

Article

ROS2-Based Low-Cost Mobile Robot for Educational Assistance with Reactive Navigation and Semantic-Cached Language Processing

Sebastián Alexis Aucapiña ¹, Nataly Cecilia Benalcázar ¹, José Varela-Aldás ^{1,*} and Ramiro Isa-Jara ^{2,*}

¹ Centro de Investigación MIST, Facultad de Ingenierías, Maestría en Robótica y Automatización Industrial, Universidad Tecnológica Indoamérica, Ambato 180212, Ecuador; saucapina2@indoamerica.edu.ec (S.A.A.); nbenalcazar3@indoamerica.edu.ec (N.C.B.)

² Laboratory of AI “A.M. Turing”-GITEA, Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Riobamba 060104, Ecuador

* Correspondence: josevarela@uti.edu.ec (J.V.-A.); ramiro.isa@epoch.edu.ec (R.I.-J.)

Abstract

Educational environments, particularly those with limited resources, require affordable mobile robots capable of combining human–robot interaction, autonomous assistance, and academic support without continuous dependence on cloud services. This work presents a low-cost ROS2-based mobile robot implemented on a Raspberry Pi 4B to provide educational assistance in Spanish within controlled classroom environments. The system integrates voice interaction, text-to-speech synthesis, YOLOv8n-based object perception, a specialized door detection model, ultrasonic and inertial sensing, differential-drive control, and a hybrid natural language processing architecture based on semantic caching, local inference, and optional cloud connectivity. Two task-dependent operating modes, education and navigation, selectively activate ROS2 nodes to reduce computational load and energy consumption. Experimental tests conducted in a university classroom evaluated speech recognition, vision models, natural language processing alternatives, sensor behavior, and battery life. The speech recognition module achieved 98% accuracy under both quiet and noisy conditions. YOLOv8n achieved an F1-score of 0.975 for common classroom objects, while the specialized door detector achieved 100% recall with 58.7% precision. The semantic cache correctly resolved recurrent academic queries in the exact-match evaluation, with an average latency of 3.8 s, reducing the need for external language models in known-question scenarios. The robot operated for 96 min in education mode and 75.6 min in navigation mode. These results demonstrate that Spanish voice interaction, reactive navigation, academic question answering, and resource-aware operation can be integrated into a single low-cost edge robotic platform for educational environments.

Keywords: educational mobile robot; human–robot interaction; ROS2; reactive navigation; semantic cache; speech interaction; SDG 9



Academic Editor: Sunan Huang

Received: 13 May 2026

Revised: 30 June 2026

Accepted: 5 July 2026

Published: 8 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Mobile robots have been widely used in several domains, including education, personal assistance, and task automation. In education, they have been used to promote programming, language, and social skill learning, as well as to increase student motivation through interaction with robotic systems [1–4]. However, many of these solutions, such as NAO or Pepper robots, are associated with high costs that limit their adoption

in resource-constrained settings, including public schools and rural areas in Latin America. This situation highlights a gap between available technological capabilities and their practical implementation in low-resource educational settings.

For Spanish speech recognition, local processing has gained relevance because of connectivity and privacy constraints. Tools such as VOSK can run locally on embedded devices such as the Raspberry Pi [5] and support Latin American Spanish, making them suitable for predefined command recognition; this is a significant advantage in view of responsiveness benchmarks for TTS systems such as those reported by Dinh et al. [6]. However, most systems are optimized for English and require an internet connection [7,8]. Furthermore, limitations in robust speech recognition have been reported due to linguistic variability and out-of-domain vocabulary [9]. Although edge-based recognition and embedded assistants have been proposed, they tend to focus on other languages, transcription tasks, or architectures that depend on external devices [2,10–12].

Robotic navigation commonly relies on techniques such as SLAM, advanced planning, or large language models (LLMs) integrated with NAV2 for navigation [13]. These approaches involve high computational requirements [14–16]. Modern vision- and learning-based methods have demonstrated improvements in dynamic environments [17–19]; however, their implementation on embedded platforms remains limited, particularly on resource-constrained devices such as the Raspberry Pi, where they may restrict interaction fluidity. In classrooms, where obstacles are dynamic, approaches based on static maps may be inefficient; therefore, this work adopts a more reactive approach based on sensors and machine learning for real-time obstacle avoidance [20,21]. Furthermore, incorporating language models enables command-based interaction and responses to open-ended questions. However, large models such as GPT-4 are difficult to deploy on embedded devices, while smaller models such as TinyLlama may lack the reasoning capabilities required for educational questions. Recent research has explored lightweight models and hybrid strategies for edge processing [22–24], as well as dynamic selection mechanisms and semantic caching to optimize responses [25–27].

Similarly, the integration of language models into robotic and human–robot interaction systems has been studied [28–30], together with edge-optimization techniques based on caching [31,32]. However, these solutions primarily focus on cloud services or task-specific applications, without addressing academic queries in educational settings. Despite these significant advances, few studies attempt to integrate all these capabilities into a single, accessible, low-cost system that supports Latin American Spanish and offline operation. The closest available examples have addressed these capabilities only partially, either through educational robots without advanced NLP [33], English-language conversational systems [7], solutions for specific users [5], or natural-language control limited to commands in multilingual settings [34–36].

In this context, this paper presents a low-cost ROS2-based educational mobile robot running on a Raspberry Pi 4B that integrates Spanish voice interaction, reactive sensor-based navigation, vision-based object and door perception, and a hybrid language-processing architecture based on semantic caching. Unlike educational robotic platforms that mainly focus on learning activities or multimodal interaction without embedded academic question answering [1–3,33], speech assistant systems primarily oriented toward recognition, transcription, or user-specific assistance [5,10–12], and natural-language robot-control approaches mainly limited to navigation commands or multilingual low-resource scenarios [34–36], the proposed system unifies academic question answering and navigation within a single low-cost edge platform designed for Latin American Spanish. Furthermore, in contrast to SLAM- or LLM-intensive navigation and interaction strategies [13–19] and edge-optimization or caching frameworks evaluated mostly outside embodied educational

robots [22–27,31,32], the proposed architecture uses task-dependent ROS2 node orchestration and a semantic cache to reduce latency and energy consumption while preserving offline operation for recurrent academic queries. The system is experimentally validated under real classroom conditions using component-level and integrated metrics, including speech recognition accuracy, object and door detection performance, NLP accuracy and latency, and battery life, providing an accessible robotic solution for resource-constrained educational environments.

The JUNO platform [37] combines navigation and natural interaction, but it depends on cloud services and targets elderly care rather than educational content delivery. To the best of the authors' knowledge, no existing work integrates offline Spanish-language academic question answering, reactive navigation, and semantic caching within a single low-cost embedded robotic platform. Table 1 summarizes the differences between the proposed system and related works.

Table 1. Comparison between the proposed system and related low-cost or language-integrated robotic platforms.

Reference	Language Support	Offline Operation	Mobile/Reactive Navigation	Academic Q&A
Baksh et al. [33]	Not specified	Not specified	No	Partial (general engagement)
Mekonnen et al. [5]	Not specified	Yes (fully offline)	No	No (object/environment awareness)
Hernández et al. [34]	Spanish and Nahuatl	Not specified	Commands only; no deployed robot	No (navigation commands only)
Pavón-Pulido et al. [38]	Not specified	No (cloud-dependent)	Yes (autonomous, pre-mapped)	No (cognitive stimulation only)
This work	Latin American Spanish	Yes (cache + local model; optional cloud)	Yes (reactive, sensor-based)	Yes (curriculum-based)

As a representative interaction, a student activates the robot by voice, asks “What is photosynthesis?”, and the robot responds verbally in approximately 3.8 s using its semantic cache, without requiring internet connectivity. In navigation mode, commands such as “robot forward” move the robot through the classroom, while the ultrasonic sensor handles obstacle avoidance.

The main contributions of this work can be summarized as follows:

- (I) The design and implementation of a low-cost (under \$250) ROS2-based educational mobile robot that integrates speech interaction, object and door perception, reactive navigation, and natural language processing within a single embedded platform.
- (II) A hybrid, energy-aware natural language processing architecture that combines semantic caching, local inference, and optional cloud connectivity, enabling offline academic question answering in Latin American Spanish.
- (III) A dual-mode (education/navigation) ROS2 node-orchestration strategy that selectively activates system components to reduce computational load and energy consumption.
- (IV) A specialized door detection model trained through transfer learning to support safe reactive navigation in dynamic indoor environments.
- (V) An experimental evaluation covering component-level metrics (speech recognition, object detection, NLP latency and accuracy, and sensor characterization), system-level performance (energy consumption and operating autonomy), and integrated navigation behavior, including command response, obstacle avoidance, and object following.

