avançados, além disso a UML permite definir quais serão as classes que criadas dentro do projeto e esquemas de banco de dados, a UML é o esqueleto dos projetos de software.

Consoante a Craig Larman (2005), a UML se tornou extremamente importante dentro de documentações de softwares por conta da requisição de padrões, essas normas se tornam importante no desenvolvimento criam melhores práticas para os programadores.

3. Método

Neste capítulo, será introduzido o método por trás do projeto, explicando como ele chegou a ser criado.

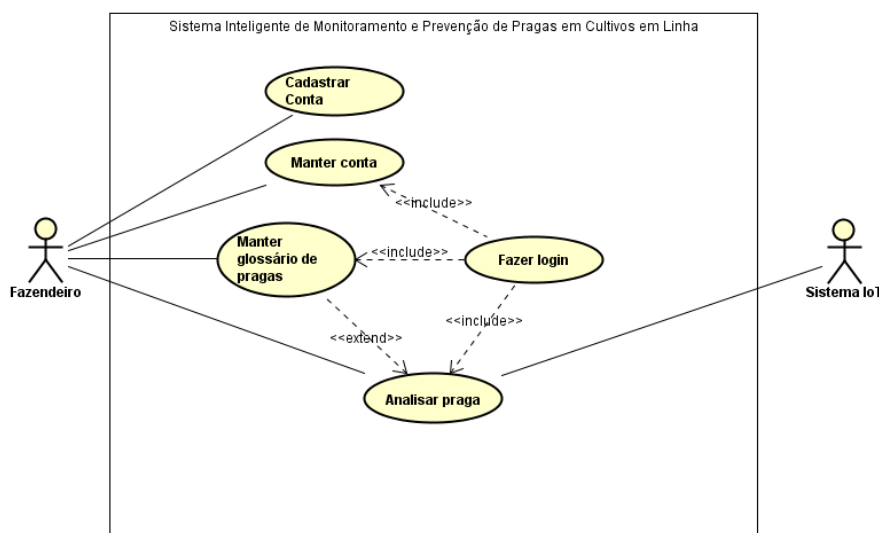
3.1. Diagramação

Um processo importante na criação de sistemas é a documentação, feita com uma norma chamada UML (*Unified Modeling Language*). Esta norma internacional envolve criar diagramas para a visualização concreta do projeto, visando trazer uma informação clara e consistente para todos os que estão envolvidos.

O início da diagramação do projeto envolve criar um Diagrama de Caso de Uso, que mostra a relação das ações que um ator (usuário ou sistema adjacente) pode fazer no sistema principal. O diagrama exemplifica os requisitos funcionais do projeto, ou seja, todas as funções que o projeto precisa exercer para ser utilizado corretamente.

A Figura 3 mostra todos os requisitos funcionais do sistema, além de dois atores: Fazendeiro, o usuário do sistema, e Sistema IoT, que representa a máquina monitora de pragas.

Figura 3 – Diagrama de Caso de Uso do AGRO-G.E.S.F.



Fonte: Autores, 2025.

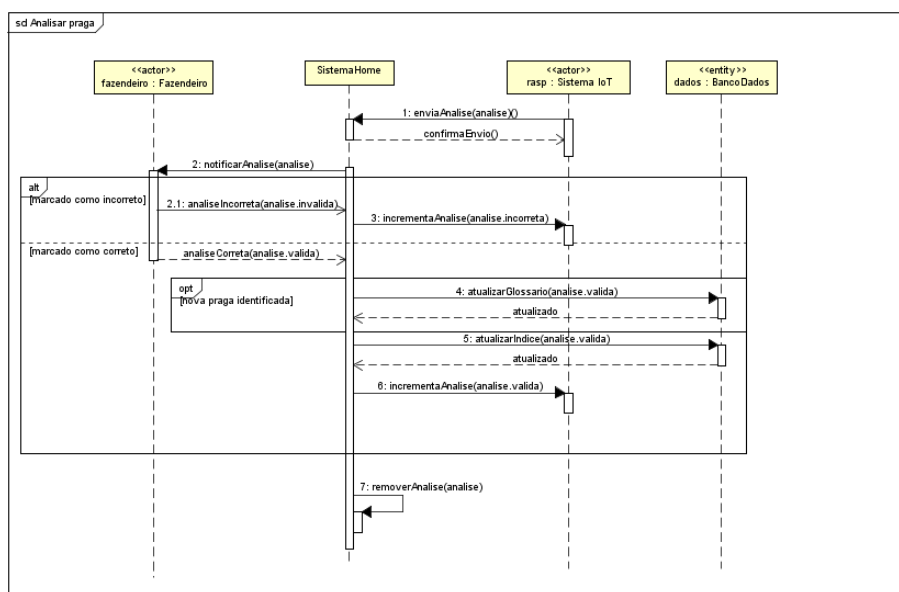
O ator Fazendeiro, o usuário do sistema, pode interagir com o sistema de várias maneiras. Para isso, porém, a primeira coisa que deve ser feita é o cadastro de uma conta no sistema, para segurança e autoridade. A partir disso, o usuário pode editar, visualizar e excluir a sua conta, tanto quanto pode manter o glossário de pragas interno do sistema, uma espécie de enciclopédia de pragas.

