Faculdade de Tecnologia da Zona Leste | Fatec Zona Leste	8º EnGeTec | 2025
8° 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

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

Enzo Costa Paz[1]
enzo.paz@etec.sp.gov.br

Felipe Vieira de Oliveira[1]
felipe.oliveira849@etec.sp.gov.br

Gustavo de Souza Morais[1]
gustavo.morais42@etec.sp.gov.br

Sakiri Moon Payao Cestari[1]
sakiri.cestari@etec.sp.gov.br

Jeferson Roberto de Lima[1]
jeferson.lima17@etec.sp.gov.br

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 farmers, the crops, 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] ETEC da Zona Leste

1. Introduction

The *Intelligent System for Monitoring and Prevention of Pests in Row Crops* is a device capable of traversing crop fields and identifying early signs of pests and diseases on leaves. Among the detectable cases are the Green-Yellow Chrysomelid Beetle (*Vaquinha-Verde-Amarela*), Late Blight (*Requeima*), and Early Blight (*Pinta-Preta*). Once signs are identified, the device transmits the information to a desktop dashboard via USB Type-C connection. Its goal is to detect early symptoms of crop diseases and pests, allowing small farmers to monitor their fields efficiently and understand the historical occurrence of these issues. The project aims to make precision agriculture accessible to small producers, promoting sustainability and reducing costs.

The main problem in small-scale agriculture lies in the late detection of pests and diseases, which causes severe crop losses and leads to the excessive use of pesticides. This overuse endangers workers, contaminates food, and harms the environment. Continuous exposure to chemicals can also increase pest resistance, reducing the effectiveness of pesticides in future applications. Between 2000 and 2012, the global pesticide market grew by 93%, while the Brazilian market increased by 190%. In 2008, Brazil became the world's largest consumer of pesticides and continues to occupy this position today (Carneiro et al., 2015). Rigotto, Vasconcelos, and Rocha (2014) affirm that pesticides represent a serious public health concern, since many people are exposed to them through agricultural work and food consumption. Lopes and Albuquerque (2018) further highlight that pesticide residues contaminate water sources, and even banned substances such as Hexachlorocyclohexane (HCH) are still found in rivers and reservoirs. These issues demonstrate that pesticide use not only generates health problems but also threatens ecosystems and biodiversity.

The problem is intensified by the lack of technical training among small farmers and by the cultural tendency to apply pesticides close to harvest time, in an attempt to eliminate remaining pests and increase yield. This practice, however, results in chemical waste, higher production costs, and contamination of both crops and soil. Thus, the lack of early detection tools and technical education perpetuates dependence on harmful agricultural methods.

The general objective of this research is to develop and implement an intelligent system capable of identifying early signs of pests and diseases in row crops using computer vision and machine learning. The system seeks to offer small farmers a low-cost technological solution for monitoring crops and managing their plantations sustainably. The specific objectives include designing a portable device to capture leaf images in real time, applying machine learning techniques in Python using the PyTorch framework (as discussed by Jeronimo), developing a visual dashboard for analysis, and evaluating the system's potential to reduce pesticide use through early identification.

The hypotheses guiding this research are that early detection systems based on artificial intelligence can reduce pesticide applications, that small farmers can integrate such systems with minimal training, and that precision monitoring can lower environmental impact and production costs while maintaining yield quality.

The methodology adopted is qualitative, relying on the analysis of scientific studies addressing the occurrence of pests, disease management, and the consequences of pesticide misuse. The theoretical basis is supported by Martins, Fontes, and Fornazier (2013) and Martins, Fornazier, and Fanton (2013), who examine pest management and agricultural impacts, and by Brito, Gomide, and Câmara, who discuss the effects of pesticide exposure on workers. Lakatos and Marconi (2017) provide the methodological foundation for qualitative research, while Menezes highlights Python's effectiveness in technological development.

Recognizing these issues, this article presents a question: what can be done to reduce the use of pesticides in cultivation, consequently reducing their impact?

2. Theoretical Foundation

In the current section, the main concepts and technologies for building the theoretical basis for the development of Agro-G.E.S.F. will be presented.

2.1. Pests in Plantations

Pests, as established by Embrapa (2024), are any species, race, or biotype of an animal or pathogenic agent that may cause any type of damage to plants or crops. Additionally, Senar (2017) complements that pests are usually characterized as mites, scale insects, and nematodes, while fungi, viruses, and bacteria are usually associated with diseases. Furthermore, Martins, Fontes, and Fornazier (2013 apud FERRÃO et al., 2017) inform that pests can occur sporadically or systematically. Those that occur sporadically and cause significant damage are known as secondary pests, while those that occur systematically every time the culture is implemented in the field and cause quantitative/qualitative damage are known as key pests.

Barros et al. (2019) point out that pests and diseases, in cases such as soybean cultivation, can cause a loss of up to 30% of national agricultural production if proper control is not exercised.

Agriculture, agricultural production methods, cultural and even social situations are inserted in different contexts historically, politically, and economically, so that Nogueira, Szwarchald, Damacena (2019) show that this tends to affect the life of rural residents, so that the combination of these variables tends to generate higher levels of exposure for such farmers to pesticides.

2.2. Internet of Things (IoT)

Oliveira (2017) states that the Internet of Things (IoT) is a concept that is becoming increasingly present in everyday life. In the last two decades, IoT has assisted human life in various areas through task automation, or in objects such as electronic badges and vehicles. IoT consists of any object capable of manipulating data on a network, aiming to assist and simplify complex actions (MAGRANI, 2018).

2.3. Raspberry Pi 4

As Dobbin (2022) indicates, the Raspberry Pi is a single-board computer (SBC), mounted on a printed circuit board (PCB). It is characterized by being able to perform the same functions as a common home computer, such as accessing the internet, browsing websites, and watching videos, with only the caveat of taking slightly longer to perform such functions due to its size.

As Santos, Lopes, and Dias (2024) point out, the Raspberry Pi 4 presents a medium level of processing due to its higher capacity processor in multitasking activities.

Additionally, Lauxen, Lovatto, and Rosa (2023) state that the Raspberry Pi 4 can be allocated within a system as the brain of the project, allowing image capture and control of elements connected to it.

2.4. Python

According to Luiz Eduardo Borges (2014), the Python language was created in 1990 by Guido van Rossum at the National Research Institute for Mathematics and Computer Science of the Netherlands (CWI), focusing on assisting professionals such as physicists and engineers.

