An Interactive Autonomous Visitor Guide Robot in a University Scenario

Carlos Vinicius Boduar de Alcantara - 60543

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Electrotechnical and Computer Engineering. Work developed during the double degree exchange program between the Polytechnic Institute of Bragança (IPB) and the Federal Technological University of Paraná (UTFPR).

Work oriented by:

Prof. PhD. Paulo Jorge Pinto Leitão

Prof. PhD. Lucas Ricken

Bragança

2025

An Interactive Autonomous Visitor Guide Robot in a University Scenario

Carlos Vinicius Boduar de Alcantara - 60543

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Electrotechnical and Computer Engineering. Work developed during the double degree exchange program between the Polytechnic Institute of Bragança (IPB) and the Federal Technological University of Paraná (UTFPR).

Work oriented by:

Prof. PhD. Paulo Jorge Pinto Leitão

Prof. PhD. Lucas Ricken

Bragança

2025

Don't be pushed around by the fears in your mind. Be led by the dreams in your heart.

Roy T. Bennett

Dedication

I dedicate this work to my father, who allowed me to follow this path by always believing in me and doing everything for my happiness. I also dedicate it to my mother, sister, friends, and teachers who helped me get there and were part of my journey.

Acknowledgement

First, I would like to thank my family, especially my mother Fátima, father Carlos, and sister Adrielly, who have always been there for me, enabling me to become who I am today.

I also thank my supervisor, PhD Professor Paulo Leitão, for all his support, trust, and patience. You are an inspiration, and I hope to learn much more to reach your level of knowledge and capabilities.

I thank my co-supervisor, PhD Professor Lucas Ricken, for all his support throughout my academic career, and for being a great role model throughout these years.

I am grateful for all the support that PhD student Alexandre Júnior gave me throughout this period, I hope that one day I will be able to express my gratitude and repay him for each teaching, encouragement and true friendship, I admire you.

I am grateful to everyone who was part of my academic life at CeDRI, being mentors, sharing moments and allowing me to learn so much from each one.

I thank the people who lived with me in Bragança, becoming a source of support in difficult times, changing my life and giving me strength, especially Beatriz Leão, Lorena Davantel, João Rosa, João Victor, Ana Karolina, Maria Fernanda.

I am grateful to all those who were part of my journey at UTFPR. Over the years, we have built friendships that I will cherish forever. I am especially grateful to Juliana Simões, Juliana Moreira, Giovana Ronqui, Gabriele Takano, Luana, João, Karina Caetano, Bruno Maiolli, Eberton, and so many others who have been part of this story.

I would like to thank UTFPR University in Campo Mourão, which welcomed me and provided all the support I needed to become the professional I want to be. I thank all the

professors and staff of the Electronic Engineering department for their teachings.

Finally, I thank IPB for welcoming me and offering the support to get where I am, especially the professors in Electrical and Computer Engineering.

Abstract

Service robots has seen advances with the use of Artificial Intelligence technologies, enabling the development of robots capable of interacting more naturally with humans. This dissertation presents the proposal and implementation of an autonomous service robot designed to act as an interactive guide in university environments, focusing on personalized interaction with users. The proposed architecture integrates multiple software and hardware modules, including a chatbot developed with the RASA platform, integration with an embedded Large Language Model (LLM), dynamic facial recognition with local updating, ROS-based navigation, and environmental perception with LiDAR sensors and RGB cameras. Furthermore, the system queries external APIs to provide contextual information, such as weather and class schedules. To evaluate the developed robot, tests were conducted with 25 users, who evaluated the user experience. The observed limitations were discussed as opportunities for improvement for future work. The presented architecture represents a relevant contribution to the field of service robots by proposing a reproducible, scalable, and user experience-centric approach.

Keywords: Service Robots; Human-Robot Interaction; Intelligent Chatbot; Speech Recognition; ROS; LLM; Facial Recognition; RASA; Autonomous Navigation; Embedded Systems.

Resumo

A robótica de serviço tem tido avanços com o uso de tecnologias de Inteligência Artificial, permitindo o desenvolvimento de robôs capazes de interagir de maneira mais natural com os seres humanos. Este trabalho apresenta a proposta e implementação de um robô de serviço autônomo desenvolvido para atuar como guia interativo em ambientes universitários, com foco na interação personalizada com os usuários. A arquitetura proposta integra múltiplos módulos de software e hardware, incluindo um chatbot desenvolvido com a plataforma RASA, integração com um modelo de linguagem (LLM) embarcado, reconhecimento facial dinâmico com atualização local, navegação baseada em ROS e percepção do ambiente com sensores LiDAR e câmeras RGB. Além disso, o sistema realiza consultas a APIs externas para fornecer informações contextuais, como clima e horários de aula. Para avaliação do robô desenvolvido foram realizados testes com 25 usuários, que avaliaram a experiência de uso, as limitações observadas foram discutidas como oportunidades de melhoria para trabalhos futuros. A arquitetura apresentada representa uma contribuição relevante para a área de robótica de serviço, ao propor uma abordagem reprodutível, escalável e centrada na experiência do usuário.

Palavras-chave: Robótica de Serviço; Interação Humano-Robô; Chatbot Inteligente; Reconhecimento de Fala; ROS; LLM; Reconhecimento Facial; RASA; Navegação Autônoma; Sistemas Embarcados.

Contents

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence.

ASR Automatic Speech Recognition.

BMS Battery Management System.

CeDRI Research Center in Digitalization and Intelligent Robotics.

CPU Central Processing Unit.

DPM Dynamic Power Management.

ESTIG School of Technology and Management.

FPS Frames Per Second.

GPU Graphics Processing Unit.

gTTS Google Text-to-Speech.

HRI Human-Robot Interaction.

IMU Inertial Measurement Unit.

IPA Intelligent Personal Assistants.

IPB Polytechnic Institute of Bragança.

LLM Large Language Model.

LoRA Low-Rank Adaptation.

ML Machine Learning.

NLP Natural Language Processing.

NLU Natural Language Understanding.

NPU Neural Processing Unit.

RAG Retrieval-Augmented Generation.

ROS Robot Operating System.

SBC Single Board Computer.

SLAM Simultaneous Localization and Mapping.

SLM Small Language Model.

TTS Text-to-Speech.

UTFPR Federal Technological University of Paraná.

VAD Voice Activity Detection.

Chapter 1

Introduction

Over the last few decades, software and hardware have undergone significant changes, allowing different technologies to evolve, highliting Artificial Intelligence (AI), which is now used almost everywhere in a way that penetrates our daily lives. It has also played a fundamental role in different areas and has contributed in a disruptive way for technological advancement [1].

One of the sectors affected is mobile robotics, driven by advances in AI and the increasing automation of tasks in multiple sectors, which has seen significant growth. It is possible to observe its use in various fields, including industrial automation, surveillance, planetary exploration, construction, museum guides, personal services, medical care, as well as many other industrial and non-industrial applications [2].

Throughout this evolution, new technologies have emerged, such as Intelligent Personal Assistants (IPA), which are advanced computer systems driven by AI techniques capable of performing tasks to help with everyday life, such as online searches, shopping and reminders. Their popularity is due to the virtual assistants established on the market, such as Google Assistant by Google, Alexa by Amazon, Siri by Apple and Cortana by Microsoft [3].

The impact of new AI technologies on society is so significant that the ChatGPT chatbot reached over 100 million active users just two months after its launch, making it the fastest-growing application in history. Large Language Models (LLM) have made

it possible to generate text and images using human language and apply them to a wide variety of tasks, providing an opportunity for the field [4].

Another impacted field is service robots, according to the International Standard Organization [5] a service robot is a *"robot in personal use or professional use that performs useful tasks for humans or equipment"*, in which they can perform tasks such as transportation, physical support and guidance.

Despite all the advances, several challenges need to be solved, such as Human-Robot Interaction (HRI) as many systems fail to understand complex contexts and emotional variations. The failure of robotic conversation can lead to negative experiences and decrease social acceptance, as users usually expect robots to operate with high precision and consistency [6].

In this context, this work seeks to develop a scalable and modular architecture to mitigate the gaps in service robots, such as the lack of understanding of contextual nuances in rule-based chatbots, the lack of repeatability in action executions in LLMs, integration between different modules, and energy efficiency. This architecture integrates service robot systems with AI mechanisms, allowing the creation of an interactive chatbot capable of having the flexibility to perform tasks and use natural language generation for a better user experience. It is also proposed to implement this architecture in a university guide robot at the School of Technology and Management (ESTIG) of the Polytechnic Institute of Bragança (IPB) in Portugal.

1.1 Objectives

The main purpose of this thesis is to design and implement an autonomous service-robot guide for university settings, built on a modular, resource-efficient architecture that simplifies the integration of heterogeneous subsystems, supports multiple tasks, and delivers a safe, natural, and dependable user experience.

To achieve this goal, the work is structured into the following specific objectives:

1. Analyze the state of the art on service robots.

2. Develop a modular architecture for service robots.
3. Implement the chatbot and an intuitive, user-friendly interface.
4. Integrate HRI systems for multimodal user interaction, e.g., speech recognition and speech generation.
5. Integrate the navigation system, improving performance for safe navigation in indoor environments.
6. Integrate LLM for text generation.
7. Integrate all the systems and run them on the robot.
8. Test and validate the integration of the system, evaluating its efficiency in interacting with users.

1.2 Document Structure

The remaning of the document is organized as follow:

In Chapter 2, there are theoretical discussions related to service robots and a brief literature review of related works.

Chapter 3 defines an architecture for an AI-supported interactive autonomous service robot, covering different integration layers with modules for interfacing, chatbot, navigation, and information requests to APIs and remote servers.

In Chapter 4 describes the developed visitor guide service robot in a university environment.

Chapter 5 presents the experimental results of the solutions developed, highlighting the challenges and lessons learned from users' reviews.

Chapter 6 summarizes the main findings and points out future work.

Chapter 2

State of the Art

The term *"robot"* was introduced into popular culture in the early 1920s by brothers Joseph and Karel Kapek. However, it was only in the late 1950s that General Motors managed to transform what was fiction into reality with the *Unimate*, a robot used in automobile production. It was an important milestone for the use of robotics in manufacturing, allowing automation to increase efficiency and reduce human error. Since then, advances in knowledge and technology have led to the application of robots in various areas, from space exploration to medicine [7].

The terminology adopted distinguishes industrial robots, aimed at automating production processes, from service robots, designed to perform tasks in personal or professional environments [5]. The evolution of service robots can be observed with the consolidation of sensing and control systems capable of operating outside the factory floor, the popularization of open frameworks and libraries, reducing development and integration barriers, and the incorporation of AI and embedded computing techniques. Service robots have become increasingly sophisticated, combining advanced perception systems, Machine Learning (ML) and real-time connectivity, allowing them to operate more efficiently in unpredictable environments, respond to human interactions more naturally and adapt to different tasks [8].

An example of a service robot is the *Atlas*, developed by Boston Dynamics [9]. In its demonstrations, it uses and integrates visual models of ML, electrical actuators along

with vision, force and proprioceptive sensors to autonomously navigate the environment and perform tasks, while also showing the ability to respond to faults and changes in the environment in real-time. Although it is an experimental platform, it demonstrates the potential of today's technological capabilities.