Para isso, porém, o usuário necessita fazer login, visualizado no diagrama como uma seta com a marca <<include>>. Essa marca implica que o caso de uso apontado pela seta necessariamente inclui o caso de uso no início da seta – por exemplo, manter conta “inclui” Fazer login. Observe que quase todos os casos de uso envolvidos com o Fazendeiro incluem Fazer Login, menos o Cadastrar conta, pois contradiz o caso.

A função principal do sistema é o Analisar praga, que envolve tanto o Fazendeiro quanto o sistema IoT. Este caso de uso tanto inclui fazer login quanto pode envolver Manter glossário de pragas, exemplificado pela seta com marca <<extend>>, que significa que o caso de uso apontado pela seta pode (mas não necessariamente precisa) utilizar o caso de uso no início da seta.

Todos os casos de uso incluem ou um Diagrama de Sequência ou Diagrama de Atividade, que mostram o passo a passo das funções destes casos. Observe o diagrama de sequência do caso de uso analisar praga, essencial ao projeto:

Figura 4 – Diagrama de Sequência: Analisar praga



Fonte: Autores, 2025.

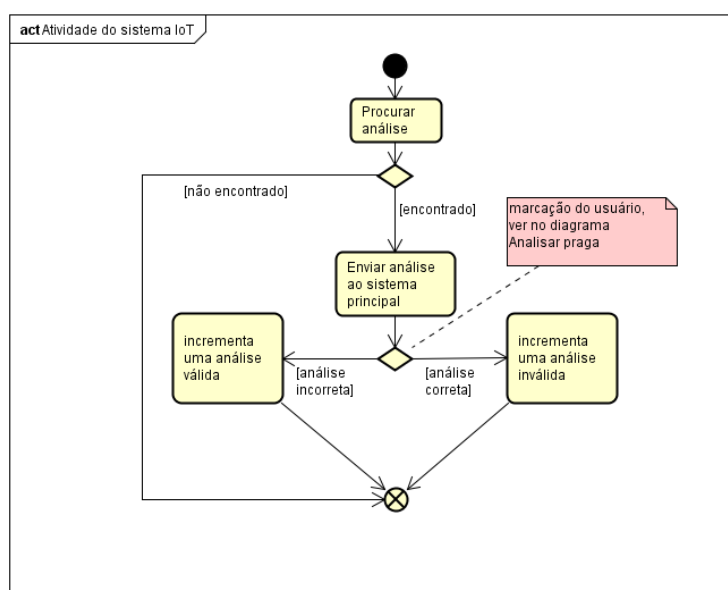
O diagrama mostra o fluxo de mensagens entre partes do sistema e atores, demonstrado pelas caixas no topo. Cada uma delas tem uma *lifeline* (linha de vida), que exemplifica a passagem do tempo para cada uma das partes, sendo preenchidas com mensagens dentre o sistema. A exemplo, a primeira mensagem no sistema é do IoT ao sistema principal. Ele envia uma análise de uma praga do campo ao sistema com o método `enviaAnalise()`, do qual o sistema responde ter confirmado o envio.

A caixa “alt” tem como função mostrar casos diferentes no diagrama, separados por caixas que englobam as mensagens. Ela indica uma exclusividade entre um caso e outro, ou seja, se um caso acontecer, o outro necessariamente não acontece. Além disso, a caixa “opt” indica uma sequência de eventos optativa, ou seja, que não é necessariamente obrigatória para o funcionamento do requisito.