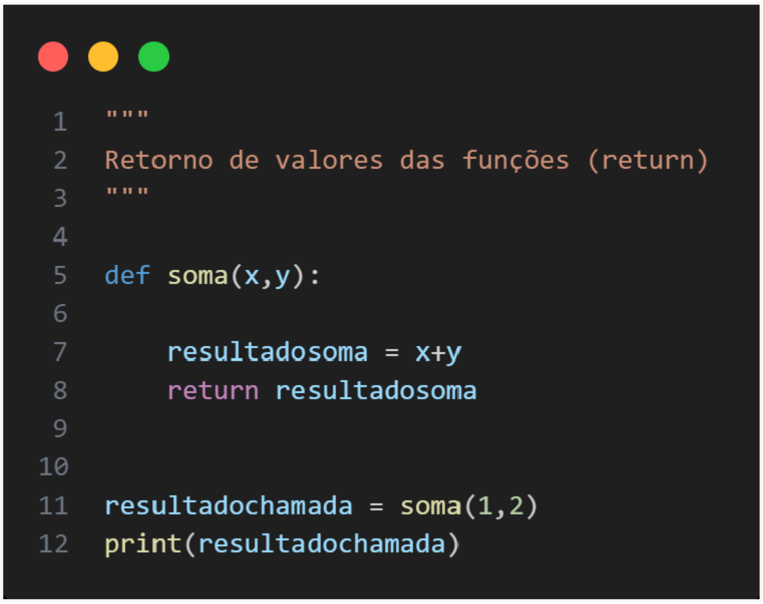
In accordance with Paiva et al. (2020), Python is a language with interesting characteristics and simple learning. Initially, it had the initial goal of being a more concise and less verbose code language. Supporting this simple programming, Python allows the use of various extension modules, which provide Python with a strong presence and power.

According to Menezes (2014), Python has been becoming a strong programming language in various areas of computing, such as artificial intelligence, databases, biotechnology, and even web applications.

In support, Python is a language known for having "batteries included," that is, it has batteries included, a programming language ready to be used. In addition to all this, Python is a programming language for obtaining results in a short time, due to its simple complexity and high density of modules for importation.

The code example below is a Python code that serves to add two numbers, which are inserted by the user themselves previously within the code itself. Below is a brief explanation of the code and its operation:

Figure 1 – Example of Python code for sum



```python
 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)
```

Source:Own authorship, 2025

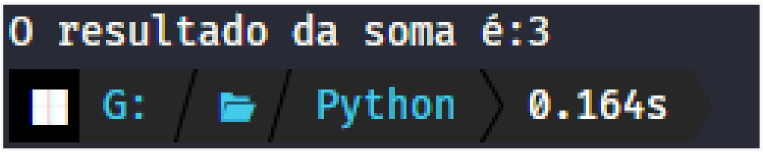
Lines 1 to 3: Creates a "Docstring" (type of string), which in the current context serves as a way to create comments; in this case, Python will go through these lines and ignore them;

Line 5: Creates the "sum" function, which serves to modularize the code and make it more practical, allowing the programmer not to have to keep repeating the same lines of code and calling the function only when necessary. In this application, the sum function receives two parameters, two items, and saves them in variables (variables are ways to save specific values within computer memory) named "x" and "y";

Lines 7 to 8: On line 7, a variable is created to receive the value of the sum of the two initially declared variables, then on line 8, the value of this variable is returned to the environment that called it;

Lines 11 to 12: A variable called "resultadochamada" is created, which is called the sum function, sending numbers 1 and 2 as parameters, while receiving the return value given by the function. On line 12, through the "print()" function, the value of the "resultadochamada" variable is displayed to the user through the integrated command prompt of the IDE.

Figure 2 – Result of the code in Python



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

Source: Own authorship, 2025

As a consequence of executing the previously illustrated code, in the above image there is the return of the Python code; in this case, numbers one and two were passed, and the return of the sum of such numbers was the number three.

2.5. Kotlin

Kotlin had its origin in 2011 with the first announcement about its existence; however, its first version was released in 2016, according to Resende (2018), aiming to be a more pragmatic programming language with a much simpler syntax to learn.

Kotlin brings with it an outstanding advantage over Java, which is the characteristic of being null-safe, as Toledo (2019) shows, the compiler is able to identify possible null objects before code execution, which avoids errors during code execution.

In 2019, as shown on the official Android Developers website (2025), through a Google action, Kotlin officially became the preferred language for developing mobile applications.

2.6. Machine Learning

Machine learning is according to Paixão et al. (2022), a subarea of computer science that works with the union of mathematical and statistical techniques applied to computational algorithms. Paraphrasing Escovedo and Koshiyama (2020), the concept of Machine Learning focuses on discovering patterns through this union of techniques, regardless of the degree of usefulness of such discoveries, in addition to using such patterns for automating certain tasks that could previously prove difficult for humans to perform.

When it comes to machine learning, it is proven through Ludemir (2021) that a large volume of data is necessary for the algorithm to learn automatically. Additionally, the accuracy of a trained model can vary according to the amount of data provided to the model, along with the quality of such data, as Tsunoda et al. (2020) say. Furthermore, it is noteworthy that trained models have stood out within the precision agriculture scenario, which are supported by large volumes of data and processing power, and are also widely used for detecting diseases and weeds in crops, predicting productivity, and managing soil health, as demonstrated by Santos, Beko, and Leithardt (2022).

2.7. Computer Vision

According to Barelli (2018), in 1982, Ballard and Brown defined computer vision in their work called Computer Vision as the science that studies and develops means to allow computers through sensors to see and extract information from what they see of the world. As Silva (2020) establishes, computer vision is commonly associated with the collection, analysis, and processing of visual data through computers for various purposes, ranging from face identification to object identification.

2.8. UML

Created in 1995 with the unification of three modeling languages, the Unified Modeling Language, or UML, is a theoretical language for modeling software that follows the object-oriented model. (Guedes, 2009).

According to authors Booch, Rumbaugh, and Jacobson (2012), UML is used in complex structure software to determine requirements, processes, and functions of simple or advanced level systems; furthermore, UML allows defining which classes will be created within the project and database schemas; UML is the skeleton of software projects.

According to Craig Larman (2005), UML became extremely important within software documentation due to the requirement of standards; these norms become important in development and create best practices for programmers.

3. Method

In this chapter, the method behind the project will be introduced, explaining how it came to be created.