2.1 Technologies Used in Service Robots

The architecture of a service robot is conceived from the combination of hardware and software elements to obtain the capacity for perception, cognition, localization, navigation and action. According to [8], three main technical groups enable the service robot to act in dynamic and unpredictable environments. In Figure 2.1 the technology stack and some of the available technologies are represented, and are made up of the software layer, which is responsible for integrating, connecting and communicating the different components of the robot. The second set, contextualization, brings together elements that allow the robot to situate itself and interpret the environment, using sensors and location and mapping methods. The human-robot interface set encompasses input and output channels that mediate interaction with the user.

In a service robot, the integration layer functions as middleware: a set of libraries, tools, and communication contracts that allow sensors, algorithms, and actuators to work in a coordinated manner, even when developed by different teams. One of the references in this field is Robot Operating System (ROS), available at [10], which, through an ecosystem composed of a publish-subscribe bus structure with typed messages, services, development tools, and an ecosystem of reusable packages, enables the integration of a modular system.

Although this dissertation uses ROS due to the need for some legacy packages and limitations of the hardware used, it is pertinent to situate the role of ROS 2 as an evolution that expands security, robustness, interoperability, and real-time requirements, critical aspects in dynamic and distributed environments, such as service robots [11].

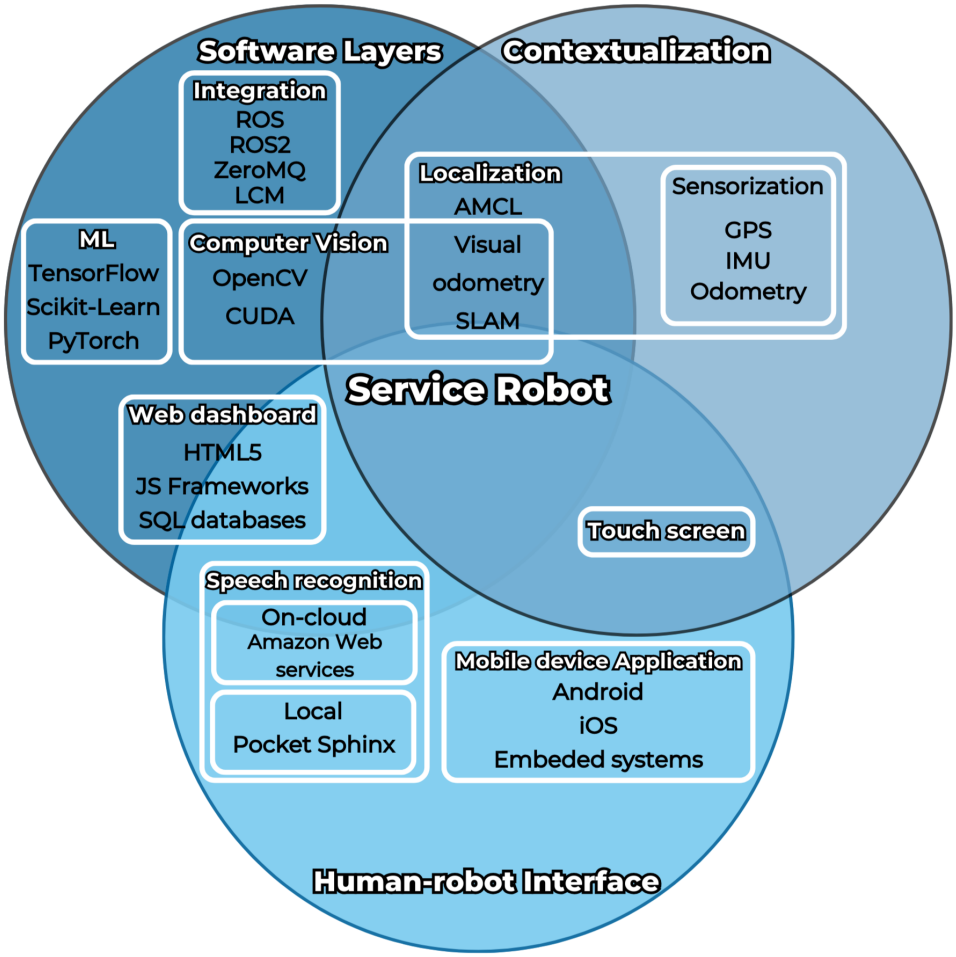

Figure 2.1: Common components of a service robot with examples of the technological stack available. Adapted from [8].

Within this area of study, sensors are divided into proprioceptive sensors, which measure the robot itself, such as Inertial Measurement Unit (IMU) and odometry, and exteroceptive sensors, which measure the environment, such as cameras, LiDAR, and infrared sensors. Sensors such as LiDAR and cameras are responsible for the spatial perception used in Simultaneous Localization and Mapping (SLAM). IMU, in turn, help stabilize position estimation, and ultrasound sensors are used to detect nearby objects and avoid collisions [12]. The selection of sensors directly conditions the performance of the SLAM and obstacle detection. To be able to handle robustly indoors it is common to use the sensory fusion of cameras, LiDAR, IMU and ultrasound [13].

The choice of locomotion architecture and actuators is based on energy efficiency, mechanical simplicity, physical safety, and navigation performance in built environments. The paper [14] highlights that motorized wheels have high efficiency on flat surfaces, directly influencing battery range and sizing, justifying the prevalence of wheeled platforms in humanized spaces, such as corridors and hospitals.

Furthermore, the robotic base can be either differential or omnidirectional, which directly impact the minimum radius of curvature, lateral maneuverability, energy consumption, and wheel wear. Differential platforms offer greater simplicity, good traction, and rotation on their own axis with low control complexity, being widely used in corridors, doorways, and narrow passages. Omnidirectional bases, on the other hand, allow lateral movement and micro-adjustments of pose, useful in dense service areas. However, they are more sensitive to floor irregularities, vibration, and require careful calibration to avoid drifting. The robust performance of these platforms depends on the closure of the sensory loop and the management of odometry errors, which accumulate through slippage along the path [14].

One of the requirements of service robots is real-time processing for perception, localization, planning, and HRI, with energy and connectivity constraints. The solution is to adopt the edge paradigm, in which processing is moved closer to the data source, reducing latency, improving resilience against network failures, and protecting data privacy. This processing is enabled by increased Central Processing Unit (CPU) computing power and the integration of Graphics Processing Unit (GPU) and Neural Processing Unit (NPU) in Single Board Computer (SBC), which achieve higher performance per watt and can now support more complex tasks locally without sacrificing autonomy [15]. However, not all tasks need to occur onboard; one trend is to combine processing with the cloud for large data that is not essential to the robot's immediate cycle, allowing for scale without compromising system performance [16].

Navigation in service robots aims to guide the system efficiently and reliably to defined objects, using partial knowledge of the environment and sensor readings. Thus, it can be categorized into four blocks: perception (extracting useful information from sensors

and location), localization (estimating the robot's position and orientation), planning (defining paths and making decisions along the way), and motion control (executing the path within physical limits) [14].

When the map is nonexistent or dynamic, it's necessary to integrate mapping and localization. This technique, called SLAM, involves the robot constructing or updating the map while simultaneously estimating its position. This is essential for the robot's autonomy, allowing it to adapt to changes. These systems typically work on position graphs, seeking to reduce accumulated uncertainty. Various methods, such as extended Kalman filters and particles, can be used to achieve this. The choices are determined by the type of environment and the available computational resources [14].

Robot path planning can be divided into two layers: the global layer, which plots routes on the map, and the local layer, which adjusts the trajectory at runtime to avoid collisions and navigate changes in the landscape. Widely used options include visibility graphs, which tend to provide paths with minimum length, but can bring the robot too close to obstacles, or Voronoi diagrams, which tend to maximize the distance to obstacles, at the cost of longer paths. For motion control, execution must respect the kinematic constraints of the mobile platform and the uncertainties of the sensors and actuators. Therefore, an increase in obstacle size is applied to compensate for errors and generate feasible trajectories [14].

Following Stanford's formulation [17], artificial intelligence is the science and engineering of making intelligent machines, with a contemporary focus on learning agents that adapt to changing conditions. In service robots, this means systems that perceive their surroundings, decide among alternatives, and act to achieve goals under uncertainty and resource limits (safety, latency, energy).

At a high level, modern AI in robotics blends four strands. Deep Learning provides hierarchical representations that turn raw signals (images, audio, point clouds) into useful abstractions for decision-making [18]. Reinforcement learning learns policies that map situations to actions to maximize cumulative reward, useful for refining behaviors and

handling long-horizon trade-offs [19]. Probabilistic inference models uncertainty explicitly (Bayesian filters, particle and factor-graph methods), the backbone of robust state estimation in the real world [20]. Transformers, introduced in [21], use attention rather than recurrence or convolution, underpinning today's language and vision models that interact naturally with users and scenes.

AI turns multimodal sensor data into environment understanding: detecting and segmenting people and objects, estimating traversable space, and recognizing places. Deep models improve robustness to illumination changes, occlusions, and domain shifts, making outputs stable enough to drive state estimation and motion planning in real time [18]. In SLAM integrates mapping with continuous pose estimation so a robot can operate without prior maps. An important mechanism is loop closure—recognizing previously visited places—which ties the pose graph and shrinks accumulated uncertainty, yielding consistent maps and reliable long missions [22]. Learned components (e.g., data association and descriptors) increasingly complement classical estimators to handle dynamics and perceptual aliasing. Another AI application is in planning and control by learning preferences, such as comfort zones and social compliance, adjusting speeds around people, and exploiting short-horizon predictions of occupancy and intent.

In service settings, users expect natural, context-aware interaction. Transformers bring strong language understanding, but language alone is not enough: behavior must be grounded in what the robot can actually do and perceive to keep actions safe and correct. Two complementary patterns have proved useful. First, LLM-orchestrated tool-use, in which the model decomposes a user request into high-level steps while affordance/skill functions gate each step by feasibility in the current scene [23]. Second, vision–language–action models, which place images, text, and action tokens in a shared sequence so that web-scale knowledge transfers to control, improving generalization to unseen objects and instructions [24]. These approaches enable flexible dialogue that maps to reliable behavior—interpreting intent broadly yet executing conservatively—and their impact should be tracked with established HRI instruments to close the loop on usability and social acceptance.

2.2 Challenges in Service Robots

On university campuses, hospitals, and libraries, service robots coexist with dense human flows, heterogeneous infrastructure (elevators, doors, shiny floors, glass), and saturated Wi-Fi networks. The adoption rate is positive, but sustainable operation depends on resolving technical bottlenecks that still emerge daily in the field (reliability, security, social navigation, energy, computing, reliable HRI, building integration, and maintenance). Annual IFR reports [25] document the expansion and diversity of the sector and serve as a basis for prioritizing where the impact of improvements is greatest.

