Federal University of Technology of Paraná – UTFPR
Academic Department of Electronics – DAELN
Academic Department of Informatics – DAINF
Computer Engineering
Integration Workshop 3 (EEX23) – S71 – 2024/2

Technical Report
PrimeCorn: an automatic corn seed sorting machine

Anna C. R. Martins – annamartins.2002@alunos.utfpr.edu.br
Eduardo V. Ximenes – eduardoximenes@alunos.utfpr.edu.br
Gabriel C. Bach – gabrielbach@alunos.utfpr.edu.br
Gustavo A. Domingues – gustavoadame@alunos.utfpr.edu.br
Suzana P. B. da Costa – suzanapachelle@alunos.utfpr.edu.br

February 2025

Abstract

In the context of agriculture, low-quality seeds for planting can lead to significant losses to producers, lowering the productivity and increasing the probability of crop diseases. Many technologies able to sort through the seeds and select the adequate seeds for planting are very expensive, forcing small-scale producers to recur to a manual separation methods. Considering this factor, and that corn is one of the most important crops in Brazil and the world, the PrimeCorn project was elaborated to classify and separate regular from irregular corn seeds (of the popcorn type) using image processing and machine learning. The machine receives a batch of seeds from the user, processes each seed, classify them in regular or irregular, and redirects them to one of two containers. A mobile app, connected to the machine via Bluetooth, is also included in the project, to register the analysis for further consultation. This way, PrimeCorn offers to the user the practicality of an automatic seed sorting method, facilitating the planting of the seeds.

1 Introduction

Agriculture is one of the main pillars of the Brazilian economy, being responsible for moving billions of reais every year, reflecting heavily on the country's GDP with a contribution of around 20%[1]. However, the agricultural field faces many challenges, such as the cultivation of defective seeds, which increases the risk of crop disease and decreases productivity.

Therefore, the separation of defective seeds from those that are good enough for planting is an important process. In the case of small farmers — who are responsible for around 25% of the country's agricultural production[1] —, this process relies on manual methods, which not only slows down operations but also increases the risk of human error, potentially compromising the overall quality of the selection process.

In this context, the PrimeCorn project aims to address this challenge faced by the producers, more specifically the small-scale ones. The type of seed used in the project is a common type of corn seed: popcorn, which is a variety that is heavily produced in Brazil (around 300 thousand tons[2]). PrimeCorn was designed to enhance the quality of the farmers' work through a machine designed to automatically sort regular from irregular seeds.

In essence, the project consists in three parts: an embedded computer vision system that, using a machine learning algorithm, classify the corn seeds; a mechanical system that allows the seeds to be classified and separated; and a mobile application where the user can see information about the processes of the machine.

2 Overview and Project Specification

2.1 Overview

PrimeCorn consists of a machine that sorts through an amount of corn seeds provided by the user, detecting and classifying them into two categories: regular and irregular. The two types of seeds are, then, dropped into their designated slots after they are processed. The machine also collects and stores data from each batch, and those results can be sent to the user's phone via Bluetooth and accessed using a mobile app. Figure 1 indicates a simplified view of the project.

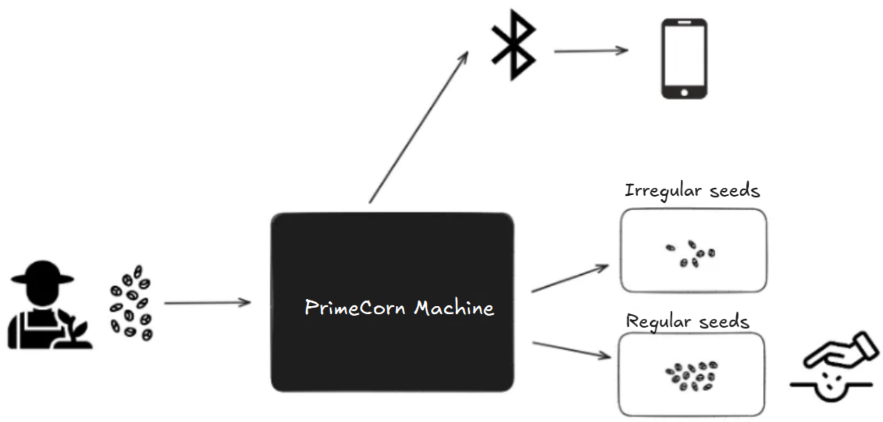

Figure 1: Initial view of the project.

