INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




From Relational Waters to Intelligent Oceans: A Lakehouse-Centric Approach to Conversational Artificial Intelligence

JOANA RODRIGUES FIGUEIREDO
julho de 2025

P.PORTO

From Relational Waters to Intelligent Oceans: A Lakehouse-Centric Approach to Conversational Artificial Intelligence

Joana Rodrigues Figueiredo

Student No.: 1201150

Dissertation for the Master's Degree in Artificial Intelligence Engineering

Supervisor: Luis Filipe de Oliveira Gomes
Advisor: João Pedro Martins Vieira e Moreira

Evaluation Committee:
President:
Doctor Carlos Fernando da Silva Ramos; Full Professor, Polytechnic of Porto - School of Engineering

Members:
Doctor Maria Goreti Carvalho Marreiros; Full Professor, Polytechnic of Porto - School of Engineering
Doctor Luís Filipe de Oliveira Gomes; Assistant Professor, Polytechnic of Porto - School of Engineering

Porto, July 14, 2025

"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring"

- Marilyn Monroe

Abstract

The digital transformation of data-intensive operational systems demands architectures capable of handling large volumes of heterogeneous and unstructured data while enabling real-time intelligent decision-making. In the water management domain, where legacy systems and operational complexity often obstruct innovation, there is an increasing need to adopt artificial intelligence-powered solutions that promote efficiency, traceability, and accessibility. Responding to this challenge, this dissertation presents CLARA — a Conversational Lakehouse Architecture supported by Real-time Artificial intelligence. CLARA is a modular solution that integrates modern data infrastructures, artificial intelligence models, and natural language interaction to support intelligent management in water utility operations.

CLARA was conceived and developed from scratch, following the data lakehouse paradigm to consolidate structured and unstructured data, such as field images. The infrastructure adopts a medallion architecture (Bronze, Silver, Gold) and includes pipelines for ingestion, loading, and transformation. Particular attention was given to documentation of transformations, and integration of flows for experiment tracking, enabling a robust foundation for artificial intelligence development and data governance.

The solution currently features two artificial intelligence models that demonstrate how the lakehouse paradigm can support intelligent reasoning beyond conventional structured data processing. The first is an optical character recognition model, which enables the automated interpretation of water meter readings directly from field images, a type of unstructured data typically excluded from traditional storage systems. This model exemplifies how AI can be embedded into the data architecture to support validation and data quality assurance workflows. The second is a predictive model based on neural networks, designed to anticipate the symptom of the next operational intervention by analyzing historical maintenance sequences. Together, these models illustrate the potential of unifying data storage and artificial intelligence reasoning within a single environment.

At the user interaction layer, a custom-built conversational assistant leverages a cascade of large language models to classify and respond to user queries in real-time. The system routes each input to one of four specialized modules: (1) to access structure data in real-time, (2) to execute and access artificial intelligence models, (3) to consult software support manuals, and (4) to provide fallback conversational support only on water-related topics. The assistant also integrates multilingual support and a semantic permission-verification mechanism that maps the user's intent and role to the structure of the underlying database, preventing unauthorized actions.

Developed in partnership with A2O – Água, Ambiente e Organização, Lda., and validated through four real-world case studies, CLARA demonstrated how a carefully orchestrated artificial intelligence pipeline, backed by an efficient data infrastructure, can modernize and improve decision-making, enhance transparency, and simplify access to complex systems through natural language.

Keywords: data lakehouse, conversational AI, AI model integration, OCR validation, permission-aware assistant

Resumo

A transformação digital de sistemas operacionais com elevada densidade de dados exige arquiteturas capazes de processar grandes volumes de informação heterogénea e não estruturada, assegurando simultaneamente a tomada de decisão inteligente em tempo real. No domínio da gestão da água, onde os sistemas legados e a complexidade operacional frequentemente dificultam a inovação, torna-se cada vez mais urgente a adoção de soluções inteligentes que promovam eficiência, rastreabilidade e acessibilidade. Em resposta a este desafio, esta dissertação apresenta a CLARA — a Conversational Lakehouse Architecture supported by Real-time Artificial intelligence. Trata-se de uma solução modular que combina infraestruturas de dados modernas, modelos de inteligência artificial e interação em linguagem natural para apoiar a gestão inteligente nas operações dos serviços de água.

A CLARA foi desenvolvida de raiz, segundo o paradigma lakehouse, para consolidar dados estruturados e não estruturados, como imagens recolhidas no terreno. A infraestrutura adota uma arquitetura medalhão (Bronze, Silver, Gold) e inclui pipelines para ingestão, carregamento e transformação. Foi dada especial atenção à documentação das transformações e à integração de fluxos para rastreamento de experiências, assegurando uma base sólida para o desenvolvimento de modelos de inteligência artificial e para a governação de dados.

A solução conta atualmente com dois modelos de inteligência artificial que demonstram como o paradigma lakehouse pode suportar raciocínio inteligente para além do processamento convencional de dados estruturados. Um é um modelo de reconhecimento ótico de caracteres, que permite a leitura automática de contadores de água a partir de imagens recolhidas no terreno, um tipo de dado normalmente excluído de sistemas tradicionais. O segundo é um modelo preditivo baseado em redes neuronais, concebido para antecipar o sintoma da próxima intervenção operacional com base em sequências históricas de manutenção. Em conjunto, estes modelos ilustram o potencial da unificação entre armazenamento de dados e raciocínio artificial num único ambiente.

Na camada de interação com o utilizador, foi desenvolvido um assistente conversacional que recorre a uma cascata de modelos de linguagem de grande escala para classificar e responder, em tempo real, às perguntas formuladas. O sistema encaminha cada input para um de quatro módulos especializados: (1) acesso a dados estruturados, (2) execução e consulta de modelos de inteligência artificial, (3) consulta de manuais de apoio ao software e (4) suporte conversacional de retaguarda sobre temas do domínio da água. O assistente integra ainda suporte multilingue e um mecanismo semântico de verificação de permissões, que cruza a intenção e o perfil do utilizador com a estrutura da base de dados, prevenindo ações não autorizadas.

Desenvolvida em parceria com a A2O – Água, Ambiente e Organização, Lda., e validada através de quatro casos de estudo em contexto real, a CLARA demonstrou como uma pipeline de inteligência artificial cuidadosamente orquestrada, apoiada por uma infraestrutura de dados eficiente, pode modernizar a tomada de decisão, aumentar a transparência e simplificar o acesso a sistemas complexos por via da linguagem natural.

Palavras-chave: data lakehouse, conversational AI, AI model integration, OCR validation, permission-aware assistant

Acknowledgement

The completion of this dissertation marks the culmination of a demanding yet deeply enriching academic journey. Reaching this point was only possible thanks to the support, guidance, and inspiration of several people, to whom I would like to express my sincere gratitude.

First and foremost, I would like to thank my academic advisor, Doctor Luis Gomes, for his rigorous supervision, constant availability, and valuable suggestions that challenged me to raise the quality of this work. I am also grateful to my supervisor, Engineer Pedro Vieira, for being an exceptional mentor and for encouraging me every day to think outside the box.

I would also like to express my heartfelt appreciation to Engineer Jorge Tavares, for believing in me from the beginning, for giving me the opportunity to carry out my master's degree in a business environment, and for playing such a significant role in my academic, professional, and personal growth.

I am also thankful to my parents, who brought me into this world and have never let me feel anything but love, care, and dedication ever since. Without you, I would not be where I am today, and for that, I will be forever grateful.

A special thank you goes to all my colleagues at A2O – Água, Ambiente e Organização, Lda., for offering me such a positive first professional experience. When you love what you do, you'll never work a day in your life, and with a team like this, that becomes easy.

And last but certainly not least, I would like to thank my best friend, quite possibly my male version and, by happy coincidence, my boyfriend. João, thank you for making me laugh during the hardest moments, for giving the right advice at the right time, and for never letting me take life too seriously.

Finally, I would like to thank myself for the resilience, the consistent effort, and for never giving up, even on the days when giving up seemed easier. Thank you, Joana.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

ACID	atomicity, consistency, isolation and durability.
AI	artificial intelligence.
AIS	automatic identification systems.
APDA	Portuguese Water Distribution and Drainage Association.
APDA	Associação Portuguesa de Distribuição e Drenagem de Águas.
API	application programming interface.
BD	big data.
BDA	big data analytics.
BI	business intelligence.
CAD	computer-aided design.
CDC	change data capture.
CLARA	Conversational Lakehouse Architecture supported by Real-time AI.
CPU	central processing unit.
CRM	customer relationship management.
CVAT	Computer Vision Annotation Tool.
DC-AI	data-centric artificial intelligence.
DDO	data-driven organization.
DL	data lake.
DRL	deep reinforcement learning.
DSS	decision support systems.
DSSL	deep semi-supervised learning.
DW	data warehouse.
EDW	enterprise data warehouses.
ELT	extraction, loading and transformation.
ERP	enterprise resource planning.
GDPR	general data protection regulation.
GIS	geographic information system.
GNN	graph neural network.
GPU	graphic processing unit.
HRL	hierarchical reinforcement learning.
IoD	internet of drones.
IoT	internet of things.

IQR	interquartile range.
IT	information technology.
LH	data lakehouse.
LIMS	laboratory information management system.
LLM	large language model.
LSTM	long short-term memory.
MARL	multi-agent reinforcement learning.
MC-AI	model-centric artificial intelligence.
MDM	master data management.
ML	machine learning.
MM-LLM	multimodal large language model.
MPP	massively parallel processing.
NLI	natural language interface.
NLP	natural language processing.
OBB	oriented bounding boxes.
OCR	optical character recognition.
OLAP	online analytical processing.
OLTP	online transaction processing.
PES	power and energy systems.
PPO	proximal policy pptimization.
RAG	retrieval-augmented generation.
RAM	rapid access memory.
RDBMS	relational database management system.
RLHF	reinforcement learning with human feedback.
RQ	research question.
SCADA	supervisory control and data acquisition.
SGD	stochastic gradient descent.
SQ	specific question.
SSD	solid state drivers.
SSL	semi-supervised learning.
SSL	self-supervised learning.

Chapter 1

Introduction

This chapter provides an overview of the motivations, objectives, contributions, and structure of this dissertation. The chapter is divided into six sections. It begins with the Contextualization section where the background and motivations for this work are presented, particularly the importance of data in the new era of artificial intelligence (AI) applications. The Market Contextualization and Impact on the Water Sector discusses the specific needs and potential benefits of these innovations in the water management sector. The Problem Statement assesses the need for a solution to the specific problem this dissertation is aiming to solve. In the Research Questions and Objective section all the questions and goals that guided the investigation and development are presented. The Contributions section highlights all the outcomes of the present dissertation. And finally, the Document Structure section explains how this dissertation is structured, as well as, a light description of what the reader will find amongst each chapter.

1.1 Contextualization

We are currently going through a significant shift in the fundamental paradigm of software engineering, in which machine learning (ML) is beginning to be considered the new software paradigm, often referred to as software 2.0 (Whang Steven Euijong et al. 2023). While traditional software development follows a typical sequence that includes stages, such as design, implementation, and debugging, ML has its starting point in the data that will feed the models during the training process. Since the performance of AI models heavily depends on data quality, a substantial part of ML system development is dedicated to its preparation, as even the best ML algorithms cannot achieve satisfactory performance without quality data.

Data has become an extremely valuable resource in the new era we are going through, so it is not surprising to see that it is being generated at astronomical rates every second. This huge amount of data is known as big data (BD) (Naeem et al. 2022). The term was first introduced by Roger Magoulas in 2005, who described it as large volumes of data whose quantity and complexity of structure make it impossible to store and process in conventional databases and applications. This is due to the fact that the data generated today has an increasingly complex structure, such as structured and unstructured data, which brings with it various problems, such as indexing, organization, search, analysis, and visualization.

As data becomes increasingly complex and is generated at extremely high rates, it is crucial that organizations invest time and resources in its correct storage and subsequent processing. After all, data is only valuable if it allows useful conclusions and insights to be drawn (Harby and Zulkernine 2025). With this goal in mind, various hierarchical systems have emerged over the years, fueled by the constant evolution of technologies. These systems have expanded the capabilities of traditional databases, giving rise to solutions capable of storing, managing, analyzing and consulting large volumes of structured, unstructured and semi-structured data,

as well as data from continuous flows, using architectures such as data warehouse (DW) and data lake (DL).