In real operation, the robot is expected to complete 4–8 hour shifts without noticeable loss of perception and navigation performance, but the energy budget is contested by locomotion, sensing, computing, communications, and peripherals. Failures can occur during peaks in simultaneous system usage, causing thermal throttling that reduces Frames Per Second (FPS) and degrades the robot's performance. Studies such as [26] show that computing alone can account for around 30% of a mobile robot's total consumption, justifying Dynamic Power Management (DPM) techniques to maintain FPS and autonomy under control. Therefore, in addition to adequately sizing batteries and Battery Management System (BMS) to meet the needs of actuators and devices, conscious management of computational resources is important, scaling modules to avoid competition for resources and wasted energy.

Operating near people requires compliance with safety and risk assessment requirements. The ISO 13482 standard [27] defines safety requirements for service robots (personal and professional), covering categories such as mobile servants, physical assistants, and people movers, with guidelines for physical contact, force/speed limitations, and protective measures. It is necessary to identify contact hazards, falls, and loss of stability, as well as electrical risks, and define safety functions consistent with each robot's task. Therefore, it is necessary to limit the robot's speed and force, implement emergency stop devices, and use redundant and complementary sensors. This is combined with testing and inspections of brakes, response times, detection range, and visual and audible signaling

to ensure the safety of the robot and its users.

Reliable navigation results from the complementarity between planning and reaction. This architecture is a necessity because maps are always out of date in living environments, and human flow predictions are imperfect. Localization is the primary source of fragility in indoor environments. Sensor noise from visual illumination variations, sonar echoes, gain jitter, sensor aliasing, odometry errors—slippage, uneven wheel radius, and uneven flooring—accumulate uncertainty. The robot loses pose and degrades trajectory tracking. To address this challenge, multisensor fusion is used: 2D LiDAR stabilizes corridor geometry, cameras add features for location recognition, and IMU/encoders smooth short intervals [14]. In teaching platforms, such as TurtleBot 4 [28], the combination is given by 2D LiDAR + OAK-D + IMU, and it is in this that rate and filter adjustments of each sensor need to be tuned to the environment.

In mapping and SLAM, sections of the environment can mislead estimators due to their similarity, leading to the creation of inappropriate routes. Visual approaches can therefore be applied; position estimation can use fiducial markers for relocation and loop closure, allowing the creation of absolute anchors that stabilize maps, reduce drift, and accelerate recovery when odometry degrades [29].

The representation of the environment influences the fluidity of movement, so different techniques need to be applied to achieve good results. The use of occupancy grids allows the robot to have the necessary control to pass through doors, bypass furniture, and align docking stations. This technique consists of dividing the space into cells and storing the probability of occupancy or establishing a traversal cost. Topological maps consist of a graph of places and connections used to prevent the global planner from using CPU scanning the entire floor, capturing connectivity between areas and shortening high-level routes. Semantic layers are a set of labels and rules to make the robot behave correctly, establishing points of interest, prohibited areas, and navigation direction. The combination of these techniques allows control over which locations to navigate, how to access them, and which rules to follow for correct navigation [14].

HRI is the scientific field that investigates, designs, and evaluates how humans and

robots work together. HRI is not limited to talking robots or beautiful interfaces: it integrates perception of human signals (speech, gestures, distances), cognitive models of task and context, decision-making under uncertainty, and real-time communication of intentions, always with humans within the control loop [30]. This interaction is crucial in the implementation of a service robot; studies show that good results emerge when interaction design, social navigation, safety safeguards, and assessment with standardized instruments work together. Reviews of HRI reinforce this framework, highlighting the central role of AI in perception, language, and adaptation, but also the practical limits of trust, ethics, and safety that accompany these gains [31].

In this scenario, for the robot to function properly, HRI needs to understand noisy and incomplete requests, move in a predictable and socially appropriate manner, communicate intent and state; recover from errors without generating risk or frustration, and preserve privacy. Its relevance has increased in recent years because advances in AI, such as contextual speech recognition, gesture/face vision, reinforcement learning, and language models — have expanded what the robot can perceive and decide. At the same time, they have exposed new limits: trust-building, explainability, cybersecurity, and ethical data management [32].

Therefore, the robot must be able to understand user behavior, whether through voice, touch, and/or gestures. For voice, it's important to use a microphone array to handle noisy environments. Combined with beamforming and Voice Activity Detection (VAD), as well as reliable wake words, Natural Language Understanding (NLU) identifies user intent to make the most appropriate decision [33]. Furthermore, it's possible to use touchscreens so the user can interactively select buttons on a clear interface, or use QR codes or even physical buttons. These options should be selected in the way that makes most sense for each scenario, aiming to achieve multimodal HRI appropriate to the context and user experience [34]. The work [35] in the healthcare context shows that simple voice+screen confirmation flows increase perceived usability and reduce errors, especially with lay audiences.

Robot behavior must be legible: it must demonstrate intent before acting. Minimal

cues, such as orienting the head or body in the direction of the detour, a brief directional LED, and a short tone, reduce startles and negotiate passage. Formalizing movement legibility as a trajectory that facilitates human inference of the goal provides the basis for designing microbehaviors that users understand effortlessly, especially in shared spaces [36].

Moving among people requires integrating social costs into the local planner, the controller adjusts speed/curvature based on human density, yields when detecting persistent blockage, and re-presents intention when the plan changes. The goal is not to arrive faster at any cost, but to arrive predictably while maintaining comfort and low near-miss [37]. On campuses, most interactions occur in motion. Assistive robots designed with HRI at the center, such as Lio, document design choices (compact base, speed/force limits, soft skin, contact detection) that facilitate safe movement among people and support tasks [38].

In HRI, feedback is the set of signals with which the robot makes its intention and state legible: appearance, voice, gestures, lights, sounds, and on-screen messages. These signals must reduce ambiguity, increase comfort, and avoid the uncanny valley. Regarding faces, strategies have favored stylized faces and controlled expressiveness over hyperrealism. The study [39] compares the effect of facial features on perceived trustworthiness and differences in judgment compared to humanoid robots like Pepper, influencing users' perceptions of various factors, such as the robot's trustworthiness and intelligence.

Another point of attention is the robot's voice, which must match the form and context: moderate timbre and an accent understandable to the target audience. Studies with the Misty II platform [40] show that perceived name, origin, and accent modulate traits attributed to the robot, recommending transparency and neutral choices in sensitive scenarios. Signals of intention also influence user perception. Before moving the base, the robot anticipates the direction with its face, a slight torso turn, and a glance toward the planned path; brief gestures and short LEDs/tones signal state changes. In platforms designed for safe circulation, such as [38], the design favors smooth and limited movements, which translates into greater legibility and less cognitive load for the user.

Natural Language Processing (NLP) is the bridge between user communication and the robot's executable plans and can be divided into several strands. Automatic Speech Recognition (ASR) is responsible for transforming speech into text, NLU is responsible for categorizing user input into intents and slots, managing dialogue for decisions, and Text-to-Speech (TTS) is responsible for generating speech. Deterministic dialogue policies are found in service robots because they offer low latency and decision traceability, one of the open-source platforms available on the market is Rasa [41], which consists of Rasa NLU, responsible for classifying intents and extracting entities, and Rasa Core, responsible for choosing the next action based on the dialogue state, stories, and rules, allowing simple flows to be executed deterministically based on training data. Another approach is to use LLM for open requests. In the work [23], LLM plays the role of a skill orchestrator, but to do so, it needs to be grounded according to the robot's limitations. This way, each action is only executed if a feasibility evaluator (learned from skills and environmental feedback) indicates that the precondition is met. The work [42] teaches the model how to invoke tools and how to use their output, in a self-supervised regime.

To make LLM viable in SBC, different techniques can be used, such as Retrieval-Augmented Generation (RAG): before generating, the system retrieves passages from a non-parametric memory, such as campus documents, and injects this content into the prompt; the RAG-Sequence and RAG-Token variants show consistent gains in knowledge-intensive tasks, with improved factuality and provenance without retraining the base model [43]. Fine-tuning: Low-Rank Adaptation (LoRA) consists of retraining all the parameters of the original model, inserting a small number of new lightweight parameters that are trained for the new task, drastically reducing the required memory while maintaining quality [44]. These techniques can be used together to allow Small Language Model (SLM) to achieve better performance, mitigating the possibility of hallucinations and providing more appropriate responses.

In summary, the state-of-the-art in service robots for academic indoor environments points to a cohesive, multidisciplinary project that aligns (i) a technical foundation: efficient actuators and bases, energy constraints, and conscious use of computing resources,

combining on-board and cloud processing. With (ii) perception, localization, and navigation: SLAM with sensor fusion, maps that combine grid/topology/semantics, and planning that respects people and the environment. (iii) User-centric HRI: enabling multimodal input (voice, screen, gestures) and clear feedback (voice, gestures, LEDs) for the robot to be understood; (iv) robust NLU — using appropriate techniques to correctly understand the user and respond appropriately. (v) Risk governance: physical security (force/speed limits, E-stop), cybersecurity, and data policies. Therefore, these aspects represent challenges in the implementation and use of these robots in the academic environment, such as long-term robustness, having to deal with changes in layout and flow of people, energy autonomy, integration with the environment, user experience, safety, repeatability.

Chapter 3

Architecture of the System

This chapter presents the proposed architecture for the service robot system developed in this thesis. To solve the difficulties related to the effective integration of multiple technologies and systems, we designed an architecture that prioritizes modularity and expandability, exemplified in Figure 3.1. This schematic diagram illustrates how the different layers interact, providing an overview that will be further explored throughout this chapter.

The proposed architecture is divided into five major layers, as shown in Figure 3.1: Robot Peripherals, Virtual Assistant Chatbot, Customized Actions, Navigation System and Remote Server APIs. Each of these layers has specific responsibilities, but they act in an integrated and complementary way, ensuring an efficient flow of data and actions within the robot. This modular structure aims not only to facilitate understanding of the system but also to enable scalability, allowing adaptations and the addition of new functionalities. In the following sections of this chapter, we will detail each layer separately, presenting their characteristics, functionalities and how they contribute to the system's overall performance.