The PrimeCorn machine mechanically separate seeds into two categories — regular and irregular - while detailed data on specific irregular subtypes can be accessed through the mobile app for further analysis. This functionality enables the producer to understand the specific issues affecting the discarded seeds, supporting informed decisions on corrective actions. Designed for practical daily use, this system primarily serves the farmer's need for efficient seed sorting, yet also provides valuable insights into field conditions whenever the farmer wishes to explore the root causes of seed irregularities.

Figure 2 displays an example of the seed classification criteria used by PrimeCorn.

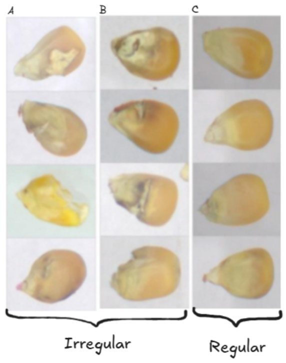

Figure 2: Seeds types: A - Broken, B - Silkcut, C - Regular

To achieve its objectives, PrimeCorn was developed using the modules displayed in the block diagram in Figure 3.

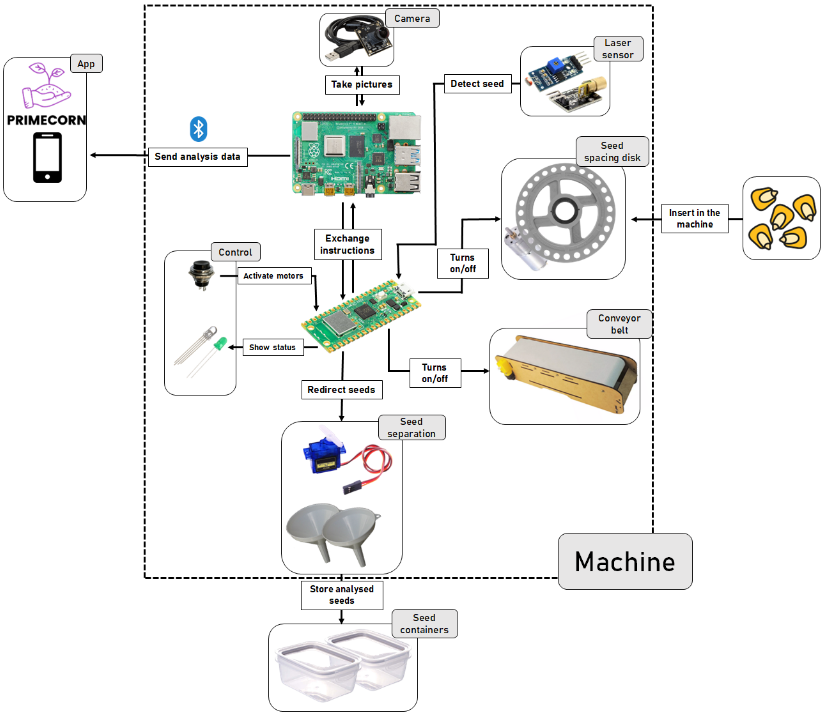

Figure 3: PrimeCorn block diagram.

2.2 Project Specification

The project specifications were defined based on the functionalities it should provide and resources at the team's disposal, leading to a list of functional and non-functional requirements, as well as anti-requirements. More details of the project specifications can be found in the team's blog [3].

Table 1 displays the list of functional requirements specified at the beginning of the project.

Table 1: Functional Requirements.

ID	Requirement
FR01	The seed sorter must include an RGB LED to indicate the PrimeCorn machine's states, with the following color codes: PURPLE for "Ready", RED for "Idle", GREEN for "Processing", and BLUE for "Saving".
FR02	The Seed Sorter must shut off the motors automatically after 15 seconds of inactivity, when no seeds are detected on the transportation mechanism.