The evolution of databases began with relational database management system (RDBMS), which were designed to manage structured data and online transaction processing (OLTP), but had significant limitations in intensive analytical loads (Mazumdar, Hughes, and Onofre 2023). These limitations prompted the emergence of DW, which centralized structured data from multiple sources, enabling complex analyses and supporting data-based decision making. However, the exponential growth of unstructured and semi-structured data from sources, such as internet of things (IoT) sensors and social media, revealed the shortcomings of DW in terms of flexibility and scalability, which fueled the development of DL (Harby and Zulkernine 2025). Despite being able to store large volumes of raw data, DL often faced governance and organization problems, often turning into data swamps, which essentially means that due to the vast volume of unnecessary data, the DL start to have too much noise that turns the useful data into useless one, making it difficult to extract analytical value (Oreščanin and Hlupić 2021). To solve these limitations, the data lakehouse (LH) architecture emerged, combining the scalability of DL with the structure and governance of DW, making it possible to store and process structured, semi-structured, and unstructured data in an efficient and integrated way (Mazumdar, Hughes, and Onofre 2023).

Several studies have explored the benefits that a LH can bring to an organization, not only in terms of data management, maintenance, and governance, but also in training AI models. A relevant example is the Deep Lake platform, designed as a specific LH for training deep learning models. This platform makes it possible to store all kinds of data, convert it into tensors which facilitates the compatibility with deep learning libraries such as TensorFlow and train models directly with the data stored in the LH (Hambardzumyan et al. 2022). This eliminates the need to export to local files, promoting greater efficiency and enabling strict control of data versions. This example represents just one of the many possibilities that modern data storage infrastructures offer in the field of AI. By allowing data to be better structured and organized, these infrastructures make it significantly easier for AI models to interpret it.

One of the fields that has benefited most from this evolution is generative AI. The use of natural language in data environments has been widely studied, especially since generative models gained prominence with the popularization of ChatGPT (N. Gupta and Yip 2024). Large models, such as those from the GPT family or BERT, known as large language models (LLMs), are based on transformer architecture and have been trained with billions of parameters and massive volumes of data (Shanahan 2024). These models have been transforming the development of AI solutions, especially in the way they interpret and generate human language. One of the most promising applications of these models is the development of natural language interfaces (NLIs), which allow users to interact with complex systems through instructions written in natural language (Őzcan et al. 2020). These interfaces make it possible to explore databases, perform queries and obtain interpreted answers, even by users without advanced technical knowledge.

Among NLIs, systems based on chatbots have received particular attention, as they support continuous conversational interactions with users (Kavaz, Puig, and Rodríguez 2023). However, the development of these systems involves overcoming relevant challenges, such as the ambiguity of natural language, maintaining context throughout the conversation and clearly communicating the system's capabilities. Recent studies, such as those presented in (Affolter, Stockinger, and Bernstein 2019) and (Franciscatto et al. 2022), analyze examples of existing NLIs, highlighting their strengths and limitations. Even so, most of the work to date has focused on traditional databases that operate with structured data, leaving solutions oriented towards heterogeneous

data, such as those that integrate structured and unstructured data simultaneously, such as LH, unexplored.

Applying natural language-based technologies to a LH infrastructure would bring a number of challenges, besides the heterogeneity of the data, as mentioned previously, these infrastructures involve a greater amount of metadata. While traditional databases deal mainly with tables and columns, LHs often include hierarchically organized data layers. These factors, among others, add complexity to the integration of these technologies in these types of systems. Therefore, this dissertation will explore the seamless integration of AI-based models in LH while including an NLP layer for user interaction.

In this scenario of increasing data complexity and advances in AI, this dissertation proposes an innovative solution that combines the LH paradigm with AI models and NLIs, enabling effective interaction with large volumes of heterogeneous data. The proposed solution explores in a practical way how to integrate query, reasoning, and prediction mechanisms in a unified, scalable, and conversational environment geared toward the management of critical infrastructure in the water sector. This approach represents a step forward from traditional solutions by demonstrating the potential of emerging technologies in the digital transformation of complex systems.

1.2 Market Contextualization and Impact on the Water Sector

This dissertation is being developed in partnership with the company A2O - Água, Ambiente e Organização, Lda., based in Porto - Portugal, founded in 1993, that markets the Navia software, which is widely used by multiple water management entities at a national level as well as a consolidated presence in Spain and Brazil. This software emerged as a response to a critical need for modernization in a sector where, until then, was widely dispersed and often poorly or not at all digitized.

Navia addresses several challenges faced by water management entities, including the difficulty in accessing field data necessary for calculating performance indicators, the lack of effective tools for organizing and planning teams and tasks, and the absence of centralized information, which was traditionally distributed across multiple systems used in the sector, such as supervisory control and data acquisition (SCADA), enterprise resource planning (ERP), customer relationship management (CRM), laboratory information management system (LIMS), and geographic information system (GIS). This software is further distinguished by its high degree of customization, enabling it to meet the specific needs of each client, supporting extensive databases and, in many cases, storing historical data spanning more than a decade.

In a context of constant technological evolution, where BD plays a crucial role, Navia stands out as a strategic tool to keep pace with sectoral transformations. The vast amount of historical data stored within the software constitutes a valuable resource for the development of AI models, which have the potential to make water resource management more efficient, precise, and practical.

Despite past technological developments in the water sector, the Portuguese Water Distribution and Drainage Association (APDA) [1] believes that this sector in Portugal needs a real commitment to innovation and the adoption of new technologies, such as those powered by AI, in order to make it more efficient and sustainable (Associação Portuguesa de Distribuição e Drenagem de Águas 2024). However, this is not just a national challenge, since an analysis carried out by a group of researchers in the United States reached similar conclusions (Vekaria and Sinha

[1] translated by the author from the Portuguese form Associação Portuguesa de Distribuição e Drenagem de Águas (APDA)

2024). The study conducted by these authors revealed that many United States companies still do not have AI solutions in place, and when they do, they often rely on external suppliers. This situation leads to significant dependencies and promotes a 'black box' perception of AI, which creates problems related to trust and the sustainability of solutions.

It is undeniable that AI can have an extremely positive impact on this sector, making it more effective, adaptable and sustainable (Kamyab et al. 2023). The various potential applications of this technology include: optimizing operations in water infrastructures, asset management, leak detection, predicting water demand and even significant improvements in water treatment (Garrido-Baserba et al. 2020) (Sela et al. 2025).

For AI to be successfully implemented in the sector, it is essential that management organizations understand its characteristics, benefits and challenges. At the same time, it is essential to ensure the availability of quality data, as well as investing time and resources in its preparation, as this plays a central role in the development of quality AI-based solutions. This concept is often summarized by the principle of 'Garbage In, Garbage Out' (Vekaria and Sinha 2024). In addition, the integration of human knowledge should be a priority, since the operators who work with the infrastructure on a daily basis have a wealth of practical knowledge that can be of enormous value in generating relevant insights and improving predictive models.

Another important aspect to highlight is the transformative potential of generative AI in the operation and management of companies in the water sector. Although its use is still limited in this field, this technology presents promising possibilities. The use of natural language allows for more intuitive and accessible interactions, contributing to the democratization of software, the simplification of interactions with systems and greater efficiency in data queries, among other benefits (Sela et al. 2025).

Finally, it is important to emphasize that the application of such powerful technology will have a significant impact on decision support and daily operations carried out by humans in the sector. Since water infrastructures are considered critical infrastructures under the AI Act, total control of these infrastructures by AI systems would be classified as an unacceptable risk (Salim, S., and B. 2024). This means that any decision with the potential to jeopardize the population's water supply or public health must always be validated by a human expert.

The proposed solution in this dissertation provides a practical response to the challenges identified in the water sector, promoting operational efficiency, valuing the knowledge of operators, and democratizing access to critical information, always with humans at the center of decisions.

1.3 Problem Statement

This dissertation stems from a real problem identified in the water sector, where the high potential of applying AI-based solutions to transform raw data into useful and actionable knowledge is recognized. As discussed earlier, the evolution of data architectures and natural language models has opened up new possibilities for accessing and exploring complex data. However, for these cutting-edge technologies to be successfully applied, it is essential to ensure their viability in real contexts, considering the limitations and specificities of existing infrastructures.

It was in this context that Navia was identified as a concrete case of application: a consolidated platform in the sector, rich in data, but limited by its current structure. Navia serves a wide and diverse range of customers in the water sector, which amplifies the potential impact of integrating AI-based functionalities into the system. Such integration could significantly enhance operational efficiency, improve information quality, and streamline processes through automation. Yet, unlocking this potential requires addressing a critical barrier: the underlying

structure of Navia's database. The quality and organization of this data repository plays a decisive role in the success of any AI initiative, and currently, they present considerable challenges that must be overcome.

Considering that Navia has been around for over 20 years, it is understandable that, in the early stages of its development, not enough time was invested in the structured design of the database or its documentation. Over time, the database has grown in a disorganized manner, accumulating hundreds of tables, many of which are empty, redundant, or without a clear purpose. The lack of documentation of the data structure, the ambiguity in the naming of fields, and the poor semantic normalization make its interpretation extremely difficult, even for experienced technical profiles. This issue complicates not only the development of ML models, but also the implementation of natural language-based features that allow users to perform quick and autonomous queries.

Most current management systems, including Navia, continue to rely on non-intuitive interfaces that force users to navigate through rigid menus or request technical support to access relevant data. This reality compromises agility in decision-making and prevents the democratization of access to information. At the same time, the water sector is increasingly data-intensive, with sources as diverse as IoT sensors, maintenance records, images, and technical forms. This complexity requires modern data infrastructures and intelligent solutions that can transform large volumes of raw data into actionable insights.

However, the current database structure acts as a real bottleneck for this transformation. The lack of accessibility, context, and standardization makes it difficult to extract value from data and slows down the technological evolution of organizations. This scenario is particularly limiting for the adoption of modern AI-based approaches, such as prediction models or autonomous validation systems for operational information.

At the same time, recent advances in LLMs have opened up new possibilities for interacting with complex systems through natural language. These models, trained with billions of parameters and massive volumes of text, demonstrate a remarkable ability to understand human instructions, maintain the context of dialogue, and generate useful responses. Their integration into management platforms can enable a new paradigm of information access that is simpler, more accessible, and user-centered.

For the sector to take advantage of these advances, it is essential to overcome current structural constraints and create a solid foundation for the development of AI-based solutions. It was precisely with this goal in mind that CLARA - Conversational Lakehouse Architecture supported by Real-time AI – was conceived during this dissertation. To do so, an innovative architecture was conceived to combine a modern LH data infrastructure with a layer of conversational intelligence based on LLMs, and with AI models dedicated to operational tasks, such as intervention prediction, and automatic reading validation. This solution was tested and evaluated to assess how it is possible to transform raw and scattered data into actionable, accessible, and useful knowledge.

1.4 Research Questions and Objectives

In order to frame and guide the research process as well as the development of this dissertation, a main research question (RQ) was defined, complemented by specific questions (SQ) aimed at delving deeper into the different aspects of the problem being analyzed. These elements serve as a basis for structuring the work and ensuring that the defined objectives are achieved in a coherent and integrated manner throughout the study. The list of Research Questions is the following:

- **RQ1**: How can modern data management infrastructures be used to support advanced analyses and artificial intelligence applications, while providing an intelligent virtual assistant for business aid and real-time analyses?
  - **SQ1**: What are the main differences between data lakes, data warehouses, and data lakehouses, and how can these infrastructures influence data analysis processes and the application of artificial intelligence?
  - **SQ2**: What are the challenges and best practices in designing and implementing modern data infrastructures to integrate data from relational and non-relational sources in order to support AI technologies?
  - **SQ3**: What are the capabilities and limitations of virtual assistants for analysing data in real-time through conversational interfaces?
  - **SQ4**: How can virtual assistants manage multiple large language models with different purposes, and how can real-time integration with business data infrastructures support this orchestration effectively?

To support the dissertation, a list of objectives was formulated according to the RQ and SQs with the aim of guiding the design and implementation of the CLARA solution. The proposed objectives are as follows:

- **O1:** Explore and evaluate different modern data infrastructures (i.e. data lake, data warehouse, and data lakehouse) to identify their characteristics, benefits, and limitations in supporting advanced analyses and the application of artificial intelligence.
- **O2:** Design and implement a modern data infrastructure solution to store, organize, and integrate data from relational and non-relational sources, ensuring scalability, flexibility, and compatibility with artificial intelligence technologies.
- **O3:** Develop an extraction, loading and transformation (ELT) pipeline that operates seamlessly between relational databases and modern data infrastructures, enabling ingestion and organization of data in different formats, with a focus on reuse for analysis.
- **O4:** Develop and test artificial intelligence models to validate their integration and execution within the data infrastructure.
- **O5:** Develop and test a virtual assistant capable of using different large language models adapted to different purposes, including information retrieval, direct access to artificial intelligence models, clarification of questions based on technical documentation, and a conversational fallback mechanism restricted to a specific domain.
- **O6:** Integrate the virtual assistant with the modern data infrastructure, using methods that guarantee consistent data retrieval and consultation, without jeopardizing the robustness of responses.
- **O7:** Test and validate the integrated solution (i.e., modern data infrastructure, AI models, and virtual assistant) in real scenarios in the water sector, assessing its applicability in analysing data and supporting business decision-making.