2. Materials and Methods

The proposed system follows a modular architecture based on ROS2 Humble (Hawksbill release) and is organized into four functional layers: (1) physical (embedded hardware), (2) abstraction (ROS2 interface nodes), (3) processing (vision, speech, and NLP), and (4) application (orchestration of operating modes). This structure decouples functional modules and optimizes resource use on the embedded platform.

2.1. Abstraction

The software architecture presented in Figure 1 is organized into ROS2 nodes grouped by functionality: hardware interface, speech processing, NLP, vision, and orchestration.

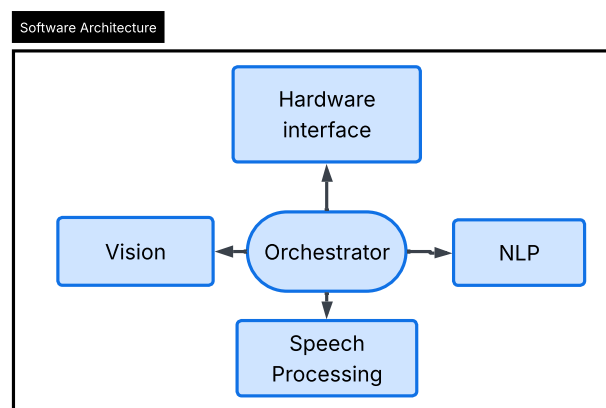


Figure 1. Software architecture based on ROS2 nodes.

The hardware interface manages sensor data acquisition and actuator control through dedicated nodes, while the vision module implements object detection using deep learning models such as YOLO and generates navigation commands based on visual and sensory information. Voice processing is performed by a node that integrates speech recognition, keyword activation, and speech synthesis, enabling natural interaction with the user in environments with or without connectivity.

This stage includes an orchestrator node that dynamically manages the operating modes. In education mode, voice processing is prioritized and input is provided by users through a microphone, whereas in navigation mode, the perception, locomotion, and voice processing modules are activated. Dynamic node management is achieved through process control, allowing ROS2 nodes to be activated or deactivated to reduce resource consumption on the embedded platform, a common strategy in resource-constrained robotic systems [25,26].

Figure 2 shows the complete ROS2 node-based structure. The system is organized into functional modules: the `usb_camera_node` acquires real-time visual data, which are processed by the `object_detector_node` using a YOLO-based model for object recognition. The `object_follower_node` integrates visual detections with distance data from the `ultra-sonic_node` and orientation data from the `imu_node` to generate motion commands. These commands are sent to the `motor_controller_node`, which converts them into PWM signals to drive the motors.

Additionally, the `voice_node` acts as an interaction hub by integrating speech recognition (VOSK, `vosk-model-small-es-0.42`), command interpretation, and text-to-speech synthesis, thereby enabling natural communication with the user. This node also interfaces with the orchestrator to trigger mode transitions based on detected commands. Such modular ROS2 node-based architectures are widely used to decouple perception, control, and interaction processes in robotic systems, improving scalability and maintainability [11,12].

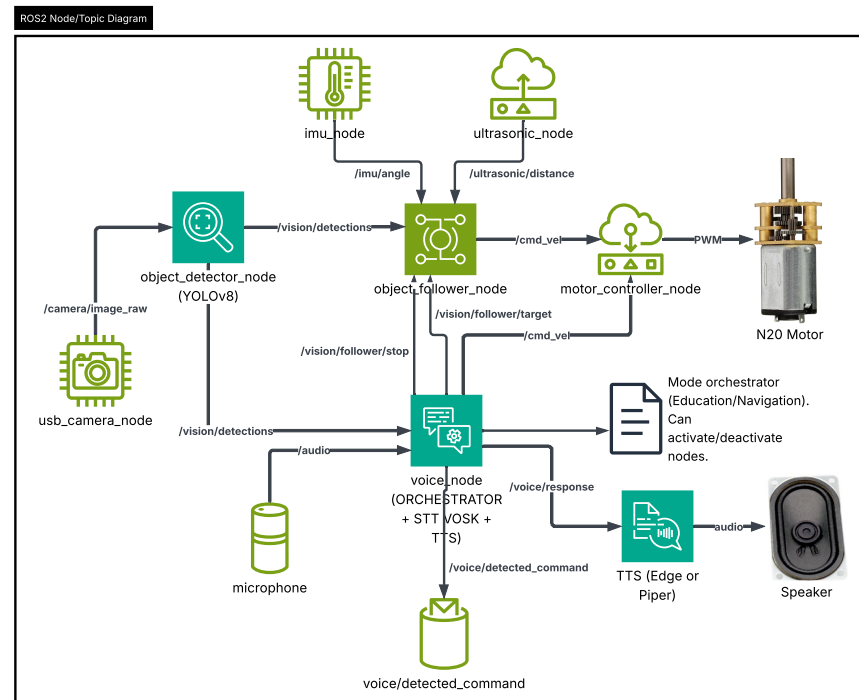


Figure 2. ROS2 node diagram for the education and navigation modes.

2.2. Hardware Interface

The robot chassis was fully designed using the open-source Blender v4.5.1 LTS software. An iterative prototyping approach was followed, in which preliminary versions were printed in PETG to verify component tolerances and fit until the final dimensions were achieved. All parts were assembled using standard screws, as shown in Figure 3.

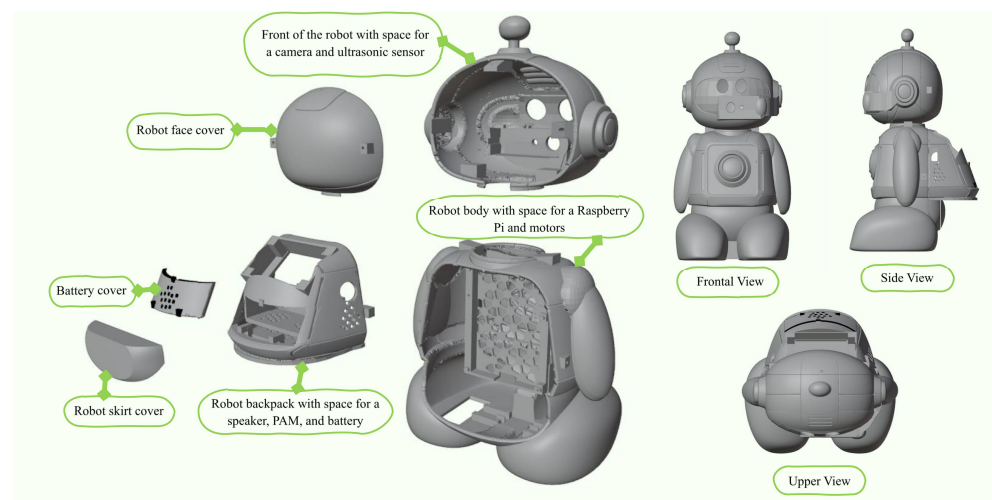


Figure 3. 3D-printed chassis design: (left) exploded view showing the individual components and their designated spaces for electronic modules; (right) orthographic views of the fully assembled robot.

2.3. Embedded Hardware

The robot's architecture is shown in Figure 4. The system is based on a Raspberry Pi 4B (4 GB RAM) (Raspberry Pi Holdings, Cambridge, UK) running Ubuntu 22.04 Server, which was selected for its compatibility with ROS2 and its edge-computing capabilities.