Table 1 continued from previous page

ID	Requirement
FR03	The seed sorter must be able to receive up to 100g of seeds at the start of the process.
FR04	The seed sorter must include a receptacle for users to load seeds to the machine, with a capacity of up to 100g of corn seeds.
FR05	The seed sorter must process at least 20 seeds per minute.
FR06	The seed sorter is designed to classify seeds into four categories: Pure, Broken, Discolored, and Silkcut. Seeds identified as Broken, Discolored or Silkcut will be directed to a "Irregular" seeds container, while Pure seeds will be sorted into a "Regular" seeds container.
FR07	The seed sorter system should be configured with a classification confidence threshold of 50 per cent. If the confidence level for an item passing along the conveyor belt falls below this threshold, the item will automatically be redirected to the irregular seeds container.
FR08	The seed sorter must have another LED that is on when the machine is paired with the mobile app and off when it is not paired.
FR09	The seed sorter must have a rotating mechanism that spaces the seeds up to 10 cm apart on the transport system.
FR10	The rotating mechanism of the seed sorter must be enclosed with walls to prevent seeds from falling off during operation.
FR11	The seed sorter must transport the seeds from the beginning to the end of the entire processing sequence.
FR12	The seed sorter transportation mechanism must not allow the seeds to fall from the sides.
FR13	The seed sorter transportation mechanism must move at a constant speed of at least 2 cm/s.
FR14	The seed sorter must capture images of the seeds at a maximum size of 1920x1080 every 3 seconds or less.
FR15	The seed sorter must be equipped with a sensor to detect the transported seeds, which trigger the camera to capture images at the appropriate moment.
FR16	The seed sorter must be equipped with a funnel connected to tubes at the end of the transportation system, allowing gravity to direct the seeds into their respective containers.
FR17	The seed sorter must distinguish regular from irregular seeds with at least 70% of accuracy.
FR18	The classification error of each class of irregular seed being classified as pure must not exceed 25%.
FR19	The F1 Score for the 'pure' class must be greater than 75%.

Table 1 continued from previous page

ID	Requirement
FR20	Each seed sorter container must have a minimum volume of 200 cm^3.
FR21	The seed sorter must accurately count the number of processed seeds based on their classification, allowing for an error margin of up to 10%.
FR22	The seed sorter must transmit the seed count information from each stored operation via Bluetooth to a mobile application and then clear the data locally on the machine.
FR23	The mobile app must receive the estimated count of the seeds to later calculate the percentages to the user.
FR24	The mobile app must have a page responsible to show the past processes and the percentage of regular and irregular seeds of each process.
FR25	The mobile app must have a page that allow the user to see all types of irregular classified seeds in a chart.
FR26	The mobile app must have a page that allow the user to syncronize to the PrimeCorn machine via bluetooth.
FR27	The mobile app must store the data locally on the device.
FR28	The mobile app must be capable of storing up to 100 individual processes.

3 Development

The following section details the development of the project.

3.1 Mechanics

The first part of the development was the design and assembly of the project's mechanical structure. It started with the creation of the CAD project using *TinkerCAD* and *Fusion 360*, where the team was able to get an overview of the mechanical part, as shown in Figure 4.

3.1.1 Rotating Disk

For the rotating disc part, *Fusion 360* allowed us to design the seed tank of the disc so that the seeds would enter correctly in the holes and the axle responsible for the connection between the rotating disc and the DC motor. After that, the team was able to 3D print the parts, as shown in Figure 5.

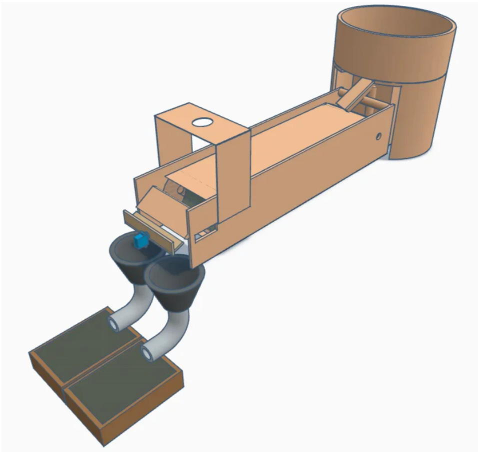

Figure 4: Mechanic overview.

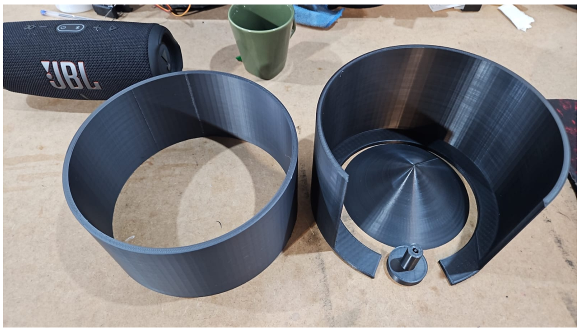

Figure 5: 3D printed parts.

3.1.2 Conveyor Belt

The conveyor belt part is composed of an assembled conveyor belt, walls on the sides to avoid the seeds from falling and a support for the camera above the detection area. The conveyor belt, whose parts are pre-made, were bought by the team and assembled following this manual. The walls and the camera support were made using MDF sheets, as shown in Figure 6.