The relationships between research questions and the objectives are demonstrated in Table 1.1 below.

Table 1.1: Relationships between Research Questions and Objectives

Research Question	Objectives
RQ1	O1, O2, O3, O4, O5, O6, O7
SQ1	O1
SQ2	O2, O3, O4
SQ3	O5
SQ4	O5, O6

1.5 Contributions

This dissertation resulted in a significant set of practical, structural, and scientific contributions, both from the technological evolution made into Navia software and to the strategic development of the company in the field of AI applied to water systems management. These contributions are divided into four main areas: technological products developed, modernization of data infrastructure, contribution to two Portugal 2030–funded national projects, and academic and sector involvement.

1.5.1 Modernization of the data infrastructure

The present dissertation included the construction of a data infrastructure based on the LH paradigm, which allows for the consolidation of structured and unstructured data, with layers dedicated to ingestion, cleaning, modeling, and analysis. This architecture was designed from the ground up to support the integration of AI models and will serve as a reference for future implementations at Navia customers who wish to adopt AI-based solutions.

The model layer and its technical organization, also developed as part of this dissertation, will enable work sharing among developers, centralized access to metrics, versions, artifacts, and model documentation, promoting a collaborative, auditable, and scalable environment. This contribution is particularly relevant considering that Navia's AI team is being created from scratch, with this dissertation representing the first structural step. As part of this foundational effort, a set of guiding principles and technical standards was established to define how AI development will be structured within the organization, including model versioning practices, documentation of training data and evaluation metrics, and procedures for deployment and governance. These foundations provide not only technical consistency but also long-term alignment with best practices in responsible and transparent AI, serving as a strategic basis for the gradual formation and consolidation of a dedicated AI team.

1.5.2 Technological products developed

One of the main results of this dissertation was the design and implementation of the intelligent virtual assistant based on multiple LLMs, capable of interpreting natural language and answering questions related to the operation of Navia and the operational data of the managing entity. This virtual assistant will be marketed as an independent product of A20 as an add-on of Navia, representing a new channel of interaction between users and the system, which is more accessible and intuitive.

In addition, an optical character recognition (OCR) model was developed that can automatically extract water meter readings from photos taken by operators in the field. This model aims to replace manual data entry, reducing errors and increasing the efficiency and reliability of the process. The model will be integrated in the Navia software and will be marketed as a new functionality.

A model for predicting the next intervention was also designed, which uses historical task sequences to automatically suggest the next recommended action after the completion of a given task. This new model will be integrated into the operators' workflow in the form of contextual notifications, contributing to preventive operationalization and avoiding oversights or planning failures. This will also be marketed as a new Navia functionality.

1.5.3 Scientific contribution and impact on Portugal 2030 projects

The work developed in this dissertation also serves as the foundation for Navia's participation in two large-scale projects funded under the Portugal 2030 programme: STREAM - Smart Temporal and Relational Environmental Analysis and Management (COMPETE2030-FEDER-01485000) and AQUASHIELD (COMPETE2030-FEDER-02236600). While the specific objectives of each project differ, both share a common starting point: the need to explore and analyze data that, until now, remained largely underutilized within Navia's operational context.

In this regard, the analyses conducted during the dissertation, together with the restructuring of the data infrastructure, provided a robust foundation for subsequent developments. The unified architecture, the implementation of semantic layers, and the integration of AI models offer a starting point upon which both projects will build their data pipelines, algorithms, and experimental frameworks. All future work to be developed within these projects will rely on the technical groundwork and design principles established throughout this dissertation, including data organization, governance mechanisms, and model integration strategies.

Additionally, these projects are being developed in collaboration with the University of Minho and foresee the supervision of master's and doctoral theses aligned with the themes explored in this work. The continuity of this research through academic collaboration ensures the consolidation of applied innovation in real operational contexts and supports the training of future specialists at the intersection of AI and water system management.

1.5.4 Academic and sector recognition

Finally, it is important to note that the results of this dissertation, as well as the technological advances that Navia is promoting in the sector, will be presented in an article submitted to the national conference ENEG – National Meeting of Management Entities [2], to be held in November 2025. This event is one of the most important forums for technological discussion in the water sector in Portugal, which demonstrates the importance and recognition that this work has already begun to generate in the community.

1.6 Document Structure

Chapter 1 contextualizes the problem under analysis, addressing its relevance in today's market. An in-depth analysis of the problem is carried out, followed by a definition of the research questions and objectives that will guide the development of the dissertation. A list of the dissertation contributions is also presented. Finally, the organization of the document, this section, clarifies and helps to visualize how this dissertation is constructed and articulated to answer the problem in question, as well as to clarify and convey a clear message to the reader.

Chapter 2 is dedicated to the Literature Review. This section aims to provide an understanding of the current state of the art of the research questions mentioned in Chapter 1, as well as to identify future research directions in various areas, such as LH, virtual assistants, the integration of AI with modern data infrastructures, among other topics. This chapter aims to guide the

[2] translated by the author from the Portuguese form Encontro Nacional de Entidades Gestoras

reader through the concepts relevant to the dissertation, as well as presenting some examples of case studies carried out by other authors.

Chapter 3 focuses on the Methods and Tools adopted, as well as the security and ethical considerations. It begins by describing the main libraries, frameworks and software environments employed to implement the LH infrastructure, train machine learning models and support the integration of large language models. It also covers the datasets used in each case study including their source, structure and role in model validation. The chapter concludes with a reflection on the technological and ethical considerations involved, particularly in what concerns data privacy, model transparency and responsible AI integration.

Chapter 4 describes the proposed solution, Conversational Lakehouse Architecture supported by Real-time AI (CLARA), a modular and intelligent system designed to enable real-time interaction with both structured and unstructured data through the integration of LLMs, predictive algorithms and a modern LH architecture. This chapter explains in detail the motivation, architectural design, components, and internal logic of CLARA.

Chapter 5 presents all the case studies conducted to validate the proposed solution, each designed to test a key capability of CLARA. These include the creation of the LH infrastructure, the development of a permission-aware virtual assistant, the implementation of AI models, and the integration of all components into a functional end-to-end system. For each case, the methodology, experimental results and impact are discussed in detail, showcasing the feasibility and robustness of the solution in real-world operational contexts.

Chapter 6 presents the main conclusions drawn from this dissertation, reflecting on how the initial objectives were achieved and how the proposed architecture successfully addressed the challenges identified in the early stages. It also provides a critical analysis of the limitations encountered throughout the work and highlights multiple directions for future development.

Chapter 2

Literature Review

This chapter covers all the literature analysis carried out as part of the dissertation. Essentially, four topics were explored, each with its own relevance to the intended final solution. During the research conducted, it was possible to analyze case studies carried out by other authors, as well as identify gaps in the literature and future research directions in each of the areas covered.

2.1 Data Management Infrastructures and Their Evolution

The increasing complexity and volume of data generated in contemporary digital environments have significantly transformed the way organizations operate and make decisions. In recent years, data has evolved from a secondary support asset to a central strategic resource, capable of driving innovation, operational efficiency, and competitive advantage. This evolution has given rise to new paradigms of organizational culture and technological infrastructureS, where data becomes the foundation of decision-making processes. This section explores the emergence of data-driven organizations and the corresponding evolution of data management infrastructures. Starting from traditional systems optimized for transaction processing, it examines the emergence of analytical systems and large-scale data repositories, culminating in the unified architecture of data lakehouse (LH). Each paradigm is analyzed in terms of its technical characteristics, strengths, limitations, and relevance in the context of modern data-intensive applications.

2.1.1 The Rise of Data-Driven Organizations

We are currently living in times in which strategic decisions made within organizations are no longer made solely on the basis of intuition and opinion, but on the basis of data and factual information (Berntsson Svensson and Taghavianfar 2020). Consequently, the application of big data (BD) and big data analytics (BDA) in organizational decision-making has been attracting increasing interest in recent years, given that the benefits associated with these approaches extend to organizations in all sectors and fields of activity. This is when the concept of a data-driven organization (DDO) emerged.

The concept of a DDO can have different definitions depending on the author. For some, it refers simply to an organization with a data-centric culture, where data collection, quality, and analysis are embedded in everyday decision-making in order to generate competitive advantage. The authors in (Frederik Möller et al. 2021) defines a DDO as an organization that incorporates five key elements: digital transformation, data science, data-driven business model, data-driven innovation and data analytics. According to this definition, an organization that claims to be a DDO can be expected to integrate data into new or existing processes through a well-defined, goal-oriented digital transformation strategy, supported by a data-centric organizational culture. This approach also presupposes the involvement of information technology

(IT) professionals to transform data science prototypes into products through new infrastructures. At the same time, it implies converting insights into business models that recognize data as a central strategic resource for creating economic value, as well as the intensive use of BD to enable the transformation of operational, marketing and decision-making practices into processes guided by empirical evidence, supported by descriptive analyses - what has happened, predictive - what could happen and prescriptive - what should be done, giving the organization an increased capacity for anticipation and informed intervention.

The authors in (Frederik Möller et al. 2021) also identify a set of four conceptual requirements that are essential for an organization to be classified as a DDO: tasks, people, technology and structure. With regard to the technological domain, the need for an infrastructure capable of supporting data processes and automated activities such as data generation, analysis, visualization and storage stands out. This perspective is also corroborated by the authors in (Ali 2023), who highlight the central role of IT in the BD era, as it enables the operationalization of business activities and the provision of services to customers. This stems from the fact that it is the area responsible for providing the resources and capacities that are essential for storing, processing and transmitting information. In addition, the authors recognize that although data-driven marketing has been gaining prominence in recent years, there is still a need for research to develop new techniques and approaches that promote more effective management and use of large volumes of data.

As DDOs emerge and consolidate, data management has undergone a significant transformation, driven by the growing need to extract value from data in real time and at scale (Simitsis, Skiadopoulos, and Vassiliadis 2023). This evolution has been marked by the introduction of different paradigms designed to store, organize and analyze data efficiently, responding to the growing need to support complex analysis and artificial intelligence (AI) applications. Among the main paradigms that have emerged in this context are data warehouses (DWs), data lakes (DLs) and LHs.

2.1.2 Traditional Architectures: From OLTP to Data Warehouses

With the technological advances made in the field of databases during the 1980s and 1990s, online transaction processing systems, commonly known as OLTPs, emerged (Simitsis, Skiadopoulos, and Vassiliadis 2023). These systems provided an efficient way of storing, consulting and updating operational and transactional data in business contexts. In most cases, OLTPs were based on relational database management systems (RDBMS), which were designed to support applications aimed at collecting and recording data, as well as maintaining the most up-to-date state of corporate information. These systems have been extensively optimized to allow multiple simultaneous accesses, ensuring read and write operations with robust guarantees of transactional integrity, in accordance with the ACID properties: atomicity, consistency, isolation and durability.

With the consolidation of OLTPs, the need to exploit stored data in order to extract knowledge and support decision-making became evident (Simitsis, Skiadopoulos, and Vassiliadis 2023). This new requirement led, from the mid-1990s onwards, to the emergence of a new generation of systems geared towards analytical processes: online analytical processing (OLAP) systems. OLAPs have established themselves as central elements of business intelligence solutions, based on specialized infrastructures such as DW or enterprise data warehouses (EDW), designed to consolidate, store and manipulate large volumes of data in order to carry out complex analyses and generate strategic insights.

DW have played a central role in the field of business intelligence (BI) for several years (Dulam, Allam, and Gade 2021). Introduced in the 1990s, they emerged as centralized repositories of structured data, gathered from multiple sources through extraction, transformation and

loading (ETL) processes, as shown in Figure 2.1a (Harby and Zulkernine 2025). These systems are designed to optimize high-performance queries, quickly becoming the preferred option for reporting and data analysis of many organizations (Gade 2022).

DWs store data in a highly organized way, using the schema-on-write format, which ensures that the data is clean, consistent and easy to analyse. However, they have significant limitations: they only support structured data, i.e. data that can be organized into tables with rows and columns. This makes it difficult to store unstructured data, such as documents, images and videos, or semi-structured data, such as JSON files, XML or logs. These forms of data are increasingly prevalent due to the constant generation of information by internet of things (IoT) sensors, social networks and other modern sources (Dulam and Allam 2024)

2.1.3 The Emergence and Challenges of Data Lakes

To overcome the limitations of DW, DL emerged in 2010 (Harby and Zulkernine 2025). Essentially, these infrastructures constitute a flexible and scalable data storage and management system, capable of ingesting and storing raw data from heterogeneous sources while preserving its original format. In addition, DLs offer maintenance, query processing and real-time data analysis functionalities, supported by rich metadata (Hai et al. 2023). These infrastructures have been widely adopted in a variety of real-world scenarios, such as medicine, smart cities and industry. What unites these distinct areas is the predominant nature of their data, which is mostly unstructured or semi-structured.

