Federal University of Technology - Paraná – UTFPR
Academic Department of Eletronics – DAELN
Academic Department of Informatics – DAINF
Computer Engineering
Integration Workshop 3 (ELEX22) – S72 – 2025/1

Technical Report
VERTix

Cauê P. dos Santos – cauepiesanti@alunos.utfpr.edu.br
Gustavo F. Goltz – goltz@alunos.utfpr.edu.br
Henrique G. P. Bubniak – bubniak@alunos.utfpr.edu.br
Lucas de L. Nogueira – lucnog@alunos.utfpr.edu.br
Lucas N. Pio – lucaspio@alunos.utfpr.edu.br
Marcelo H. L. da Silva – marcelohenrique@alunos.utfpr.edu.br

June 2025

Abstract

VERTix is an innovative platform for physical assessment based on vertical jumps, developed to enhance the evaluation of performance and rehabilitation in physiotherapy and sports environments. It provides a fast, accurate, and user-friendly way to measure key physiological parameters such as jump height, flight time, and applied force. The system generates detailed visual data, including force-vs-time graphs for each leg and automated technical reports, supporting objective and consistent analysis. By eliminating the uncertainties and limitations of manual tools, VERTix improves the reliability of assessments and facilitates the detection of muscular imbalances, asymmetries, and incorrect movement patterns. These features are particularly valuable in clinical settings, where visual and quantitative feedback is often limited. Additionally, the platform enables professionals to compare multiple test sessions, helping track progress over time and guide informed decisions regarding training adjustments or rehabilitation strategies [1]. With its accessible interface and emphasis on precise measurement, VERTix contributes to more effective injury prevention, recovery monitoring, and performance optimization for athletes and patients alike.

1 Introduction

Vertical jump assessment is a widely used method to measure lower limb explosive strength, being particularly relevant in physiotherapy rehabilitation and physical training contexts [2]. This type of analysis provides crucial information about motor coordination, muscular balance, reaction time, and symmetry between the sides of the body [3]. Despite its importance, many physiotherapy clinics and gyms still lack accessible, precise, and visual tools that allow accurate measurement and monitoring of these parameters over time.

In the absence of objective instruments, professionals often rely on manual methods or subjective evaluations, compromising data reliability. Moreover, the lack of clear visual feedback hinders patient engagement and makes it difficult to identify gradual improvements or asymmetries that could indicate risks of recurring injuries.

1.1 Proposed Solution

To address these limitations, this report presents VERTix, an innovative digital platform focused on vertical jump evaluation. The system enables automated and precise measurement of variables such as jump height, flight time, applied force, and symmetry between lower limbs [4]. Through an intuitive interface and detailed graphical reports, VERTix supports physiotherapists, trainers, and other professionals in making well-informed decisions, tracking patient progress, and optimizing rehabilitation and injury prevention processes. The solution combines accessible technology with technical rigor, promoting a transformation in how physical performance is monitored and analyzed.

2 Project Specification

This section outlines the essential requirements identified for the system, encompassing both functional and non-functional specifications. These requirements were established based on market research, technical constraints, and user experience considerations. The specifications are categorized to address various aspects of the system, including mechanical functionality, user interaction, payment processing, and safety measures. Tables 1, 2, 3 and 4 presents the list of these requirements, which guided the system's development and served as benchmarks for evaluating the final implementation. These requirements are categorized into different types, including Functional Requirements (FR), which define the specific functionalities the system must perform, and Non-Functional Requirements (NFR). The tables only show the obligatory requirements of the system; a full list of the requirements is in our blog [5].

Table 1: Mechanical Requirements

ID	Description of Requirement	Achieved
FR101	The structure must contain two independent jumping platforms, one for each foot	Yes
FR102	Platforms must be able to withstand the weight and impact of multiple sets of repeated vertical jumps	Yes
FR103	The bases must have internal space or appropriate compartments for fixing the load cells and wiring, preventing displacement	Yes

ID	Description of Requirement	Achieved
FR104	The structure must support the camera at a height and angle suitable for capturing the jump	Yes
FR105	The structure must contain space to accommodate the hardware, with ventilation and space for maintenance	Yes
FR106	The structure must contain a support with LEDs, serving as a visual stimulus system	Yes
FR107	The structure must absorb impact and prevent slipping, using non-slip rubber and cushioning material	Yes
FR108	The structure must contain a system to guide and protect the cables that connect sensors, LEDs and modules, preventing breakage	Yes
NFR101	The structure must have dimensions suitable for the natural positioning of the feet, with comfort and stability	Yes
NFR102	The mechanical structure must be modular to facilitate transportation and maintenance	Yes
NFR103	The structure must be made of a material that is resistant to impacts and fatigue from continuous use	Yes
NFR104	The external finish of the structure must avoid sharp surfaces or corners, ensuring user safety	Yes
NFR105	The camera support must have an adjustable or fixed height between 1 m and 1.5 m, ensuring visibility of the body during the jump	Yes
NFR106	The entire structure must be light enough for mobility, yet firm enough for stability during use	Yes
NFR107	The layout of the elements must follow the principle of ergonomics and easy interaction with the user	Yes

Table 2: Embedded System and CPU Requirements

ID	Description of Requirement	Achieved
FR201	The system must include two force measurement units, one for each leg	Yes
FR202	Each force measurement unit must accurately measure the force applied by each leg	Yes
FR203	The system must be capable of comparing the data obtained from both platforms (legs)	Yes
FR204	The hardware must be powered by a single power supply	Yes
FR205	The hardware must be soldered onto a phenolic board	Yes