3.1.3 Seed Separation Mechanism

The seed separation mechanism, in Figure 7, was designed to sort seeds using a servo motor-driven platform that redirects seeds in to designated funnels. The platform, controlled by the servo motor mounted on a sturdy support, directs seeds into one of two funnels based on the desired sorting criteria. Each funnel is connected to a tube that channels the seeds to external containers, ensuring organized storage.

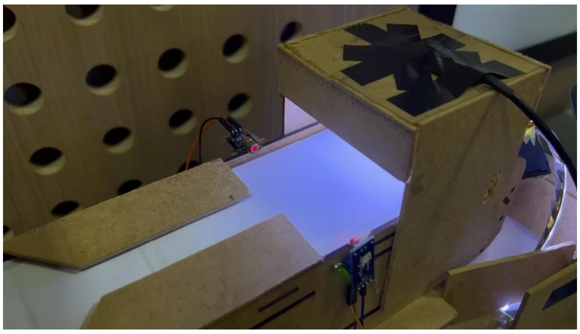

Figure 6: Conveyor integrated.

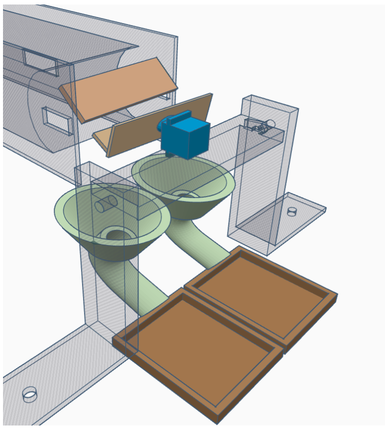

Figure 7: 3D overview of the servo part.

3.1.4 Box of the project

The project box was designed using *Fusion 360*, and from there we were able to pass the project on to a locksmith who was responsible for cutting the MDF sheets, as shown in Figures 8, 9 and 10.

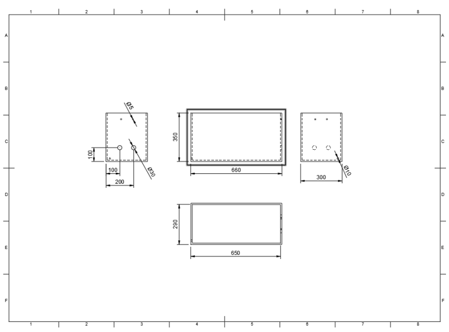

Figure 8: Side view

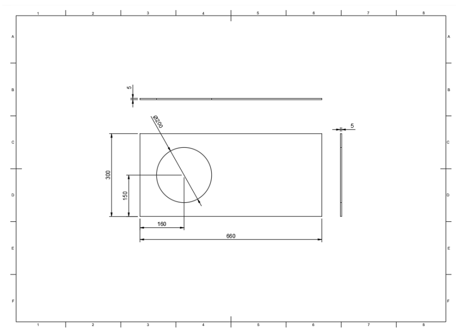

Figure 9: Top view

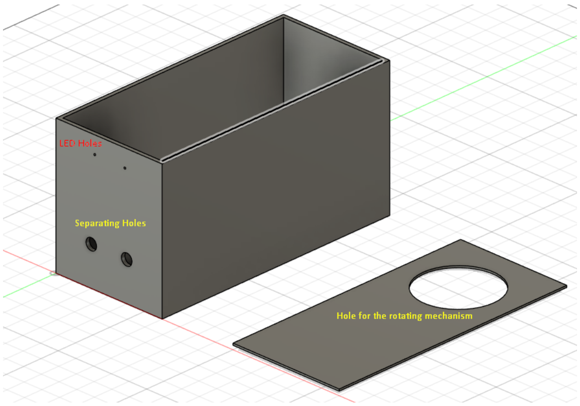

Figure 10: 3D design view.

3.2 Electronic/Hardware

The circuitry of the project was designed on a schematic and soldered in a universal Printed Circuit Board. The components that could not be soldered to the board were connected to the rest of the circuit using pin headers.

Figure 11 shows the schematic for the electronic circuit used in the project.