With the growth in the application of AI in a wide variety of sectors, it has become clear that machine learning (ML) models often require the processing and analysis of large volumes of data in order to achieve effective results. This reality has once again brought to light the limitations of DWs, which face significant challenges in this context, including lack of scalability, inadequate processing models, capacity constraints and high operating costs (Mazumdar, Hughes, and Onofre 2023). Consequently, data scientists have started looking for ways to access raw data directly, reinforcing the relevance of DL. In these systems, data is stored in open file formats, allowing experts to carry out exploratory analysis, feature engineering and model training more efficiently. To do this, they use processing engines that are better suited to supporting the complex analyses required by ML, overcoming the limitations imposed by traditional data storage and processing models.

However, the same characteristics that make DL the preferred storage method for data scientists can, over time, compromise its usability. The absence of robust data governance and quality assurance mechanisms means that these infrastructures gradually turn into what is often referred to as data swamps (Oreščanin and Hlupić 2021). This term refers to a state in which, among the useful data, there is a significant amount of noise and unnecessary information, which hinders future analysis and drastically reduces the usefulness of the data for developing quality models. In addition, DLs can lead to increased storage and processing costs, and the data it contains has to be cleaned, tagged and structured before it can be used, which can be a very time-consuming process (Harby and Zulkernine 2025).

2.1.4 The Lakehouse Paradigm: A Unified Approach

It is clear that neither DW nor DL were able to fully satisfy the growing needs of organizations. Some opted to maintain both infrastructures in parallel, as shown in Figure 2.1b. However, this approach incurs additional costs such as financial, storage related, governance related, and pipeline management which highlights the ongoing need for innovation and improvement. It was then that the LH emerged, first introduced in 2021 by the authors in (Armbrust Michael et al. 2021), these infrastructures were defined as being able to combine the flexibility and

scalability of DL, while maintaining the consistency, performance and governance characteristics of DWs (Dulam, Allam, and Gade 2021). This innovative approach allows organizations to store structured, unstructured and semi-structured data in a single infrastructure, while providing the ability to handle transactional and analytical workloads simultaneously. In addition, LHs offer ACID guarantees, ensuring reliable, high-performance analytical queries without compromising data integrity or reliability. This hybrid approach not only reduces the complexity of managing and maintaining two separate infrastructures, but also creates a simpler and more flexible data ecosystem (Gade 2022), as shown in Figure 2.1c.

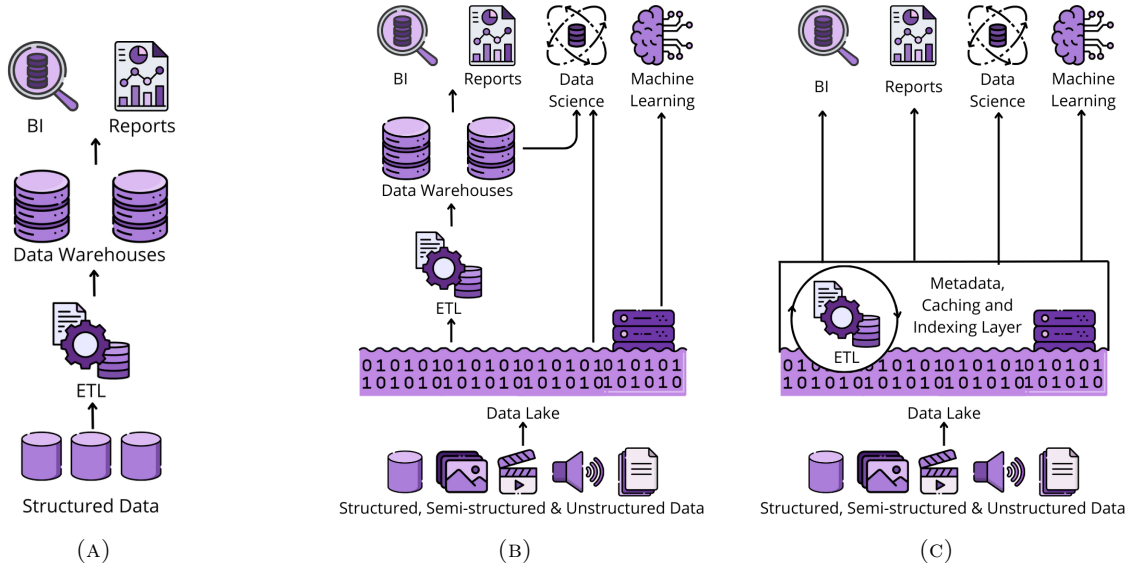


FIGURE 2.1: Evolution of data platform architectures to today's two-tier model (A) First-generation plataforms (B) Current two-tier architecture (C) Lakehouse plataforms
(Armbrust Michael et al. 2021)

With the obvious advantages associated with the LH architecture, applications quickly began to emerge in a wide variety of areas. A relevant study to mention is the one conducted by the authors in (Schneider et al. 2024), which analyzed different data storage and management architectures, such as DW, DL and LHs, as well as the platforms on which they can be implemented, namely Delta Lake, Apache Hudi and Apache Iceberg. The authors concluded that LHs have the ability to simplify corporate analytical architectures, reduce costs and improve the quality of the results of the analyses carried out. In addition, they argue that this recent architecture presents itself as a viable alternative to conventional DW and DL.

In the area of healthcare, the authors in (Li Ruichen, Choy Murphy, and Ma Nang Laik 2022) carried out a study in which they implemented an AI-based medical diagnosis system, using a LH architecture in a cloud environment, aimed at small and medium-sized healthcare units. The authors chose this architecture over a DL or a traditional DW because of its ability to combine the scalable, low-cost storage of a DL with the high-performance query and analysis capabilities of a DW. This choice proved particularly pertinent in this context, given that the IT resources available were limited.

Another relevant example of a case study is the master's thesis presented in (Philip Salqvist 2024), which carries out a comparative analysis between DW and LH architectures. The main objective of the dissertation was to evaluate the performance of both architectures in reading datasets of different sizes, allowing the readers to understand for which scenarios each would be more suitable. The author concluded that DW is more efficient for decision support systems (DSS) workloads that require low read latencies, while LH is more appropriate for scenarios that

require high scalability. However, the study was limited to a 64 GB data set and the latency metric, which led the author to recommend future research that includes larger datasets and the use of additional metrics, such as throughput, for a more comprehensive analysis.

With regard to the management and maintenance of LHs, the authors of the article in (Paras Jain et al. 2023) carried out a study to analyze different aspects of these infrastructures such as: transaction coordination, metadata storage and consultation, as well as the efficient handling of updates. The study showed that the design of LH systems requires significant technical compromises, particularly with regard to transaction coordination, data ingestion strategies and metadata storage, which have a direct impact on performance. The authors also highlighted the importance of strategically navigating these trade-offs to ensure the efficient execution of diverse workloads in real environments. Finally, it was concluded that, despite the immense potential demonstrated by the LH architecture, this is still a recent approach, with a significant need for future research to answer open technical questions. Suggested directions for future research include: balancing ingest latency and query latency by merging data asynchronously; developing automatic adaptive summarization strategies suitable for different workloads; using cost models to plan queries efficiently, switching between indexed strategies for small queries or distributed strategies for larger queries; and improving support for high concurrent write rates by mitigating the impact of high latency associated with updating metadata in the underlying storage.

In line with this, the authors in (Nathalie Janssen et al. 2024) corroborated these conclusions, highlighting the need for further research into the technical aspects of implementing LHs, including optimization strategies and best practices for their performance in real-world scenarios.

After the analysis of these studies and the literature reviewed, a summary table of all the key differences between DWs, DLs and LH was made and it is present below on table 2.1.

Table 2.1: Simplified comparison table between Data Warehouse, Data Lake, and Data Lakehouse

Feature	**Data Warehouse (DW)**	**Data Lake (DL)**	**Data Lakehouse (LH)**
Schema-on-read	×	✓	✓
Schema-on-write	✓	×	✓
Support for structured data	✓	×	✓
Support for unstructured data	×	✓	✓
ACID transactions	×	×	✓
Batch data ingestion	✓	✓	✓
Streaming data ingestion	×	×	✓
Metadata management	✓	×	✓
SQL queries	✓	×	✓
Real-time processing support	×	×	✓
Cost-effectiveness	✓	✓	✓
Advanced security	✓	×	✓
Optimized for advanced analytics	✓	×	✓

The analysis of these studies highlights that LHs have the potential to simplify analytical architectures, reduce costs and improve the quality of analysis, presenting themselves as a viable alternative to DW and DL. In addition, LHs combine the scalability and low cost of DLs with the analytical performance of DWs, making them particularly useful in contexts with limited

resources. However, the architecture is still new and faces significant technical challenges, such as transaction coordination, metadata management and support for high write rates. Future research should explore adaptive summarization strategies, more efficient query planning and solutions that balance ingest and query latency.

2.2 Semantic and Structural Challenges in Heterogeneous Data Integration

The integration of heterogeneous data is a central challenge in modern information systems, particularly when aiming to ensure semantic consistency across distributed and diverse sources. This section explores the multiple dimensions of heterogeneity, examines ontology-based strategies for achieving semantic interoperability, and highlights how recent advances in natural language processing (NLP) and large language models (LLMs) are enhancing alignment and integration processes. Additionally, it analyses the evolution of data pipelines, from traditional ETL to real-time extraction, loading and transformation (ELT) and streaming approaches, emphasizing their role as foundational mechanisms for scalable and resilient data integration.

2.2.1 Dimensions of Heterogeneity and Semantic Interoperability

Addressing the heterogeneity present in data sources is fundamental to ensure the consistency and reliability of the integration process (Chad 2025). From this perspective, it is possible to distinguish different forms of heterogeneity that compromise interoperability between systems. Structural heterogeneity arises when data is organized according to different schemas, models or formats, making it difficult to combine them directly. In addition, syntactic heterogeneity refers to the use of different coding formats or languages to represent the same information, which can lead to incompatibilities when reading or interpreting the data. Finally, semantic heterogeneity, perhaps the most challenging, concerns discrepancies in the meaning, interpretation and terminology attributed to the data, resulting from variations in the contexts of origin.

Achieving semantic interoperability in the context of integrating heterogeneous data is a highly complex challenge (Chad 2025). One of the main obstacles is the incompatibility between schemas and ontologies adopted by different data sources. These may use different organizational structures, vocabularies and classification standards, making semantic alignment substantially more difficult. Differences in terminology, concept definitions and units of measurement introduce ambiguities that compromise the coherent interpretation of information. Added to this complexity is the fact that the meaning of data can be influenced by the context in which it is produced or applied, and it is common for the same term to take on different interpretations depending on the domain or industry. The lack of widely agreed semantic frameworks exacerbates these difficulties by limiting the possibility of establishing a shared understanding. At the same time, data quality and consistency issues such as missing values, duplicates or internal inconsistencie represent additional barriers to achieving true semantic interoperability.

2.2.2 Ontology-Based Integration and Its Limitations

In response to the challenges posed by heterogeneous data, ontologies have emerged as a particularly promising solution in the field of data science and computing (Smaili and Kabbaj 2025). In this context, an ontology is understood as a formal, machine-readable representation of knowledge relating to a given domain, functioning as a structured, shared vocabulary that allows computer systems to interpret data semantically and communicate with each other effectively. By providing an organized structure that describes concepts, the relationships between these concepts and the applicable constraints, ontologies establish a common basis of understanding that is essential for achieving semantic interoperability between diverse systems. This

structure is made up of several fundamental elements (Morse, Information, and Gouda 2025): the classes or concepts that represent categories or groups of entities with common properties; the object properties, which specify the relationships between these classes; the data properties, which describe attributes of the objects with literal values such as text or numbers; the individuals, which correspond to concrete instances of classes; as well as a hierarchical structure that organizes the concepts into a coherent taxonomy; and, also, rules and restrictions that define the logic and conditions governing the relationships and properties.

As far as approaches to data integration using ontologies are concerned, several strategies can be identified (Morse, Information, and Gouda 2025). The single-ontology approach assumes the existence of a single vocabulary shared by all sources, promoting a unified view of the domain. On the other hand, multiple-ontology is based on local ontologies specific to each data source, and interoperability is ensured through mappings between these ontologies. The hybrid-ontology approach combines a common vocabulary with local extensions adapted to specific needs, while the global-as-view ontology establishes a global ontology that serves as a reference, with mappings coming from the local sources. In addition, it is possible to distinguish between higher-level ontologies, which have a high degree of abstraction and cross-domain applicability, and application ontologies, whose definition and structure are geared towards specific contexts and concrete purposes (Hees 2024). These different categories and application models demonstrate the versatility of ontologies as a fundamental tool for semantic harmonization and effective data integration in complex and distributed environments.