ID	Description of Requirement	Achieved
FR206	The main computer must acquire video from a connected camera	Yes
NFR201	The main computer must be a RaspberryPi 4	Yes
NFR202	The main computer must have an LED connected to its GPIO to indicate jump permission	Yes
NFR203	The main computer must transmit video data for processing in the cloud	Yes
NFR204	The microcontroller must be an ESP	Yes
NFR205	The system shall be powered by a 9V power supply	Yes
NFR206	The system shall use load cells to measure the force applied to the platforms	Yes
NFR207	The system shall use the SPI protocol to communicate between the load cells and the microcontroller	Yes

Table 3: Software (Mobile App) Requirements

ID	Description of Requirement	Achieved
FR3101	The system must allow the user to create and edit circuits	Yes
FR3102	The system must allow the user to create and edit volunteer's data	Yes
FR3104	The system must allow the user to view the video recordings of jumps	Yes
FR3105	The system must generate pre-processed reports with charts, trends and baseline comparisons using the jump data acquired	Yes
FR3106	The system must allow the user to export the report to the email of the volunteer	Yes
FR3107	The system must notify the user when a new report can be viewed	Yes
FR3108	The system must notify the user when a system error occurs	Yes
FR3109	The system must allow the user to view historical data per volunteer	Yes
FR3110	The system must permit the user to create an account to access the application	Yes
FR3111	The system must provide secure storage and access to user information	Yes
FR3113	The system must send circuit data to the embedded system	Yes

ID	Description of Requirement	Achieved
NFR3101	The system should be integrated with a cloud-based server hosted in Firebase or alike	Yes
NFR3103	The system should export the report as a pdf file format	Yes
NFR3105	The Mobile App should be built with Flutter	Yes
NFR3106	The system should show the user when there's incomplete information in reports of any volunteer	Yes
NFR3107	The system should communicate with the embedded system using the cloud interface	Yes
NFR3109	The report should use text, graphs and images to display volunteer data and results	Yes

Table 4: Software (Cloud and Computer Vision) Requirements

ID	Description of Requirement	Achieved
FR3201	The system must generate the user's joint points in the examination video	Yes
FR3202	The system must be able to calculate the height reached by the user in his jumps	Yes
FR3203	The system must be able to calculate the angles of the leg joints during jumps	Yes
FR3205	The system must flag incomplete/missing data received	Yes
FR3206	The system must be able to recognize asymmetries between the user's legs before jumps	Yes
FR3207	The system must automatically detect the start and end of each jump during the video	Yes
FR3208	The system must calculate the suspension time of the jump	Yes
FR3209	The system must allow uploading of video and/or images via RESful API	Yes
NFR3201	The system must use MediaPipe to recognize gestures in videos of jumps	Yes
NFR3202	Videos and jump information must be saved in a Firebase database	Yes
NFR3203	The system must have low cost	Yes
NFR3204	The system must be scalable to allow ondemand processing	Yes
NFR3205	System response time must be fast for asynchronous tasks	Yes

ID	Description of Requirement	Achieved
NFR3206	The system must allow temporary storage of unnecessary data, with possibility of automatic/scheduled cleaning	Yes
NFR3207	The system must support the Python programming language	Yes

3 System Architecture

The system architecture is organized into three interdependent layers: the mechanical structure, the hardware, and the software. The mechanical structure provides the physical foundation for the jump platform and the recording and signaling totem. The hardware layer integrates the electronic components and microcontrollers used in the project. Finally, the software layer, composed of the mobile application and the cloud system, is responsible for user interaction and processing of the data received from the microcontrollers.

3.1 Mechanical Structure

The entire structure of the VERTix is primarily built using MDF, which was chosen for its durability and cost-effectiveness. Various processes were employed during the mechanical construction, including cutting and machining, assembly, LED installation, and finishing. The mechanical structure consists of four main interrelated modules:

- Individual jump platforms
- LED and video capture tower
- Protective case
- Supports and fasteners

The mechanical design was developed with a focus on stability, protection of electronic components, and ease of assembly and transport.

3.1.1 Platforms, supports, fasteners, and protective case

This section is intended to provide a technical overview of the dimensions of the platforms, supports, fasteners, and protective case, which can be seen in Figure 1. The measurement platforms have dimensions of 500 x 300 x 30 mm, consisting of two 15 mm MDF boards glued together. The load cells are installed on the platforms using dedicated supports to ensure an even distribution of weight. The supports were repurposed from the original scales from which the load cells were removed. They are made of plastic and rubber and have a height of 17.5

mm. The fasteners, which connect the two platforms, have dimensions of 300 x 50 x 10 mm. The protective case, whose main function is to house the board with the electronic components, has dimensions of 200 x 200 x 50 mm and features a removable lid, making system maintenance easier.