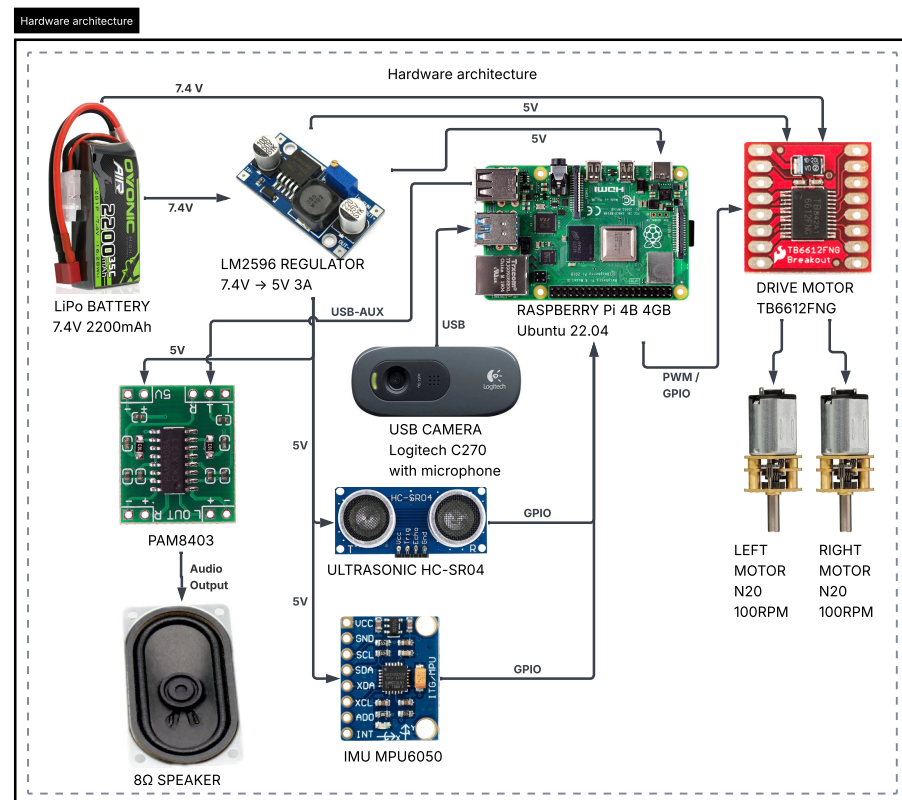


Figure 4. Embedded hardware architecture diagram of the mobile robot.

Locomotion is implemented using a differential drive system with N20 DC motors, which provide a compact size and sufficient torque for indoor mobile robots. These motors are controlled by a TB6612FNG driver, which enables bidirectional control and PWM-based speed regulation. A ROS2 node translates high-level velocity commands into low-level PWM signals, facilitating modular control within the robot architecture [11].

Table 2 shows the complete hardware bill of materials with individual component costs.

Table 2. Hardware bill of materials and component costs.

Component	Model	Manufacturer	Cost (USD)
Computing unit	Raspberry Pi 4B (4 GB)	Raspberry Pi Holdings, Cambridge, UK	63.36
Motor driver	TB6612FNG	Toshiba, Kawasaki, Japan	6.00
IMU	MPU6050	TDK InvenSense, San Jose, CA, USA	6.00
Camera	Logitech C270 USB	Logitech, Lausanne, Switzerland	23.18
Audio amplifier	PAM8403	Diodes Incorporated, Plano, TX, USA	2.05
Voltage regulator	LM2596 DC-DC	Texas Instruments, Dallas, TX, USA	2.49
Drive motors	N20 DC 100 RPM ×2	Generic component	15.98
Ultrasonic sensor	HC-SR04	Generic component	3.10
Battery	LiPo 7.4 V 2200 mAh ×2	Generic component	28.49
Speaker	8 Ω	Generic component	2.99
Wheels ×2	—	Generic component	4.10
Caster wheels ×2	—	Generic component	4.10
Total			161.84

Differential-Drive Kinematics and Motor Control

The robot employs a differential drive locomotion scheme controlled by two independent N20 DC motors actuated through a TB6612FNG driver. Let v and ω denote the desired

linear and angular velocities of the robot, respectively, as received through the `cmd_vel` ROS2 topic. The corresponding linear velocities of the left and right wheels, v_L and v_R , are obtained from the inverse kinematics of the differential drive model:

$$v_L = v - \frac{\omega L}{2}, \quad v_R = v + \frac{\omega L}{2} \quad (1)$$

where L is the distance between the wheel centers (track width). The corresponding wheel angular velocities are computed as:

$$\omega_L = \frac{v_L}{r}, \quad \omega_R = \frac{v_R}{r} \quad (2)$$

where r is the wheel radius. For the developed platform, $r = 0.0215$ m (43 mm wheel diameter) and $L = 0.10$ m, based on physical measurements of the assembled chassis.

The N20 motors used in this work have a nominal speed of 100 RPM, corresponding to a theoretical maximum wheel angular velocity of approximately 10.47 rad/s. However, due to mechanical load, friction, and voltage drop under operating conditions, an empirically calibrated maximum angular velocity, $\omega_{max} = 7.5$ rad/s, was used to map the computed wheel angular velocities to PWM duty-cycle commands:

$$\text{PWM}_i = \text{clip}\left(\frac{\omega_i}{\omega_{max}} \times 100, -100, 100\right), \quad i \in \{L, R\} \quad (3)$$

This PWM value, ranging from -100% to 100% , is sent to the TB6612FNG motor driver together with the corresponding direction signals, enabling proportional and bidirectional control of each wheel independently. This formulation allows the robot to execute both straight-line motion ($v_L = v_R$) and in-place or arc turns ($v_L \neq v_R$) by appropriately combining the linear and angular velocity commands received through the ROS2 `cmd_vel` interface.

For motion stabilization and orientation estimation, an MPU6050 IMU is integrated. This sensor provides accelerometer and gyroscope data, allowing basic trajectory correction and short-term pose estimation suitable for reactive navigation approaches in dynamic environments [20].

The perception system combines multiple sensors to enable multimodal environmental understanding. A USB camera enables visual perception and supports object detection through deep learning models, as commonly implemented in robotic vision systems [17]. Additionally, an HC-SR04 ultrasonic sensor is incorporated for short-range obstacle detection, providing consistent real-time distance measurements, particularly in indoor environments [20]. The IMU also contributes to perception by complementing motion awareness.

For human–robot interaction, audio output is implemented using a PAM8403-based audio amplifier connected to an $8\ \Omega$ speaker. This module enables efficient low-power sound amplification, which is suitable for embedded voice assistant systems [12].

The computational core of the system is a Raspberry Pi 4B, selected for its compatibility with ROS2 and its capability to perform edge-computing tasks such as vision processing and natural language interaction [5]. Power is supplied by a 7.4 V LiPo battery, which is regulated to 5 V using a DC–DC step-down converter (LM2596). This regulation ensures the stable voltage levels required by the embedded system and its peripherals. Such configurations are widely used in mobile robotic platforms to ensure autonomy and reliability [16].

2.4. Information Flow

The information flow depends on the operating mode. During navigation, voice commands are translated into movement actions by integrating visual and sensory perception. Figure 5 illustrates the data flow for the movement-command execution process. The sequence begins when the user issues a voice command, which is captured by the microphone and processed by the voice_node. This node performs speech recognition using VOSK and extracts the intended navigation command.

Once the command is interpreted, it is sent as a cmd_vel message to the object_follower_node, which is responsible for generating motion actions. This node integrates perception data from multiple sources, including visual detections from the object_detector_node, distance measurements from the ultrasonic_node, and orientation data from the imu_node, allowing the system to adapt movement decisions to the environment.

The resulting motion commands are then transmitted to the motor_controller_node, which converts the high-level velocity commands into PWM signals to control the N20 motors.

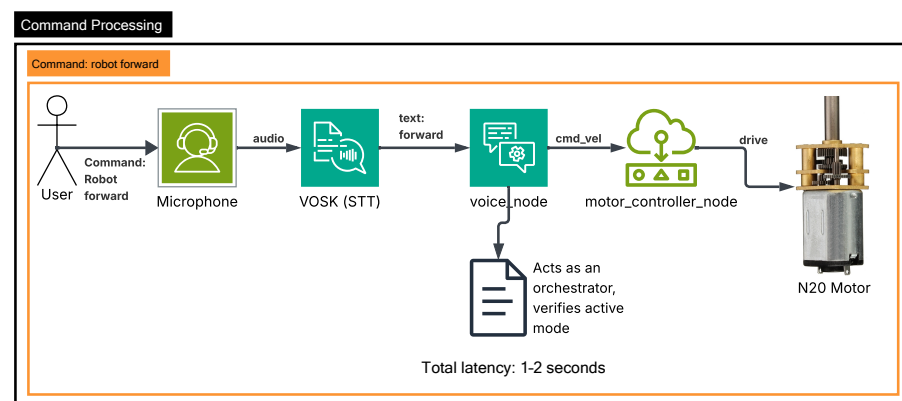


Figure 5. Data flow for the movement-command case.

Figures 6 and 7 illustrate the data flow for the academic question-answering process. The sequence begins when the user issues a spoken query, which is captured through the microphone and processed by the voice_node. This node performs speech recognition using VOSK, converts the audio input into text, and identifies the user's intent.