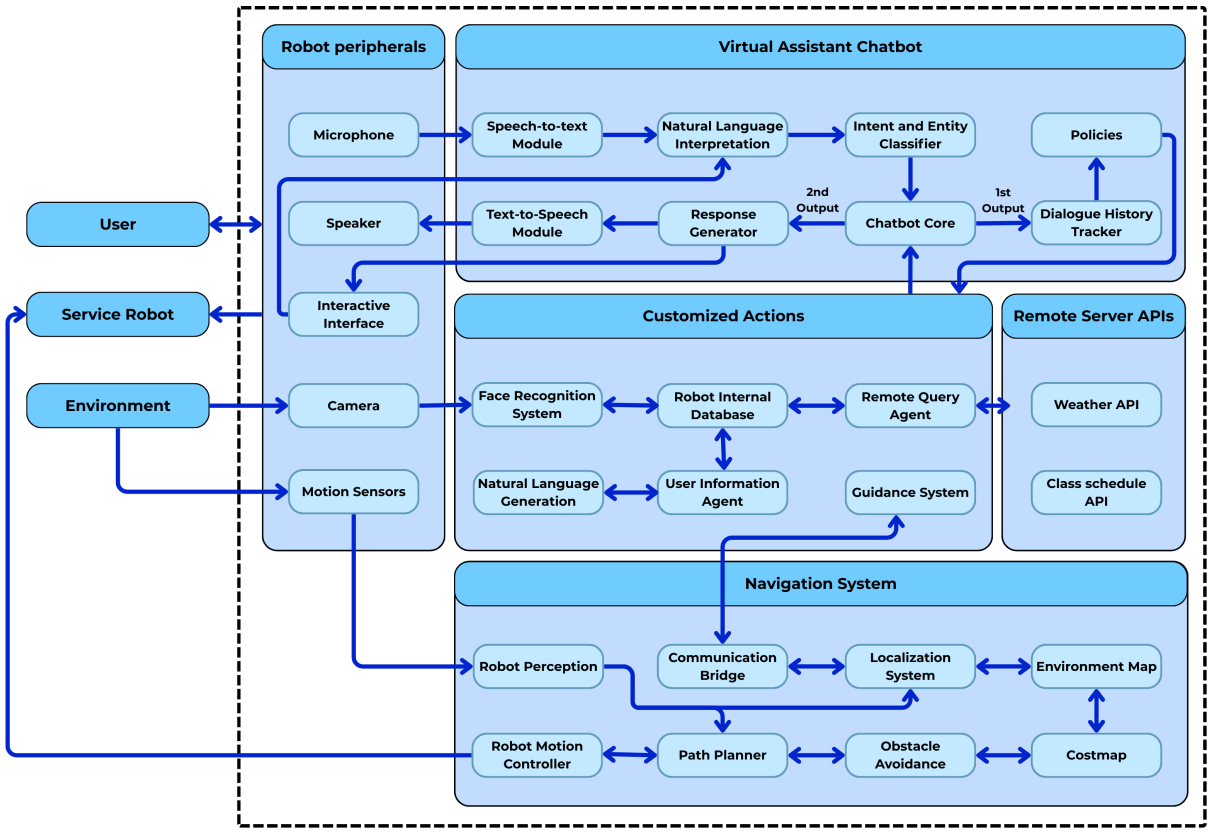

Figure 3.1: Modular architecture for an autonomous service robot with AI-based interaction system.

3.1 Interactive Interface

The Robot Interactive Interface layer comprises elements responsible for interacting with the user and the environment within the service robot's operational domain. The HRI refers to how people and robots communicate and interact in shared environments. Especially in the case of service robots, which deal directly with people without specific technical knowledge, it is crucial that they have an intuitive and easy-to-understand interface. A suitable interface not only improves the quality of interactions, but also broadens social acceptance, significantly increasing people's trust and comfort when using robots on a daily basis [45].

In order to establish fluid and friendly communication in service robots, the architecture developed foresees the use of multiple interaction devices, such as cameras, microphones, interactive screens and speakers. These devices act as essential channels that allow the robot to capture different input forms and offer comprehensible responses to users. Combining these features promotes an intuitive and positive experience for users, which is fundamental to the acceptance and successful integration of robots into everyday environments.

In human-robot communication, aspects related to voice recognition and generation play a critical role, through the microphone and speaker, directly influencing the quality and fluidity of interactions. In this way, systems that have good voice recognition tend to provide more positive interactions, as they minimize failures and misunderstandings, factors that can lead to frustration and rejection on the part of users.

The interactive interface is the direct visual communication channel with the user. The screen has the role of being a stable reference point for interaction, making it possible to present clear, accessible information in real time. This makes it possible to display commands, responses, system status and interaction options in a clear and organized way, making it easier to understand what the robot is doing or needs. By allowing direct touch interaction, the screen also increases the accessibility of the interface, offering an alternative to voice communication and making the experience more inclusive for different user profiles.

The interactive interface follows user-centered design principles, seeking to present information in a clear and visually organized way. One of the central elements of this interface is the use of an animated avatar that acts as a mediator of verbal communication. The avatar is able to blink, smile and synchronize lip movements with the robot's speech, which helps to reinforce the naturalness of the interaction and increase the user's emotional engagement. This approach follows recommendations in the HRI literature, which point out that visual signals and social expressions increase the feeling of empathy and acceptance of the robot [45].

The proposed architecture incorporates environmental perception sensors such as LiDAR, odometry sensors and RGB cameras to provide the robot with an accurate understanding of the space around it. These sensors allow it to recognize obstacles, calculate distances, identify passable surfaces and navigate safely in complex environments. This spatial perception is fundamental not only for avoiding collisions, but also for planning efficient trajectories, adapting behaviors and ensuring that the robot acts autonomously and reliably in different contexts.

Sensors also play an important role in the robot's social adaptation. The ability to perceive the presence of nearby users, for example, can be used to automatically start a dialog or to position the robot in a less invasive way in space. This attention to context reinforces the responsive nature of the interaction and helps the robot to act in a way that is more in line with social norms and human expectations, increasing user acceptance and comfort.

3.2 Virtual Assistant Chatbot

The virtual assistant layer mediates communication between the user and the system, interpreting commands in natural language and converting them into understandable actions for the robot. This layer is responsible for supporting dialog in human language, whether by voice or text, and is structured in a modular way to allow integration with different tools for interpretation, dialog management and response generation. As [46] points out, modern conversational systems need not only to understand the content of the user's speech, but also to manage the context and decide how to respond coherently and appropriately.

The architecture of this conversational layer is based on a well-defined pipeline, which includes: (i) the input mechanism, which can be via text or audio (in this case, preceded by a speech transcription module); (ii) the natural language interpreter, which extracts the intention and entities from the input; (iii) the dialog management core, which decides what should be done next based on the context; and (iv) the response module, which

presents the user with the system's output, both in textual and audio form. This structure reflects the fundamental elements of dialogue systems described by [46], who emphasizes the separation of understanding, management and response as key to flexible and reusable systems.

3.3 Customized Actions

Although the dialog module allows commands to be interpreted in natural language, it alone is unable to meet the diversity of tasks demanded by real users. The inclusion of the personalized actions layer reflects the growing need to make service robots not only conversational agents, but also operational and contextually useful. Whenever the content of the conversation requires access to external information, the use of sensors or the execution of complex logic, this layer comes into play. This allows for more adaptive responses, connected to the real world and the context in which the robot is inserted. This separation also allows for greater scalability, since new modules can be added without having to restructure the entire conversation logic.

Facial recognition is one of the most important personalized action modules that extends the robot's social intelligence. This module makes it possible to identify people based on previously registered images or to make new registrations during the interaction, associating a name or profile with the detected user. This contributes directly to increasing the user's perception of empathy, familiarity and context, fundamental aspects for social acceptance.

Another example of personalized action envisaged in the architecture is consulting external services via APIs, allowing the robot to offer dynamic information that evolves over time. This module is designed to respond to requests that require up-to-date data, such as the weather forecast, class timetables or institutional events. The flexibility of this mechanism allows it to be adapted to different domains, since the APIs consulted can be replaced or extended depending on the application context. This makes the robot more informative, functional and capable of providing relevant answers even when the

necessary knowledge is not stored locally.

To ensure coherence in the dialogue and personalized responses, the architecture incorporates a module dedicated to managing user information. This agent acts as an intermediary between the history of interactions, the local database and the other action modules. It is responsible for retrieving previously stored data - such as names, preferences or past interactions - and deciding, based on the current context, whether an action should be taken or whether external information needs to be consulted. By isolating this logic in a separate component, the architecture promotes greater organization, modularity and reusability, making it easier to apply in other domains.

Complementing the more operational functions, the architecture provides for the possibility of social actions, designed to strengthen the emotional bond between the robot and users. During movements or pauses in the dialog, the system can activate brief conversation modules, generating a sense of continuity and presence. This approach not only contributes to a more pleasant experience, but also reduces the feeling of waiting or inactivity, promoting a perception of continuous and empathetic intelligence.

To complete the layer of personalized actions, the architecture includes an autonomous guide module, which acts as a bridge between the dialogue and the navigation system, being triggered when the user asks, for example, to be guided to a certain location. The existence of this bridge between language and locomotion is fundamental for service robots in environments such as schools, hospitals or cultural centers, where mobility is an essential part of the system's usefulness. The architecture responsible for making this navigation autonomous and safe is discussed below.

3.4 Navigation System

The guidance system in the custom actions connects to the Navigation System layer via a communication bridge that translates the user requested destinations to the robot's low-level localization and control system. The perception module processes raw data from the robot's sensors (i.e. the LiDAR, the odometry and RGB cameras) and make available the

information needed by the path planning and localization systems, the latter being used to take advantage of sensory fusion and methods such as SLAM, Kalman or particle filters to determine the position and orientation of the robot using a map of the environment as a reference.

The path planner determine the optimum route for the robot to reach its destination. The obstacle avoidance system, adjusts the robot's path in real-time to avoid collisions with dynamic and static obstacles. Costmaps help in this process by being a local derivation of the environment map that determines the difficulty of moving through a location and perceive variations in the original map. The motion controller receives trajectory commands and generates the necessary signals for the robot's actuators to make the required trajectory using PID control.

The architecture presented in this chapter brings together the main components needed for a service robot to be able to interact naturally with users and perform contextualized tasks. By adopting a modular structure, the architecture allows new modules to be easily incorporated, whether to integrate additional sensors, new languages or specific functionalities. In the next chapter, the details of the practical implementation of this system in a university service robot will be explored, illustrating the challenges and decisions adopted in each module.

Chapter 4

Implementation

Transforming a conceptual architecture into a functional robotic system requires a series of technical decisions, adaptations and practical validations. The architecture was transformed into a functional system, made up of multiple integrated subsystems, each responsible for a specific function. This chapter describes how these modules were implemented on a physical platform equipped with multiple sensors and processing devices, detailing the technologies used, the communication between the layers and the challenges encountered during the construction of the autonomous service robot.

Initially, the robotic platform and hardware infrastructure that serve as the basis for the system are described, followed by the implementation of the user interface, chatbot, integration with language models, facial recognition and external queries. Speech Generation, Speech Recognition, autonomous navigation and, finally, the total integration of the subsystems are then discussed.

4.1 Robot Platform and Infrastructure

This project continues the work carried out by [47], who implemented the robotic platform's navigation base and the localization system based on fiducial markers and SLAM. The choice to keep the same physical platform and part of the software solutions developed previously made it possible to speed up the implementation and concentrate efforts

on the evolution of the upper layers of the architecture, such as the dialog and customized actions.

The Figure 4.1 shows the robotic platform developed, equipped with all the sensors, processors and interaction devices needed to implement the system.

Figure 4.1: Preliminary version of the visitor guidance robot.

The robotic base chosen for this work is the Magni Silver platform, from Ubiquity Robotics [48], which offers a robust mechanical structure, differential traction and support for loads of up to 100 kg. This model has been designed to facilitate the development of robotic applications. It has native integration with the ROS framework, which simplifies the adaptation of software for navigation control and integration with other modules. The Magni base is equipped with an on-board control system based on Raspberry Pi and odometry sensors with an accuracy of 2 mm, guaranteeing smooth and reliable movement indoors. In addition, the platform has a set of sonars for detecting nearby obstacles and 12V and 5V power inputs for powering other embedded components.