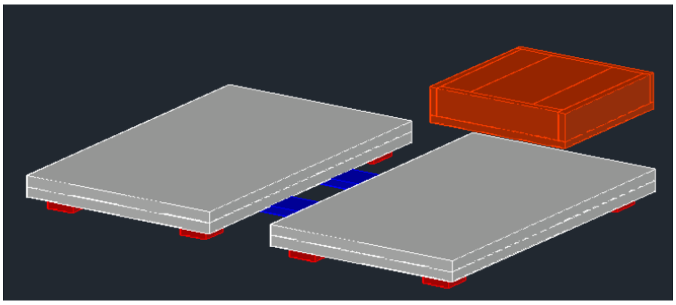

Figure 1: Platforms, supports, fasteners, and protective case

3.1.2 LED and video capture tower

The LED and video capture tower has overall dimensions of 300 x 300 x 1000 mm and is made of 15/17.5 mm MDF. Figure 2 shows the overall structure, with the three colored sections representing the base, body, and top of the tower. The base was designed to ensure the stability of the tower, the body houses the LEDs used as visual indicators, and the top section is intended to house the camera, which will be responsible for video capture together with the microcontroller.

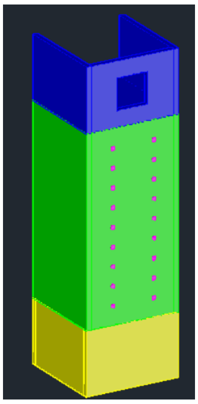

Figure 2: LED and video capture tower

3.1.3 Results

In order to ensure the proper functioning of the structure, several important parameters were calculated, as shown in Table 5. The analysis of these parameters, combined with the 30 mm MDF platform, ensures support for jumps of up to

150 kg with a wide safety margin. In addition, practical tests were conducted with a series of 20 jumps (10 regular + 10 with bent knees) performed by a 70 kg volunteer on a single platform. These tests confirmed the efficiency and reliability of the platform.

Table 5: Simplified structural analysis

Parameter	Value	Result
Dynamic load (150 kg × 3 g)	4414 N	$\sigma_{max} = 4,1 MPa$
Allowable stress for MDF	10 MPa	MS ≈ 2,5
Maximum deflection	0,38 mm	≪ L/200 (2,5 mm)

Figure 3 shows the fully assembled system. Through testing, it was found that a distance of approximately 4 meters between the platform and the tower proved to be excellent, although it can be adjusted to optimize video capture.

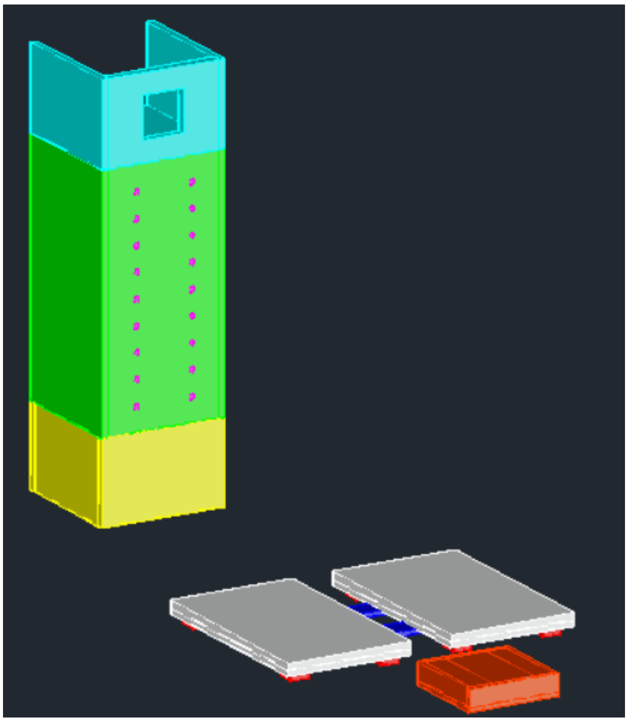

Figure 3: Completely assembled system

3.2 Hardware and Firmware

The Hardware and Firmware module of VERTix is divided into two submodules: the Embedded System module and the Raspberry Pi module, both represented in the schematic diagram shown in Figure 4.