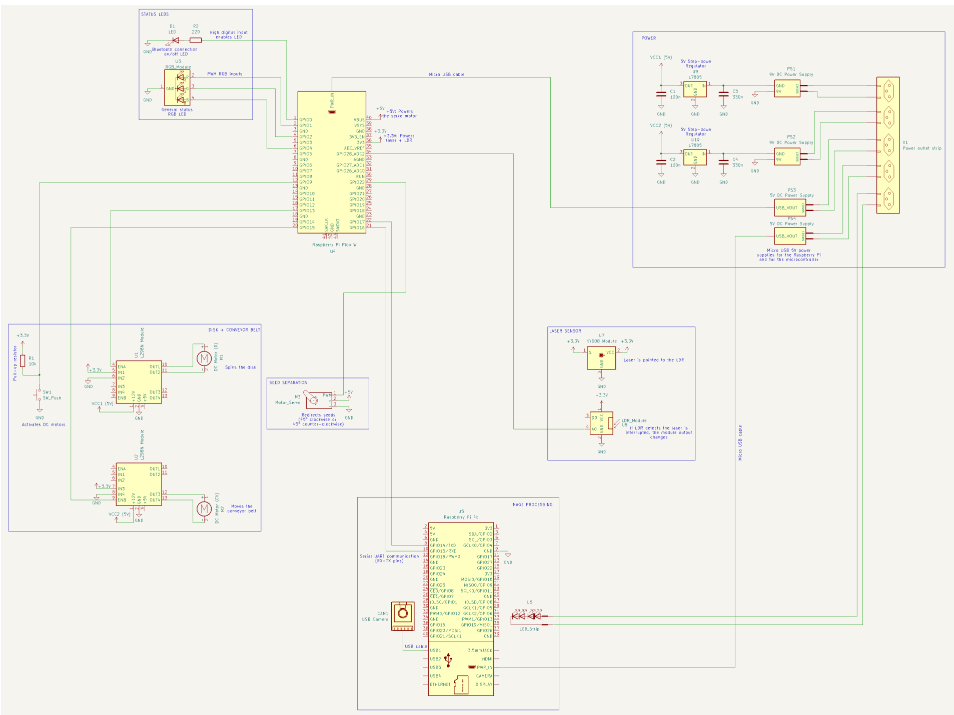

Figure 11: Full circuit schematic.

3.2.1 Embedded system

The embedded system of the project is centered around a Raspberry Pi 4B[4] and a micro controller (Raspberry Pi Pico W[5]). The Raspberry Pi is responsible for taking pictures of the seeds, running the computer vision model, and dealing with the analysis data. The micro controller, on the other hand, deals with the rest of the circuit, which relieves the Raspberry Pi from the processing cost of controlling all of the modules while running an image processing algorithm. Both are in sync during the whole process, and they communicate via their RX/TX pins, using the UART protocol.

3.2.2 DC motors

There are two DC motors in the project: one that spins a perforated disk that spaces the seeds, and one that moves a conveyor belt that transports them. Both operate under 5V DC, with dedicated power supplies of 5A and 3A respectively. They are driven each by an H-Bridge module and controlled by the Raspberry Pi Pico and the signals of a push-button.

3.2.3 Laser sensor

To detect the presence of a passing seed on the conveyor belt, the circuit counts with a laser sensor, composed of a KY-008 laser module[6] aligned with a LDR sensor module[7]. The system works as a simple tripwire sensor, in which, when a seed passes, blocking the laser, the variation of the signal received by the LDR is sent to the micro controller.

3.2.4 Taking pictures

The photos are taken by an USB camera[8] with adjustable focus connected to the Raspberry Pi. A white LED strip attached with the camera is responsible for the lighting.

3.2.5 Seed separation

A 9g servo motor[9] was used to control the mechanic separation system, also connected to the micro controller. The servo motor is directly powered by the 5V VBUS output of the Raspberry Pi Pico.

3.2.6 Status LEDs

The machine counts with two status LEDs: a RGB LED that indicates its current state (one for four), and an on/off LED that indicates a Bluetooth connection to the user's phone.

3.2.7 Power

The whole circuit is powered using a power outlet strip, which in turn provides power to the components. Each DC motor is powered by separate 5V DC supplies, to avoid current limitations, while the Raspberry Pi 4 and the Raspberry Pi Pico are powered by their own power sources, using USB cables, and the LED strip receives energy from its own 12V cable. The other components, such as the servo motor and the laser, are powered by the 5V or 3.3V outputs from the micro controller itself.

3.3 Software

The software of the project was developed in Python (for the firmware and the computer vision algorithm) and Javascript (for the app). The code repositories can be found in GitHub [10].

3.3.1 Firmware

The project firmware is divided into two components: the *Raspberry Pi 4 firmware* and the *Raspberry Pi Pico firmware*, each responsible for specific system functionalities. This division optimizes the distribution of computational load, preventing a single component from being overloaded, particularly with resource-intensive tasks such as image processing. Both parts communicate continuously through the UART protocol to ensure synchronization throughout the process.