Despite their recognized potential, traditional ontology-based approaches have significant limitations, especially in the context of integrating complex and distributed data (Smaili and Kabbaj 2025). In its classic form, ontology construction still relies heavily on manual processes conducted by domain experts, which compromises the efficiency and scalability of this approach. In fact, designing coherent and applicable ontologies from multiple databases has been identified as one of the main bottlenecks of semantic integration (Chuangtao Ma and Bálint Molnár 2022). This limitation becomes particularly evident when different systems independently develop their own ontologies, aggravating interoperability difficulties and accentuating already existing semantic flaws. The resulting fragmentation compromises the integration of information between different organizations, mainly due to the lack of a common vocabulary to harmonize meanings and build bridges between different contexts. The tools currently available to support this process are largely limited to simple mappings, usually one-to-one, and have difficulty dealing with more extensive ontologies or more demanding integration scenarios. Furthermore, the very nature of semantic heterogeneity, which is often very specific to each domain, makes it difficult to develop integration solutions that are both reusable and scalable, without resorting to considerable manual effort to map, reconcile and align the different conceptual elements (Kumar 2025). To mitigate some of the limitations identified above and increase the accuracy of the semantic matching and integration processes, NLP techniques have begun to be incorporated into the methodologies associated with the use of ontologies (Morse, Information, and Gouda 2025).

2.2.3 Enhancing Ontology Alignment with NLP and LLMs

NLP allows computer systems to analyze and understand the structure and meaning of human language, thus offering a richer basis for identifying semantic relationships between terms, even when their syntactic formulation differs substantially (Liu, Hogan, and Crowley 2011). In the context of ontology matching, these techniques are particularly useful for detecting equivalences or conceptual divergences that would be difficult to identify with purely symbolic methods. These strategies consist of the use of edit distance measures or distributed word representations, such as word embeddings, have shown clear improvements in the quality and accuracy of semantic data integration. More recently, the use of AI-based approaches, in which PLN

plays a central role, has made it possible to automate processes such as semantic reconciliation or the extraction of relevant concepts from unstructured data more effectively (Chad 2025). In this context, LLMs play a promising role by providing a deeper and more contextualized understanding of data and by enabling more robust semantic reasoning about the relationships between entities.

Research in applied domains has highlighted the potential of integrated approaches based on ontologies to promote semantic interoperability in heterogeneous systems. In the agricultural sector, for example, the authors in (Smaili and Kabbaj 2025) developed a methodology centered on merging existing ontologies with the aim of overcoming fragmentation and creating a unified ontology. The result of this process was gistAgro, a merged ontology made up of 67,373 classes and 124,192 logical axioms, capable of supporting complex queries in the context of agricultural management. Similarly, in the internet of things (IoT) domain, the authors in (Ranpara 2025) proposed a semantic ontology-based framework to improve interoperability and automation in technologically heterogeneous environments. The results of the simulation study showed an interoperability success rate of around 98% between different devices, demonstrating the effectiveness of the approach.

In the energy sector, the work carried out by (Santos et al. 2021) used ontologies and web semantic technologies as a central axis to mitigate the problems of semantic interoperability between different tools and agents in the context of power and energy systems (PES). The creation and sharing of a common vocabulary allowed the agents (of the multi-agent system developed) to correctly interpret the messages exchanged, ensuring effective communication between systems with different data models. The approach also included the reuse and extension of already existing ontologies, promoting coherent integration between agents specialized in different PES subdomains. As well as facilitating interoperability, this strategy has contributed to greater flexibility and intelligence in the systems, by decoupling them from rigid models and rules.

In the context of multidomain master data management (MDM), semantic reconciliation is widely recognized as essential to ensure data consistency and accuracy. In this context, (Kumar 2025) analyzed several existing tools for aligning domain ontologies, highlighting LogMap for its ability to combine high processing speed with accurate semantic matching.

More recently, exploratory research has emerged that evaluates the use of large-scale language models, such as GPTs, in the task of aligning non-ontological data with existing ontologies. An example of this line of work can be found in the thesis presented by (Hees 2024), which concludes that although these approaches are not yet mature enough to fully replace autonomous mapping systems, they show great potential as support tools, offering useful suggestions that can significantly assist experts in the semantic mapping process.

2.2.4 Data Pipelines and Modern Approaches to Real-Time Integration

When it comes to the data integration itself, in order to mitigate problems associated with semantic heterogeneity, data pipelines have become essential components, as they allow data to be transported, transformed and delivered between various systems (Aryan Gupta and Meera Patel 2021). However, as the complexity and scale of data operations increase, pipelines face new challenges, requiring increased attention to their resilience.

A data pipeline is generally structured into four main stages: data ingestion, data processing, data storage and data analysis (Bilal Khan et al. 2024). Regarding data ingestion, one of the most recognized processes is ETL, which consists of extracting data from various sources, transforming it into a consistent format and loading it into a single data infrastructure, as per shown in Figure 2.2a. In addition, these continuous ingestion methods allow the stored data to

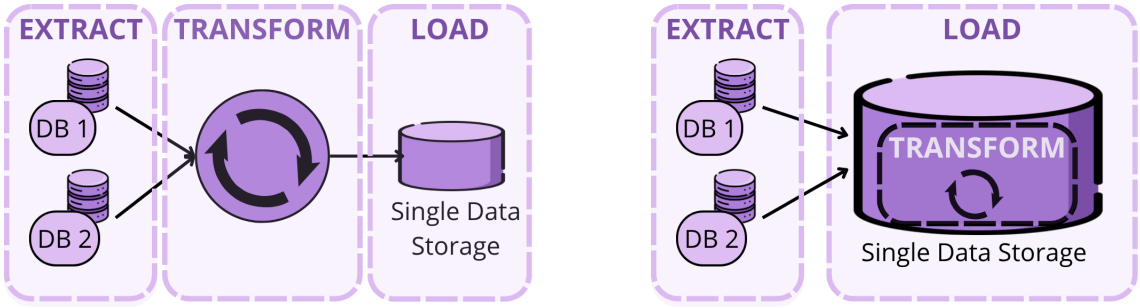
be constantly updated. However, for real-time data integration, more sophisticated mechanisms are still needed.

One example of such mechanisms would be the study carried out by the authors in (de Assis Vilela Flávio et al. 2023). This article introduces an innovative architecture called Data Magnet, designed for real-time ETL processing for DW. Unlike traditional, often intrusive approaches that rely on the use of wrappers, triggers and log analysis directly on the data sources, Data Magnet uses a publish-subscribe system combined with a tag system to extract data in a non-intrusive and reactive way, responding automatically to insertion events on the data sources. The authors identified a gap in research into non-intrusive alternatives for real-time data ingestion, and to fill it, Data Magnet was designed as a real-time ETL system, challenging the traditional ETL paradigm that operates on the basis of periodic synchronizations.

Although the ETL paradigm has historically been the most widely used for moving data between operating systems and analytical environments, another alternative approach that has been gaining particular prominence, when it comes to modern data infrastructures, is ELT (Mazumdar, Hughes, and Onofre 2023). In this way, data is initially extracted and loaded directly into the target infrastructure, and only transformed later, as needed, as per show in Figure 2.2b. This reversal in the order of operations offers greater flexibility, as it allows different types of data like structured, semi-structured or even unstructured to be stored in their original form. In addition, by postponing the transformation, it is possible to take advantage of the massively parallel processing (MPP) capabilities provided by modern data systems, which allows complex transformations to be applied only to relevant subsets, optimizing resources and speeding up response times for specific use cases.

Another factor contributing to the growing adoption of the ELT paradigm is the progressive migration of data infrastructures to cloud environments (Singhal and A. Aggarwal 2022). This change allows organizations to benefit from highly scalable and configurable computing resources, such as adjustable rapid access memory (RAM), multicore central processing units (CPUs), fast solid state drivers (SSD) storage, and even state-of-the-art graphic processing units (GPUs), all without requiring a significant initial investment in hardware. In addition, cloud computing offers fault tolerance mechanisms, ensuring system continuity even in the event of physical component failures.

These features are particularly relevant in the data transformation phase, which is, by nature, computationally intensive (Sivabalan and Minu 2021). Thus, by choosing to first load the raw data into the cloud infrastructure and only then apply the necessary transformations, organizations can take full advantage of the available processing capacity, optimizing both performance and process efficiency. For this reason, in cloud-based scenarios, the ELT model is generally more advantageous than the traditional ETL.



(A) (B)

Figure 2.2: Comparison between ETL and ELT (A) ETL (B) ELT

However, neither ETL or ELT are famously know for being great in real-time ingestion pipelines scenarios, for these situations a promising solution is the use of data streams (Martín et al. 2022). Data streams allow organizations to create pipelines that efficiently capture and process data as it arrives, facilitating analysis and decision-making in real time (Aryan Gupta and Meera Patel 2021). One of the most widely used tools for managing data streams is Apache Kafka, which, in addition to facilitating the management and processing of data streams, offers a distributed, scalable and fault-tolerant messaging system. Authors such as (Marcu Ovidiu-Cristian and Bouvry Pascal 2024) point out that the use of data streams will be essential in the development of AI models. To fill the gap in managing data pipelines for AI supported by data streams, the open-source Kafka-ML platform was developed, using Apache Kafka. This platform allows ML models to be defined, trained, evaluated and implemented using a user-friendly web interface.

As for the integration of heterogeneous data, the study by (Putrama and Martinek 2024) presents an analysis of the literature in this area, covering the period from 2012 to 2023. The authors concluded that although most of the papers analyzed focus on BD, few studies address the structure of the data involved in detail, namely the distinction between structured, semi-structured and unstructured data. Only a small number of articles explicitly identify the nature of the data as structured, which reveals a significant gap in research into the integration of data of different natures. In addition, the authors noted that while innovative techniques such as ML and data privacy management techniques have occasionally been explored, they are not yet widely adopted. This was identified as a promising future direction for research. Another relevant factor highlighted was the scarcity of case-based studies illustrating critical factors for implementing complete heterogeneous data integration solutions in industrial contexts.

Another relevant study is presented by (Wrembel 2024), motivated by the growing challenges associated with the complexity and volume of data in modern systems, which make data integration processes increasingly demanding and difficult to optimize. In this study, the authors explored innovative approaches based on ML techniques, with the aim of improving the efficiency and performance of integration processes. Broadly speaking, the research focused on optimizing tasks related to modern data sources and improving the handling of user-defined functions, which are often highly complex. The results highlight the potential of using ML to address the challenges of data integration, contributing to more efficient and adaptable processes. However, the authors identified relevant gaps, such as the need to further research modern data sources, including NoSQL systems, and to develop more robust solutions to handle complex tasks. These future directions reinforce the importance of exploring new approaches and the transformative role of advanced techniques in overcoming the challenges inherent in the big data era.

In summary, heterogeneous data integration remains a multifaceted challenge, particularly in contexts involving real-time ingestion, semantic heterogeneity, and large-scale distributed systems. Ontologies have proven to be a robust mechanism for harmonizing meanings and facilitating interoperability, especially when combined with NLP and AI techniques. Data pipelines, whether based on traditional ETL, non-intrusive architectures like Data Magnet, or real-time streaming platforms such as Kafka, serve as the operational backbone of these integration efforts. Nevertheless, future work must address structural data classification, broader adoption of ML techniques, and the development of robust case-based methodologies to translate theoretical advances into practical, scalable solutions.

2.3 Artificial Intelligence and Data Infrastructures

The integration of AI and data infrastructures has profoundly transformed the way organizations extract value from information. On the one hand, the development of new ML paradigms has made it possible to solve increasingly complex problems through systems that learn from data

autonomously. On the other hand, the evolution of data architectures, which are increasingly flexible, scalable, and analysis-oriented, has enabled the effective application of AI models on a large scale and in near real time. This section explores the main ML paradigms, the growing importance of the data-centric paradigm, and the critical role of modern infrastructures such as LHs in the operationalization of AI-based solutions.

2.3.1 ML Paradigms

The field of ML encompasses different paradigms, each with distinct characteristics, applications, and challenges. These paradigms, presented in Figure 2.3, define the strategies used by intelligent systems to extract patterns, make predictions, or make decisions based on data. Understanding these approaches is essential to contextualize the solutions developed in AI and understand their advantages and limitations in different scenarios. The following subsections analyze the three main types of learning: supervised, unsupervised, and reinforcement learning.