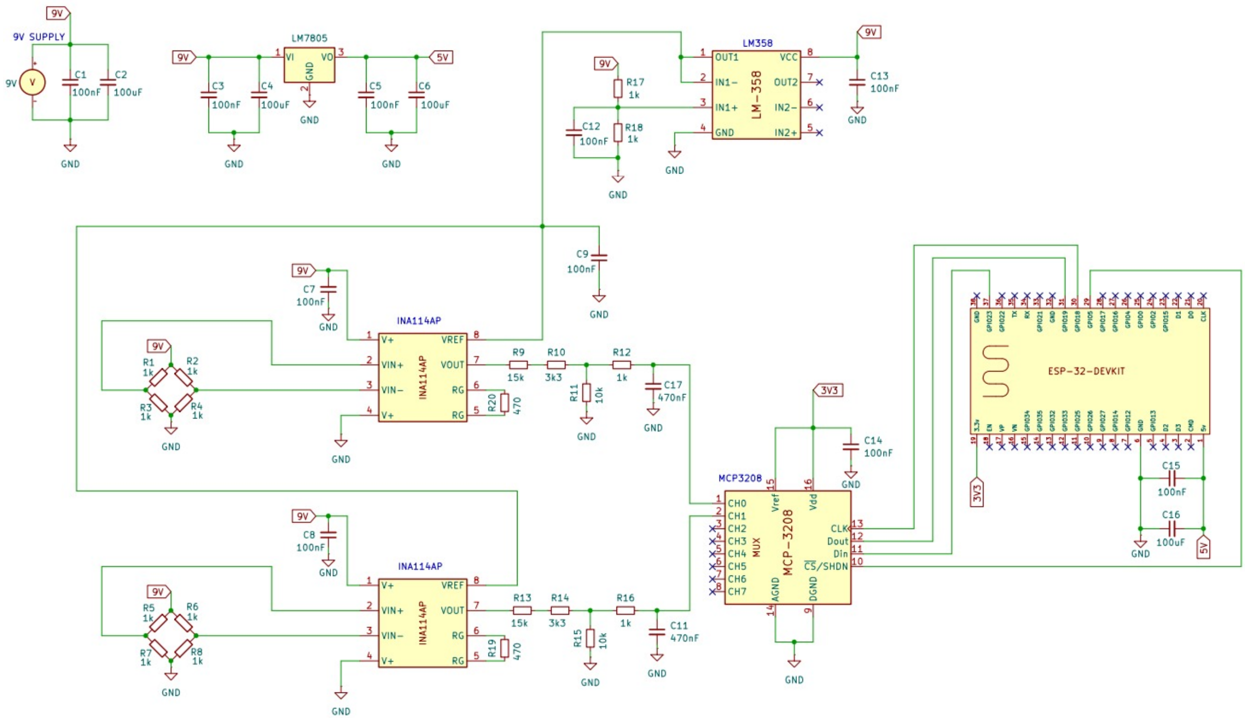

Figure 4: Schematic diagram

The diagram in Figure 5 illustrates the connection between the modules and the cloud.

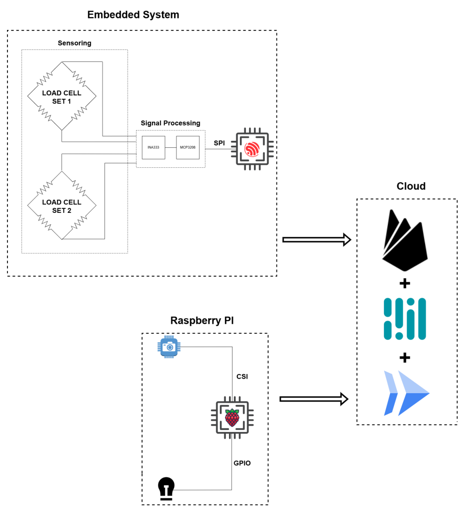

Figure 5: Hardware/Cloud diagram

3.2.1 Embedded System module

The Embedded System module is integrated into the platforms and comprises eight load cells (four per platform) and a circuit board equipped with an ESP32 microcontroller. The load cells acquire analog signals corresponding to the force applied by each leg, whereas the board performs signal amplification, filtering, digitization, local processing, and transmission to the cloud. Figure 6 and 7 present the circuit board and the load cells, respectively.

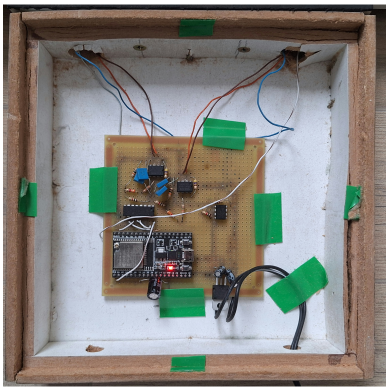

Figure 6: Electronic circuit board

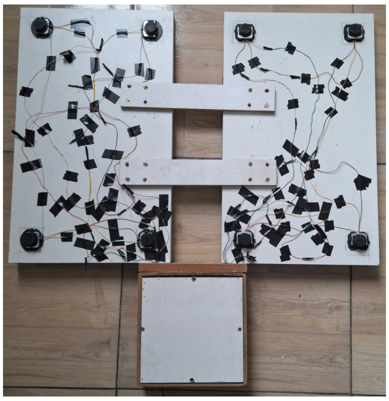

Figure 7: Load cells

Note: All components are listed in the project blog.

3.2.2 Raspberry PI module

The Raspberry Pi module is integrated into the Totem and is responsible for recording a video of the jump using a Raspberry Pi camera. It also interacts with the user by displaying different colors on an LED strip throughout the jump routine. Once the jump is completed, the recorded video is sent to the cloud for further analysis. Figure 8 illustrates the phases of the jump routine, including their respective durations and the corresponding LED colors.

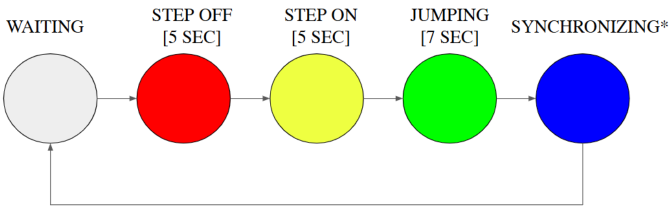

Figure 8: Totem LEDs diagram

Note: In case of any failure during the jump, the LEDs will flash red.

*The LEDs will flash blue during the synchronization phase

3.2.3 Power Consumption

Regarding the embedded system, including the electronic circuit board and the eight load cells, the peak current consumption is approximately 500mA, which corresponds to the maximum drawn by the ESP32. Other components, such as the ADC, amplifiers, and load cells, have negligible power consumption.

As for the Raspberry Pi module, its maximum current consumption is determined by the peak values of the Raspberry Pi (3A) and the LED strip (2A).

Table 6 summarizes the power consumption of the hardware components.