Há também o diagrama de atividade, que mostra eventos de forma a criar um certo fluxograma. Este não é necessariamente linear, sendo mais utilizado para sistemas onde podem ser feitas várias coisas simultaneamente. Observe o diagrama abaixo, que descreve as funções do sistema IoT:

O diagrama começa com a bola preenchida, que significa o início da atividade. Dele, o IoT segue para Procurar Análise, que leva para uma condição, indicada por um losango. Ele ilustra que, caso o sistema não encontre uma praga, deve terminar a atividade, indicada pela bola com o X no meio. Se encontrado, ela envia a análise ao sistema, que torna com mais uma opção. Agora, o usuário deve marcar a análise como correta ou incorreta, o que fará com que o sistema IoT atualize sua visão computacional com dados do usuário, conforme as ideias do Machine Learning. Ambas as opções também terminarão a atividade.

Figura 5 – Diagrama de Atividade do sistema IoT



Fonte: Autores, 2025.

4. Metodologia

A pesquisa do projeto envolve a metodologia qualitativa, que utiliza uma participação do próprio público-alvo nos experimentos. Como os autores Lakatos e Marconi (2017) apontam, este tipo de pesquisa social envolve entender um fenômeno específico com mais vigor, utilizando depoimentos, interpretações e comparativos. Portanto, como não utiliza estatísticas e regras concretas, é mais suscetível a mudanças, focando em análises detalhadas ao invés de generalizadas.

O projeto ilustrado no artigo utilizou essa metodologia pois seu objetivo é focar em ajudar o pequeno agricultor, focando mais nos problemas encontrados em seu cotidiano e pretendendo criar uma ligação mais íntima entre o projeto e o público ao focar em suas questões pessoais.

5. Resultados e Discussões

O projeto foi desenvolvido durante o ano de 2025, tendo como foco a ajuda aos pequenos agricultores no controle de pragas e a redução do uso de agrotóxicos com a utilização de agricultura de precisão. Ele foi dividido em dois sistemas principais: uma aplicação feita para computador e celular para o agricultor, e um sistema IoT automatizado que servirá como o agente principal no aviso prévio de pragas.

5.1 Aplicação a nível usuário

O software criado para o computador (desenvolvido em *Python*) e o de celular (desenvolvido em *Kotlin*) visam criar a conexão necessária entre o usuário e o sistema IoT, visto que o IoT é completamente automatizado e pode ser deixado sozinho no campo. Este software visa ajudar o agricultor a monitorar a atividade de pragas em seu cultivo, elaborar um gráfico de evidências de pragas e gerenciar a comunicação entre usuário e IoT.

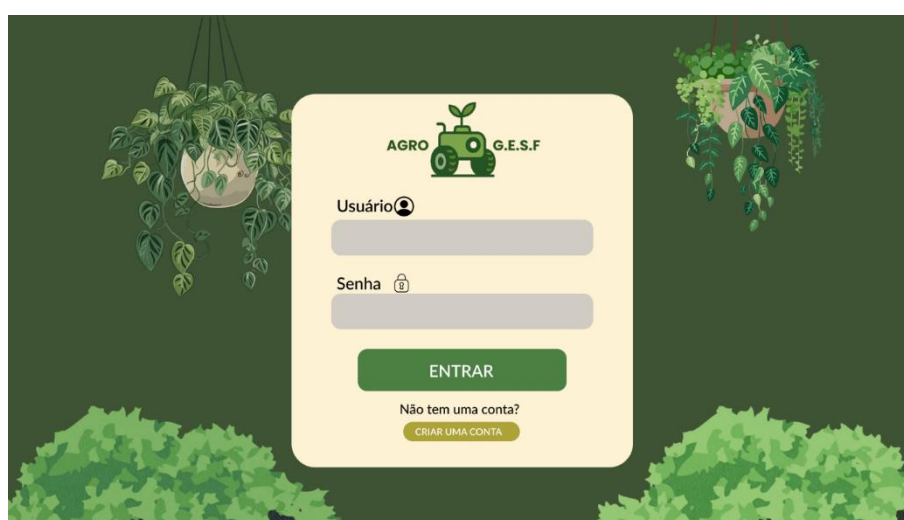
As seguintes figuras não são o produto, mas sim um protótipo utilizado para visualização. Maioria das figuras a seguir estarão tanto no produto para computador quanto para celular, e para simplificação, será mostrado majoritariamente as telas de computador, visto que ambas as telas serão praticamente idênticas.

Ao iniciar o sistema, a primeira tela que aparecerá será a de login, que o usuário deve preencher com as informações de sua conta antes de continuar a utilizar o software.