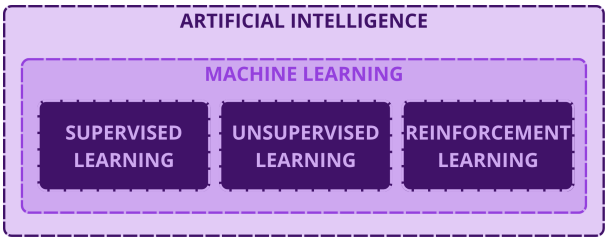


FIGURE 2.3: Main paradigms of ML (Hiran et al. 2021)

Supervised Learning

The evolution of AI has been marked by significant advances in both algorithms and the infrastructure that supports them. At the heart of this transformation is ML, whose operating paradigms determine how systems extract knowledge from data. With the growing complexity of real-world problems and the diversity of application contexts, it has become essential to understand the different learning approaches that underpin AI systems.

Supervised learning is the most common type of ML, where a model is trained using labeled data to optimize performance on a single task (Tiwari 2022). This method allows systems to learn from concrete examples and is widely used in classification problems, which assign predefined categories, and in regression problems, which predict continuous results. Its effectiveness depends heavily on the quality and quantity of labeled data available, which is not always easy to guarantee in real-world contexts.

In contrast, semi-supervised learning (SSL) is a paradigm that builds models using both labeled and unlabeled data (Yang et al. 2023). This approach is particularly desirable because obtaining large volumes of labeled data can be difficult, expensive, or time-consuming, while unlabeled data is typically abundant and accessible. By combining both, the model can improve performance even when the amount of annotated data is limited, taking advantage of the underlying structure of the unlabeled data.

Within SSL, there is also a subfield dedicated to deep semi-supervised learning (DSSL) methods, which exploit the potential of deep neural networks in contexts where labeled data is scarce (Yang et al. 2023). DSSL methods can be grouped into several categories: deep generative approaches, which attempt to model the distribution of data; consistency regularization methods, which encourage the model to maintain similar predictions in the face of small changes in inputs; graph-based methods, which exploit relationships between labeled and unlabeled samples;

pseudo-labeling techniques, which assign artificial labels based on the model's own predictions; and hybrid methods that combine several of the above strategies.

Within this field, self-supervised learning (SSL) is a recent pre-training technique that uses artificial tasks—known as pretext tasks—to extract useful features from unlabeled data (Zhao et al. 2024). Instead of relying on external labels, supervision is automatically extracted from the structure of the data itself, for example, by predicting missing parts in an image or omitted words in a text. These tasks allow the model to learn rich and transferable representations, which can then be fine-tuned for specific tasks with a small number of labeled examples. In this way, SSL helps mitigate dependence on manually annotated data and, in many cases, achieves performance comparable to fully supervised models.

The applications of supervised learning have been explored in detail by the authors in (Tiwari 2022), who identify a wide range of practical cases in different domains. In the field of computer vision, image recognition, handwriting analysis, and disease detection in medical images, such as early identification of blindness or skin cancer, stand out. In signal processing, applications such as voice and audio analysis and speech recognition are mentioned. In natural language processing, examples include sentiment analysis, hate speech detection, and the classification of multimodal memes, which combine text and images. In cross-cutting contexts, supervised learning is also used in the detection of fraudulent credit card transactions, in personalized recommendation systems such as those used by Netflix and Amazon, in virtual personal assistants such as Alexa and Siri, and in autonomous vehicles, where models are trained with large volumes of labeled data to interpret the environment and make decisions in real time.

Unsupervised Learning

The other main branch of ML is unsupervised learning, which is based on the use of unlabeled data sets, i.e., without a clearly defined response variable (Valkenborg et al. 2023a). In this paradigm, the system is trained to identify patterns, relationships, or underlying structures in the data without resorting to any type of external supervision. Instead of learning from annotated examples, as in supervised learning, the model attempts to autonomously explore the internal organization of the data.

Unsupervised learning algorithms can generally be grouped into two main categories: clustering methods and dimensionality reduction or transformation techniques (Valkenborg et al. 2023b). The former aim to bring similar samples together into groups (clusters), while the latter seek to represent the data in a more compact or informative way, facilitating visualization or subsequent processing. Rather than testing specific hypotheses or building predictive models, this approach focuses on exploring the intrinsic structure of the data and generating new hypotheses, often used as a basis for further analysis or supervised learning.

Among the advantages of unsupervised learning is the ability to discover patterns or regularities in unlabeled data, which is particularly valuable in contexts where manual annotation is difficult, time-consuming, or expensive (Valkenborg et al. 2023a). These methods also overcome limitations associated with single-view approaches by promoting richer and more integrated representations of data (Moujahid and Dornaika 2025). When correctly applied, they can contribute to increasing the accuracy of supervised models, improving understanding of the latent structure of data, and revealing unexpected relationships.

However, this approach also presents significant challenges. One of the main challenges is related to the interpretation of results: without a response variable, it becomes more difficult to assess the quality of the clusters or transformations obtained (Kauffmann et al. 2025). In addition, models are subject to phenomena such as the so-called "Clever Hans" effect, in which the system learns to make decisions based on irrelevant or misleading clues present in the data,

which can compromise robustness and generalization under new conditions. The results can also be influenced by factors such as the presence of noise, incomplete data (Moujahid and Dornaika 2025), or the scale of the variables, which can distort the behavior of the algorithms (Valkenborg et al. 2023a).

Despite these challenges, unsupervised learning has a wide range of practical applications, as identified by the authors in (Moujahid and Dornaika 2025) who explored the application of this form of learning in different contexts. In healthcare, it is used for medical diagnosis, combining images with molecular data (omics data), for example. In the field of computer vision, it enables object recognition and automatic scene analysis. In natural language processing, it is applied to the clustering of documents or topics. In social networks, it facilitates the identification of communities with similar interests. And in recommendation systems, it contributes to personalizing content based on user behavior patterns. Even in more specific areas, such as dentistry, it has been used in the analysis of dental malocclusions, revealing its versatility and cross-cutting applicability (Valkenborg et al. 2023a).

Reinforcement Learning

The third and final ML paradigm is reinforcement learning, which is an ML technique that allows an agent to learn to make sequential decisions in complex environments based on acquired experience (Wang et al. 2024). Inspired by the trial-and-error learning observed in humans and animals, this paradigm is based on continuous interaction between the agent and the environment. For each action performed, the agent receives a reward signal that guides it in adapting its behavior. The central goal of reinforcement learning is to discover an optimal policy, that is, a decision strategy that maximizes the total reward over time.

The conceptual origin of reinforcement learning dates back to the so-called "Law of Effect," formulated by Edward Thorndike in 1911, which suggested that behaviors followed by positive consequences tend to be repeated (Shakya, Pillai, and Chakrabarty 2023). This intuition motivated the development of techniques that allow sequential decision-making tasks to be solved, especially in scenarios where building an explicit model of the system is difficult, costly, or impractical.

Early reinforcement learning methods were based on tables, such as Q-tables, where the best action to take for each possible situation (or state) was explicitly stored. However, this type of approach quickly became unfeasible in more complex problems, where the number of possible states or actions is very high or even continuous (Shakya, Pillai, and Chakrabarty 2023). In these cases, it would be necessary to store an impractical number of combinations, which would make the learning and decision-making process too slow or impossible.

In addition, the very functioning of reinforcement learning has characteristics that make direct training difficult. On the one hand, feedback is not immediate; often the agent only realizes whether it has made a good decision after several interactions, which complicates the attribution of merit to a specific action (Le et al. 2022). On the other hand, there is no external supervisor to tell the agent what the correct action would be in each situation. This means that the agent has to learn autonomously, trying out different options, evaluating the results, and gradually adjusting its strategy, a demanding process that has motivated the development of more sophisticated and scalable techniques, such as deep reinforcement learning (DRL).

In the field of reinforcement learning, there are two main approaches: model-free and model-based (Wang et al. 2024). Model-free methods learn solely from the agent's experiences, without any prior knowledge of the dynamics of the environment. Model-based methods, on the other hand, build an internal representation of the environment (a model) that allows the agent to simulate future actions and make more informed decisions.

In recent years, the emergence of DRL has revolutionized the field by integrating reinforcement learning with deep neural networks (Le et al. 2022). This combination allows for the resolution of highly complex problems, with continuous or large state and action spaces, where classical methods are not effective. An emblematic example of this approach is AlphaGo, the system that defeated human champions in the game of Go (Shakya, Pillai, and Chakrabarty 2023).

Another important aspect is multi-agent reinforcement learning (MARL), which studies the learning and behavior of multiple agents that interact with each other in a shared environment (T. Li et al. 2022). This area is particularly relevant in distributed systems, such as autonomous vehicles or cooperative sensor networks. Hierarchical reinforcement learning (HRL) allows complex, long-horizon tasks to be divided into simpler subtasks, organized in a hierarchical structure of policies, facilitating learning and increasing the scalability of systems (Pateria et al. 2021). More recently, the integration of graph neural networks (GNNs) with DRL has attracted great interest, allowing RL to be applied to graph-structured environments, such as logistics networks, communication networks, or combinatorial optimization problems (Munikoti et al. 2024).

Reinforcement learning and its variants have shown great potential in multiple areas. Notable applications include gaming, robotics, finance, medicine, NLP, computer vision, and recommendation systems (Le et al. 2022).

2.3.2 The Data-Centric AI Paradigm

Data-centric AI (DC-AI) is at the core of a paradigmatic transformation in the field of software engineering, where ML has become the new core of systems development, supported by large volumes of data and advanced computing infrastructures (Whang Steven Euijong et al. 2023). In this new context, software engineering has had to be rethought, giving data the status of a first-order element, alongside code. Since a substantial part of the life cycle of a ML system is dedicated to preparing and processing data, without quality data, even the most sophisticated algorithms tend to perform poorly. Thus, the practices associated with the DC-AI approach are progressively being consolidated as an integral part of modern processes for developing solutions based on AI.

Before the emergence of the data-centric approach, the lifecycle of an ML system predominantly followed the model-centric artificial intelligence (MC-AI) paradigm, in which the reference data set, in most cases, remains virtually unchanged (Zha et al. 2023). In this paradigm, the main goal of researchers and practitioners was to iterate on the model in order to improve its performance. Although MC-AI encourages significant advances in models, it places excessive trust in the data and unfortunately, in practice, many real-world data sets are small, skewed, noisy and even corrupted, which compromises the robustness and reliability of the models generated.

The expression “garbage in, garbage out” is often used in the context of ML precisely because data is not just a fuel that feeds models, but is a critical factor in defining their final quality (Zha et al. 2023). In addition, the MC-AI approach also proved to be ineffective in improving model results, since performance gains occurred sporadically and generally only in contexts where the volume of data was extremely high (Majeed and Hwang 2024). Furthermore, this exclusive focus on improving the model, to the detriment of the quality of the data that feeds it, can lead to significant risks for human users, such as biased decisions, low accuracy in prediction/classification tasks, data/concept drift, higher CO2 emissions, technological fragmentation and poor control over the systems developed.

The truth is, historically, most of the scientific efforts in the field of AI have focused on models rather than data; it is estimated that around 90% of current specialized literature focuses on models, while only 10% is dedicated to data (Whang Steven Euijong et al. 2023). It was only in 2021 that Andrew Ng formally introduced the concept of DC-AI during a live streaming

session (Majeed and Hwang 2024). According to the author, the central premise of DC-AI is to pay substantial attention to the quality and suitability of the data, while making the most of the advances already consolidated in AI models and code, namely through the use of pre-trained models. Since the introduction of the new paradigm, much progress has been made that would have been difficult to achieve with MC-AI alone. With the affirmation of DC-AI, the attention of the scientific and professional community in AI has progressively shifted from the mere optimization of models to a growing appreciation of data quality. The fundamental aim of this repositioning is to build more robust, reliable and transformative AI systems, capable of responding effectively to the complex challenges of the real world.

2.3.3 AI and Modern Data Infrastructures

This paradigm shift, driven by the focus on data quality promoted by the data-centric approach, has profound implications for the data infrastructures that support AI systems. The authors in (Patil, Mahajan, and Sakhare 2024) stress the importance of organizations starting to integrate data pipelines that take into account not only the collection and preparation of data, but also crucial aspects such as its quality, privacy, security, governance, as well as the scalability and flexibility of the underlying infrastructures. For the principles of DC-AI to be fully applied, it is therefore essential to have architectures that allow for the efficient ingestion, organization and analysis of large-scale data in heterogeneous formats. In this context, LHs are a particularly promising solution, combining the flexibility of DL with the quality control, structuring and governance mechanisms typical of DW.

The development and use of AI models directly on LHs presents the potential for transforming advanced analytics, optimizing processes and providing valuable insights. However, implementing these models in LHs involves significant technical challenges that need to be addressed to ensure their effectiveness.