Table 6: Hardware Power Consumption

Component	Module	Current Draw	Voltage Supply	Power Consumption
Circuit board	Embedded System	~500 mA	9 V	~4.5 W
Load cells	Embedded System	~0 mA	9 V	~0 W
Raspberry Pi 4	Raspberry Pi	~3 A	5.1 V	~15.3 W
LED strip	Raspberry Pi	~2 A	5 V	~10 W
Total				**~29.8 W**

3.2.4 Calibration

In order to convert the digitized data into a physical quantity (Newtons), a calibration procedure was carried out. The adopted method involved simultaneously applying twenty different loads to each platform, simulating various body weights. The measurements ranged from 0kg to 50kg per platform, increasing by 2.5kg with each step. Following data collection, linear regression was applied, resulting in a linear function, along with the determination coefficient (R^2) and Pearson correlation coefficient (r) for each platform. These linear functions were subsequently implemented in the microcontroller's force computation algorithm. Figure 9 and 10 present the results for both platforms, including the graphs, linear equations, and the corresponding coefficients.

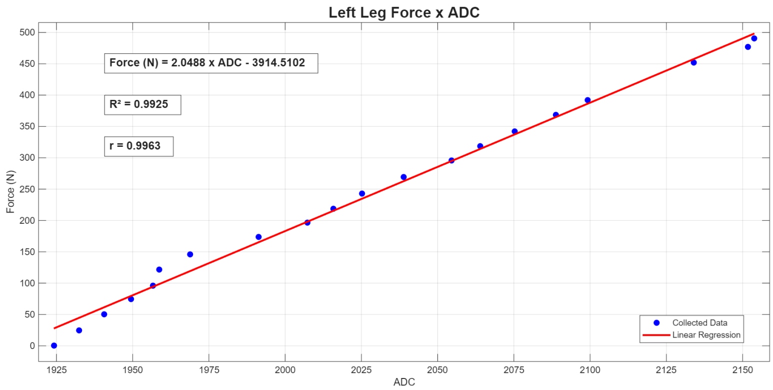

Figure 9: Calibration of left platform

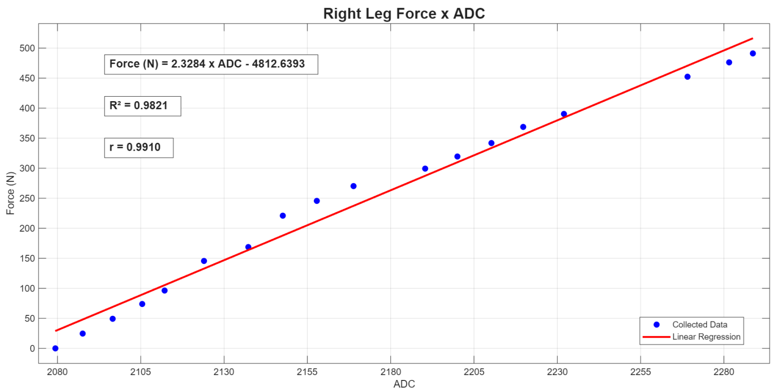

Figure 10: Calibration of right platform

Note: A kitchen scale was used to measure the weight of the objects applied during the calibration procedure.

Note: All collected data and calculations can be found in the project blog

3.2.5 Results

Regarding force measurement and weight acquisition, the system performs adequately, despite certain limitations such as electrical noise from the power grid and interference from nearby devices. After extensive testing, a discrepancy of up to approximately 10% was observed between the weight measured by the platform and that indicated by standard commercial body scales. Additionally, the power consumption is relatively low, ensuring low energy costs during extended usage sessions.

3.3 Computer Vision and Cloud

3.3.1 Cloud Architecture

Google Cloud was adopted for the cloud architecture, serving as the platform responsible for managing the API and processing the videos and signals received from the microcontrollers.

The API was developed in Python using the Flask framework, and the entire system was containerized with Docker to ensure easy deployment across different environments. Requests from both the mobile application and microcontrollers include status checks, routine creation, data submission, and information processing. All communications occur over the HTTPS protocol to ensure security.

All API endpoints were designed with robust error and exception handling mechanisms to notify the requester in case of any issues or inconsistencies during the request process.

In addition, Google Cloud handles the execution of MediaPipe, signal analysis, and video uploads to Vimeo. These operations run in the background using threads, allowing the API to remain responsive and capable of handling multiple processing requests concurrently without performance degradation.

3.3.2 Force Analysis

For the Force Analysis, the cloud receives from the ESP32 an array containing the sampled force and weight data collected from each load cell platform during the execution of the jump. These samples represent the vertical ground reaction forces applied by the volunteer throughout the movement. By adding the force values from each platform, we can calculate the total force exerted by the volunteer during the entire jump action. Similarly, summing the static weight measurements prior to the jump allows us to estimate the volunteer's body weight.

To estimate the air time of the jump, we analyze the list force samples to identify the two key force peaks: the take-off and the landing. These peaks correspond to the highest force values recorded before the loss of contact with the platform and upon ground recontact, respectively. The number of samples between these two peaks represents the duration for which the individual was airborne. Given that the system operates at a fixed and known sampling rate, we can accurately estimate the air time by dividing the number of samples between the peaks by the sampling frequency. The result is the total air time of the jump expressed in seconds.