3.3.1.1 Raspberry Pi 4 firmware

The Raspberry Pi 4 firmware is responsible for handling high-level computational tasks critical to the system's operation. It manages the camera module to capture images, processes these images using an onboard computer vision model, and performs real-time inferences to classify the detected objects. Additionally, the Raspberry Pi 4 functions as a GATT server, enabling Bluetooth communication for external device interaction. To ensure traceability and allow for further analysis, all captured data, including images and classification results, are stored in its local storage. A more detailed flow diagram illustrating the execution process of the Raspberry Pi 4 firmware can be found in the team's blog [3].

3.3.1.2 Raspberry Pi Pico firmware

The Raspberry Pi Pico firmware is dedicated to real-time control of the system's mechanical and electronic components. It regulates the DC motor drivers, ensuring precise conveyor belt movement, and operates a laser sensor for accurate seed detection. Additionally, it controls the servo motor, adjusting its angle to sort the seeds based on classification results. The Pico also manages an LED indicator to reflect the machine's current operational state and processes user inputs from the start button, allowing manual intervention when needed.

A more detailed flow diagram illustrating the functional execution of the Raspberry Pi Pico firmware can be found in the team's blog [3].

3.3.2 Computer vision model

For the development of the computer vision model, the team utilized *Edge Impulse*[11], a specialized platform for machine learning on edge devices. Initially, a public dataset[12] was used for training and testing the model. However, after multiple attempts, the team determined that capturing custom images and creating a new dataset would yield a model better suited to the project's specific scenario. The final model was trained using 3,000 images captured by the team and leveraged the *MobileNetV2* neural network with the transfer learning technique to enhance performance and adaptability.

The model is running on a *Raspberry Pi 4* with 2GB of RAM, using the *AARCH64* architecture. The inference time of the model is approximately 30ms.

Figure 12 presents the confusion matrix of the trained model, providing a detailed overview of its classification performance. Tests executed with seed images never used in training procedure achieved an accuracy of 91%.

Figure 12: Confusion matrix of the model

3.3.3 Mobile app

The mobile application was developed using React Native[13] and integrates a Bluetooth Low Energy library[14] for connecting to the device via Bluetooth. It provides an intuitive user interface with three primary screens designed to manage processes, handle device synchronization, and generate reports.

On the processes screen, users can view the latest 100 processes, each displaying key details such as the total number of seeds processed, the percentage of regular versus irregular seeds, and two action buttons. The first button allows users to delete a specific process they no longer need, while the second button directs them to the report screen, where they can see more detailed information about that process. A general button on this screen enables users to navigate to the synchronization screen. The process screen is shown in Figure 13.

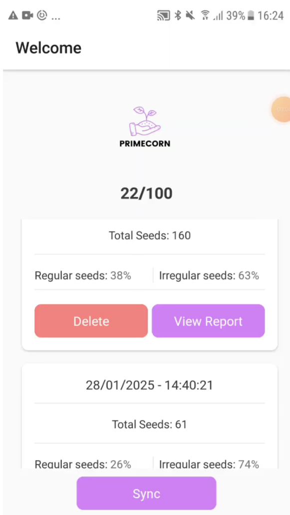

Figure 13: Process history screen.

The synchronization screen is where users can connect to available external devices. By pressing a button, they trigger a search for nearby devices, and a new screen appears showing a list of detected devices. Users can select their preferred device from this list, and once chosen, an animation is displayed to indicate that the app is in the process of synchronizing with the device. After the synchronization is complete, users can go back to the processes screen, where any new data from the connected device is displayed. The synchronizations screens are shown in Figure 14.