Caso o usuário não tenha uma conta cadastrada no sistema, ele deve clicar no botão “Criar uma conta”. Então, deverá preencher as informações necessárias na tela de cadastro, tomando cautela para que certos campos sejam verídicos e não sejam condizentes com outras contas no banco de dados do sistema.

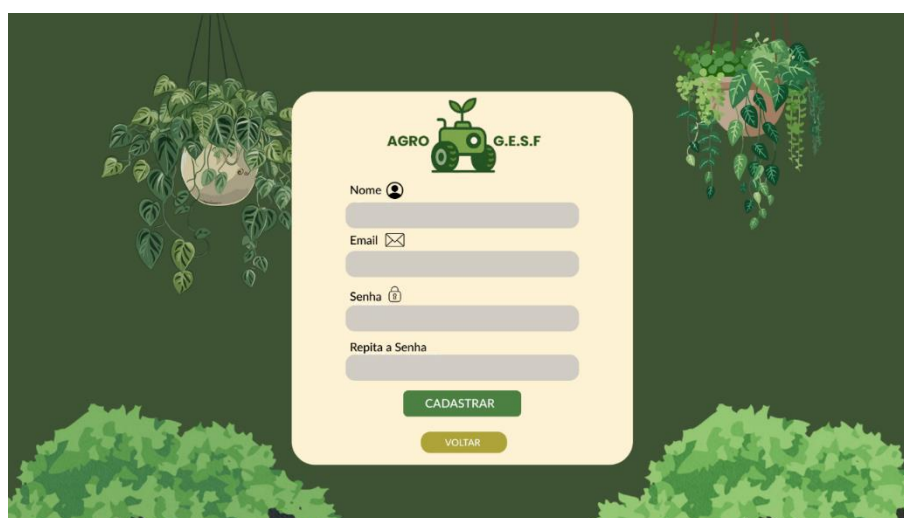
Ao se cadastrar, o usuário volta a tela de login, onde pode agora preencher com suas informações e entrar definitivamente no sistema.

Figura 6 – Tela de Login do sistema

A interface de login do sistema 'AGRO G.E.S.F.' apresenta um formulário centralizado em um fundo verde escuro decorado com plantas pendentes e arbustos. O formulário, em um cartão amarelo claro, contém o logo da empresa no topo, seguido por campos de entrada para 'Usuário' e 'Senha'. Abaixo dos campos, há um botão verde 'ENTRAR'. Na base do formulário, um link 'Não tem uma conta?' aponta para um botão amarelo 'CRIAR UMA CONTA'.

Fonte: Autores 2025.

Figura 7 – Tela de Cadastro do sistema

A interface de cadastro do sistema 'AGRO G.E.S.F.' mantém o mesmo design visual da tela de login. O formulário amarelo centralizado solicita o preenchimento dos campos 'Nome', 'Email', 'Senha' e 'Repita a Senha'. Na base, há dois botões: um verde 'CADASTRAR' e um amarelo 'VOLTAR'.

Fonte: Autores, 2025.

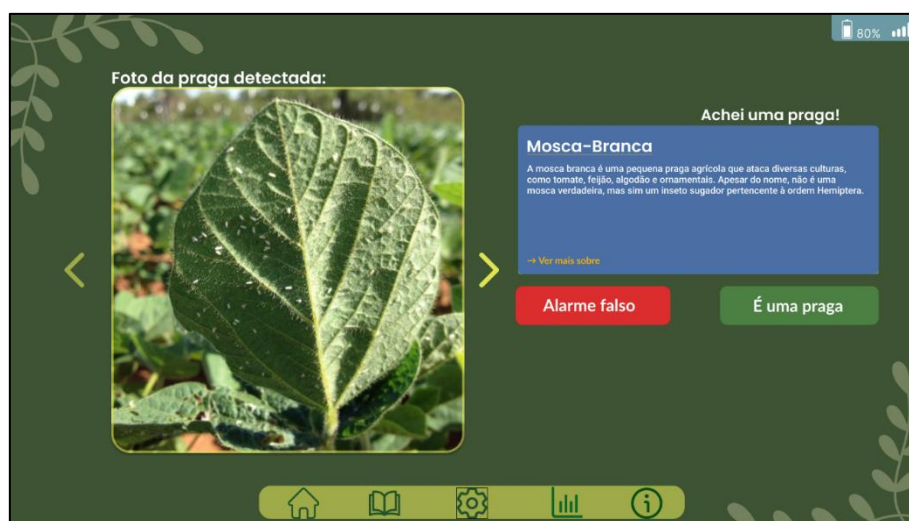
A primeira tela a ser evidenciada é a que chamamos de *Dashboard* (painel). Essa é a tela que mostra ao usuário as análises feitas pelo IoT, e inclui informações para que o usuário possa confirmar ou não que a praga foi marcada corretamente. Note que no canto superior direito há um indicador de bateria e de sinal do IoT, para manutenção correta dele.