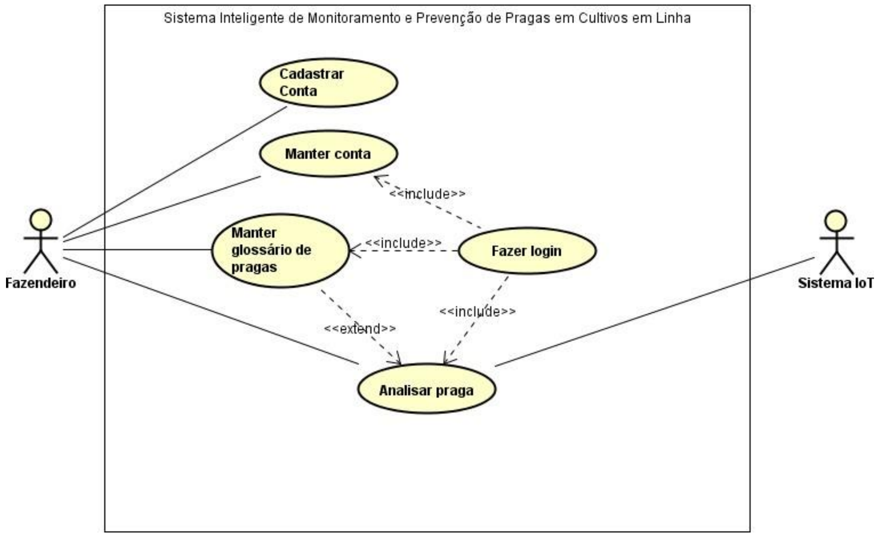
3.1. Diagramming

An important process in system creation is documentation, done with a standard called UML (Unified Modeling Language). This international standard involves creating diagrams for concrete visualization of the project, aiming to bring clear and consistent information to all those involved.

The beginning of project diagramming involves creating a Use Case Diagram, which shows the relationship of actions that an actor (user or adjacent system) can perform in the main system. The diagram exemplifies the functional requirements of the project, that is, all functions that the project needs to perform to be used correctly.

Figure 3 shows all functional requirements of the system, in addition to two actors: Farmer, the system user, and IoT System, which represents the pest monitoring machine.

Figure 3 – AGRO-G.E.S.F. Use Case Diagram.



Source: Own authorship, 2025.

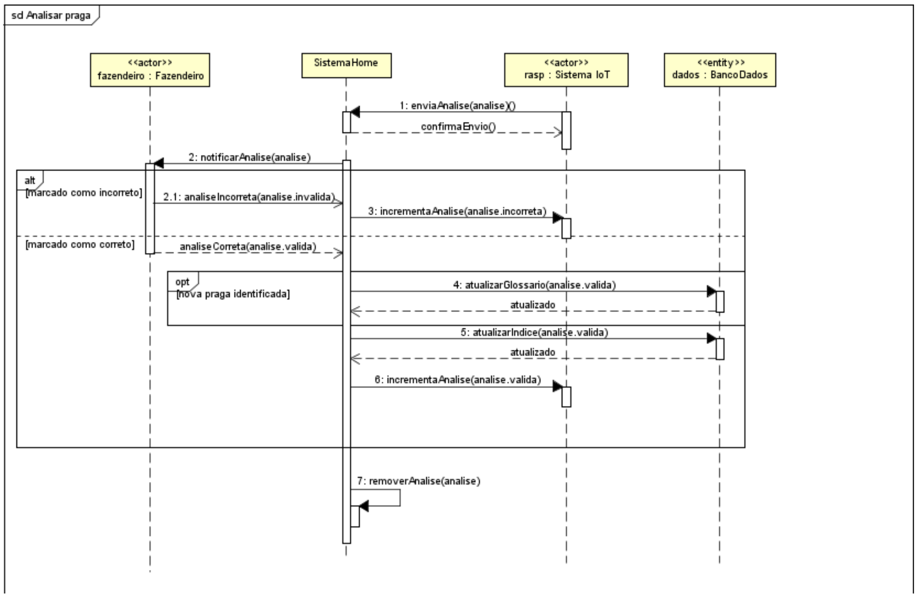
The Farmer actor, the system user, can interact with the system in various ways. For this, however, the first thing that must be done is registering an account in the system, for security and authority. From this, the user can edit, view, and delete their account, as well as maintain the system's internal pest glossary, a kind of pest encyclopedia.

For this, however, the user needs to log in, visualized in the diagram as an arrow with the <<include>> mark. This mark implies that the use case pointed to by the arrow necessarily includes the use case at the beginning of the arrow -- for example, maintain account "includes" Login. Note that almost all use cases involved with the Farmer include Login, except Register account, as it contradicts the case.

The main function of the system is Analyze pest, which involves both the Farmer and the IoT system. This use case both includes login and may involve Maintain pest glossary, exemplified by the arrow with the <<extend>> mark, which means that the use case pointed to by the arrow can (but does not necessarily need to) use the use case at the beginning of the arrow.

All use cases include either a Sequence Diagram or Activity Diagram, which show the step-by-step of these cases' functions. Observe the sequence diagram of the analyze pest use case, essential to the project:

Figure 4 – Sequence Diagram: Analyze pest


sd Analisar praga
<<actor>>
fazendeiro : Fazendeiro
SistemaHome
<<actor>>
rasp : Sistema IoT
<<entity>>
dados : BancoDados
1: enviaAnalise(analise)()
confirmaEnvio()
2: notificarAnalise(analise)
alt
[marcado como incorreto]
2.1: analiseIncorreta(analise.invalida)
3: incrementaAnalise(analise.incorreta)
[marcado como correto]
analiseCorreta(analise.valida)
opt
[nova praga identificada]
4: atualizarGlossario(analise.valida)
atualizado
5: atualizarIndice(analise.valida)
atualizado
6: incrementaAnalise(analise.valida)
7: removerAnalise(analise)

Source: Own authorship, 2025.