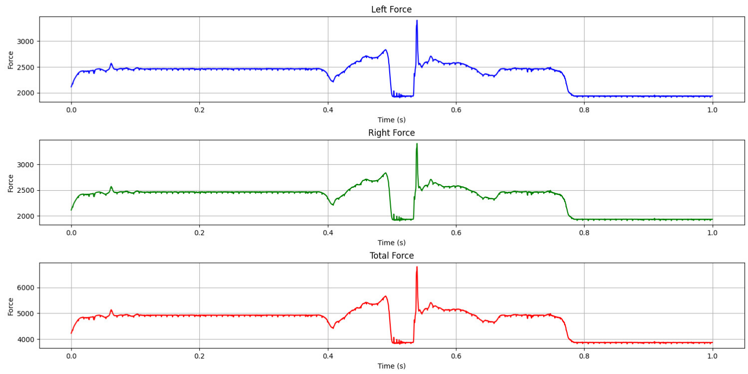

Figure 11: Force Samples during a jump

3.3.3 Computer Vision Analysis

The computer vision component of the VERTix platform is structured into three main modules: the **Detector**, the **Analyzer**, and the **Visualizer**. This modular approach enables a robust and extensible pipeline for extracting and interpreting biomechanical data from video recordings of vertical jumps.

After the cloud server receives the video stream from the Raspberry Pi, it forwards the footage to the Detector module. The Detector is responsible for running inference on every frame using two state-of-the-art models: MediaPipe for extracting joint landmarks, and YOLOv5 for detecting the bounding box of the person. The raw inference data, including the pixel coordinates of key joints and the bounding box, is then passed to the Analyzer module.

The Analyzer module first converts the pixel-based coordinates to real-world measurements in centimeters. This is achieved by dividing the number of vertical pixels between the person's heels (from MediaPipe) and the top of the head (from YOLOv5) by the person's actual height, which is provided by the app at the start of the assessment. Using this scaling factor, the Analyzer computes the center of mass (CoM) position for each frame, based on a biomechanical model described in the literature (to be cited). The vertical (y-axis) position of the CoM is thus expressed in centimeters, generating a time series signal of the CoM's vertical displacement throughout the jump. [6]

This signal is then filtered to reduce noise. The Analyzer calculates the frame-to-frame difference (the discrete derivative) of the CoM position to estimate the vertical velocity. A threshold is applied to this derivative to detect the start of the jump: when the vertical velocity of the CoM reaches a certain negative value (to be specified), the jump is considered to have begun. The CoM height at this frame is used as a baseline.

The moment the CoM height exceeds this baseline marks take-off (loss of ground contact). The maximum CoM height after take-off corresponds to the jump's peak, and the return to baseline indicates landing. This process allows the system to determine the jump height, airtime, and other relevant metrics. Additionally, the Analyzer uses MediaPipe inferences to track the angle between the thigh and the vertical axis, enabling the detection of possible knee valgus during the jump.

All extracted information is then passed to the Visualizer module. The Visualizer generates plots of the CoM position, its derivative, and the knee valgus angle over time. It also creates annotated videos visualizing the detected landmarks and bounding boxes. The processed video with overlaid landmarks is uploaded to Vimeo, and all jump metrics are stored in Firebase, making them accessible to the app for report generation.

3.3.4 Results

To validate the computer vision analysis module of the VERTix platform, we conducted an experiment involving 20 vertical jumps. One jump was excluded from the analysis due to poor filming, resulting in a total of 19 valid jumps. For each jump, the height was measured using a tape measure as a reference, and the result was compared to the height estimated by the computer vision system.

The table below presents the key measurements for each jump, including the subject's height, the reference jump height (tape), the height estimated by the system, and the percentage error.

Table 7: Simplified structural analysis

ID	Height (cm)	Ref. (cm)	Vision (cm)	Error (%)
1	166	29.5	37.55	27.29
2	166	34.5	42.81	24.09
3	166	36.0	38.11	5.86
4	166	40.0	42.22	5.55
5	166	42.5	42.28	-0.52
6	166	44.5	43.72	-1.75
7	166	44.0	45.33	3.02
8	166	47.5	45.37	-4.48
9	166	44.0	49.22	11.86
10	166	49.5	45.53	-8.02
11	196	37.0	42.76	15.57
12	196	42.0	44.78	6.62
13	196	46.5	49.62	6.71
14	196	41.0	50.68	23.61
15	196	45.5	46.04	1.19
16	196	42.0	49.55	17.98
17	196	47.5	49.56	4.34
18	196	46.0	49.42	7.43
19	196	50.5	49.10	-2.77

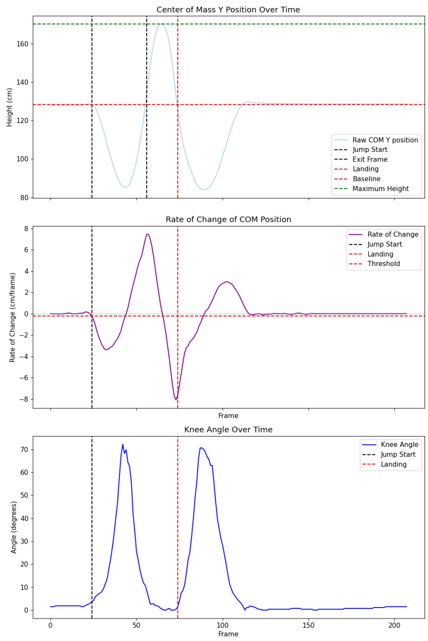

Figure 12: Plots generated by the Visualizer module

Figure 12 illustrates the typical outputs generated by the Visualizer module, including the CoM vertical position (A), its derivative (B), and the knee valgus angle for a jump (C). These plots provide valuable insights for both performance analysis and injury prevention.