Na parte inferior, há uma série de botões que o usuário pode clicar para seguir a outras telas do sistema.

O glossário de pragas, indicado com o ícone de livro, servirá como uma base para que o sistema IoT consiga classificar a praga encontrada precisamente, e poderá ser administrado pelo usuário. A princípio, o sistema já virá com uma bagagem de informações neste glossário, mas o usuário tem liberdade para colocar ou retirar pesquisas dele quando bem entender.

A tela de perfil, indicada pelo ícone de engrenagem, permite ao usuário administrar as informações de sua conta, colocar foto, evidenciar seu gênero etc. É importante notar que o sistema pode ter mais de uma conta cadastrada, o que significa que a tela de perfil apenas mostrará a conta que estiver atualmente logada no sistema.

Figura 8 – Dashboard do Sistema



Fonte: Autores, 2025.

Figura 9 – Tela do Glossário de Pragas



Fonte: Autores, 2025.

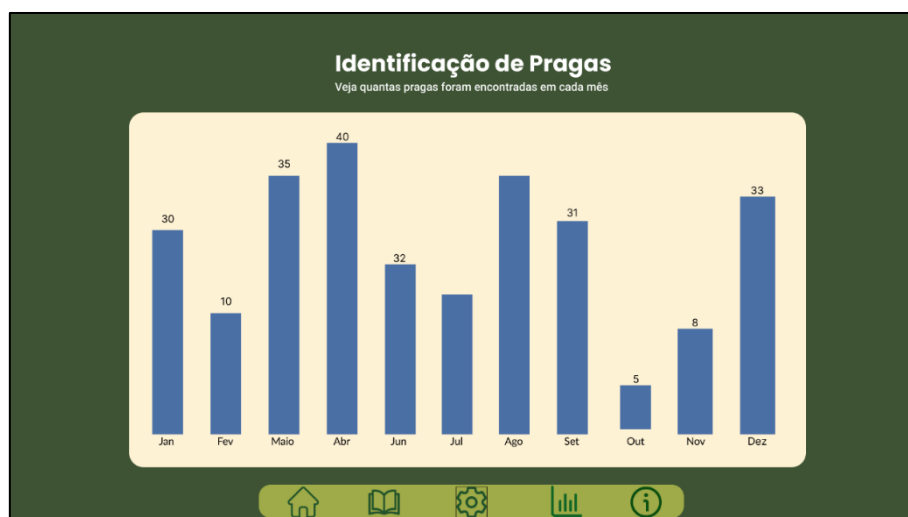
Figura 10 – Tela de Perfil



Fonte: Autores, 2025.

A tela do índice de pragas, indicada pelo ícone de gráfico mostra uma estatística automatizada que evidencia a quantidade de pragas detectadas pelo IoT, separados por mês.

Figura 11 – Tela de Índice de Pragas



Fonte: Autores 2025.

Por fim, a tela Sobre Nós, indicada pelo ícone de informação, mostrará a respeito dos criadores do projeto, com nome, imagem e descrição individuais.

Figura 12 – Tela Sobre Nós



Fonte: Autores, 2025.

Além disso, a aplicação para celular contará com uma tela extra, que mostrará um mapa de calor baseado nos locais onde foram identificadas pragas no cultivo. Essa tela é baseada em GPS (*Global Positioning System*) e é gerenciado pelo IoT.

5.2 Sistema IoT

A segunda parte do projeto é um sistema embarcado capaz de andar por um caminho especificado a procura de pragas nas plantações, utilizando de visão computacional e machine learning para enviar suas análises a aplicação. O nome AGRO-G.E.S.F. vem deste sistema, e é um acrônimo para sua funcionalidade (*Agronomical Ground-Unit for Environmental Surveillance Framework*). Ele é um carrinho em formato de jeep, de aproximadamente 155 x 100 x 35mm de caçamba, 60 x 30 x 15mm de módulo conversor.

O carrinho, feito para se locomover no terreno difícil rural, se locomove utilizando uma placa SBC Raspberry Pi, que pode ser considerado o “cérebro” do sistema. É ele quem desenvolve todas as funções que o carrinho pode fazer, se comunica com as outras peças, e faz a análise.