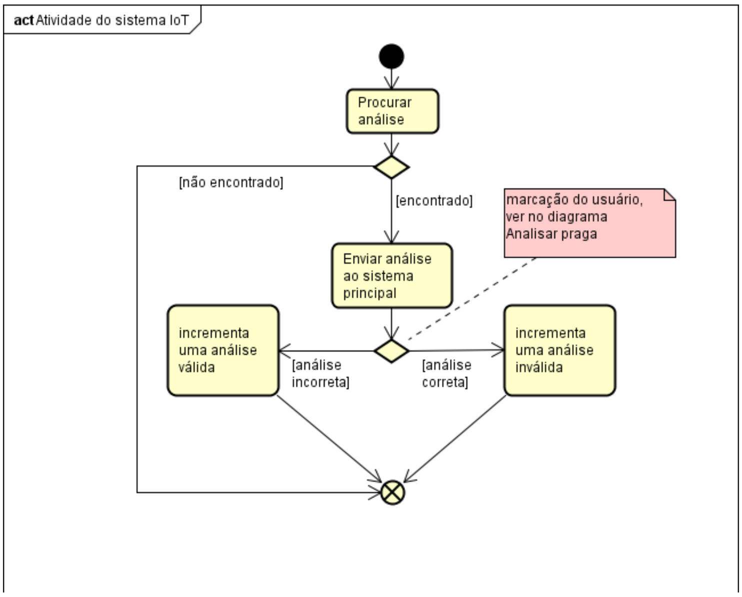
The diagram shows the message flow between parts of the system and actors, demonstrated by the boxes at the top. Each of them has a lifeline, which exemplifies the passage of time for each of the parts, being filled with messages within the system. For example, the first message in the system is from the IoT to the main system. It sends an analysis of a field pest to the system with the enviaAnalise() method, to which the system responds having confirmed the sending.

The "alt" box has the function of showing different cases in the diagram, separated by boxes that encompass the messages. It indicates exclusivity between one case and another, that is, if one case happens, the other necessarily does not happen. Additionally, the "opt" box indicates an optional sequence of events, that is, which is not necessarily mandatory for the requirement to function.

There is also the activity diagram, which shows events in a way to create a certain flowchart. This is not necessarily linear, being more used for systems where several things can be done simultaneously. Observe the diagram below, which describes the IoT system functions:

The diagram begins with the filled ball, which means the beginning of the activity. From it, the IoT goes to Search Analysis, which leads to a condition, indicated by a diamond. It illustrates that if the system does not find a pest, it should end the activity, indicated by the ball with the X in the middle. If found, it sends the analysis to the system, which returns with one more option. Now, the user must mark the analysis as correct or incorrect, which will cause the IoT system to update its computer vision with user data, according to Machine Learning ideas. Both options will also end the activity.

Figure 5 – IoT system Activity Diagram


act Atividade do sistema IoT
Procurar análise
[não encontrado]
[encontrado]
marcação do usuário, ver no diagrama Analisar praga
Enviar análise ao sistema principal
incrementa uma análise válida
[análise incorreta]
[análise correta]
incrementa uma análise inválida

Source: Own authorship, 2025.

4. Methodology

The project's research involves qualitative methodology, which, according to Lakatos and Marconi (2017), uses the participation of the target audience itself in experiments. This type of social research involves a more vigorous understanding of a specific phenomenon, through testimonies, interpretations and comparisons. Therefore, by not using statistics and concrete rules, it is more susceptible to changes, focusing on detailed and not generalized analyses. The project illustrated in the article used this methodology because its objective is to focus on helping small farmers, focusing more on the problems encountered in their daily lives. day and intending to create a more intimate connection between the project and the public, focusing on their personal issues

5. Results and Discussions

The project was developed during 2025, focusing on helping small farmers control pests and reduce pesticide use through precision agriculture. It was divided into two main systems: an application made for computer and mobile for the farmer, and an automated IoT system that will serve as the main agent in pest early warning.

5.1 User-level Application

The software created for the computer (developed in Python) and mobile (developed in Kotlin) aim to create the necessary connection between the user and the IoT system, since the IoT is completely automated and can be left alone in the field. This software aims to help the farmer monitor pest activity in their crop, create a graph of pest evidence, and manage communication between user and IoT.

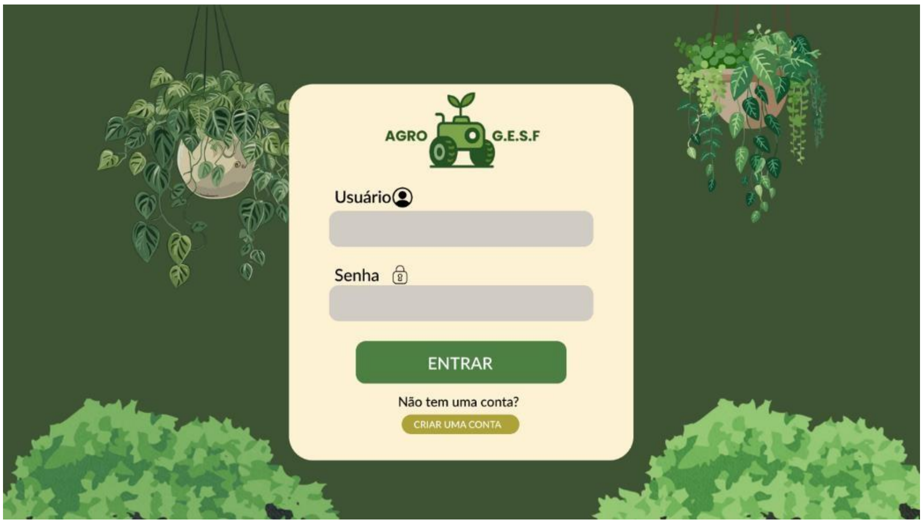
The following figures are not the product, but rather a prototype used for visualization. Most of the following figures will be in both the computer and mobile product, and for simplification, computer screens will be shown mostly, since both screens will be practically identical.

When starting the system, the first screen that will appear will be the login screen, which the user must fill in with their account information before continuing to use the software.

If the user does not have an account registered in the system, they must click the "Create an account" button. Then, they must fill in the necessary information on the registration screen, taking care that certain fields are truthful and not consistent with other accounts in the system database.