The recognized query is then forwarded to the natural language processing module, where it is evaluated using a hybrid architecture that includes a semantic cache, a local model, and an optional cloud-based model. The system first attempts to resolve the query using the semantic cache to reduce latency and computational cost; if no match is found, the request is processed by either the local or the cloud-based model, depending on availability and complexity [25,26].

The semantic cache uses the sentence-transformers library with the all-MiniLM-L6-v2 model to compute query embeddings and retrieve semantically similar cached responses. Formally, a query q is matched against cached entries c_i using cosine similarity:

$$\text{sim}(q, c_i) = \frac{\text{emb}(q) \cdot \text{emb}(c_i)}{\|\text{emb}(q)\| \|\text{emb}(c_i)\|} \quad (4)$$

where $\text{emb}(\cdot)$ denotes the sentence embedding produced by the all-MiniLM-L6-v2 model. A cache hit is triggered when $\text{sim}(q, c^*) \geq \theta = 0.7$, where $c^* = \arg \max_i \text{sim}(q, c_i)$. The local language model is Qwen2.5-1.5B (qwen2.5:1.5b-instruct-q4_0), deployed via Ollama with Q4_0 quantization to reduce the memory footprint on the Raspberry Pi 4B. The cloud-based

model is the Trinity API (Trinity Large Preview), which is accessed only when internet connectivity is available.

Once a response is generated, it is sent back to the voice_node, which performs text-to-speech synthesis using either an offline or an online TTS system. The resulting audio signal is then transmitted to the speaker, enabling the robot to provide a verbal response to the user.

Throughout this process, the orchestrator node manages the activation of the required modules, prioritizing voice processing while disabling unnecessary components to optimize resource use. This structured data flow enables efficient human–robot interaction and supports responsive behavior under embedded-system constraints [12].

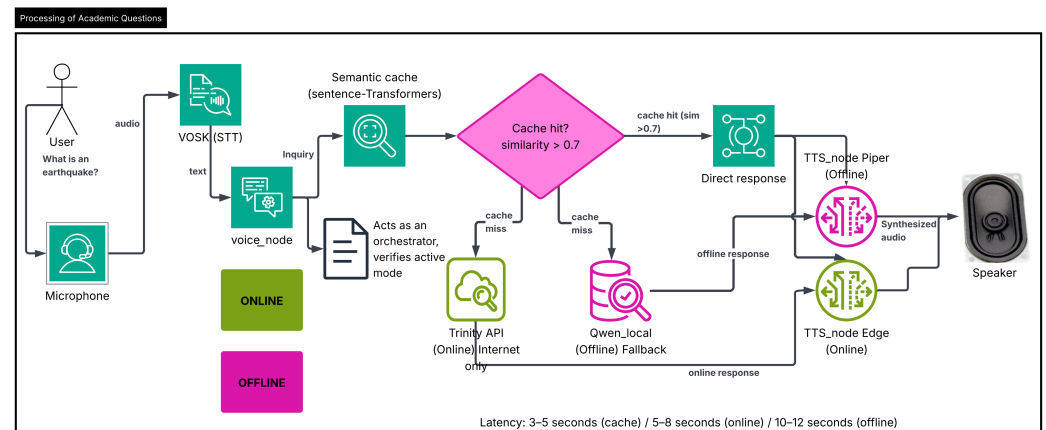


Figure 6. Data flow for the academic question answering case.

Therefore, this system was designed with a focus on Spanish-language operation, offline functionality, and energy efficiency. Optimization strategies include reduced logging, automatic system startup, and basic security mechanisms to ensure stability and autonomy in educational settings.

2.5. Experimental Design

The integrated system was evaluated in a university classroom under typical lighting and noise conditions, considering the education and navigation operating modes. The sensors were calibrated using standard procedures.

The ultrasonic sensor was evaluated by comparing its measurements with known distances and calculating the absolute error, relative error, and standard deviation.

The IMU was validated under stationary and controlled-motion conditions to estimate drift and noise.

The camera was calibrated using objects with known dimensions to establish the relationship between pixels and real-world size.

The YOLOv8n object detection model was configured with a confidence threshold of 0.25 and a publish rate of 2.0 Hz on the Raspberry Pi 4B (Raspberry Pi Holdings, Cambridge, UK). It was evaluated using a set of images representative of the environment, focusing on performance metrics and inference latency on the Raspberry Pi. Additionally, a specialized door detection model was trained using transfer learning on a dataset obtained from [37] and evaluated on a separate validation set of 62 images comprising a mixture of web-sourced and locally captured images, following a sourcing composition similar to that of the general object detection validation set.

The speech recognition system was evaluated under quiet and noisy conditions. Testing was conducted primarily with one speaker, with occasional additional trials involving two other speakers (male and female) to introduce some variability in voice characteristics.

All speakers were positioned approximately 30 cm from the microphone. Test queries combined navigation commands and academic questions to reflect realistic usage scenarios. Noisy conditions were generated by playing pre-recorded videos of people speaking at high volume from a television located directly above the robot, simulating background classroom chatter, while a mobile phone placed next to the microphone introduced additional electronic interference; sound pressure level was not quantitatively measured.

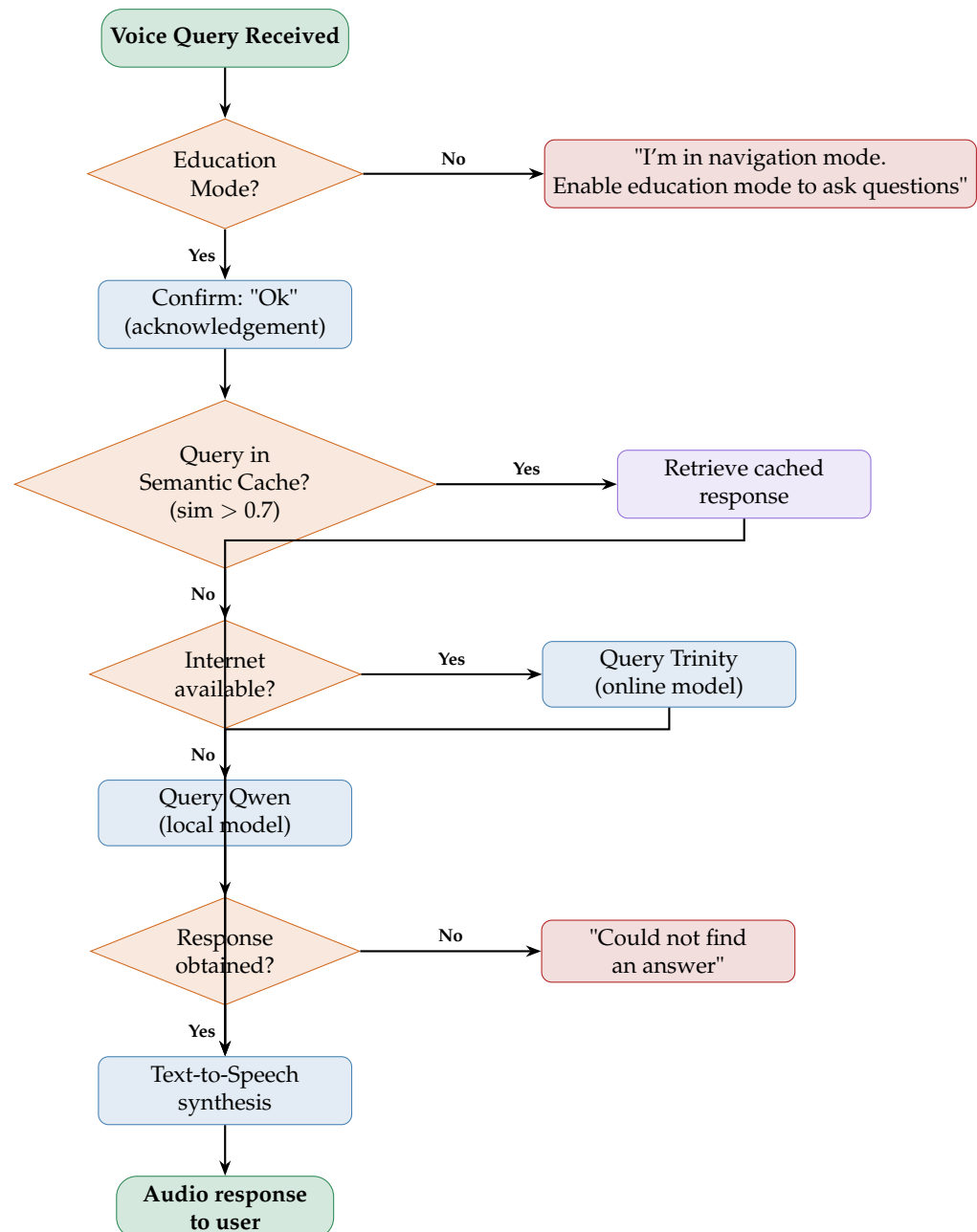


Figure 7. NLP query-routing algorithm flowchart. Queries are first checked against the semantic cache (similarity threshold > 0.7); upon a cache miss, routing depends on internet availability: the online Trinity model is prioritized when connectivity is detected, while the local Qwen model serves as a fallback for offline operation.