Além disso, o sistema possui um receptor de GPS e uma câmera, que serão utilizados para monitorar o cultivo a partir da visão computacional. Eles podem ser considerados os “olhos” do sistema, que irão ser utilizados pelo Raspberry Pi para as análises em campo.

Para que o carrinho funcione, está instalado uma bateria de lítio 2200mAh, com um *stepdown* XL4015, um módulo regulador de tensão com corrente de saída máxima de 5v. Isso permite que o sistema funcione por um longo tempo, diminuindo o risco de curto-circuito e permitindo manutenções mais espaçadas do carrinho. Não obstante, a comunicação da SBC com o software a nível usuário incluirá um indicador de porcentagem da bateria.

Figura 13 – Carrinho AGRO-G.E.S.F.



Fonte: Autores, 2025.

6. Considerações Finais

Observa-se que o projeto inclui a acessibilidade necessária aos pequenos agricultores do Brasil, com aplicações fáceis de utilizar e que não necessitam de grandes unidades de processamento, além de trazer um baixo custo comparado a outros projetos que têm função similar.

Embora o sistema seja inicialmente preparado para identificar poucas pragas e focado na região brasileira, não é difícil incrementar ele com pesquisas mais aprofundadas e monitoramento de outras pragas, mediante suficientes dados para a adaptação da máquina, e deve ser capaz de atuar em diversas áreas, visto que não há necessidade de sinal de rede e internet para que o sistema funcione completamente.

O sistema presente, porém, não é capaz de produzir estes resultados, tendo apenas funcionalidade em áreas específicas de atuação. Não é possível, por exemplo, identificar pragas em cultivos que não sejam de linha, pois o carrinho precisa de um caminho liso para se locomover.

Há também possibilidade de crescimento nas funcionalidades do sistema, aumentando sua atuação em outras áreas. O sistema, com maior desenvolvimento, poderia ser capaz de servir como pesquisador de campo, introduzindo uma automatização nas pesquisas de pragas na agricultura, além de ser modular o bastante para poder conter peças com maior desempenho e processamento.

Assim, espera-se que o projeto seja de grande ajuda a agricultura brasileira, ajudando a resolver o problema supracitado no artigo.

Referências

Android Developers. **Abordagem Kotlin ao Android**. Android Developers, 2025. Disponível em:

<https://developer.android.com/kotlin/first?hl=pt-br>. Acesso em: 07 novembro 2025.

BARELLI, Felipe. **Introdução à Visão Computacional**: Uma abordagem prática com Python e OpenCV. [S.l.]: Casa do Código, 2018. Disponível em:

[https://www.kufunda.net/publicdocs/Introducao%20a%20Visao%20Computacional%20-%20Uma%20abordagem%20pratica%20com%20Python%20e%20OpenCV%20\(Casa%20do%20Codigo\).pdf](https://www.kufunda.net/publicdocs/Introducao%20a%20Visao%20Computacional%20-%20Uma%20abordagem%20pratica%20com%20Python%20e%20OpenCV%20(Casa%20do%20Codigo).pdf).

Acesso em: 06 novembro 2025.

BARROS, Geraldo Sant'Ana de Camargo *et al.* **Efeito do não tratamento de pragas e doenças sobre preços ao consumidor e produtos da cadeia produtiva de soja (parte 3)**. Piracicaba, São Paulo: CEPEA, 2019. Disponível em:

https://www.cepea.org.br/upload/kceditor/files/Cepea_EstudoPragaseDoencas_Parte%203.pdf. Acesso em: 06 novembro 2025.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Rio de Janeiro: Elsevier, 2012.

BORGES, Eduardo Luiz. **Python para desenvolvedores**. Rio de Janeiro: Edição do Autor, 2009. Disponível em:

<https://www.kufunda.net/publicdocs/Python-Luiz%20Eduardo%20Borges.pdf>. Acesso em: 06 novembro 2025.

CARNEIRO, Fernando Ferreira; AUGUSTO, Lia Giraldo da Silva; RIGOTTO, Raquel Maria; FRIEDRICH, Karen; BÚRIGO, André Campos. **Dossiê ABRASCO: Um alerta sobre os impactos dos agrotóxicos na saúde**. Escola Politécnica de Saúde Joaquim Venâncio: Expressão Popular, 2015. Disponível em:

<https://abrasco.org.br/download/dossie-abrasco-um-alerta-sobre-os-impactos-dos-agrotoxicos-na-saude/?wpdmdl=79148&refresh=690d0994e28cb1762462100>. Acesso em: 06 novembro 2025.