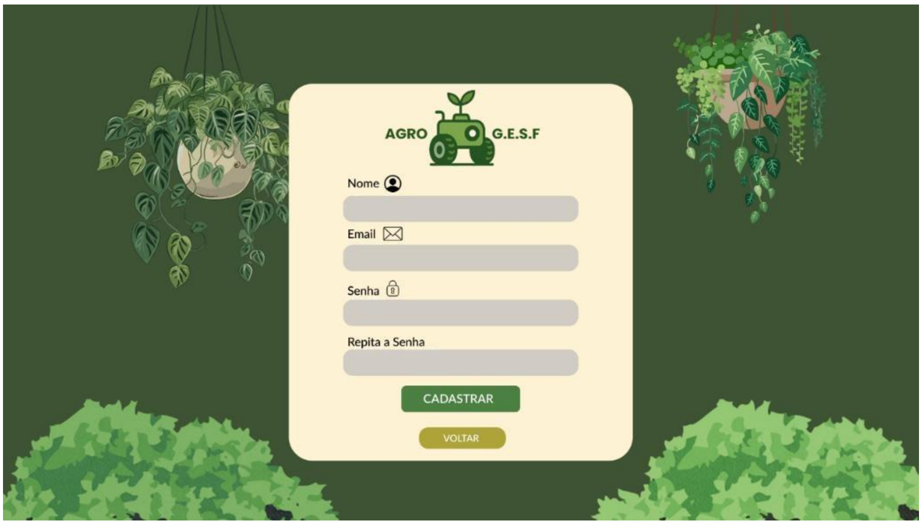
Upon registering, the user returns to the login screen, where they can now fill in with their information and definitively enter the system.

Figure 6 – System Login Screen



Source: Own authorship,2025.

Figure 7 – System Registration Scree



Source: Own authorship, 2025.

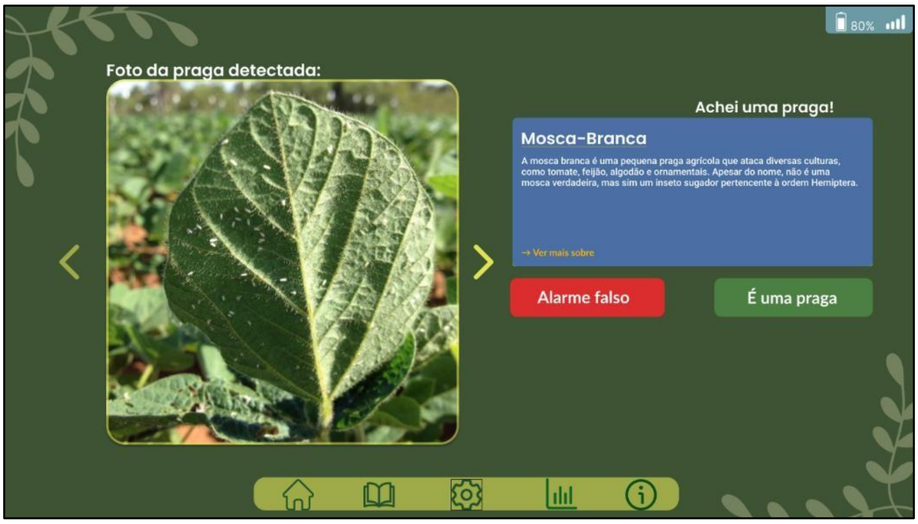
The first screen to be highlighted is what we call Dashboard (panel). This is the screen that shows the user the analyses made by the IoT, and includes information so that the user can confirm whether or not the pest was marked correctly. Note that in the upper right corner there is a battery and signal indicator for the IoT, for proper maintenance.

At the bottom, there are a series of buttons that the user can click to go to other system screens.

The pest glossary, indicated with the book icon, will serve as a basis for the IoT system to be able to classify the found pest precisely, and can be administered by the user. Initially, the system will already come with a baggage of information in this glossary, but the user has freedom to add or remove research from it whenever they want.

The profile screen, indicated by the gear icon, allows the user to manage their account information, add photo, show their gender, etc. It is important to note that the system can have more than one registered account, which means that the profile screen will only show the account that is currently logged into the system.

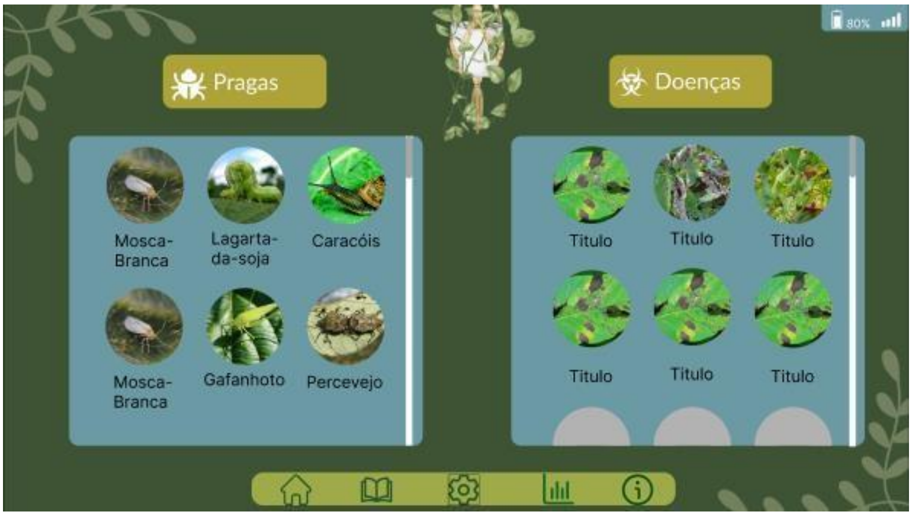
Figure 8 – System Dashboard



A mosca branca é uma pequena praga agrícola que ataca diversas culturas, como tomate, feijão, algodão e ornamentais. Apesar do nome, não é uma mosca verdadeira, mas sim um inseto sugador pertencente à ordem Hemiptera.

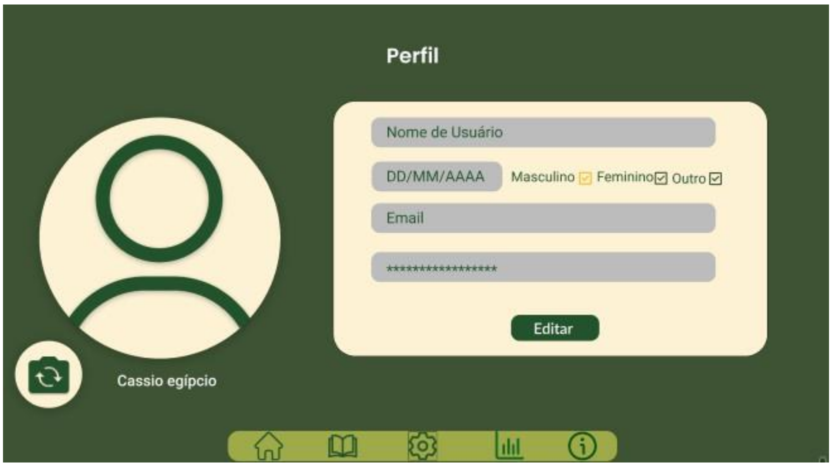
Source: Own authorship, 2025.