One of the main challenges relates to data preparation. The heterogeneous nature of the data stored in a LH, which can include structured, semi-structured and unstructured data, requires rigorous cleansing, transformation and formatting processes so that AI models can use it effectively (Dulam, Allam, and Gade 2021). It is therefore essential to invest time and resources in properly ingesting data into the infrastructure, avoiding cases of semantic heterogeneity, as discussed in section 2.2.

In the water sector, AI applications have shown significant potential. Exemplary case studies include water treatment, as demonstrated by the authors in (Matthew Lowe, Ruwen Qin, and Xinwei Mao 2024), who analyzed the performance of different AI models, including classical and deep learning models, in various areas related to water treatment. In addition, AI has proven useful in predicting consumption, using historical usage data, weather patterns and socio-economic events, thus enabling the optimization of water management (Kannan Nova 2023). Another relevant application is the detection of anomalies, such as leaks, through the analysis of sensor data, as investigated by the authors in (Mahdi et al. 2024), contributing to the prevention of water losses and ensuring its quality.

Modern data infrastructures, such as LHs, offer significant advantages for training and applying AI models in near real time. These infrastructures stand out for their ability to process large volumes of streaming data and the integration of different data sources, which makes it possible to create more accurate and adaptable models (Hermanus et al. 2024). However, implementing AI models directly in LHs still requires cloud platforms with scalable computing resources and pre-trained AI services that make them easier to manage and operate.

A notable example of a case study that explored the advantages provided by LHs for AI applications was conducted by the authors in (Hermanus et al. 2024). In this work, the researchers

developed a LH capable of integrating and hosting data from multiple sources, with different structures, including data collected by various IoT sensors. This system is designed to facilitate more efficient data management and enable its future use in implementing and optimizing the management of a smart city in Indonesia, with applications in areas such as transport management, energy optimization, public safety, waste and water management, environmental monitoring and air quality control. The research also highlighted the importance of creating robust governance and data security mechanisms, as well as ensuring the scalability and adaptability of the solution to meet the growing and dynamic needs of smart cities.

A case study that doesn't focus on any specific data management architecture but aims to highlight how AI can be used to transform the way organizations implement and use AI was the study conducted by the authors in (George 2022). This article was motivated by the growing relevance of integrating AI with multi-cloud and hybrid architectures, the authors investigated the main challenges associated with this paradigm, including interoperability issues, data management and governance, performance optimization, fault tolerance and cost management. The study highlighted the opportunities provided by the integration of AI in multi-cloud environments, namely in terms of scalability, resilience and innovation, and emphasized the need to adopt innovative solutions, such as federated learning and edge computing, to overcome the challenges identified. However, important gaps were pointed out, such as the need to develop more robust security strategies, comprehensive data governance models and advanced cost optimization techniques, as well as exploring the impact of emerging technologies and carrying out case studies of implementations in real contexts. This work underlines the importance of continuing research in this area, with a view to unlocking the full transformative potential of AI in multi-cloud environments.

The authors in (Justin Levandoski et al. 2024), motivated by the growing need to unify the functionalities of DL and DW, have created a solution that combines the flexibility of an open source format with robust security and governance and the potential of AI/ML integration in data processing, this solution called BigLake is a multi-cloud LH platform that offers integration of structured and unstructured data, offering advanced support for inference and processing with AI/ML, as well as performance acceleration and detailed governance. The authors concluded that BigLake effectively addresses modern data management and AI needs, but identified improving data processing efficiency and reducing data transfer times through optimized strategies as future research directions.

In short, the integration of AI models into LHs shows significant potential for transforming advanced analytics in various sectors, enabling the efficient management of large volumes of data and the generation of insights in near real time. Despite the advantages offered by these infrastructures, technical challenges such as data preparation, scalability and governance need to be addressed to maximize their effectiveness. Recent case studies highlight the transformative impact of AI in areas such as urban management, water resources and multi-cloud environments, highlighting the continued need for technological innovation and research into security strategies, performance optimization and practical implementation to unlock the full potential of this convergence.

2.4 Chatbots and Conversational AI

The field of conversational AI has undergone significant transformations in recent decades, evolving from simple rule-based scripts to complex generative systems capable of understanding and producing human-like language. Chatbots, in particular, have emerged as key applications of this evolution, enabling interaction between humans and machines in a natural and intuitive

way. This section provides a comprehensive overview of the historical and technological trajectory of chatbots, the foundational architectures that underpin modern conversational agents, and their capacity to access and reason over structured and unstructured data through natural language. Special attention is given to the role of LLMs in enabling these advances, as well as to the future directions and challenges facing the development of intelligent conversational systems.

2.4.1 From Rule-Based Systems to Generative AI

Chatbots represent one of the first practical manifestations of AI and Human-Computer Interaction (Adamopoulou and Moussiades 2020). The starting point for this evolution dates back to ELIZA, developed in 1966 by Joseph Weizenbaum, widely recognized as the first chatbot in history (Sadhu, Burman, and Mandal 2022). It simulated a psychotherapist, using simple pattern matching rules, and surprised users with its ability to hold seemingly coherent conversations. Years later, PARRY, created in 1972 by Kenneth Colby, emerged as an evolution of ELIZA, incorporating a more complex personality and simulating a patient with schizophrenia (Adamopoulou and Moussiades 2020). This progression demonstrated the potential of AI to model distinct human behaviors, even with limited computational resources, marking the beginning of a line of research that would grow exponentially.

This technological evolution was accompanied by significant changes in the development models adopted. Initially, rule-based models predominated, in which responses were defined by fixed lexical patterns and manually coded (Rane et al. 2022). Subsequently, these were replaced by models based on information retrieval, which use knowledge bases or APIs to select relevant responses, introducing greater flexibility and adaptability to the context (Adamopoulou and Moussiades 2020). However, the real turning point came with the emergence of generative models, capable of producing new and contextually appropriate responses based on deep neural networks. Supported by advanced architectures such as LLMs, these models paved the way for a new paradigm of AI-based conversation, allowing for personalization, consistency, and adaptation to the user's communication style (Zhu et al. 2025). Examples such as ChatGPT and Google Bard, later renamed Gemini, have demonstrated high proficiency in generating coherent and context-sensitive responses (Al-Amin et al. 2024; Tamim Mahmud Al-Hasan et al. 2024).

2.4.2 Foundational Architectures and Learning Paradigms

The advancement of these systems was made possible by a fundamental architectural leap: the introduction of the Transformer model, proposed by Vaswani et al. (2017) (Vaswani et al. 2017). This architecture replaced recurrent and convolutional neural networks with mechanisms based exclusively on attention, allowing for more efficient parallelization and better use of long-range dependencies. At the heart of this model is the self-attention mechanism, which weighs the relevance of each word in the sequence relative to the others, capturing complex relationships regardless of their positional distance. Positional encodings, multi-head self-attention layers, and feedforward neural networks make up this highly modular and scalable structure.

The effectiveness of Transformers is intrinsically linked to large-scale pre-training, where models are exposed to massive volumes of unlabeled text (Wolf et al. 2020). This process allows them to acquire a generalist understanding of language before being fine-tuned for specific tasks. Models such as BERT, the various versions of GPT (including GPT-3 and GPT-4), LLaMA, and PaLM were all developed using this approach, demonstrating remarkable capabilities in tasks such as machine translation, summarization, classification, and question answering (Zhou and Camba 2025). The scalability of Transformers with the number of parameters and the amount of

training data has been instrumental in the advances seen in AI-based linguistic applications (Wolf et al. 2020).

Despite the impact of pre-training, it is often necessary to refine the behavior of models for specific application contexts. For this, fine-tuning is used, which allows models to be adapted to more targeted data sets (Ouyang et al. 2022). A particularly relevant technique in this context is reinforcement learning with human feedback (RLHF), which incorporates human preferences into the reinforcement learning process. The process involves, in a first phase, supervised training with examples of human responses; in a second phase, the training of a reward model that evaluates the quality of the outputs; and, finally, the adjustment of the main model using algorithms such as proximal policy pptimization (PPO), in order to maximize the generation of responses preferred by users. This approach, used in InstructGPT, has demonstrated significant improvements in the usefulness, accuracy, and safety of the responses generated.

In addition to traditional fine-tuning, the adaptation of LLMs to different tasks is also facilitated by transfer learning techniques, which reuse knowledge acquired in other tasks to improve performance in new contexts (Ruder et al. 2019). One of the emerging strategies in this field is the use of adapters which are lightweight modules integrated between the layers of Transformers that allow the model to be customized without modifying its original parameters (Houlsby et al. 2019). This technique not only promotes efficiency but also reduces the risk of catastrophic forgetting, a recurring problem in continuous learning.

2.4.3 Multimodality and Continual Adaptation

With the maturation of linguistic capabilities, modern LLMs have expanded beyond text, integrating additional modalities such as image, audio, or video. These models, known as multimodal large language models (MM-LLMs), significantly increase the expressiveness and contextual reasoning of interactions by allowing inputs and outputs in multiple formats (Zhang et al. 2024). This evolution involves converting different types of data into a common representational space, allowing the model to understand and generate content in an integrated manner. The applications are vast, including accessibility, medical analysis, and computer-aided design.

A concrete example of this applicability was explored by Zhou and Camba (2025), who investigated the use of MM-LLMs in the context of computer-aided design (CAD). The authors concluded that these models have the potential to transform design processes by promoting greater accessibility and collaboration among users with different levels of technological literacy. By enabling natural language interactions and inputs such as images and sketches, MM-LLMs reduce the technical barriers associated with traditional CAD tools, including non-technical professionals in the creative process and accelerating prototyping and iteration cycles.

This multimodal capability is linked to the growing need for continuous adaptation of models. Continuous learning aims to enable LLMs to evolve over time without the need for complete retraining. However, this approach raises challenges such as catastrophic forgetting, in which the model loses previously acquired skills. To mitigate this problem, approaches such as parameter-efficient fine-tuning, the use of external information retrieval mechanisms such as retrieval-augmented generation (RAG) (Zhou and Camba 2025), and knowledge editing techniques have been explored. In addition, learning by imitation with linguistic feedback allows for the iterative integration of human user contributions, promoting the gradual evolution of the system based on successive interactions.

2.4.4 Data Access, Natural Language Interfaces, and Future Directions

Complementarily, LLMs have demonstrated the ability to process and extract knowledge from structured data, such as tables or knowledge bases, which allows them to respond to complex

queries and provide accurate information. A relevant example is StrucGPT, developed based on the GPT architecture, which allows intelligent querying of structured data in natural language (Jiang et al. 2023). This functionality makes chatbots valuable tools for real-time data access and decision support.

However, integrating chatbots with dynamic data poses significant challenges. One of the main challenges is the ambiguity of natural language queries, which can be interpreted in multiple ways (Kimiya Keyvan and Jimmy Xiangji Huang 2022). This difficulty is amplified in voice-only interfaces, where a single sentence can contain multiple intentions (Zhizhong Wu 2024). Additionally, ensuring fast response times is essential to providing a satisfactory user experience, especially in contexts that require real-time interactions.

Despite these limitations, chatbots have been applied in various sectors, providing decision support and efficient access to information. In the water sector, for example, recent studies show how chatbots have contributed to improving resource management and decision-making (Ray Saikat Sinha et al. 2024). With their increasing influence, it is essential to ensure that their development is responsible, ethical, and transparent (Al-Amin et al. 2024).

In this context, the emergence of natural language interface (NLI) is particularly relevant, as they expand the role of chatbots by allowing users to interact with complex data, such as databases or APIs, through human language, without the need for knowledge of formal languages such as SQL (Affolter, Stockinger, and Bernstein 2019). The central objective of these interfaces is to democratize access to information, reducing the technical barriers associated with querying structured data (Őzcan et al. 2020). Similar to the transition seen in chatbots themselves, NLIs have evolved from rule-based approaches and traditional statistical methods to solutions supported by LLMs, capable of understanding user intent and automatically generating structured commands. A paradigmatic example of this integration is the use of architectures with RAG, as mentioned previously.

In this context, integration with modern data architectures, such as LH, is particularly relevant (J. Aggarwal 2025). These systems combine the flexibility of data lakes with the robustness of data warehouses, as discussed earlier in Section 2.1, creating a unified infrastructure that supports different types of data - including text, image, video, and audio - and enables their efficient analysis in real time. The ability of these infrastructures to ensure quality, apply schemas, and provide uniform access via SQL is crucial for NLP applications, as it ensures a solid and scalable foundation for the extraction of useful knowledge by conversational models. By enabling the transformation of raw data into AI-ready information assets, LH become key players in the operationalization of natural language interfaces, especially in dynamic, large-scale environments.