Speech synthesis was analyzed in online and offline modes by measuring the response time from text generation to the start of audio playback. The natural language processing module was evaluated in three configurations: semantic cache, local model, and cloud model, comparing accuracy and latency for academic queries.

The metrics used for object detection and speech recognition included accuracy, latency, and error, depending on the component analyzed. Finally, power consumption was evaluated through a single continuous discharge test per operating mode using a 7.4 V LiPo battery with a nominal capacity of 2200 mAh. Average current was calculated based on the battery's initial and final voltage variation under typical operating conditions for each mode.

3. Results

This section presents the experimental results obtained from the evaluation of the proposed system. Figure 8 shows the developed physical prototype, including the layout of the sensors, actuators, and computing unit.

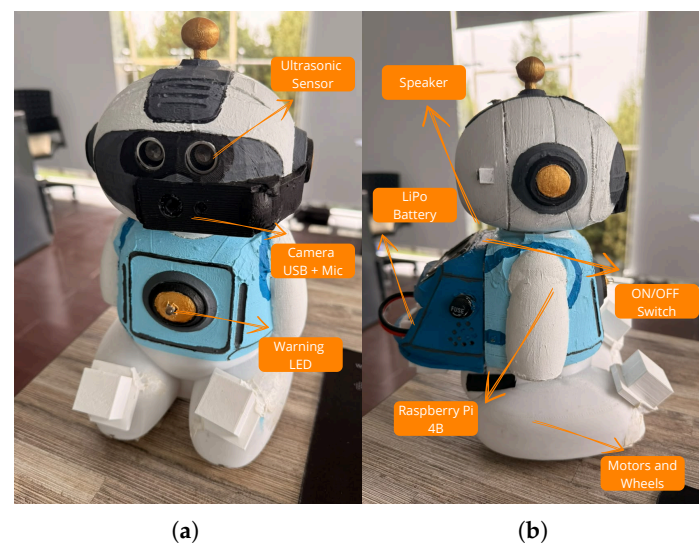


Figure 8. Physical prototype of the developed educational robot. (a) Front view showing the ultrasonic sensor (HC-SR04), USB camera with integrated microphone (Logitech C270), and warning LED; (b) side view showing the speaker (8 Ω), ON/OFF switch, LiPo battery, N20 motors and wheels, and Raspberry Pi 4B computing unit (located internally in the torso).

3.1. Sensor Characterization

3.1.1. Ultrasonic Sensor

Measurements from the ultrasonic sensor show distance-dependent behavior. At 5 cm, an overestimation is observed (70% error), which is attributed to near-field effects. Between 10 and 25 cm, the error remains below 2%, indicating high accuracy at short distances. However, beyond 50 cm, increasing underestimation and greater dispersion are observed, with errors reaching up to 19.6%. The results presented in Table 3 indicate that the sensor is reliable for short-distance obstacle detection (<40 cm), but limited at longer distances.

Table 3. Results of the ultrasonic sensor characterization.

Target Distance	Mean Measured Distance	Standard Deviation	Absolute Error	Relative Error
5 cm	8.5 cm	± 0.52 cm	3.5 cm	70%
10 cm	9.8 cm	± 1.06 cm	0.2 cm	2%
25 cm	25.3 cm	± 3.80 cm	0.3 cm	1.2%
50 cm	40.2 cm	± 11.40 cm	9.8 cm	19.6%
100 cm	91.1 cm	± 19.59 cm	8.9 cm	8.9%

3.1.2. IMU Sensor

The IMU sensor shows low noise at rest ($\leq 0.14^\circ$) and a drift of $0.45^\circ/\text{min}$. During 90° rotations, the average error was 6.2° , which remained within an acceptable range for basic navigation. Overall, the sensor exhibits short-term stability and an adequate response for indoor applications. These results are presented in Table 4.

Table 4. Results of the IMU sensor characterization.

Metric	Value
Quiescent noise (Roll)	$\pm 0.05^\circ$
Quiescent noise (Pitch)	$\pm 0.14^\circ$
Quiescent noise (Yaw)	$\pm 0.12^\circ$
Quiescent drift (60 s)	0.45°
Stability in fixed position (30 s)	$\sigma = 0.57^\circ$
Accuracy in 90° rotation	83.8° (error 6.2°)
Roll measurement range	$\pm 35^\circ$
Pitch measurement range	-28° to $+40^\circ$
Sampling frequency	9.7 Hz

3.1.3. USB Camera

The camera calibration yielded an average error of 0.23 cm (6%). The largest deviations were associated with variations in the orientation and effective distance of the objects, as shown in Figure 9, where the size of the 10-cent coin is measured in pixels and then compared with its actual size.



Figure 9. Pixel-based measurement of the 10-cent coin.

This level of accuracy is sufficient for detection and size-estimation tasks in controlled environments, as shown in Table 5.

Table 5. USB camera test results.

Object	Actual Size	Estimated Size	Error	Error (%)
ID-card	8.50 cm	9.20 cm	0.70 cm	8.2%
5-cent coin	2.10 cm	2.19 cm	0.09 cm	4.3%
10-cent coin	1.75 cm	1.82 cm	0.07 cm	4%
25-cent coin	2.35 cm	2.52 cm	0.17 cm	7.2%
50-cent coin	3 cm	3.18 cm	0.18 cm	6%
1-dollar coin	2.60 cm	2.77 cm	0.17 cm	6.5%

3.1.4. Power Consumption and Battery Life

Power consumption reveals key differences between operating modes. Education mode had a current consumption of 1.46 A (7.3 W) and a runtime of 96 min, while navigation mode increased consumption to 1.85 A (9.25 W), reducing runtime to 75.6 min (21% less). This confirms the effectiveness of the mode-selection strategy for optimizing energy use, as shown in Table 6.

Table 6. Energy consumption results.

Operating Mode	Initial Voltage	Final Voltage	Average Current (A)	Average Power (W)	Estimated Runtime
Education	8.42 V	8.01 V	1.46 A	7.3 W	1.6 = 96 min
Navigation	8.42 V	7.90 V	1.85 A	9.25 W	1.26 = 75.6 min

3.1.5. Object Detection Performance

The YOLOv8n model was evaluated for the recognition of three object categories relevant to classroom environments: person, chair, and backpack, which are commonly present in classrooms and toward which the robot could move. Objects such as pencil cases or erasers were omitted because they are not commonly found on the floor and would be detected by the sensor as obstacles. Because the pretrained YOLOv8n model follows the COCO class taxonomy, in which backpacks were occasionally classified under the visually similar “suitcase” category, detections corresponding to either label were considered valid identifications of the backpack class for evaluation purposes. The model achieved precision and recall values of 97.5%, with an accuracy of 97% and an average latency of 1064.4 ms, corresponding to 0.9 FPS on the Raspberry Pi 4B. It should be noted that the 0.9 FPS inference rate is not suitable for real-time dynamic obstacle tracking; camera-based detection serves primarily for object and target identification, while reactive obstacle avoidance is handled independently by the ultrasonic sensor node.

The metrics are summarized in Table 7.

In Figure 10, the confusion matrix shows only two errors (one FP and one FN), indicating promising performance. The class-wise analysis shows a recall of 100% for people and chairs, while a single false negative was observed for backpacks. A single false positive was detected in empty scenarios. Overall, the model shows promising performance under prototype-level evaluation in controlled classroom conditions.