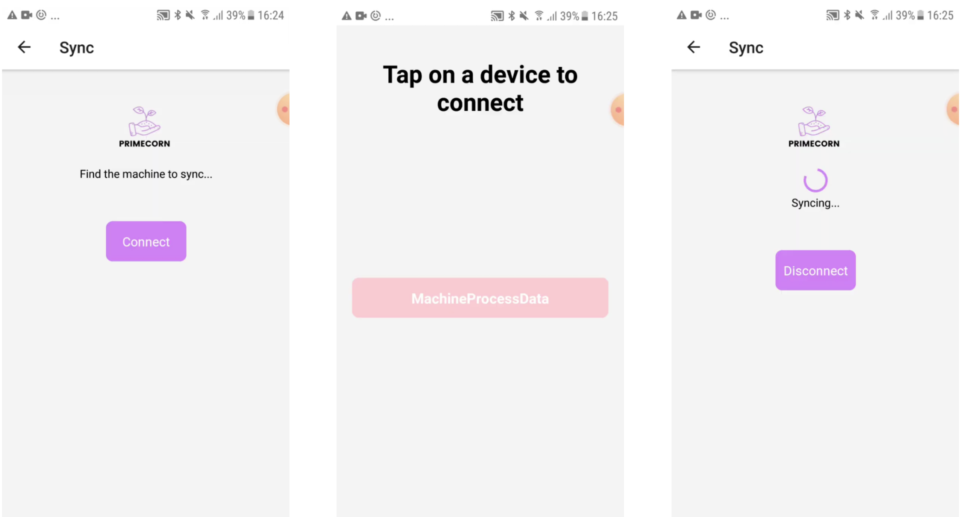

Figure 14: Sync screens.

When users select the "View Report" option for a specific process, they are redirected to the report screen, where they can view detailed information about the selected process. The screen displays the date when the process was recorded, along with a graph that illustrates the breakdown of irregular seeds by type. This feature helps users gain a deeper understanding of each process and provides valuable insights into the data generated by the connected devices. The report screen is shown in Figure 15.

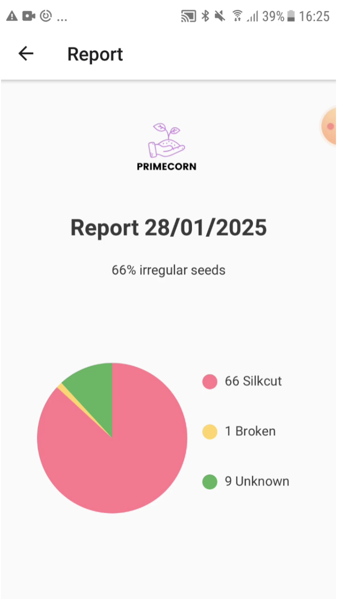

Figure 15: Report screen.

4 Results

The results of the PrimeCorn project were satisfactory. All the mandatory requirements were fulfilled and the team worked mostly on schedule throughout the development of the project. The entire process of development can be seen in the team's blog [3].

The performance of the machine also met the teams expectations, separating corn seeds with a reasonable precision and speed, according to what was promised in the Project Charter. It's usability is also satisfactory, being simple and intuitive to the user, added with the application where the information generated by the machine can be easily be seen.

The final look of the project can be seen in Figures 16, 17 and 18. The team also worked on a brief overview video for a general presentation [15].

4.1 Adaptations

From the planning phase to the execution, some minor adjustments were made:

- **Mechanics:** using all the holes in the perforated disk proved to be a challenge, because the flow of seeds was too heavy for the image processing and the mechanical separation system. So, the team chose to cover most of the holes on the disk, to maintain a more regulated flow.

- **Electronics:** the circuit worked as planned, but the main issue faced was the power supply, mainly for the DC motors. The problem was solved with dedicated supplies for each of them, and another for the other components.

- **Software:** the pre-made dataset of seed pictures was insufficient, so the team created a custom dataset with 3,000 images, using their own samples of popcorn seeds, using their own samples, the dataset.

4.2 Schedule

The project development started in October and finished in December 2024, with some minor adjustments in the computer vision model until February 2025. The work was done following six deliverables during this time:

- **Deliverable 1:** Mechanical design;

- **Deliverable 2:** Mechanical project and Electronic design;

- **Deliverable 3:** Electronic project and Software design;

- **Deliverable 4:** Software project;

- **Deliverable 5:** Mechanical, Electronic and Software integration;

- **Deliverable 6:** Overall integration

Even though, in the last deliverables, the team extrapolated the scheduled time, the total amount of hours were less than than originally planned, showing that the execution and planning phases were successful.

Deliverable	Estimated hours	Worked hours
Deliverable 1	33.8	33.7
Deliverable 2	58.45	57
Deliverable 3	78.15	70.3
Deliverable 4	71.5	55
Deliverable 5	57.2	60
Deliverable 6	38	45.5
Total	**337.1**	**321.5**

Table 2: Work Hours Summary

4.3 Final view of the project

The following pictures show some of the views of the PrimeCorn machine. In Figure 16, it is possible to see the place where the user may insert the seeds, Figure 17 shows the button to start the motors and the status LEDs, while in Figure 18 it's shown the labeled outputs through which the sorted seeds are dropped out of the machine.

Figure 16: Top view

Figure 17: Side view

Figure 18: Final View

4.4 Budget

In the original plan, the budget was of R$998,50 plus R$200,00 of extra emergency budget. The actual cost of the project, summing up all expenses, was R$1.095,50. This value is inside the foreseen price estimated by the Project Charter.