In order to increase the autonomy of use and meet the energy consumption requirements of the new modules added, improvements were made to the platform's power supply system, such improvements can be seen in the Figure 4.2. The original batteries were replaced with three 12V, 30Ah lithium iron phosphate batteries each, model $LiFePO_4$ from Eco-Worthy [49]. Two of them were connected in parallel to power the Magni base and its locomotion systems, while the third battery was allocated to meet the needs of the touchscreen and the Jetson Orin NX. The latter is connected to two DC-DC converter boards that stabilize the output voltage. Selector switches have also been added to turn off the system and allow each battery to be powered individually.

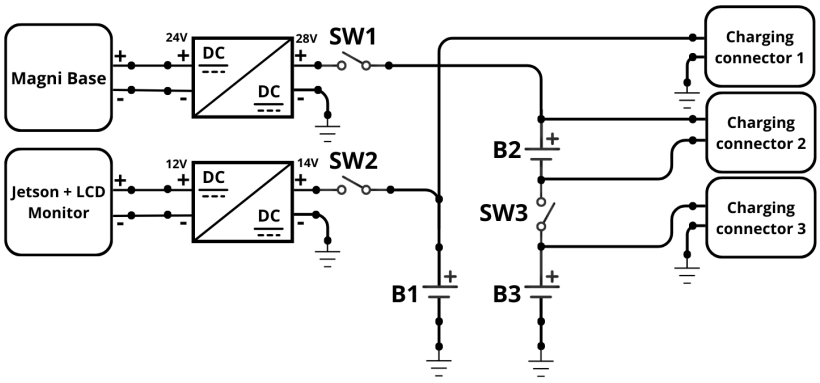

Figure 4.2: Robot power supply wiring diagram.

The robot's navigation layer was implemented on the Raspberry Pi 5 [50], a development board with a four-core ARM Cortex-A76 architecture clocked at up to 2.4 GHz. This board was chosen for its robustness, low power consumption and ease of integration with the Magni Silver platform. However, as the Magni silver platform only has support up to ROS Noetic and the Raspberry Pi 5 does not have support for Ubuntu 20.04, it was necessary to containerize the application in Docker so that it could be used on the Raspberry. This approach ensures compatibility with the navigation base and makes it

easier to maintain the proposed modular architecture.

To meet the processing requirements of artificial intelligence and machine learning, the system relies on the NVIDIA Jetson Orin NX 8GB [51], a high-performance embedded computing module. The Orin NX offers up to 117 TOPS (trillion operations per second) of computing power and is equipped with a 6-core ARM Cortex-A78AE CPU and a 1024-core CUDA Ampere GPU, features that make it ideal for running natural language models and convolutional neural networks in real time. As well as supporting frameworks such as TensorRT, PyTorch and ROS, Orin NX integrates Ethernet connectivity, USB and PCIe expansion slots, making it easy to communicate with the other modules in the system. This platform was chosen for its ability to support embedded cognitive tasks with low latency and high energy performance, essential for service robots.

To provide the robot with the ability to perceive the environment and support autonomous navigation, the Hokuyo UST-10LX 2D [52] LiDAR sensor was integrated. The UST-10LX offers a 270º field of view, a detection range of 0.06 to 10 meters and an accuracy of ±40 mm, with repeatability of less than 30 mm. It is powered by a 12V DC input and communicates via an Ethernet interface, allowing integration with the ROS framework. The sensor was connected to the Raspberry Pi 5 board via the Ethernet port, enabling the use of ROS packages such as urg_node [53], which provides support for reading and publishing sensor data in the ROS environment.

The Intel RealSense D435i [54] depth camera has been incorporated into the platform to extend spatial perception capabilities and enable 3D vision applications. The sensor offers a resolution of up to 1280x720 pixels, with an effective range of up to 10 meters and a field of view of 86° x 57°, as well as a capture rate of up to 90 fps. Direct integration with the Raspberry Pi 5 was achieved via a USB 3.0 connection, managed by the ROS package realsense2_camera [55].

The system has a 10.1-inch touchscreen with HD resolution (1280x720), installed on the front frame of the robot and connected directly to the NVIDIA Jetson Orin NX. This screen was chosen for its ability to display detailed graphics and support touch interactions, allowing the user to access menus and receive visual information while interacting

with the robot.

The system also incorporates a USB camera connected directly to the NVIDIA Jetson Orin NX, designed for social interaction and visual monitoring tasks. This camera is used for facial recognition, but can also be used for expression detection and presence recording.

To support voice interaction, the system relies on the ReSpeaker Mic Array v2.0 [56], connected directly to the NVIDIA Jetson Orin NX. This module features four MEMS microphones with omnidirectional pickup, enabling 360° audio pickup, as well as support for beamforming and noise cancellation techniques, key features for speech understanding in noisy environments.

4.2 User Interface

The implementation of the graphic interface plays a fundamental role in the proposed robotic system, as it is through it that the user interacts visually and textually with the robot. At this stage, we decided to develop a web application using HTML, CSS and JavaScript. The interface was hosted and executed on the NVIDIA Jetson Orin NX, which, in addition to managing natural language processing, also manages the rendering of the interface. The complete code is available in the project repository, and this chapter presents the main functionalities implemented, along with descriptions of their logical flows. Extensive technical details, such as code files, external libraries and specific configurations, can be consulted directly in the repository.

The architecture of the graphic interface was structured in three main layers: the HTML file defines the basic structure of the page and organizes the interaction elements, such as buttons, menus and text areas; the CSS file is responsible for visual stylization, defining colors, fonts, spacing and the responsiveness of the interface; finally, the JavaScript file implements the interaction logic, including message processing, switching visual states (idle/talking) and integration with the backend for sending and receiving

messages. This division into layers ensures modularity and facilitates system maintenance, as well as allowing components to be reused in other projects.

In the graphical interface developed, there are several features that optimize the user experience. In the Figure 4.3 you can see the main menu and the action buttons that allow the user to consult information such as the weather, class schedules, laboratory guides and start conversations with the natural language model.

Figure 4.3: Main menu of the graphical interface with action buttons.

Integration between the graphical interface and the processing modules was carried out via asynchronous HTTP calls using the Fetch API, enabling agile communication with the chatbot and the TTS service. When the user enters a message or selects an option, the frontend sends the request to the chatbot, which processes the intention and returns a response. The frontend, in turn, updates the message area and, if necessary, triggers audio playback via TTS to complement the visual interaction with audio feedback.

4.3 Chatbot Using RASA

To implement the chatbot, the RASA [57] platform was chosen, an open-source solution that offers flexibility, scalability and total control over the dialog flow. This choice was based on the need to customize the system and the possibility of running it locally, which is essential for mobile robots that can operate in environments without constant internet access. In addition, RASA allows different natural language processing (NLU) and dialog management (Core) modules to be integrated, making it easier to create complex dialogs that are adaptable to the context. Its modular architecture also makes it possible to add customized actions, such as integration with external APIs and the use of fallback with advanced language models, such as TinyLlama, increasing the solution's robustness and adaptability.

The conversation flow implemented in RASA is illustrated in Figure 4.4 and visually represents how the dialog is processed within the system. RASA's structure is based on a modular division that facilitates the development and maintenance of complex dialog flows. The first module, RASA NLU, interprets natural language, identifying the intent of the message and extracting useful information (entities). The second module, RASA Core, is responsible for managing the state of the dialog, deciding which actions to take based on the history of interactions and the policies defined. This approach allows the robot to understand the context of the conversation and react appropriately to the user's needs. Complementing this structure, RASA supports intents, entities, stories and custom actions.

The process begins with the reception of the user's message, which is sent to the RASA NLU module for analysis of the intention and extraction of entities. RASA Core then uses this information, together with the dialog history, to determine the next action to be taken. This action can be a direct response to the user or the triggering of external modules, such as the facial recognition system or the weather and timetable APIs. When the dialog model shows low confidence in the response generated, the flow is redirected to the fallback language model, thus ensuring a more natural interaction and avoiding

blockages in the conversation. This modular integration allows the chatbot to coordinate different functionalities and maintain the coherence of the conversation, even in complex scenarios.

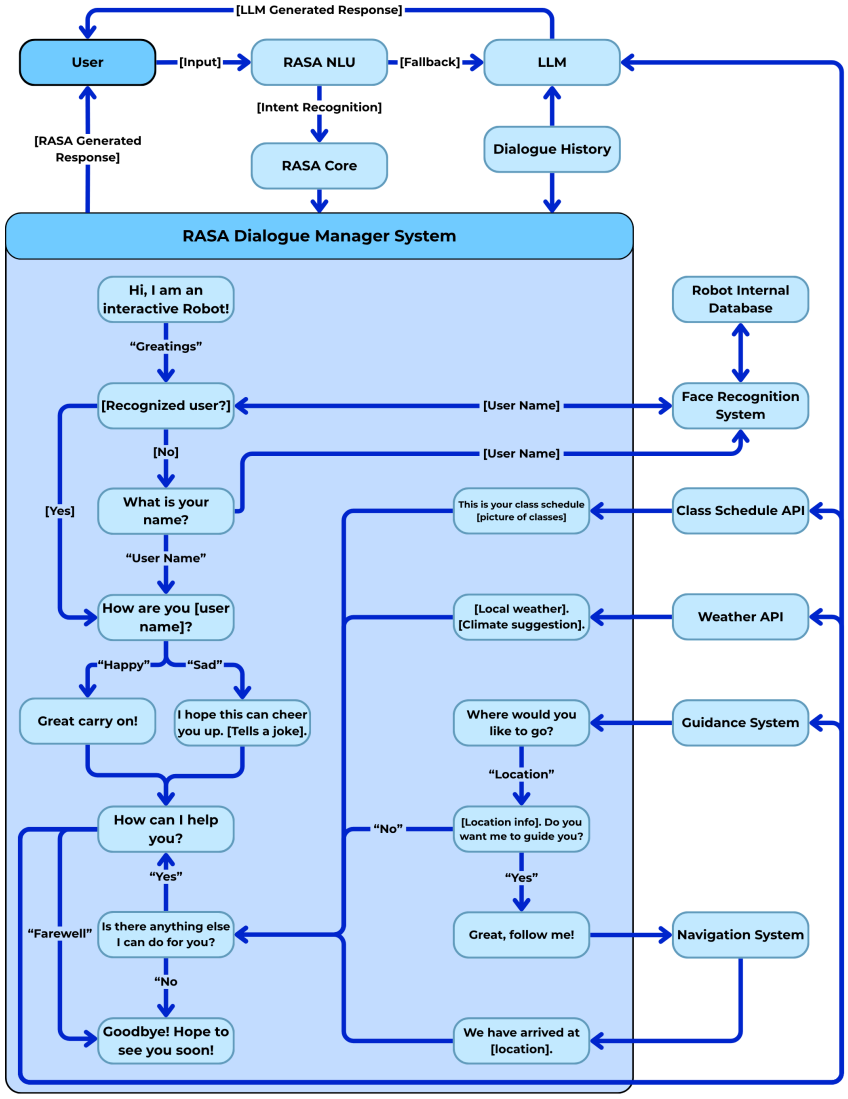

Figure 4.4: Exemplification of the service robot's dialogue system.