3.1.6. Performance of the Specialized Door Detector

The specialized model achieved a recall of 100% and a precision of 58.7%, indicating high sensitivity but also the presence of false positives, as presented in Table 8. It is important to note that the system architecture prevents false-positive door detections from

causing unintended movement: the robot does not autonomously navigate toward detected doors. All movement commands are issued explicitly by the user via voice after reviewing the detection output. False positives observed during evaluation were primarily caused by door-shaped objects, such as windows, cabinets, and wall panels with similar rectangular geometry, consistent with the limited training dataset size.

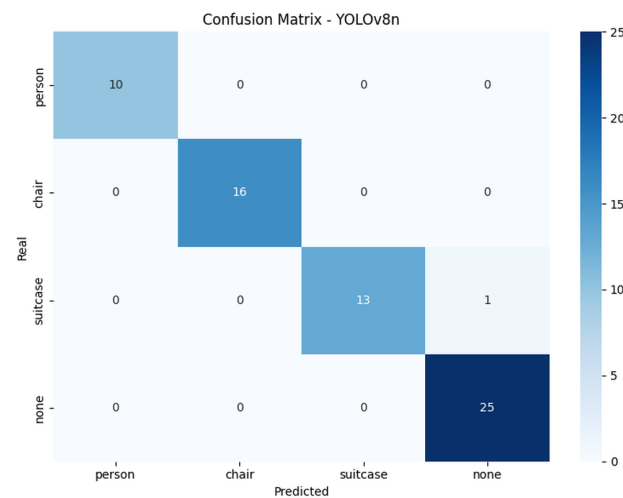


Figure 10. YOLOv8n confusion matrix with object-classification results. The “suitcase” label corresponds to the backpack object class, as detections of backpacks were occasionally classified under this visually similar COCO category (see main text for details).

Table 7. YOLOv8n object detection test results.

Metric	Value
Precision	97.5%
Recall	97.5%
F1-score	0.975
Accuracy	97%
Average latency	1064.4 ms
Average FPS	0.9
Total samples	66

This behavior is suitable for navigation, where it is preferable to detect possible routes even when false detections occur. However, confusion with similar structures was observed and is attributable to the limited size of the dataset [38].

Table 8. Door detection test results.

Metric	Value
Precision	58.7%
Recall	100%
F1-score	0.740
Accuracy	69.4%
Total samples	62

3.1.7. Evaluation of the Speech Recognition System

The VOSK system achieved 98% accuracy under both quiet and noisy conditions. Average latency increased from 935.6 ms to 1095.8 ms ($\approx 17\%$) in the presence of noise. The evaluation set comprised navigation commands (e.g., “robot adelante”, “robot para”) and academic questions across mathematics, language, and general knowledge domains (see Section 2.5 for evaluation details and Section 3.1.9 for representative examples).

These results indicate strong accuracy performance, with noise having only a moderate impact on latency, as shown in Table 9.

Table 9. Evaluation results of the VOSK system under quiet and noisy conditions.

Metric	No Noise	With Noise
Precision	98%	98%
Minimum latency	387.3 ms	366.1 ms
Maximum latency	1392.2 ms	1463.2 ms
Mean (ms)	935.6 $\pm$ 243.5	1095.8 $\pm$ 276.2
Median (ms)	998.2	1149.8
Total tests	50	50

3.1.8. Latency of Speech Synthesis Systems

Thirty latency measurements were collected for each speech-synthesis system. Table 10 shows these results:

Table 10. Latency results for the Edge TTS (online) and Piper TTS (offline) systems.

Metric	Edge (Online)	Piper (Offline)
Mean (ms)	3890.3 $\pm$ 996.6	4817.6 $\pm$ 583.0
Minimum (ms)	3280.4	4215.5
Maximum (ms)	8658.1	7305.9
Median (ms)	3674.6	4676.3
Interquartile range (ms)	3539.6–3790.3	4576.8–4815.8
95% confidence interval (ms)	[3518.1, 4262.4]	[4599.9, 5035.3]

The Edge TTS (online) system had an average latency of 3.89 s, which was approximately 19% faster than Piper (offline), at 4.82 s. However, Piper showed less variability in response times, indicating more stable performance. The latency difference was notable, and the peak values observed for Edge TTS are likely due to fluctuations in the internet connection. These results highlight a trade-off between speed and stability, as well as between online and offline operation.

3.1.9. Performance of Natural Language Processing Models

The queries were evenly distributed among mathematics, language, and general knowledge questions. Representative examples include arithmetic operations (e.g., “What is 7 times 8?”), grammar-related questions (e.g., “What is the antonym of “tall”?”), and general knowledge questions related to the local educational curriculum (e.g., “What is photosynthesis?” and “Who was Eloy Alfaro?”), most of which were resolved directly by the semantic cache. Queries not present in the cache, such as “Who was Rafael Correa?” or “What is a radio?”, were routed to the local or cloud-based language models, illustrating the system’s ability to handle both recurrent and novel academic questions.

To clarify the evaluation protocol, the 30 questions used to assess the semantic cache consisted exclusively of exact matches to entries present in the cache database, yielding 100% accuracy. The separate evaluation sets for the Trinity and Qwen models included semantically similar queries containing named entities shared with cached questions (e.g., “When did Eloy Alfaro die?” alongside the cached “Who was Eloy Alfaro?”) to verify that the routing logic correctly identifies cache misses and does not incorrectly retrieve cached responses for related but distinct queries. Genuinely novel questions absent from the cache, such as “Who is the president of Ecuador?”, were also included in these sets. Query routing follows the algorithm described in Figure 7.

The results are shown in Table 11:

Table 11. Evaluation results for the natural language processing systems.

Metric	Semantic Cache	Trinity (Online)	Qwen (Local)
Correct answers	30/30	28/30	23/30
Precision	100%	93.3%	76.7%
Minimum latency	2800.9 ms	6041.9 ms	8784.8 ms
Maximum latency	8564.3 ms	15,607.8 ms	47,588.1 ms
Mean (ms)	3795.5 ± 1412.0	8251.8 ± 1905.8	17,294.6 ± 9385.3
Median (ms)	3400.3	7596.8	13,620.9
Interquartile range (ms)	3202.9–3612.6	7271.1–8896.0	11,588.2–18,247.5
Total questions	30	30	30

Note that the 100% accuracy reported for the semantic cache reflects performance on exact-match queries present in the cache database, and does not represent general question-answering capability; the Trinity and Qwen evaluations included semantically similar and novel questions to test routing correctness.

The NLP system showed notable differences among the three evaluated modalities. The semantic cache achieved 100% accuracy with an average latency of 3.8 s; the online Trinity model achieved 93.3% accuracy with an average latency of 8.3 s; and the local Qwen model achieved 76.7% accuracy with a considerably higher latency of 17.3 s. These results demonstrate that the semantic cache is highly effective for known queries, whereas language models provide greater flexibility at the cost of higher latency. However, the language models showed limitations in specific domain knowledge and high variability in response times, attributable to hardware constraints. In addition to technical performance, the system’s economic feasibility was evaluated. The total implementation cost is below \$250, including the computing unit, sensors, actuators, and electronic components. Compared with commercial platforms or open-source educational robots [33], the proposed system integrates additional capabilities such as offline natural language processing in Spanish and energy optimization through operating modes, while maintaining a considerably lower cost without compromising system performance.

3.2. Navigation Performance Evaluation

To complement the component-level characterization, three integrated navigation tests were conducted in navigation mode, each comprising 20 trials. These tests evaluate the system-level behavior of the robot under realistic operating conditions and are summarized in Table 12.

The first test evaluated voice-command-based motion control by measuring whether the robot correctly executed movement and stopped upon explicit voice commands. Out of 20 trials, 15 were successful (75%). The five failures were attributed to two causes: in three cases, the stop command was issued before the preceding movement command had

completed processing, causing it to be ignored; in two cases, the voice level was insufficient for the VOSK recognizer to detect the command.

The second test evaluated reactive obstacle avoidance using the HC-SR04 ultrasonic sensor. The robot successfully detected and halted before obstacles in all 20 trials (100%), confirming the reliability of the ultrasonic-based reactive behavior within the characterization range reported in Section 3.1.1 (<40 cm).

The third test evaluated visual object-following behavior in navigation mode. Out of 20 trials, the robot initiated movement in 18 cases (90% following rate). In 14 trials, the target object was correctly classified and followed. In four trials, the object was misclassified under an adjacent COCO category (consistent with the behavior described in Section 3.1.5), but the robot still followed the detected bounding box, resulting in correct locomotion despite the classification error. In the remaining two trials, the detector produced no detection, preventing any robot movement.

Table 12. Navigation-performance evaluation results (20 trials per test).