Table 3 shows all of the materials used to build PrimeCorn.

5 Conclusions and future work

5.1 Conclusions

The development of PrimeCorn was valuable to the enrichment of the team's experience in multiple areas, specially computer vision, machine learning and embedded systems.

The biggest challenges faced by the team were time management, which fortunately was balanced by the scheduling, and the mechanics of the machine, specially the DC motors and the power supply issue. Also, the team had a problem related to the seed image classification, because if the pictures weren't cropped, the machine learning algorithm wouldn't work. So, it was implemented a crop algorithm inside our computer vision system and it didn't worked as expected because of dirt in the conveyor belt. This took a considerable time to be discovered and the solution was installing some adjustments in the seeds transporter. The challenges were overcame by constant team communication and consultation with friends and acquaintances in the area.

The sections of the project that were developed more smoothly were notably the ones the team had planned the most in advance, which shows how crucial the planning phase was.

Name	**Qty.**	**Cost**
Conveyor belt	1	R$ 250,00
Rotating disk and ring	1	R$ 50,00
MDF sheets	1	R$ 50,00
3D printing	-	R$ 150,00
DC motor	1	R$ 50,00
Servo Motor 9G	1	R$ 15,00
Raspberry Pi 4	1	R$ 230,00
Camera	1	R$ 50,00
Laser Module KY-008	1	R$ 5,00
LDR Sensor Module	1	R$ 6,00
Corn seeds sample bag	1	R$ 4,00
L298N Dual H-Bridge Module	2	R$ 30,00
Raspberry Pi PICO W	1	R$ 60,00
Push Button	1	R$ 3,00
RGB LED module	1	R$ 10,00
On/off green LED	1	R$ 0,50
AC-DC Power supply	2	R$ 60,00
Circuit board + components	-	R$ 40,00
Plastic recipients	2	R$ 12,00
Funnels + hose	-	R$ 20,00
Total		**R$ 1.095,50**

Table 3: Budget

The team's work on PrimeCorn was satisfactory, since the functionalities worked well and the final result delivered what was promised in the Project Charter.

5.2 Future work

PrimeCorn is open to many improvements in the future, such as:

- A better computer vision model, with a larger sample of seeds to train it, improving even more its accuracy;
- Expanding the subtypes of irregular seeds detected;
- Changes in the separation mechanism to allow a faster rate of processed seeds.

6 Acknowledgments

The authors thank all acquaintances and friends who helped in the development of PrimeCorn. Specially Gilberto Suss Neto, who provided consultancy with the mechanical system, and Celso Luiz Bach, for presenting the problem that led to our solution.

References

[1] Pequeno produtor mas com grande importância na economia. https://www.sinditabaco.com.br/item/pequeno-produtor-mas-com-grande-importancia-na-economia.

[2] Produção de milho-pipoca "estoura" no Brasil. https://globorural.globo.com/agricultura/milho/noticia/2023/11/producao-de-milho-pipoca-estoura-no-brasil.ghtml.

[3] PrimeCorn: Seed Sorting Machine. https://superficial-baseball-f05.notion.site/PrimeCorn-1230b9cf6c578036a811da4f40522db2.

[4] Raspberry Pi 4 Model B specifications. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

[5] Pico series microcontrollers Raspberry Pi documentation. https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html.

[6] KY-008 Laser Transmitter Module. https://arduinomodules.info/ky-008-laser-transmitter-module/.

[7] LDR Module Datasheet. https://adiy.in/wp-content/uploads/2022/04/A87493_LDR-Module_Datasheet.pdf.

[8] HBVCAM OV9726 720P HD OTG Free Driver PC webcam Camera module. https://www.hbvcamera.com/1MP-%20HD-usb-cameras/hbvcam-ov9726-720p-hd-otg-free-driver-pc-webcam-camera-module.html.

[9] SG90 Datasheet. https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf.

[10] PrimeCorn repositories. https://github.com/stars/annaravazzi/lists/primecorn.

[11] Edge Impulse. https://edgeimpulse.com/.

[12] Corn Seeds Dataset. https://naagar.github.io/cornseedsdataset/.

[13] React Native · A framework for building native apps using React. https://reactnative.dev/.

[14] Python GATT server example for the Raspberry Pi. https://github.com/Douglas6/cputemp.

[15] PrimeCorn Integration Workshop 3 [CC]. https://youtu.be/Gka3NWqWwBI.