The chatbot implementation began with defining intents and entities in the Rasa NLU, which listed various intents, such as greetings, schedule inquiries, and weather information. Each intent was accompanied by examples of various phrases, allowing the model to learn different forms of requests. Entities were also configured to allow the model to extract specific information, such as course names or schedules. All of these definitions

were integrated into the domain.yml file, which serves as the core of the architecture, organizing intents, entities, responses, slots, and custom actions in a single location, facilitating system maintenance and scalability.

To define conversational flows and control chatbot behavior, stories.yml and rules.yml files were used. The stories.yml file maps out complete conversations, describing them from start to end, while rules.yml specifies specific behavioral rules, such as exception handling or triggering default responses in uncertain situations. Policies, configured together in domain.yml, guide RASA Core in choosing which actions to execute, balancing execution between stories and rules to ensure the chatbot maintains consistent responses aligned with the conversation context.

Complementing the chatbot implementation, several custom actions were developed to perform dynamic tasks, such as queries to external APIs and integration with the facial recognition system. These actions are called by RASA Core based on the recognized intent and the context of the conversation.

4.4 Facial Recognition

The facial recognition module developed in this work aims to increase the naturalness and personalization of the interaction between the bot and the user, allowing it to identify previously registered users or register new users during the interaction. By recognizing the user, the bot can adjust the context of the conversation, personalize responses, and strengthen the bond of trust with the user.

Facial recognition was implemented in the system using the face_recognition library [58], based on dlib and OpenCV. It offers functions for face detection in images, facial encoding extraction, and similarity comparison, allowing for the recognition of previously registered faces with high reliability. Its integration with the rest of the architecture was facilitated by Python support and available documentation.

The facial recognition implementation flow in the system was structured around two main actions: action_face_detect and action_save_new_face, both using the face_recognition

library. When triggered, action_face_detect initiates real-time video capture, processing the frames with action_face_detect to detect and encode faces. The encodings are then compared with the local CSV database, checking for similarity and classifying the recognized face. If the similarity reaches a value above the defined threshold, the system confirms the user after three consecutive detections, avoiding false positives. If the user is not recognized, the chatbot offers the registration option, managed by action_save_new_face, which uses a function to perform artificial image enhancements (mirroring, brightness, and rotation) and save multiple samples in the CSV.

Facial encodings and user names were stored in a local database in CSV format, facilitating management and quick information retrieval. This method was chosen for its simplicity, lightweight nature, and compatibility with the embedded Python environment. This database is maintained dynamically during the conversation, allowing users to be added or removed in real time, making the system flexible and adaptable. This solution also ensures data privacy, as all information remains on the device and is not sent to external services.

The identification logic adopted follows a cycle of capture, detection, encoding, and comparison, with temporal stabilization, as evident in Algorithm 1. At each frame, faces are detected and converted into 128-dimensional embeddings; then, only the face with the largest area is considered to reduce ambiguity in frames with multiple faces. Similarity is estimated by the shortest distance to the CSV embedding set (one line per user, containing the name and the serialized vector). An acceptance threshold of $\tau = 0.55$ is defined, and identification is only confirmed after $K = 3$ consecutive readings above this threshold for the same name, which mitigates false positives in isolated frames. Once identification is confirmed, the user's registration is updated online: old CSV entries are removed and new samples generated by data augmentation (original, horizontal flip, gain/brightness adjustment, and slight rotation) are recorded, keeping the base adapted to small variations in pose and lighting.

This strategy prioritizes simplicity and reproducibility without sacrificing practical robustness on the embedded device: the use of a confirmation window stabilizes the

Algorithm 1 Face identification

Require: video stream; CSV database $\mathcal{D}$
Params: threshold $\tau = 0.55$, confirmations $K = 3$
while camera is open **do**
    grab_frame();
    convert_to_RGB();
    detect_faces();
    compute_embeddings();
    select_largest_face_embedding();
    compute_distances(); get_best_name();
    $acc \leftarrow 1 - d_{\min}$
    if $acc \geq \tau$ **then** increase counter for that name
    elsereset counter
    end if
    if counter $= K$ **then return** name; UpdateFace(name, frame)
    end if
end while
return *unknown*
UpdateFace: replace user's rows in CSV and append augmented samples.

decision, choosing the largest face reduces collisions between identities in the same frame, and incremental CSV updates keep the model adapted to the user over time. It also operates entirely on the device—preserving privacy.

4.5 External APIs

In the context of robotics architecture, integration with external APIs significantly expands the service robot's usefulness, connecting it to dynamic information and online services. This strategy allows the robot to provide personalized, up-to-date responses relevant to the environment in which it operates, such as class schedules, weather forecasts, and institutional events. By adopting this approach, the system becomes more versatile and capable of meeting different user demands, becoming a useful tool in everyday life.

The external API module was implemented through custom Rasa actions, responsible for querying HTTP services, transforming responses (JSON/PDF), and returning messages and media to the user via a dispatcher. This approach encapsulates specific

integrations without coupling the dialog logic to the specifics of each provider, allowing for evolving endpoints and formats with minimal impact on the conversational flow.

For current weather conditions, the ActionGetWeather action queries OpenWeatherMap for Bragança, PT, extracting air temperature, wind chill, and a description of the sky. The next-day forecast is processed by ActionGetTomorrowWeather, which uses the IPMA public API. The action requests the daily series for the Bragança municipal identifier and cross-references the weather type code with a table of textual descriptions, resulting in a message with a summarized forecast, maximum and minimum temperatures, and precipitation probability.

For academic schedules, ActionClassSchedule consumes the IPB institutional API. The action receives the identifiers selected by the user from the school and course slots, constructs the schedule URL, downloads the PDF, saves it locally, and converts each page into an image. It then sends the images to the frontend through the Rasa interface, ensuring immediate display of the schedule on the robot's screen. When the HTTP response is unsuccessful, the action communicates the unavailability to the user, preserving the user experience.

User parameters are collected using the ActionAskSchool and ActionAskCourse actions. The first lists the schools available through the IPB API and creates buttons with the school code abbreviations. The second searches for courses at the selected school, filters by the specified degree, and displays buttons with the course name. This button-driven interaction reduces ambiguity and avoids error-prone free entries, keeping slots consistent throughout the dialog.

4.6 Speech Generation

To make interaction more natural and accessible, a speech synthesis module was implemented that transforms the robot's responses into audio played directly on the device. The solution was designed to operate on the Jetson Orin NX, with low latency and minimal coupling to the rest of the system. We opted for an HTTP microservice in Python

Flask [59] that receives text via POST /audio and executes the speech generation and playback pipeline. In the service, the text is converted to audio with Google Text-to-Speech (gTTS) [60], saved temporarily as an MP3, accelerated 1.25× with pydub [61], and played locally before removing the temporary file. This design maintains orchestration and playback on-device, with a simple API that receives "text": "<message>" and returns the success of the operation.

Interface integration follows a speech cycle controlled by the frontend. Upon receiving responses from Rasa, the client-side script disables user input, displays a "speaking" animation, sends the text message to the TTS microservice, and, in parallel, displays the text on the screen in short blocks to maintain readability. At the end, the animation returns to its "idle" state, and UI events are reactivated. This strategy avoids duplicate clicks during speech and synchronizes the visual and audio experience.

4.7 Speech Recognition

To enable voice commands and message dictation, a local speech recognition service was developed in Python, running on the Jetson Orin NX. The module exposes the POST /start-recognition endpoint (Flask + CORS) and, with each call, runs through a pipeline: (i) multichannel recording (6 channels, 16 kHz) via sounddevice [62], (ii) per-channel noise reduction with noisereduce [63] and downmixing to mono (channel average), and (iii) transcription with Whisper (tiny model, GPU-accelerated when available). Audio files are temporarily persisted in 16-bit WAV, and the service returns the transcription in JSON. The pipeline is available at Algorithm 2.

Recording is done during a fixed window from a selected device ID corresponding to the V3.0 respeaker. After capture, each channel undergoes independent noise reduction; then, the channels are averaged to form a mono track, balancing robustness and simplicity. Finally, the processed file is delivered to Whisper, which performs automatic language detection and transcription, returning the text to the caller. This design prioritizes local

Algorithm 2 Speech recognition pipeline

Require: HTTP request to `/start-recognition`

Params: duration $T = 10$s, samplerate $f_s = 16$ kHz, channels $C = 6$, device_id $= 0$
Record: capture C-channel audio for T seconds
Denoise: for each channel $c \in \{1..C\}$ apply noise reduction
Mixdown: average denoised channels
Load model: clear GPU/CPU memory; load Whisper on GPU
Transcribe: run Whisper on processed audio
Cleanup: clear memory; return JSON {`"transcription": text`}

processing, controlled latency, and low dependence on external components, while remaining flexible: it is possible to switch the Whisper model to a higher-capacity one according to time and GPU constraints, adjust the window duration, or override the noise reduction strategy. It has been observed that aggressive filters can reduce vocal energy and impair accuracy; this can be mitigated by adjusting the noisereduce parameters, normalizing the gain after filtering, or adopting VAD/endpointing to capture only speech fragments.

4.8 LLM Integration

The integration of LLM into the conversational stack aims to expand request coverage by acting on out-of-scope messages. In this work, LLM can be triggered as a fallback when the RASA policy presents low trust, as demonstrated in Figure 4.4, or it can be triggered on demand when an open conversation is desired. In both cases, the text generated by LLM does not execute any actions, RASA continues to manage the conversation flow.

Given the embedded context, a small LLM was adopted on the device. After on-device performance testing, the TinyLlama 1.1B model [64] was chosen. This model presented the best results, with acceptable memory consumption and runtime. This choice allowed for fully local text generation, but imposed limits on accuracy and robustness.

Although the chosen model worked adequately, it presented hallucination problems. To improve the responses, the model was fine-tuned using LoRA. By adding low-rank adapters to the base model, approximately 1.5 million trainable parameters, the infrastructure also supports quantization and variants such as QLoRA to balance memory and

performance on the SBC. The data used were taken from the official Research Center in Digitalization and Intelligent Robotics (CeDRI) website [65], using information such as people and laboratories.

For domain fidelity without sacrificing edge, the API integrates a RAG Engine into the generation path: before responding, snippets are retrieved from local sources in a MongoDB index, and these snippets are included in the prompt. This reduces hallucinations, shortens responses, and, in practice, offsets the extra retrieval cost, as LLM generates fewer tokens to obtain the same information.

The application runs as a Python microservice, similar to the previous modules. Thus, an internal HTTP endpoint is exposed, where user-entered text and previous messages are forwarded to the endpoint for processing. The service then allocates the model in memory and generates text using the GPU. It then returns the generated text to RASA, freeing up device memory.

4.9 Navigation Integration

In this work, the existing navigation was integrated, based on the work of [47], and made more stable, with parameter adjustments, control improvements, and process orchestration for daily use on campus. The objective was to increase operational reliability, adapt behavior to the new weight of the platform, and the increased computational power of the robotic base.

The Localization System maintains the robot's relative pose estimation within the Environment Map generated by SLAM using the embedded robots's sensors. The system uses the AMCL algorithm to make the pose estimations and maintaining the costmap updated with obstacles that appears in the robot path. To increase the precision and performance of the AMCL estimations by addressing intrinsic errors and robot drift, fiducial marker's were positioned on the environment celling. By applying and algorithm based on the Ar Track Alvar library, the relative global position of the robot is calculated and used to update the AMCL probability of the robot's position.