Test	Successful	Failures	Success Rate
Voice command response (stop/go)	15/20	3 timing overlap; 2 low voice level	75%
Obstacle avoidance (ultrasonic)	20/20	—	100%
Object following (visual)	18/20	2 no detection	90% *

* 14/20 with correct object classification; 4/20 with object misclassification but correct following behavior.

4. Discussion

The results demonstrated the feasibility of integrating multiple capabilities—including voice interaction, reactive navigation, and natural language processing—into a single Raspberry Pi-based embedded device under real-world conditions, validating the accuracy, latency, and power consumption metrics established as objectives. Although current studies often focus on individual capabilities [12,20], the proposed approach demonstrates that a unified architecture can operate with limited computational resources, enabling implementation in educational environments, as has recently been explored in low-cost robotic platforms [35].

The decision to omit SLAM-based navigation represents a trade-off between accuracy and computational efficiency. Although this limits global localization capabilities, the reactive navigation approach was more suitable for dynamic indoor environments [14,17], such as classrooms with frequently changing objects. Similarly, the use of a semantic cache [26,27] considerably reduces response latency for recurrent requests, although its effectiveness depends on how frequently those queries are repeated.

The low inference speed of 0.9 FPS observed for the YOLOv8n model represents a limitation for continuous dynamic obstacle tracking. However, the system compensates for this limitation by combining vision-based detection with the HC-SR04 ultrasonic sensor for immediate obstacle stopping, which operates independently of the camera pipeline. Camera-based detection primarily supports object recognition and navigation-target identification, while reactive obstacle avoidance is handled by the ultrasonic node, which is not subject to the same computational constraints.

Additionally, the 6.2° rotational error observed in the IMU may accumulate over extended navigation sequences, representing a known limitation for tasks that require precise heading control. This further motivates the future integration of visual odometry or a lightweight SLAM approach to correct drift over time.

The 75% success rate for voice command-based motion control reflects a timing sensitivity in the current sequential command processing pipeline: when a stop command is issued before the preceding movement command has completed processing, it may be ignored. This has implications for safe operation and should be addressed in future work through stop-command prioritization, an interrupt-based mechanism, or a dedicated command queue to ensure safety-critical stop signals are always processed regardless of system state.

Regarding natural language processing, large-scale LLMs remain impractical for embedded platforms [22], while smaller models often lack sufficient reasoning capabilities [23]. The proposed hybrid architecture offers a balanced solution that combines local inference with basic cloud processing to maintain functionality under varying connectivity conditions, a strategy similar to that proposed in recent work on resource optimization at the edge [25]. Power consumption also plays a critical role in embedded robotic systems. The implementation of dual operating modes (education and navigation) proved effective in extending battery life by selectively activating system components according to task requirements, representing a practical optimization aligned with real-world usage scenarios and recent efforts to reduce energy consumption in autonomous navigation [16].

It is important to note that the experimental evaluation presented in this work constitutes a technical validation of the integrated platform, focused on measuring the functional performance of individual components and their interaction under controlled classroom conditions. The objective of this phase was to demonstrate that the system is technically feasible and operationally promising as a prerequisite for educational deployment. Evaluating the robot's effectiveness as a pedagogical tool—including user studies with students and teachers, measurement of learning outcomes, and assessment of engagement—represents a distinct and subsequent research phase that falls outside the scope of this technical validation study. Additional limitations include the use of relatively small datasets for computer-vision tasks, reliance on internet connectivity for complex queries [5], increased latency in local language models [7], and the absence of advanced navigation strategies such as SLAM [14,17], which limits the system to basic reactive behaviors.

5. Conclusions and Future Work

This paper presented the development of an ROS2-based educational mobile robot implemented on a Raspberry Pi 4B, integrating Spanish voice interaction (online and offline), reactive navigation, and a hybrid natural language processing architecture. The system demonstrates that Spanish voice interaction, reactive navigation, academic question answering, and resource-aware operation can be integrated into a single low-cost edge robotic platform for educational environments. Experimental results validate system performance through key metrics. The speech recognition module achieved 98% accuracy under both quiet and noisy conditions. The object detection system reached an F1-score of 0.975, while the specialized door detection model achieved 100% recall, prioritizing safe navigation. Additionally, the semantic cache resolved queries with an average latency of 3.8 s, considerably outperforming the local and cloud-based models. In terms of energy consumption, the system exhibited an average current of 1.46 A in education mode and 1.85 A in navigation mode. These results confirm the effectiveness of the proposed dual-mode strategy in optimizing energy use while maintaining system performance.

Future work will follow two complementary directions. The first direction addresses technical improvements: integrating more efficient local language models, expanding the vision and speech datasets, and incorporating lightweight navigation strategies such as visual odometry or a reduced-footprint SLAM approach to correct IMU drift over extended trajectories. The second direction addresses educational deployment: once the platform's

technical robustness is confirmed through integrated navigation trials, the system will be validated in real classroom settings through structured user studies with students and teachers, measuring interaction quality, academic support effectiveness, and engagement. This two-phase approach ensures that educational deployment is grounded in a technically reliable platform.

The novelty of the proposed system lies not in the individual component technologies, but in their integration into a unified, low-cost platform explicitly designed for Latin American Spanish educational environments. The combination of offline academic question answering through a curriculum-aligned semantic cache, reactive sensor-based navigation, dual-mode energy management, and classroom-validated experimental results constitutes a contribution not previously demonstrated in a single embedded robotic system of this cost and deployment context.

Author Contributions: Conceptualization, S.A.A. and R.I.-J.; methodology, S.A.A. and J.V.-A.; software, S.A.A.; validation, S.A.A. and R.I.-J.; formal analysis, S.A.A.; investigation, S.A.A.; resources, S.A.A. and N.C.B.; data curation, S.A.A.; writing—original draft preparation, S.A.A.; writing—review and editing, R.I.-J., N.C.B. and J.V.-A.; visualization, S.A.A. and J.V.-A.; supervision, R.I.-J. and J.V.-A.; project administration, R.I.-J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code, ROS2 launch files, configuration files, and semantic-cache database supporting this study are publicly available at <https://github.com/SesSic/MobileRobotAssistantNLPROS2> (accessed on 4 July 2026). The trained door detection model weights, evaluation scripts, and validation image sets used in this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors acknowledge the support of the Facultad de Ingenierías, Maestría en Robótica y Automatización Industrial, Universidad Tecnológica Indoamérica, Ambato, Ecuador, in the development of this research. During the preparation of this manuscript, the authors used Claude Sonnet 4.6 (Anthropic) for language editing, including grammar correction and improvement of sentence clarity in English. The authors reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tselegkaridis, S.; Sapounidis, T. Exploring the Features of Educational Robotics and STEM Research in Primary Education: A Systematic Literature Review. *Educ. Sci.* **2022**, *12*, 305. [CrossRef]
2. Park, S.; Min, D.; Nguyen, T.A.; Choi, E. Metaverse Security: LLM-Based Malicious Code Detection Through Token Optimization. In Proceedings of the 2025 International Conference on Metaverse Computing, Networking and Applications (MetaCom), Seoul, Republic of Korea, 27–29 August 2025; pp. 316–323. [CrossRef]
3. Smith, E.M.; Graham, D.; Morgan, C.; MacLachlan, M. Artificial intelligence and assistive technology: Risks, rewards, challenges, and opportunities. *Assist. Technol.* **2023**, *35*, 375–377. [CrossRef] [PubMed]
4. Varela-Aldás, J.; Collay, W.; Del-Valle-Soto, C.; Palacios-Navarro, G. Does the Use of Dedicated Mobile Devices in Magnetism Classes Improve Student Learning? *Front. Educ.* **2026**, *11*, 1796723. [CrossRef]
5. Mekonnen, F.Y.; Al Bataineh, M.F.; Abdoun, D.A.; Serag, A.; Tamiru, K.T.; Abula, W.; Darota, S. Development of a Fully Autonomous Offline Assistive System for Visually Impaired Individuals: A Privacy-First Approach. *Sensors* **2025**, *25*, 6006. [CrossRef] [PubMed]
6. Dinh, H.P.T.; Patamia, R.A.; Liu, M.; Cosgun, A. Benchmarking the Responsiveness of Open-Source Text-to-Speech Systems. *Computers* **2025**, *14*, 406. [CrossRef]
7. Pinto-Bernal, M.; Biondina, M.; Belpaeme, T. Designing Social Robots with LLMs for Engaging Human Interaction. *Appl. Sci.* **2025**, *15*, 6377. [CrossRef]