Figure 9 – Pest Glossary Screen



Source: Own authorship, 2025.

Figure 10 -Profile Screen



Perfil

Nome de Usuário

DD/MM/AAAA Masculino ☑ Feminino ☑ Outro ☑

Email

****

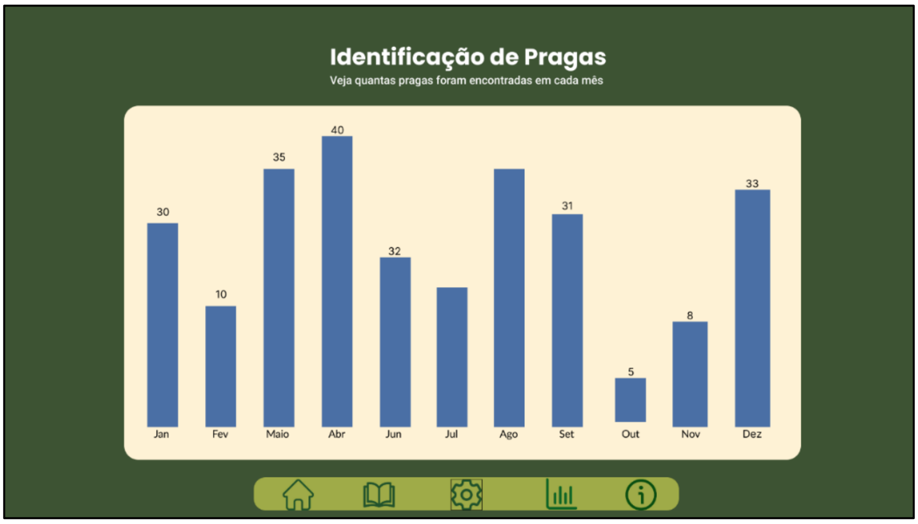
Editar

Cassio egípcio

Fonte: Autores, 2025.

The pest index screen, indicated by the graph icon, shows an automated statistic that highlights the number of pests detected by the IoT, separated by month.

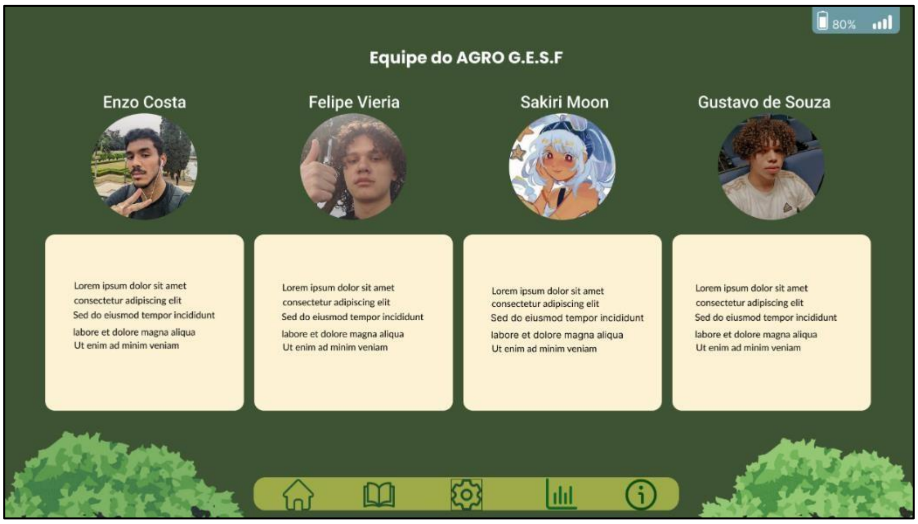
Figure 11 – Pest Index Screen



Source: Own authorship 2025.

Finally, the About Us screen, indicated by the information icon, will show the project creators, with individual name, image, and description.

Figure 12- About Us Screen



Equipe do AGRO G.E.S.F

Enzo Costa	Felipe Vieria	Sakiri Moon	Gustavo de Souza
Lorem ipsum dolor sit amet consectetur adipiscing elit Sed do eiusmod tempor incididunt labore et dolore magna aliqua Ut enim ad minim veniam	Lorem ipsum dolor sit amet consectetur adipiscing elit Sed do eiusmod tempor incididunt labore et dolore magna aliqua Ut enim ad minim veniam	Lorem ipsum dolor sit amet consectetur adipiscing elit Sed do eiusmod tempor incididunt labore et dolore magna aliqua Ut enim ad minim veniam	Lorem ipsum dolor sit amet consectetur adipiscing elit Sed do eiusmod tempor incididunt labore et dolore magna aliqua Ut enim ad minim veniam

Source: Own authorship, 2025.

Additionally, the mobile application will have an extra screen, which will show a heat map based on locations where pests were identified in the crop. This screen is based on GPS (Global Positioning System) and is managed by IoT.