Trough the publish of the destination topic a Dijkstra algorithm plan the global route to the robot achive the final destination, while Obstacle Avoidance solves local circulation in real time, respecting the building's kinematic limits and safety rules. The result goes to the Robot Motion Controller, which sends speeds to the base and, with each cycle, returns progress, replanning if it loses progress.

More sensitive parameter settings and adjustments related to the Magni database ROS package were performed [48]: increasing the controller frequency to 20 Hz for faster response; recalibrating minimum and maximum speeds on the x and theta axis for the new weight to reduce errors and increase fluidity. Adjustments were made to X, Y, and theta position tolerances to achieve oscillation-free stopping and reduce errors. Adjustments were made to the PID locomotion control to reduce oscillations at low speeds, avoid prolonged command saturation in tight turns, and stabilize the response to command timeouts.

To make the system reproducible and quick to start, a tmux bring-up was implemented. This session creates six panels and executes, in order and with controlled delays, the launches of the robotic base, laser sensor, tracking/localization system, RGB camera, map system with fiducial markers, and the Rasa connector system. The Rasa bridge receives navigation target publications from Rasa, which is physically connected to the NVIDIA Jetson Orin NX via Ethernet.

The implementation demonstrated that the proposed modular architecture is capable of successfully integrating different components, such as RASA, the facial recognition module, and external APIs, creating a cohesive and functional system. Despite the technical challenges faced, such as limited embedded hardware resources and difficulties in module compatibility, the project proved to be viable and applicable to various scenarios. This foundation allows us to move forward with evaluating the system in real use, through practical user testing, to validate the solution's effectiveness and identify opportunities for improvement.

Chapter 5

Results and Discussion

This chapter presents the results and discussions regarding the implementation and performance of the service robot, addressing its behavior in real-world usage situations, challenges, lessons learned, and improvements. Following the methodology described in [66], a test was conducted with 25 participants to validate the system and its modules. Responses were collected on a scale from 1 (strongly disagree) to 5 (strongly agree).

To evaluate the facial recognition module, a script was developed to guide participants' interaction with the robot in a controlled environment. Each participant began the test by spontaneously greeting the robot in English, using expressions such as "Hello," "Hi," or "Hey." The robot then performed facial detection and compared it with the local database. If the participant was already registered, the system greeted them by mentioning their name; otherwise, it requested their name for registration and added the new face to the database. This procedure was repeated three times per participant to verify the consistency of the facial recognition and the correct functioning of the registration.

Table 5.1 presents the results obtained in the evaluation of the service robot's facial recognition module. Overall, participants reported a positive user experience, highlighting the speed of the identification process and the reliability of face matching. Although some improvements were suggested—such as improving visual feedback and reducing errors in environments with variable lighting—the module's overall performance was considered satisfactory for the proposed objectives.

Table 5.1: Facial Recognition System Evaluation Responses.

N°	Statement	Avg.	SD
1	The robot was able to recognize my face quickly.	4.04	1.17
2	The facial recognition feature worked consistently.	3.92	1.26
3	I found the face recognition process confusing.	2.04	1.27
4	It was hard to understand whether the robot recognized me or not.	2.2	1.29
5	I would trust this system to recognize me in different situations.	3.64	1.19

Although facial recognition demonstrated good results in terms of detection and identification, it is important to emphasize the need to improve its robustness to handle different lighting conditions and viewing angles. The 0.55 threshold used managed to correctly recognize people in most attempts, but accuracy proved sensitive to movement or subtle changes in angle, resulting in considerable performance drops. Under ideal conditions, accuracy reached up to 80%, demonstrating the potential of the library used. However, to consolidate this performance, improvements to the algorithm and camera calibration will be necessary. Furthermore, it is essential to expand the face database to assess whether speed and reliability remain, thus making the module more robust and reliable for real-world environments.

The results of the external API evaluation are summarized in Table 5.2, which shows average scores above 3.8 for access efficiency and overall satisfaction, reflecting the good user acceptance of this functionality. Ease of use was also evidenced by low averages in items related to interaction difficulty, confirming that the implemented flows are intuitive. These results validate the decision to integrate weather and class schedule APIs, demonstrating that they add value to the robot and contribute to a more practical and informative user experience.

Although the external API module has proven efficient and well-received, there is still room for improvement and further value addition. One example is the limitation of the weather API, currently configured to respond only to Bragança; allowing users to select other cities would make the functionality more flexible and personalized. The class

Table 5.2: External API Assessment Responses.

N°	Statement	Avg.	SD
1	I found it difficult to ask about the weather.	1.72	1.34
2	I found the process of getting information fast and efficient.	3.88	0.93
3	I had difficulty accessing my class schedule through the robot.	1.72	1.02
4	I am satisfied with the current options offered by the robot.	3.88	1.13
5	I would like to see more features added to the robot in future versions.	4.36	1.11

schedule API offers potential for expanding the system's usability, allowing users to check professors' locations, access university maps, and obtain real-time event information.

As shown in Table 5.3, the robot's navigation system performed well during the tests. Participants gave high marks to the accuracy of the route followed and the feeling of security when following the robot. Navigation repeatability was also well evaluated, demonstrating the system's reliability in repeated tasks. The low averages for responses to environmental changes and overall reliability confirm that the system was considered stable and functional in the area where it was tested. These results reinforce the robot's potential to perform navigation tasks in controlled scenarios, serving as a basis for future expansions.

Table 5.3: Navigation System Assessment Responses.

N°	Statement	Avg.	SD
1	The robot followed the selected path accurately.	4.36	0.95
2	I felt safe following the robot.	4.24	1.05
3	The robot didn't respond well to changes in the environment.	1.88	1.05
4	The robot successfully repeated the navigation task multiple times.	4.24	1.01
5	I found the navigation system unreliable.	1.72	0.84

It's important to recognize that the navigation system still operates in a restricted area of the building, limited to a few ESTiG laboratories. While it has proven responsive to obstacles and pedestrian traffic, expansion to the rest of the building will require adjustments to routes and map configuration to ensure the system remains reliable and safe. Furthermore, it will be necessary to evaluate performance in areas with varying lighting patterns, heavy traffic, and potential sources of interference, all of which can

impact the robustness of the navigation.

As shown in Table 5.4, the evaluation of the LLM module revealed significant challenges to be overcome. The average of 2.40 for clarity and correctness of responses indicates that the system is not yet reliable enough to provide consistent information about the IPB and CeDRI. The high rate of irrelevant responses (3.88) reflects the model's difficulties in handling unexpected or out-of-domain questions. Even so, the functionality was well received by users, with an average overall satisfaction rating of 3.84, indicating that the module holds promise for further improvements.

Table 5.4: LLM System Assessment Responses.

N°	Statement	Avg.	SD
1	The robot answered my questions clearly and correctly.	2.40	1.29
2	The robot sometimes gave unrelated or incorrect answers.	3.88	1.17
3	The robot handled unexpected questions well.	2.76	1.36
4	I enjoyed using this question-and-answer feature.	3.84	1.14
5	I would rely on the robot to get information about IPB or Cedri.	3.48	1.23

During testing and development, it was observed that the language model used had significant limitations in terms of accuracy and robustness. Even with fine-tuning, due to the small size of the embedded LLM, the system struggled to answer basic questions, generating irrelevant or inaccurate responses. Furthermore, the intensive use of the device's memory when loading and using the model caused system crashes and prevented the use of other interface functions. Therefore, it was necessary to find an alternative to using an embedded LLM on the device. A viable solution would be to adopt a larger LLM, running on a local IPB server, ensuring better performance and reducing the impact on embedded processing. However, this would be subject to possible connection losses, impairing the user experience.

Table 5.5 presents the results of the evaluation of the speech recognition and generation systems implemented in the service robot. Overall, users demonstrated a preference for voice interaction over typing, with an average score of 4.00, indicating the potential of this functionality to increase the naturalness of communication with the robot. The perception of rapid response obtained an average of 3.36, suggesting room for improvement in

speech recognition. The average of 3.76 for the statement that the robot sometimes misunderstood what was said highlights limitations in speech recognition, while the difficulty in understanding speech generated by the robot obtained an average of 2.28, indicating room for improvement in the clarity and quality of the synthesized voice.

Table 5.5: Speech Recognition and Generation System Assessment Responses.

N°	Statement	Avg.	SD
1	The robot sometimes misunderstood what I said.	3.76	0.88
2	The robot responded quickly after my speech input.	3.36	1.25
3	I prefer interacting with the robot through speech instead of typing.	4.00	1.22
4	I found the robot's spoken responses difficult to understand.	2.28	1.24
5	I would feel comfortable using this voice system in real situations.	3.80	1.00

Despite offering an engaging experience, speech recognition still presents challenges, particularly when it comes to adapting to different accents and detecting background noise. Using Whisper on the GPU helped reduce response time, but it didn't completely eliminate errors in environments with multiple sound sources. Speech synthesis with gTTS also presented clarity and intonation issues, making it difficult to understand in some cases. These challenges highlight the need for improvements in both audio capture and the quality of the generated speech to achieve the expected level of usability.

Table 5.6 presents the results of the overall evaluation of the robotic system. Participants expressed general satisfaction with the robot, with high scores for aspects such as the desire to continue interacting and ease of learning. The interface was considered easy to use and well-integrated with the system's functionalities. On the other hand, some areas for improvement were highlighted, such as response consistency, signaling the need for improved dialogue processing. The low average for the need for technical support reinforces the system's autonomy of use. Overall, the results indicate a positive experience and validate the proposed architecture.

The results presented in this chapter demonstrate that the architecture developed for the service robot meets its proposed objectives, providing positive interaction with users. Although some aspects, such as response consistency and speech recognition accuracy, require improvement, the system has proven to be functional and promising. The

Table 5.6: Final System Assessment Responses.

N°	Statement	Avg.	Var.
1	I think that I would like to interact more with the robot.	4.32	0.90
2	I found the interface unnecessarily complex.	1.64	0.76
3	I thought the interface was easy to use.	3.92	1.29
4	I think that I would need the support of a technical person to be able to interact with the robot.	1.68	0.90
5	I found the various functions of the robot were well integrated.	3.60	0.91
6	I thought there was too much inconsistency in the robot's responses.	2.96	1.09
7	I imagine that most people would learn to interact with the robot very quickly.	4.00	0.95
8	I found the interface not user-friendly.	1.92	1.03
9	I felt very confident in interacting with the robot.	3.92	0.99
10	I needed to learn a lot of things before interacting with the robot.	1.76	1.05

validations performed indicate that the robot is capable of performing its functions autonomously and in an integrated manner, representing a significant contribution to the academic environment and to the advancement of service robotics.

Chapter 6

Conclusions and Future Works

This work presented the development of a modular architecture for service robots, integrating multiple subsystems, such as a chatbot with RASA and LLM, facial recognition, autonomous navigation with ROS, and an interactive graphical interface. The proposal aimed to enable the robot to operate autonomously, meeting user needs in different contexts, particularly in academia.