The average percentage error across all 19 jumps was 10.10%, with a standard deviation of 7.56%. These results demonstrate that the computer vision system provides a reliable estimate of jump height, with most errors within an acceptable range for practical use in performance and rehabilitation settings.

3.4 Mobile App

The mobile application serves as the user interface for interaction with the jump measurement platform. It allows users to register routines, select volunteers, initiate circuits, and access detailed performance reports.
The application follows the Model-View-Controller (MVC) architectural pattern, developed using Flutter and Dart.

- **View**: Contains all UI components and is responsible for displaying data and handling user interactions;
- **Controller**: Handles user input, controls the flow of data, and manages state;

- **Model**: Represents the app's data and business logic. Includes entities like volunteers, routines, reports and cloud communication handled via Firebase FireStore, Storage, and Authentication.

This way, the app offers an intuitive interface with four main screens to manage records, initialize circuits, and generate reports. On the login screen the user can have acces to all functionalities regarding account creation, recovery, and authentication.This is shown in figure 13

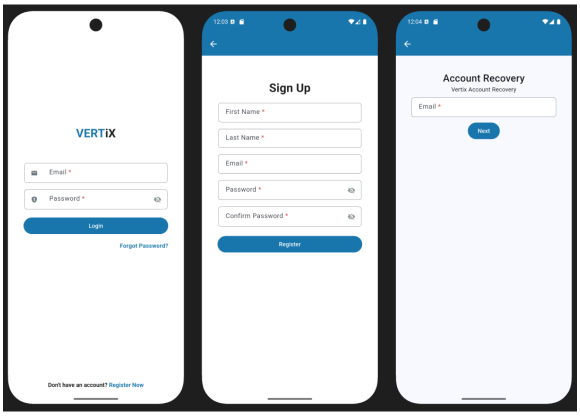

Figure 13: Login, sign-up, and password recovery screens.

The home screen (Fig. 14) is where the user can navigate to the volunteer and jump routines screens, and also see the list of generated reports. By interacting with the buttons chips he can change what type of information to be displayed, as well as navigate to a different type of report.

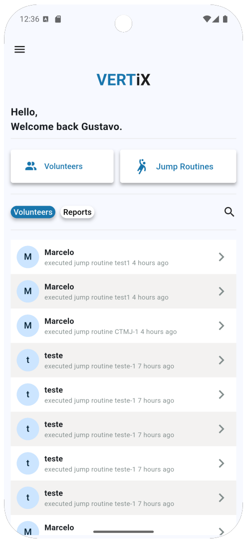

Figure 14: App home page.

Selecting the “Volunteer” option opens the volunteer screen. Tab 1 lists existing profiles; Tab 2 provides a form to create new profiles (see Figure 15).

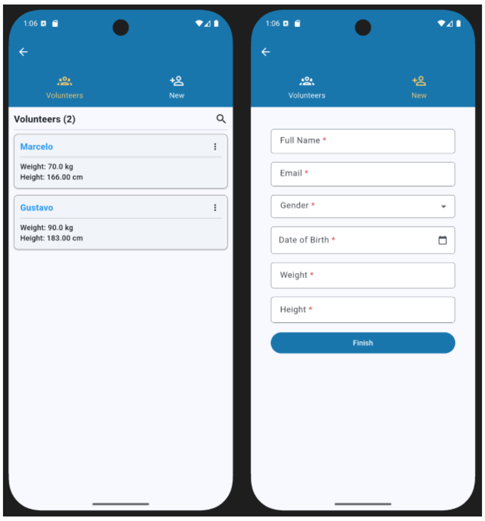

Figure 15: Volunteer tabs.

The same can be done via the "Jump Routine" option, where the user will be redirected to a page that contains all functionalities to manage jump routines (Fig. 16).

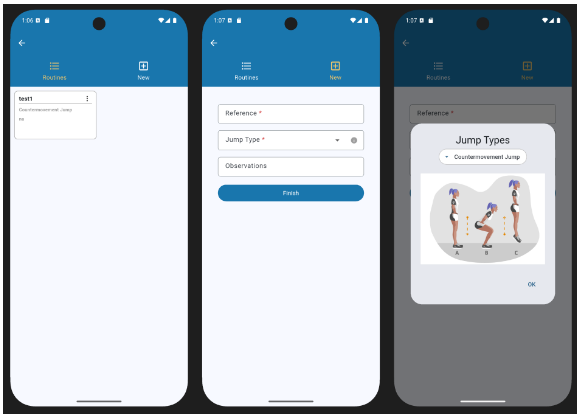

Figure 16: Jump Routine tabs.

To initiate a circuit, the user selects a volunteer and a jump routine. These two identify who is going to perform the jumps, and what kind of movement is he going to perform respectively. Once selections are made, the app notifies the cloud to start the circuit. All required data is already stored in the database. After the circuit completes, the app redirects to the home screen. When processing finishes, selecting the new report displays a detailed dashboard of the jump data. There are two types of reports:

- **Per-jump report**: Includes jump force graph, parameter extraction, performance assessment, and synchronized video.;
- **Summary report**: Aggregates session data, showing jump averages, standard deviation, and comparison by parameter.