DOBBIN, Ivan Diniz. **Desenvolvimento de software com interface gráfica em Raspberry Pi 4 para controle de uma máquina de cisalhamento**. 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software) – Universidade de Brasília, Faculdade UnB Gama, Brasília, 2022.

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária. *Pragas quarentenárias: perguntas e respostas*. 2024.

Disponível em: <https://www.embrapa.br/tema-pragas-quarentenarias/perguntas-e-respostas>. Acesso em: 26 maio 2025.

FERRÃO, Romário Gava; FONSECA, Aymbiré Francisco Almeida da; FERRÃO, Maria Amélia Gava; MUNER, Lúcio Herzog De (ed.). *Café Conilon*. 2. ed. atual. e ampl., 2. reimp. Vitória, ES: Incaper, 2017.

GUEDES, Gilleanes T.A. **UML 2: Uma abordagem prática**. São Paulo, SP: Novatec Editora, 2009. ISBN 978-85-7522-193-8. Disponível em: <https://s3.novatec.com.br/capitulos/capitulo-9788575221938.pdf>. Acesso em: 07 novembro 2025.

KOSHIYAMA, Adriano; ESCOVEDO, Tatiana. **Introdução a Data Science**: Algoritmos de Machine Learning e Métodos de Análise. [Rio de Janeiro, RJ]: Casa do Código, 2020.

LARMAN, Craig. **Utilizando UML e Padrões**. Porto Alegre: Bookman, 2000.

LAUXEN, Rafael; LOVATTO, Andressa; ROSA, Ronaldo Serpa da. **MicroscopePi: o desenvolvimento de um microscópio digital com Raspberry Pi 4**. In: SALÃO DE PESQUISA, EXTENSÃO E ENSINO DO IFRS, 8., 2023, Bento Gonçalves. Anais [...]. Bento Gonçalves: Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, 2023. p. [página inicial-final]. Disponível em:

<https://eventos.ifrs.edu.br/index.php/salao/8salao/paper/view/2456>. Acesso em: 12 ago. 2025.

LOPES, Carla Vanessa A.; ALBUQUERQUE, Guilherme Souza Cavalcanti de. **Agrotóxicos e seus impactos na saúde humana e ambiental: uma revisão sistemática**. Curitiba, Paraná, 2018. Disponível em:

<https://www.scielo.br/j/sdeb/a/bGBYRZvVVKMrV4yzqfwwKtP/?lang=pt>. Acesso em: 05 novembro 2025.

LUDEMIR, Teresa Bernarda. **Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências**.

Instituto de Estudos Avançados da Universidade de São Paulo: [s.n.], 2021. Disponível em:

<https://www.scielo.br/j/ea/a/wXBdv8yHBV9xHz8qG5RCgZd/?format=html&lang=pt>. Acesso em: 06 novembro 2025.

MAGRANI, Eduardo. **A Internet das Coisas**. Rio de Janeiro: FGV Editora, 2018. Disponível em:

<https://books.google.com.br/books?hl=pt-BR&lr=&id=qYtDwAAQBAJ&oi=fnd&pg=PA1&dq=MAGRANI,+2018+iot&ots=rhUoBAoa47&sig=8eDSelcal5fW8QGFB70dxloIf3w?v=onepage&q=MAGRANI%2C%202018%20iot&f=false>. Acesso em: 06 novembro 2025.

Faculdade de Tecnologia da Zona Leste Fatec Zona Leste	8º EnGeTec 2025
8º Encontro de Gestão e Tecnologia	ISSN 2675-4479
Organizações Ágeis em Tempos de Incerteza	Dezembro de 2025

OLIVEIRA, Sérgio de. **Internet das Coisas** com ESP8266, Arduino e Raspberry Pi. 1 ed. São Paulo, SP: Novatec Editora, 2017. Disponível em: [https://www.kufunda.net/publicdocs/Internet%20das%20Coisas%20com%20ESP8266,%20Arduino%20e%20Rasperry%20Pi%20\(S%C3%A9rgio%20de%20Oliveira\).pdf](https://www.kufunda.net/publicdocs/Internet%20das%20Coisas%20com%20ESP8266,%20Arduino%20e%20Rasperry%20Pi%20(S%C3%A9rgio%20de%20Oliveira).pdf). Acesso em: 06 novembro 2025.