5.2 Iot System

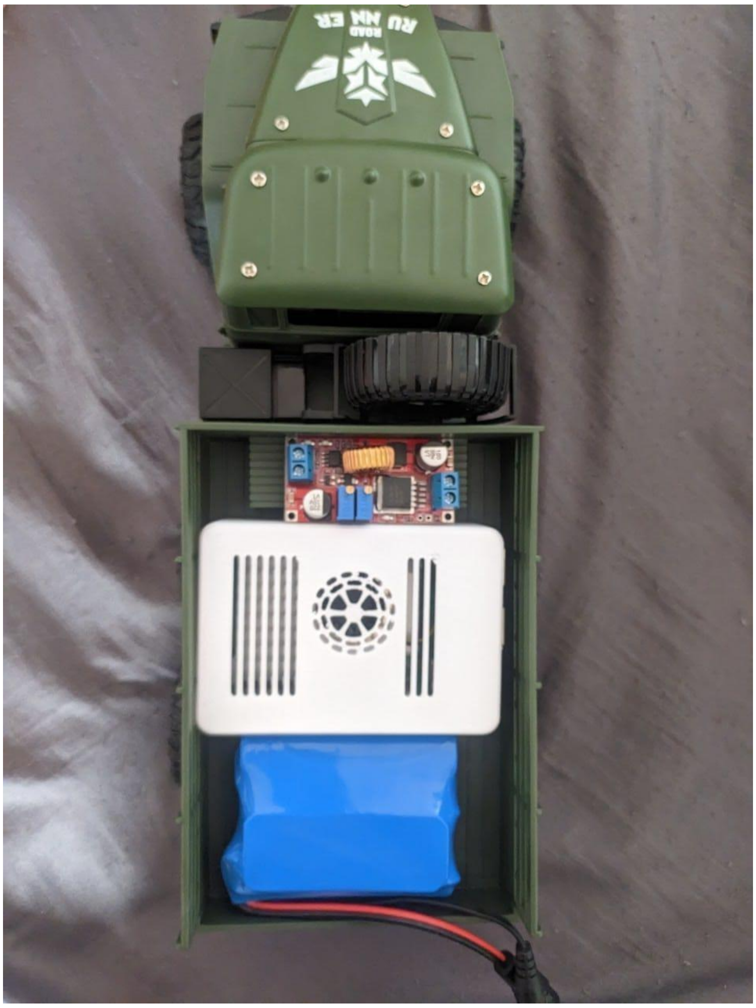
The second part of the project is an embedded system capable of walking along a specified path looking for pests in plantations, using computer vision and machine learning to send its analyses to the application. The name AGRO-G.E.S.F. comes from this system and is an acronym for its functionality (Agronomical Ground-Unit for Environmental Surveillance Framework). It is a jeep-format cart, approximately 155 x 100 x 35mm bed, 60 x 30 x 15mm converter module.

The cart, made to move in difficult rural terrain, moves using a Raspberry Pi SBC board, which can be considered the "brain" of the system. It is what develops all the functions that the cart can do, communicates with other parts, and performs the analysis.

Additionally, the system has a GPS receiver and a camera, which will be used to monitor the crop from computer vision. They can be considered the "eyes" of the system, which will be used by the Raspberry Pi for field analysis.

For the cart to work, a 2200mAh lithium battery is installed, with an XL4015 stepdown, a voltage regulator module with maximum output current of 5v. This allows the system to work for a long time, reducing the risk of short circuits and allowing more spaced maintenance of the cart. Nevertheless, the SBC communication with the user-level software will include a battery percentage indicator.

Figura 13 - AGRO-G.E.S.F. Trolley.



Source: Own authorship, 2025.

6 Final Considerations

It is observed that the project includes the necessary accessibility to small farmers in Brazil, with easy-to-use applications that do not require large processing units, in addition to bringing a low cost compared to other projects that have similar function.

Although the system is initially prepared to identify few pests and focused on the Brazilian region, it is not difficult to increment it with more in-depth research and monitoring of other pests, given sufficient data for machine adaptation, and should be able to operate in various areas, since there is no need for network signal and internet for the system to function completely.

The present system, however, is not capable of producing these results, having functionality only in specific areas of operation. It is not possible, for example, to identify pests in crops that are not row crops, as the cart needs a smooth path to move.

There is also possibility of growth in system functionalities, increasing its operation in other areas. The system, with further development, could be capable of serving as a field researcher, introducing automation in pest research in agriculture, in addition to being modular enough to be able to contain parts with greater performance and processing.

Thus, it is expected that the project will be of great help to Brazilian agriculture, helping to solve the problem mentioned above in the article.

References

Android Developers. **Kotlin Approach to Android**. Android Developers, 2025. Avalible at: https://developer.android.com/kotlin/first?hl=pt-br. Acessed on: November 07, 2025.

BARELLI, Felipe. **Introduction to Computer Vision: A Practical Approach with Python and OpenCV**. [*S.l*]: Casa do Código, 2018. Avalible on: https://www.kufunda.net/publicdocs/Introducao%20a%20Visao%20Computacional%20-%20Uma%20abordagem%20pratica%20com%20Python%20e%20OpenCV%20(Casa%20do%20Codigo).pdf. Acessed on: November 06, 2025.

BARROS, Geraldo Sant'Ana de Camargo *et al*. **The effect of not treating pests and diseases on consumer prices and products in the soybean production (part 3)**. Piracicaba, São Paulo: CEPEA, 2019. Avalible at: https://www.cepea.org.br/upload/kceditor/files/Cepea_EstudoPragaseDoencas_Parte%203.pdf. Acessed : November 06, 2025.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML User Guide.** Rio de Janeiro: Elsevier, 2012.

BORGES, Eduardo Luiz. **Python for Developers**. Rio de Janeiro: Edição do Autor, 2009.Avalible at: https://www.kufunda.net/publicdocs/Python-Luiz%20eduardo%20Borges.pdf. Acessed on: November 06, 2025.

CARNEIRO, Fernando Ferreira; AUGUSTO, Lia Giraldo da Silva; RIGOTTO, Raquel Maria; FRIEDRICH, Karen; BÚRIGO, André Campos. **ABRASCO Dossier: A Warning about the impacts of pestcides on health.** Joaquim Venâncio School of Health Sciences: Popular Expression, 2015. Avalible at: https://abrasco.org.br/download/dossie-abrasco-um-alerta-sobre-os-impactos-dos-agrotoxicos-na-saude/?wpdmdl=79148&refresh=690d0994e28cb1762462100. Acessed on: November 06, 2025.

DOBBIN, Ivan Diniz.**Development of Software with a graphical interface on Raspberry PI 4 for controlling a shearing machine**. 2022. Final Year Project(Bachelor's Degree in Engineering Software) – University of Brasília, Faculdade UnB Gama, Brasília, 2022.

EMBRAPA –Brazilian Agricultural Research Corporation. *Quarantine pests: questions and answers*. 2024. Avalible at: https://www.embrapa.br/tema-pragas-quarentenarias/perguntas-e-respostas. Acessed on: May 26, 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:** A pratical approach. São Paulo, SP: Novatec Editora, 2009. ISBN 978-85- 7522-193-8 .Avalible at: https://s3.novatec.com.br/capitulos/capitulo-9788575221938.pdf. Acessed on: November 07, 2025.

KOSHIYAMA, Adriano; ESCOVEDO, Tatiana. **Introduction to Data Science**: Machine Learning Algorithms and Analysis Methods. [Rio de Janeiro, RJ]: Casa do Código, 2020.

LARMAN, Craig. **Using UML and Patterns** . Porto Alegre: Bookman, 2000.