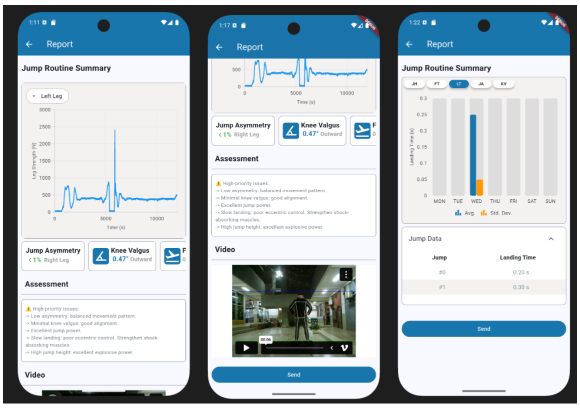

Figure 17: Generated reports.

The user can then send the report generated to the email specified on volunteer's creation (Fig. 15).

4 Results

The outcome of the work described in the previous sections is presented in the following video: https://www.youtube.com/watch?v=5ceOo-X0Qho

The project specifications and requirements were largely met, with all mandatory requirements successfully completed.

Table 7 provides a detailed cost analysis, presenting a breakdown of each component, along with their respective quantities and total expenses. This assessment includes all components necessary for the implementation of the system, covering hardware, materials, peripherals, and accessories.

Table 8: Simplified structural analysis

Item	**Qty**	**Cost (BRL)**
Raspberry Pi 4	1	250.00
Camera for Raspberry Pi	1	90.00
ESP32	1	40.00
8 load cells (50 kg each)	8	60.00
Integrated circuits (Voltage regulators, amplifiers, ADC)	-	90.00
Phenolite PCB	1	20.00
9V power supply	1	15.00
MDF boards	-	400.00
Support rubbers	-	10.00
LEDs	1	40.00
Power strip	-	20.00
General electronic components + materials	-	25.00
TOTAL		1060.00

The budget was respected, and the selected components performed effectively while staying within financial constraints. Furthermore, the risk analysis conducted during the planning phase proved valuable, helping to mitigate potential issues and ensuring the project was completed within budget.

The project schedule was also followed, with weekly tasks completed on time. In cases where delays occurred due to component availability or delivery times, these were promptly resolved at the beginning of the following week, minimizing impacts on the overall timeline. The expected number of work hours, according to the plan, was 383 hours. However, considering the challenges encountered during development, the project was completed in a total of 482.4 hours, which is close to the planned amount when considering the 30% margin of error in the estimated time, which amounted to 497.9 hours.

5 Conclusion

The VERTix project presents an innovative approach to the analysis of physical parameters based on vertical jumps, leveraging Machine Learning mechanisms to obtain more accurate data.

Since all mandatory requirements were met, along with the implementation of optional ones, it can be concluded that the project was successful. Furthermore, the budget was adhered to throughout the process, allowing for the replacement of components in case of damage. However, although the current implementation has achieved its objectives, there are areas that can be improved and further developed in the future.

5.1 Future Work

Future improvements include mechanical enhancements to better protect cables beneath the platforms, the use of pins for quick and dynamic platform alignment, as well as enabling height adjustment of the platforms. From a mechanical perspective, there is also the potential to explore alternative materials to reduce weight and improve moisture resistance, along with the use of sealing techniques to increase durability.

Regarding video and signal analysis, it would be valuable to study and apply new analysis techniques to ensure greater accuracy in the measurements performed, providing even more reliable results to VERTix users.

Additional developments could include the addition of new features in the application to enable greater user interaction with the system. Furthermore, introducing a new dynamic through the use of indicator LEDs, voice commands could be explored to facilitate interaction between the hardware + mechanics and the end user, ensuring a more dynamic and realistic experience and potentially attracting a broader market segment.

Acknowledgements

We extend our gratitude to all those who contributed to the development of this project, either directly or indirectly. Special thanks to our professors Gustavo Benvenutti Borba and Humberto Remigio Gamba, whose guidance and feedback were essential throughout the entire development process.

References

[1] Gregory D Myer, Kevin R Ford, Jensen L Brent, and Timothy E Hewett. The effects of plyometric vs. dynamic stabilization and balance training on power, balance, and landing force in female athletes. *Journal of Strength and Conditioning Research*, 20(2):345–353, May 2006.

[2] DGE Robertson, GE Caldwell, J Hamill, G Kamen, and SN Whittlesey. *Research Methods in Biomechanics*. Human Kinetics, 2nd edition, 2014.

[3] Chris Bishop, Paul Read, Shyam Chavda, and Anthony Turner. Asymmetries of the lower limb: The calculation conundrum in strength training and conditioning. *Strength and Conditioning Journal*, 38(6):27–32, December 2016.

[4] JS Dufek and BT Bates. The evaluation and prediction of impact forces during landings. *Medicine and Science in Sports and Exercise*, 22(3):370–377, Jun 1990.

[5] Vertix project overview. https://berry-hour-e9d.notion.site/VERTix-1c7189d4adf380f7b297c6091e98ecdb. Accessed: 2025-06-26.

[6] D. Lafond, M. Duarte, and F. Prince. Comparison of three methods to estimate the center of mass during balance assessment. *Journal of Biomechanics*, 37(9):1421–1426, 2004. Using the landmarks from the MediaPipe model.

[7] Vertix. YouTube. https://www.youtube.com/watch?v=5ceOo-XOQho.