PAIVA, Fábio Augusto P. *et al.* **Introdução a Python** com aplicações em sistemas operacionais. Natal: Editora IFRN, 2020. Disponível em: [https://memoria.ifrn.edu.br/bitstream/handle/1044/2090/EBOOK%20-%20INTRODU%C3%87%C3%83O%20A%20PYTHON%20\(EDITORIA%20IFRN\).pdf?sequence=1&isAllowed=y](https://memoria.ifrn.edu.br/bitstream/handle/1044/2090/EBOOK%20-%20INTRODU%C3%87%C3%83O%20A%20PYTHON%20(EDITORIA%20IFRN).pdf?sequence=1&isAllowed=y). Acesso em: 06 novembro 2025.

PAIXÃO, Gabriela et al. **Machine learning na medicina: revisão e aplicabilidade**. *Arquivos Brasileiros de Cardiologia*, [S. l.], [s. n.], [s. v.], [s. n.], [20--]. Disponível em: <https://www.scielo.br/j/abc/a/WMgVngCLbYfJrkmC65VFCKp/?format=pdf&lang=pt>. Acesso em: 11 ago. 2025.

REZENDE, Kassiano. **Kotlin com Android**: Crie aplicativos de maneira fácil e divertida. [S.l.]: Casa do Código, 2018. ISBN 978-85-94188-75-5.

RIGOTTO, Raquel Maria; VASCONCELOS, Dayse Paixão e; ROCHA, Mayara Melo. **Uso de agrotóxicos no Brasil e problemas para a saúde pública**. Ceará, Fortaleza, 2014. Disponível em: <https://www.scielo.org/article/csp/2014.v30n7/1360-1362/pt/#>. Acesso em: 05 novembro 2025.

SANTOS, P. R. S.; LOPES, A. F. G.; DIAS, W. R. A. Usando RaspBerry Pi para redução energética no IFRO. **Revista de Gestão e Secretariado**, [S. l.], v. 15, n. 4, p. e3596, 2024. DOI: 10.7769/gesec.v15i4.3596. Disponível em: <https://ojs.revistagesec.org.br/secretariado/article/view/3596>. Acesso em: 12 ago. 2025.

SANTOS, Rogério P. dos; BEKO, Marko; LEITHARDT, Valderi R. Q. **Modelo de Machine Learning em Tempo Real para Agricultura de Precisão**. In: ESCOLA REGIONAL DE ALTO DESEMPENHO DA REGIÃO SUL (ERAD-RS), 2022, Curitiba. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2022.

SENAR – SERVIÇO NACIONAL DE APRENDIZAGEM RURAL. *Café: controle de pragas, doenças e plantas daninhas*. Brasília: SENAR, 2017. 71 p. (Coleção SENAR, 190). ISBN 978-85-7664-151-3.

SILVA, Tarcízio. **VISÃO COMPUTACIONAL E RACISMO ALGORÍTMICO**: BRANQUITUDE E OPACIDADE NO APRENDIZADO DE MÁQUINA. In: *Revista da Associação Brasileira de Pesquisadores Negros (ABPN)*. [s.l.], fev. 2020. Disponível em: https://www.researchgate.net/publication/339514173_Visao_Computacional_e_Racismo_Algoritmico_Branquitude_e_Opacidade_no_Aprendizado_de_Maquina. Acesso em: 06 novembro 2025.

TOLEDO, Lucas Henrique B. **Desenvolvimento de Aplicações Kotlin Adaptáveis Utilizando a Linguagem Kotlin**. Ouro Preto: Universidade Federal de Ouro Preto, 2019. Disponível em: https://monografias.ufop.br/bitstream/35400000/3042/6/MONOGRAFIA_DesenvolvimentoAplica%C3%A7%C3%B5esAndroid.pdf. Acesso em: 07 novembro 2025.

TSUNODA, D.F.; MOREIRA, P.S.C.; GUIMARÃES, A.J.R. **Machine learning e revisão sistemática de literatura automatizada: uma revisão sistemática**. *Rev. Tecnol. Soc.*, Curitiba, v. 16, n. 45, p. 337-354, out./dez., 2020. Disponível em: <https://periodicos.utfpr.edu.br/rts/article/download/12119/7811>. Acesso em: 06 novembro 2025.

Os conteúdos expressos no trabalho, assim como os direitos autorais de figuras e dados, bem como sua revisão ortográfica e das normas são de inteira responsabilidade dos autores.

Os autores do trabalho declaram que durante a preparação do manuscrito não foram utilizadas ferramenta/serviço de Inteligência Artificial (IA), sendo todo o texto produzido e de responsabilidade dos autores.