LAUXEN, Rafael; LOVATTO, Andressa; ROSA, Ronaldo Serpa da**. MicroscopePi: the development of a digital microscope with Raspberry Pi 4**. In: SALÃO DE PESQUISA, EXTENSÃO E ENSINO DO IFRS, 8., 2023, Bento Gonçalves. Anais [...]. Bento Gonçalves: Federal Institute of Education, Science and Technology of Rio Grande do Sul, 2023. p. [página inicial-final]. Avalible at: https://eventos.ifrs.edu.br/index.php/salao/8salao/paper/view/2456. Acessed on: August 12, 2025.

LOPES, Carla Vanessa A.; ALBUQUERQUE, Guilherme Souza Cavalcanti de. **Pesticides and their impacts on human and environmental health**. Curitiba, Paraná, 2018.Avelible at: https://www.scielo.br/j/sdeb/a/bGBYRZvVVKMrV4yzqfwwKtP/?lang=pt. Acessed on: November 05, 2025.

LUDEMIR, Teresa Bernarda. **Artifical Intelligence and Machine Learning: Current State and Trends**. Institute of Advanced Studies of the Universsity of São Paulo: [*s.n.*], 2021 .Avalible at: https://www.scielo.br/j/ea/a/wXBdv8yHBV9xHz8qG5RCgZd/?format=html&lang=pt. Acessed on: November 06, 2025.

MAGRANI, Eduardo.**The Internt of Things**. Rio de Janeiro: FGV Editora, 2018. Avalible at: https://books.google.com.br/books?hl=pt-BR&lr=&id=qYtlDwAAQBAJ&oi=fnd&pg=PA1&dq=MAGRANI,+2018+iot&ots=rhUoBAoa47&sig=8eDSelcal5fW8QGFB70dxloIf3w#v=onepage&q=MAGRANI%2C%202018%20iot&f=false. Acessed on: November 06, 2025.

OLIVEIRA, Sérgio. **Internet of Things with ESP8266, Arduino and Raspberry Pi**. 1 ed. São Paulo, SP: Novatec Editora, 2017. Avalible at: https://www.kufunda.net/publicdocs/Internet%20das%20Coisas%20com%20ESP8266,%20Arduino%20e%20Raspberry%20Pi%20(S%C3%A9rgio%20de%20Oliveira).pdf. Acessed on :November 06, 2025.

PAIVA, Fábio Augusto P. *et al*. **Introdução a Python** with applications in operating systems. Natal: Editora IFRN, 2020. Avalible at: https://memoria.ifrn.edu.br/bitstream/handle/1044/2090/EBOOK%20-%20INTRODU%C3%87%C3%83O%20A%20PYTHON%20(EDITORA%20IFRN).pdf?sequence=1&isAllowed=y. Acessed on: November 06, 2025.

PAIXÃO, Gabriela et al. **Machine Learning in medicine: review and applicabilty.***Brazilian Archives of Cardiology*, [S. l.], [s. n.], [s. v.], [s. n.], [20--].Avalible at: https://www.scielo.br/j/abc/a/WMgVngCLbYfJrkmC65VFCkp/?format=pdf&lang=pt. Acessed on: August 11, 2025.

REZENDE, Kassiano. **Kotlin with Android**: Create applications easily and enjoyably. [*S.l.*]: Casa do Código, 2018. ISBN 978-85-94188-75-5.

RIGOTTO, Raquel Maria; VASCONCELOS, Dayse Paixão e; ROCHA, Mayara Melo. **Use of pestcides in Brazil and public health issues**. Ceará, Fortaleza, 2014. Avalible at: https://www.scielosp.org/article/csp/2014.v30n7/1360-1362/pt/#. Acessed on: November 05, 2025.

SANTOS, P. R. S.; LOPES, A. F. G.; DIAS, W. R. A. Using Raspberry Pi for energy reduction ar IFRO.**Journal of Management and Secretariat** , *[S. l.]*, v. 15, n. 4, p. e3596, 2024. DOI: 10.7769/gesec.v15i4.3596. Avalible at: https://ojs.revistagesec.org.br/secretariado/article/view/3596. Acessed on: August 12, 2025.

SANTOS, Rogério P. dos; BEKO, Marko; LEITHARDT, Valderi R. Q. **Real-Time Machine Learning Model for Precison Agriculture** . *In*: REGIONAL HIGH-PERFOMANCE SCHOOL OF THE SOUTH REGION (ERAD-RS), 2022, Curitiba. **Anais** [...]. Porto Alegre: Brazilian Computer Society, 2022.

SENAR – NATIONAL RURAL LEARNING SERVICE. *Coffe: pests, disease, and weed control*. Brasília: SENAR, 2017. 71 p. (Coleção SENAR, 190). ISBN 978-85-7664-151-3.

SILVA, Tarcízio. **COMPUTER VISON AND ALGORITHMIC RACISM**: WHITENESS AND OPACITY IN MACHINE LEARNING . *In: Jounal of the Brazilian Association on Black Researchers (ABPN)*. [*s.l.*], fev. 2020. Avalible at: https://www.researchgate.net/publication/339514173_Visao_Computacional_e_Racismo_Algoritmico_Branquitude_e_Opacidade_no_Aprendizado_de_Maquina. Acessed on: November 06, 2025.

TOLEDO, Lucas Henrique B. **Development of Adaptable Kotlin Applications Using the Kotlin Langauge**. Ouro Preto: Federal Universty of Ouro Preto, 2019. Avalible at: https://monografias.ufop.br/bitstream/35400000/3042/6/MONOGRAFIA_DesenvolvimentoAplica%C3%A7%C3%B5esAndroid.pdf. Acessed on: November 07, 2025.

TSUNODA, D.F.; MOREIRA, P.S.C.; GUIMARÃES, A.J.R. **Machine Learning and automated systematc literature review: a systematic review.** Rev. Tecnol. Soc., Curitiba, v. 16, n. 45, p. 337-354, out./dez., 2020. Avalible at: https://periodicos.utfpr.edu.br/rts/article/download/12119/7811. Acessed on: November 06, 2025.

“The contents expressed in the work, as well as the copyright of figures and data, as well as their spelling check, and standards are the sole responsibility of the authors.

The authors of the work declare that during the preparation of the manuscript, no Artificial Intelligence (AI) tool/service was used, with the entire text being produced and being the responsibility of the authors.”