8. Herath, D.; Grant, J.B.; Rodriguez, A.; Davis, J.L. First impressions of a humanoid social robot with natural language capabilities. *Sci. Rep.* **2025**, *15*, 19715. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Fadel, W.; Bouchentouf, T.; Buvet, P.-A.; Bourja, O. Adapting Off-the-Shelf Speech Recognition Systems for Novel Words. *Information* **2023**, *14*, 179. [\[CrossRef\]](#)
10. Di Leo, S.; De Cicco, L.; Mascolo, S. Real-Time Speech-to-Text on Edge: A Prototype System for Ultra-Low Latency Communication with AI-Powered NLP. *Information* **2025**, *16*, 685. [\[CrossRef\]](#)
11. Gupta, S.; Mamodiya, U.; Al-Gburi, A.J.A. Speech Recognition-Based Wireless Control System for Mobile Robotics: Design, Implementation, and Analysis. *Automation* **2025**, *6*, 25. [\[CrossRef\]](#)
12. Lazzaroni, L.; Bellotti, F.; Berta, R. An embedded end-to-end voice assistant. *Eng. Appl. Artif. Intell.* **2024**, *136*, 108998. [\[CrossRef\]](#)
13. Espada, J.P.; Qiu, S.Y.; Crespo, R.G.; Carús, J.L. Leveraging large language models for autonomous robotic mapping and navigation. *Int. J. Adv. Robot. Syst.* **2025**, 1–18. [\[CrossRef\]](#)
14. Al-Tawil, B.; Hempel, T.; Abdelrahman, A.; Al-Hamadi, A. A review of visual SLAM for robotics: Evolution, properties, and future applications. *Front. Robot. AI* **2024**, *11*, 1347985. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Li, Y.; Na, L.; Liang, X.; An, Q. Indoor Dynamic Environment Mapping Based on Semantic Fusion and Hierarchical Filtering. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 236. [\[CrossRef\]](#)
16. Teso-Fz-Betoño, D.; Aramendia, I.; Ramos-Hernanz, J.A.; Caballero-Martin, D.; Affou, H.; Lopez-Guede, J.M. Optimization of Energy Efficiency with a Predictive Dynamic Window Approach for Mobile Robot Navigation. *Sustainability* **2025**, *17*, 4526. [\[CrossRef\]](#)
17. Shen, Y.; Zhang, X. A dynamic SLAM system with YOLOv7 segmentation and geometric constraints for indoor environments. *Robotica* **2025**, *43*, 2527–2545. [\[CrossRef\]](#)
18. Sasaki, M.; Tsuda, Y.; Matsushita, K. Development of Autonomous Mobile Robot with 3DLidar Self-Localization Function Using Layout Map. *Electronics* **2024**, *13*, 1082. [\[CrossRef\]](#)
19. Elwaly, A.; Abdellatif, A.; El-Shaer, Y. New Eldercare Robot with Path-Planning and Fall-Detection Capabilities. *Appl. Sci.* **2024**, *14*, 2374. [\[CrossRef\]](#)
20. Medina-Santiago, A.; Morales-Rosales, L.A.; Hernández-Gracidas, C.A.; Algreto-Badillo, I.; Pano-Azucena, A.D.; Torres, J.A.O. Reactive Obstacle-Avoidance Systems for Wheeled Mobile Robots Based on Artificial Intelligence. *Appl. Sci.* **2021**, *11*, 6468. [\[CrossRef\]](#)
21. Molina-Leal, A.; Gómez-Espinosa, A.; Cabello, J.A.E.; Cuan-Urquizo, E.; Cruz-Ramírez, S.R. Trajectory Planning for a Mobile Robot in a Dynamic Environment Using an LSTM Neural Network. *Appl. Sci.* **2021**, *11*, 10689. [\[CrossRef\]](#)
22. Wang, R.; Gao, Z.; Zhang, L.; Yue, S.; Gao, Z. Empowering large language models to edge intelligence: A survey of edge efficient LLMs and techniques. *Comput. Sci. Rev.* **2025**, *57*, 100755. [\[CrossRef\]](#)
23. Jang, S.; Yang, S.; Choi, C. Analyzing Quantized Small Language Models for Efficient Edge Deployment. *J. Korean Inst. Commun. Inf. Sci.* **2025**, *50*, 1364–1380. [\[CrossRef\]](#)
24. Lu, Z.; Sun, C.; Li, X. Multimodal Large Language Model-Enabled Machine Intelligent Fault Diagnosis Method with Non-Contact Dynamic Vision Data. *Sensors* **2025**, *25*, 5898. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Tuli, S.; Casale, G.; Roveri, M. SELA: Smart Edge LLM Agent to Optimize Response Trade-offs of AI Assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2025**, *9*, 1–18. [\[CrossRef\]](#)
26. Haqiq, K.; Jahan, M.V.; Farimani, S.A.; Masoom, S.M.F. MinCache: A hybrid cache system for efficient chatbots with hierarchical embedding matching and LLM. *Future Gener. Comput. Syst.* **2025**, *170*, 107822. [\[CrossRef\]](#)
27. Song, C.; Kang, S. Snip-Cache: A code snippet caching system for LLM-based command-driven IoT systems. *Internet Things* **2026**, *36*, 101852. [\[CrossRef\]](#)
28. Asuzu, K.; Singh, H.; Idrissi, M. Human–robot interaction through joint robot planning with large language models. *Intell. Serv. Robot.* **2025**, *18*, 261–277. [\[CrossRef\]](#)
29. Yang, Z.; Du, Y.; Liu, D.; Zhao, K.; Cong, M. A human-robot interaction system for automated chemical experiments based on vision and natural language processing semantics. *Eng. Appl. Artif. Intell.* **2025**, *146*, 110226. [\[CrossRef\]](#)
30. Kotei, E.; Thirunavukarasu, R. A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. *Information* **2023**, *14*, 187. [\[CrossRef\]](#)
31. Liu, G.; Liu, Y.; Wang, J.; Du, H.; Niyato, D.; Kang, J.; Xiong, Z. Adaptive Contextual Caching for Mobile-Edge Large Language Model Service: A Proposed Framework. *IEEE Veh. Technol. Mag.* **2025**, early access. [\[CrossRef\]](#)
32. Zhu, Y.; Jia, J.; Dong, M.; Qi, J. SentireCache: Accelerate Sentiment Classification with Saliency-Based Caching. *IEEE Trans. Affect. Comput.* **2025**, *16*, 1349–1361. [\[CrossRef\]](#)
33. Baksh, F.; Zorec, M.B.; Kruusamäe, K. Open-Source Robotic Study Companion with Multimodal Human–Robot Interaction to Improve the Learning Experience of University Students. *Appl. Sci.* **2024**, *14*, 5644. [\[CrossRef\]](#)

34. Hernández, A.; Ortega-Mendoza, R.M.; Villatoro-Tello, E.; Camacho-Bello, C.J.; Pérez-Cortés, O. Natural Language Understanding for Navigation of Service Robots in Low-Resource Domains and Languages: Scenarios in Spanish and Nahuatl. *Mathematics* **2024**, *12*, 1136. [[CrossRef](#)]
35. Aram, K.; Erdemir, G.; Can, B. Formation Control of Multiple Autonomous Mobile Robots Using Turkish Natural Language Processing. *Appl. Sci.* **2024**, *14*, 3722. [[CrossRef](#)]
36. Uruj, S.; Goswami, R.; Shetty, S.D.; Venkatesan, K.; Ramanujam, K. Comparative Analysis of GPT-4 and LLaMA 3.2 Integration With Speech Processing Models for Enhancing Human–Robot Interaction and Motion Control in Real-World Applications. *IEEE Access* **2025**, *13*, 127170–127182. [[CrossRef](#)]
37. Pavón-Pulido, N.; Blasco-García, J.D.; López-Riquelme, J.A.; Feliu-Batlle, J.; Oterino-Bono, R.; Herrero, M.T. JUNO Project: Deployment and Validation of a Low-Cost Cloud-Based Robotic Platform for Reliable Smart Navigation and Natural Interaction with Humans in an Elderly Institution. *Sensors* **2023**, *23*, 483. [[CrossRef](#)] [[PubMed](#)]
38. Door Dataset. Available online: <https://universe.roboflow.com/coins-j8vwp/door-p28ku> (accessed on 6 March 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.