The integration of these modules demonstrated practical feasibility, with positive results in user testing. The chatbot proved functional and responsive, albeit limited by the size of the embedded LLM, while facial recognition demonstrated a acceptable accuracy rate in controlled environments. The navigation system, in turn, was able to guide users safely within the mapped space, validating the proposed architecture. These tests provided evidence that the service robot can act as an effective assistant in university settings.

Despite the overall success, development revealed challenges that need to be overcome to improve the system's robustness. The LLM module presented hallucinations and difficulty answering specific questions about the IPB and CeDRI, indicating the need for additional training and model adjustments. Speech recognition, while promising, proved sensitive to background noise and accents, impacting reliability in some scenarios. Furthermore, navigation is still limited to a portion of the environment, requiring expansion to cover the entire building.

Overall, the developed architecture represents a significant contribution to the field of service robotics, particularly by integrating multiple modules in a modular and scalable manner. The work highlights how a distributed approach can facilitate system maintenance, customization, and evolution. Furthermore, the research strengthens the role of service robots as assistants in academic settings, offering valuable insights for future applications and improvements in the design of human-robot interactions.

6.1 Future Works

Based on the results obtained, the limitations identified and the possibilities for system evolution, the following future works stand out:

- Improve the LLM module fine-tuning database with institutional data from IPB and CeDRI to reduce hallucinations and improve response accuracy;
- Explore running LLM on local servers or in the cloud, allowing the use of larger models without impacting the performance of the service robot;
- Expand the database of the facial recognition module and implement emotion detection and dynamic behavior adjustment strategies to improve human-robot interaction (HRI);
- Expand navigation system coverage throughout the building;
- Develop an automatic charging station to increase the robot's operational autonomy and facilitate its maintenance;
- Integrate additional features such as event APIs, multi-language support, and personalized recommendations;
- Use the robot as an experimental platform for research and innovation at IPB.

Bibliography

[1] Y. Jiang, X. Li, H. Luo, S. Yin, and O. Kaynak, "Quo vadis artificial intelligence?" *Discover Artificial Intelligence*, vol. 2, no. 1, p. 4, 2022.

[2] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1 729 881 419 839 596, 2019.

[3] E. Islas-Cota, J. O. Gutierrez-Garcia, C. O. Acosta, and L.-F. Rodríguez, "A systematic review of intelligent assistants," *Future Generation Computer Systems*, vol. 128, pp. 45–62, 2022.

[4] T. Wu et al., "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122–1136, 2023.

[5] *Robots and robotic devices – vocabulary*, Geneva, Switzerland: International Organization for Standardization, 2021.

[6] D. Liu, C. Li, J. Zhang, and W. Huang, "Robot service failure and recovery: Literature review and future directions," *International Journal of Advanced Robotic Systems*, vol. 20, no. 4, p. 17 298 806 231 191 606, 2023.

[7] N. G. Hockstein, C. Gourin, R. Faust, and D. J. Terris, "A history of robots: From science fiction to surgical robotics," *Journal of robotic surgery*, vol. 1, pp. 113–118, 2007.

[8] J. A. Gonzalez-Aguirre et al., "Service robots: Trends and technology," *Applied Sciences*, vol. 11, no. 22, p. 10 702, 2021.

[9] B. Dynamics. "Atlas: The world's most dynamic humanoid robot." Accessed: Feb. 18, 2025. [Online]. Available: https://bostondynamics.com/atlas/.

[10] Ros.org. [Online]. Available: https://www.ros.org/, Accessed: Jun. 2025.

[11] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science robotics*, vol. 7, no. 66, eabm6074, 2022.

[12] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A review of sensing technologies for indoor autonomous mobile robots," *Sensors*, vol. 24, no. 4, p. 1222, 2024.

[13] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A review of sensing technologies for indoor autonomous mobile robots," *Sensors*, vol. 24, no. 4, 2024. DOI: 10.3390/s24041222.

[14] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[15] U. Iqbal, T. Davies, and P. Perez, "A review of recent hardware and software advances in gpu-accelerated edge-computing single-board computers (sbcs) for computer vision," *Sensors*, vol. 24, no. 15, p. 4830, 2024.

[16] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[17] C. Manning. "Artificial intelligence definitions," Stanford University Human-Centered Artificial Intelligence (HAI). [Online]. Available: https://hai.stanford.edu/assets/files/2023-03/AI-Key-Terms-Glossary-Definition.pdf. Accessed: Jun. 2025.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] R. S. Sutton, A. G. Barto, et al., *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1.

[20] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[21] A. Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[22] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2017.

[23] A. Brohan et al., "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*, PMLR, 2023, pp. 287–318.

[24] B. Zitkovich et al., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, PMLR, 2023, pp. 2165–2183.

[25] I. F. of Robotics (IFR). "Record of 4 million robots working in factories worldwide." Accessed: Feb. 18, 2025. [Online]. Available: `https://ifr.org/ifr-press-releases/news/record-of-4-million-robots-working-in-factories-worldwide`.

[26] L. Liu, R. Zhong, A. Willcock, N. Fisher, and W. Shi, "An open approach to energy-efficient autonomous mobile robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023.

[27] ISO. "Iso/fdis 13482. "[Online]. Available: `https://www.iso.org/standard/83498.html`. Accessed: Jun. 2025.

[28] *Turtlebot 4*, 2023. [Online]. Available: `https://clearpathrobotics.com/turtlebot-4/`, Accessed: Jun. 2025.

[29] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[30] M. A. Goodrich, A. C. Schultz, et al., "Human–robot interaction: A survey," *Foundations and trends® in human–computer interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[31] I. H. Han et al., "Human-robot interaction and social robot: The emerging field of healthcare robotics and current and future perspectives for spinal care," *Neurospine*, vol. 21, no. 3, p. 868, 2024.

[32] A. Obaigbena, O. A. Lottu, E. D. Ugwuanyi, B. S. Jacks, E. O. Sodiya, O. D. Daraojimba, et al., "Ai and human-robot interaction: A review of recent advances and challenges," *GSC Advanced Research and Reviews*, vol. 18, no. 2, pp. 321–330, 2024.

[33] R. Haeb-Umbach, J. Heymann, L. Drude, S. Watanabe, M. Delcroix, and T. Nakatani, "Far-field automatic speech recognition," *Proceedings of the IEEE*, vol. 109, no. 2, pp. 124–148, 2020.

[34] H. Su, W. Qi, J. Chen, C. Yang, J. Sandoval, and M. A. Laribi, "Recent advancements in multimodal human–robot interaction," *Frontiers in Neurorobotics*, vol. 17, p. 1 084 000, 2023.

[35] L. Blavette et al., "Acceptability and usability of a socially assistive robot integrated with a large language model for enhanced human-robot interaction in a geriatric care institution: Mixed methods evaluation," *JMIR Human Factors*, vol. 12, no. 1, e76496, 2025.

[36] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2013, pp. 301–308.

[37] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[38] J. Mišeikis et al., "Lio-a personal robot assistant for human-robot interaction and care applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5339–5346, 2020.

[39] Z. Janeczko and M. E. Foster, "A study on human interactions with robots based on their appearance and behaviour," in *Proceedings of the 4th Conference on Conversational User Interfaces*, 2022, pp. 1–6.

[40] J. K. Barfield, "The role of name, origin, and voice accent in a robot's ethnic identity," *Sensors*, vol. 24, no. 19, p. 6421, 2024.

[41] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017.

[42] T. Schick et al., "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 539–68 551, 2023.

[43] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[44] E. J. Hu et al., "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.

[45] C. McGinn, "Why do robots need a head? the role of social interfaces on service robots," *International Journal of Social Robotics*, vol. 12, no. 1, pp. 281–295, 2020.

[46] M. McTear, *Conversational ai: Dialogue systems, conversational agents, and chatbots*. Springer Nature, 2022.

[47] A. de Oliveira Júnior, "Combining particle filter and fiducial markers in a slam-based approach to indoor localization of mobile robots," M.S. thesis, Instituto Politecnico de Braganca (Portugal), 2021.

[48] U. Robotics, *Magni silver mobile based robot.* [Online]. Available: `https://www.ubiquityrobotics.com`, Accessed: Jun. 2025.

[49] E. Worthy. [Online]. Available: https://www.eco-worthy.com, Accessed: Jun. 2025.

[50] R. Pi, *Raspberry pi 5: Product brief, 2025*. [Online]. Available: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf, Accessed: Jun. 2025.

[51] Nvidia. [Online]. Available: https://www.nvidia.com/en-eu/autonomous-machines/embedded-systems/jetson-orin/, Accessed: Jun. 2025.

[52] Hokuyo, *Ust-10lx specification*, 2015. [Online]. Available: https://www.hokuyo-aut.jp/dl/UST-10LX_Specification.pdf.

[53] T. Baltovski, *Urg_node*. [Online]. Available: http://wiki.ros.org/urg_node, Accessed: Jun. 2025.

[54] RealSense, *Intel realsense d400 series product family: Datasheet*, 2019. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf.

[55] Doronhi, *Ros wrapper for intel realsense devices*. [Online]. Available: https://github.com/IntelRealSense/realsense-ros, Accessed: Jun. 2025.

[56] S. Studio, *Respeaker mic array v2.0*. [Online]. Available: https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/, Accessed: Jun. 2025.

[57] Rasa, *Rasa documentation*. [Online]. Available: https://rasa.com/docs/, Accessed: Jun. 2025.

[58] A. Geitgey, *Face_recognition*. [Online]. Available: https://github.com/ageitgey/face_recognition, Acessad: Jun. 2025.

[59] Flask, *Flask documentation*. [Online]. Available: https://pypi.org/project/Flask/, Accessed: Jun. 2025.

[60] gTTS, *Gtts documentation*. [Online]. Available: https://pypi.org/project/gTTS/, Accessed: Jun. 2025.

[61] Pydub, *Pydub documentation*. [Online]. Available: `https://pypi.org/project/pydub/`, Accessed: Jun. 2025.

[62] Sounddevice, *Sounddevice documentation*. [Online]. Available: `https://pypi.org/project/sounddevice/`, Accessed: Jun. 2025.

[63] PyPI, *Noisereduce documentation*. [Online]. Available: `https://pypi.org/project/noisereduce/`, Accessed: Jun. 2025.

[64] TinyLlama, *Tinyllama documentation*. [Online]. Available: `https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0`, Accessed: Jun. 2025.

[65] CeDRI, *Home - cedri*. [Online]. Available: `https://cedri.ipb.pt/`, Accessed: Jun. 2025.

[66] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

Appendix A

Publications

C. B. Alcantara, A. O. Júnior, J. Costa, L. Ricken, A. Mendes and P. Leitao, "Artificial Intelligence-Based User Interaction Module for Autonomous Mobile Service Robots," 2024 IEEE 3rd Industrial Electronics Society Annual On-Line Conference (ONCON), Beijing, China, 2024, pp. 1-6, doi: 10.1109/ONCON62778.2024.10931279.

Appendix B

Repositories

The source code used in the development of this dissertation is available online at the following page:

Repository: github.com/Robot-CeDRI

Release used in results: `v1.0.0` (data `2025-10-28`).