The practical applications of these interfaces span domains as diverse as e-commerce, healthcare, education, and human resources (Jha, Anand, and Karthikeyan 2025), as well as critical systems such as internet of drones (IoD) platforms (Sezgin 2025) or automatic identification systems (AIS), used for consulting maritime trajectories (Guo et al. 2025). In the context of the IoT, NLIs have been applied in scenarios such as cyber threat detection, device management, and sensor data analysis, providing an accessible layer of abstraction over complex technical infrastructures (Zong et al. 2025). By acting as a bridge between natural language and technical data formats, LLMs substantially increase the accessibility and usefulness of these systems, converting human descriptions into executable commands and presenting structured responses in a fluid manner (Jha, Anand, and Karthikeyan 2025).

With continuous advances in conversational AI research and development, even more sophisticated systems are expected to emerge, capable of understanding natural language, reasoning contextually, and interacting in a personalized manner. Future prospects point to the creation

of proactive agents capable of anticipating user needs and acting autonomously within defined limits (Vishal and Vishalakshi Prabhu 2023). The ability to provide explanations, justify decisions, and continuously adapt will increasingly be a central requirement for the safe and reliable integration of these systems into society.

Chapter 3

Methods, Tools and Responsible AI Practices

This chapter outlines the methodologies and tools employed to address the project's challenges, including the implementation of a data lakehouse (LH) architecture, the development of artificial intelligence (AI) models, and the creation of a virtual assistant powered by large language models. Additionally, it addresses data protection, security, and ethical considerations to ensure compliance with regulatory standards and promote responsible AI usage.

3.1 Methods and Tools

The successful implementation of the Conversational Lakehouse Architecture supported by Real-time AI (CLARA) system required the integration of a diverse set of methods, technologies and tools, capable of responding to the complexity and heterogeneity of the use cases addressed. This section presents the technical foundations and components adopted throughout the development process, including the data infrastructure, natural language processing mechanisms, the libraries and programming environments used, as well as the datasets that supported the training of machine learning models. The decisions made reflect not only the functional requirements of the solution, but also concerns related to scalability, maintainability, performance and security. By combining state-of-the-art tools with robust methodological approaches, it was possible to build a cohesive and flexible system architecture capable of supporting both real-time interaction and advanced analytical tasks.

3.1.1 Data Lakehouse Infrastructure

The data infrastructure developed within the scope of this dissertation is based on the adoption of the LH paradigm, an emerging architecture that aims to overcome the limitations of both traditional data warehouse (DW) and data lake (DL) by merging their strengths into a unified solution. This approach is especially relevant in scenarios that require efficient management of large volumes of heterogeneous data, including structured, semi-structured and unstructured data, as is the case in the scope of this dissertation.

At the heart of this infrastructure is Delta Lake (Lake 2025), an open-source project initially developed by Databricks (Databricks 2023), which acts as a transactional layer over files stored in Parquet format. It is important to note that Delta Lake does not replace Parquet as a storage format, but merely acts as an extension of it. In other words, data is still physically stored in Parquet files, benefiting from columnar compression, selective reading and interoperability with various processing engines.

The use of the Parquet file format is especially relevant, since it is this file format that explains the storage space optimizations obtained during this dissertation. Parquet is a columnar format,

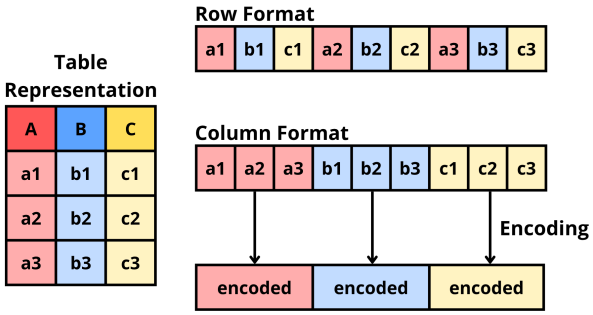


FIGURE 3.1: Regular formats vs Parquet Format (Sharma, Marjit, and Biswas 2018)

unlike more common formats such as CSV and SQL Dump which store data in rows. The columnar format allows the values in each column to be stored in sequence, in independent blocks. This organization favours the application of highly effective compression algorithms, since the values within the same column tend to repeat themselves or follow similar patterns, they undergo a compression process, which differs depending on the nature of the data in each column, but allows for a significant reduction in the space occupied on disk, often more than 80%, compared to row-oriented formats. This comparison can be seen in Figure 3.1.

However, Delta Lake doesn't just use the Parquet format it adds a transactional metadata system, through a directory called `_delta_log`, which allows file directories to be treated as complete tables, with capabilities that were previously restricted to relational databases. These capabilities include support for atomicity, consistency, isolation and durability (ACID) transactions, data versioning, the imposition and evolution of schemas and efficiency in update and removal operations. This guarantee is achieved by sequentially writing transactional logs in JSON files, which describe each change to the table such as the addition, removal or modification of data files. These logs are then combined into checkpoints in Parquet format, speeding up the time it takes to read and restore previous versions. This last aspect is directly related to the time travel feature, which allows access to previous versions of the table, being particularly useful for auditing, debugging and scientific reproducibility. In addition, Delta Lake supports the imposition of data schemas, avoiding the ingestion of files with unexpected columns or inconsistent types. It also allows these schemas to evolve over time, in a controlled manner, without compromising the integrity of historical data. Finally, traditionally costly operations in Data Lakes, such as updates and deletes, are optimized through the application of file compression techniques and efficient maintenance of metadata indexes.

The execution and orchestration of LH operations was carried out using Apache Spark (Apache Spark 2025), a distributed processing engine that guarantees high performance and horizontal scalability. The integration between Spark and Delta Lake makes it possible to process large-scale data with support for SQL operations, complex transformations and pre-processing pipelines for training machine learning models. This integration is especially relevant in an industrial context where the volume of data tends to grow exponentially.

In addition, the MLflow platform (MLFlow 2025) was integrated to manage the lifecycle of machine learning models. This tool allows the registration and comparison of different model versions, as well as the tracking of metrics obtained during training, such as accuracy, loss and f1-score. The integration of MLflow with Spark and Delta Lake makes it possible to keep all the relevant information about the experiments developed in a unified and traceable way, reinforcing the consistency and auditability of the model development process.

This set of characteristics like columnar structure, efficient compression, adaptive coding and

integration with versioning and transaction mechanisms, positions Parquet as an excellent format for modern analytical architectures such as LH. Its use as the basis of Delta Lake ensures that data is not only stored in an economical and scalable way, but is also accessible with high performance and security.

3.1.2 Natural Language Processing

The system developed in this dissertation integrates various natural language processing components, with the aim of enabling the interpretation of natural language questions written by users and its conversion into structured actions within the LH ecosystem. To this end, state-of-the-art language tools and models were used, both locally and accessible via an external application programming interface (API), making it possible to combine performance with flexibility.

The automatic generation of SQL queries from natural language was supported by the use of the Gemini API (Google 2025), a large language model (LLM) provided by Google. This tool was chosen because of its ability to understand complex, large structures, interpret intentions and produce well-formed SQL statements that are semantically aligned with the database architecture. Integration with Gemini made it possible to transform user questions into valid SQL queries, significantly increasing accessibility to the information contained in the system. To complement this query generation component, the Mistral AI API (Mistral AI 2025) was used, which is particularly effective at answering questions based on long documents. This model was used to interpret questions related to the operation of the Navia business software, based on a technical document describing its functionalities in detail. Mistral AI's ability to handle multi-modal documents, i.e. text, tables, images, etc. and provide detailed answers made it particularly useful for this purpose, enabling a user assistance system with a high degree of accuracy and usefulness.

With regard to the local execution of language models, the Ollama (Ollama 2025) tool was used, which allows large language models, such as LLaMA 3.3 (Meta 2025) to be run in a local environment, without the need to connect to the cloud. Through Ollama, it was possible to implement three different functionalities of the solution proposed in this dissertation, such as the automatic classification of questions for their proper routing, the generation of complete answers based on the user's questions and the result of the queries, as well as a relapse system that deals with all questions that do not fit into any of the predefined paths. This local execution was particularly important from the point of view of privacy and performance, ensuring low response times and complete control over the data processed.

Together, these natural language processing components provided CLARA with a robust linguistic layer, capable of interpreting, classifying, transforming and responding to human inputs with a high degree of comprehension and adaptability. This integration of LLMs was crucial in bringing the users' conversational interface closer to the system's complex data, making the experience more natural and efficient.

3.1.3 Python Libraries

All the development carried out as part of this dissertation was implemented in the Python programming language, recognized as one of the most widely used languages in the field of AI due to its vast collection of specialized libraries and the strong scientific and industrial community that supports it. This choice ensured not only high productivity in development, but also fluid integration between the various technical components of the proposed solution.

Several of the libraries used correspond to Python (Python 2025a) interfaces to the tools explored previously, such as the Delta Lake and Apache Spark libraries, which ensured the manipulation and transformation of large volumes of structured and semi-structured data. The

MLflow library also proved to be essential, allowing direct integration with the training code of the Machine Learning ML models developed, enabling the automatic recording of metrics, parameters, model versions and training artifacts, all accessible on a dedicated web interface.

In the field of language models, integration with Gemini was facilitated by Google's official library, which allows direct interactions with generative models without the need for manual configuration of API calls. Similarly, the Mistral AI library provided a simple and efficient connection to the models provided by the company, making it possible to perform inference directly from the local code. In the case of locally executed models, the Ollama tool was used, which provides a local REST API that abstracts the process of executing models such as LLaMA 3.3, maintaining access via HTTP calls, but without the need to send the data to the cloud. This solution ensured privacy, control over the execution environment and reduced response times, while keeping the integration interface simple and coherent with the other APIs used.

Some libraries also proved crucial for very specific features of the system, such as the langdetect library (Michal Danilk 2025) which was used to automatically detect the language in which the user asked the question, allowing the answer to be returned in the same language, ensuring a more natural and contextualized interaction experience. The sqlglot library (Toby Mao 2025) was essential for converting SQL queries generated in the PostgreSQL dialect into the dialect compatible with Apache Spark, ensuring that the instructions were executed correctly on the LH infrastructure.

For the permissions verification module, it was necessary to calculate the semantic similarity between transaction descriptions and database elements. This task was supported by the all-mpnet-base-v2 model, accessed via the sentence-transformers library (Tom Aarsen 2025), which allows sentences to be converted into high-dimensional vectors that preserve their semantics. These embeddings made it possible to identify correspondences between functional descriptions and real database structures, even when the vocabulary and form of expression differed significantly. The results of this semantic mapping stage were stored in the form of a serialized dictionary in a file using the pickle library (Python 2025b), which was essential due to its ability to preserve the native structure of Python objects efficiently, something difficult to achieve with more generic formats such as CSV or JSON.

With regard to model training, the Ultralytics library (Ultralytics 2025) was used to develop an OCR model based on the YOLOv8 architecture with oriented bounding boxes (OBB). This advanced version of YOLO makes it possible to detect objects in inclined positions, which proved crucial given the nature of the images captured in the field, where water meters often appear misaligned. This architecture was chosen for its high efficiency in detecting visual patterns and for the simplicity of the training interface offered by the library.
For the model to predict the next intervention, based on sequential patterns, the Keras library (Team 2025) was used, enabling the construction and training of an LSTM (Long Short-Term Memory) neural network. This type of architecture is particularly effective in modeling temporal sequences, being able to capture long-term dependencies between events, an essential feature for the problem in question.

The FastAPI library (Ramírez 2023) was used to develop the API that interconnects the different components of the CLARA solution. This library stands out for its high performance, native support for data typing with Python and automatic integration with interactive documentation in Swagger UI (SMARTBEAR 2025). Its adoption has made it possible to create efficient and secure endpoints for communication between the user interface, the language models, the prediction modules and the data infrastructure, guaranteeing a modular and scalable architecture.

Finally, libraries such as pandas (pandas development team 2025) and numpy (NumPy community 2025) played a transversal role throughout the project. Pandas was used to read, filter and manipulate CSV files, while numpy was used to visualize images with annotations directly on Jupyter notebooks, the development environment chosen throughout the process.

3.1.4 Datasets for Model Training

The development of the machine learning models in this dissertation was based on the careful construction of two different data sets: one for the optical character recognition (OCR) model and the other for the model for predicting the next intervention, based on time sequences. In the case of the OCR model, the images used were extracted directly from a file system that is segregated from the Navia database. Extracting this content required collaboration with the development team, and a specific User Story was created to allow access to the desired data. The images collected corresponded to a one-year period of activity for a specific client, with the aim of guaranteeing a representative and current volume of operational reality.

After the initial extraction, a careful manual selection was made, with 1,368 images chosen that were relevant to the model's objective. The images were annotated using the Computer Vision Annotation Tool (CVAT) platform, which allows labels to be applied using OBB, which is essential for this specific case, given the variety of angles and inclinations with which the water meters can appear in the photographs. After the annotation process, the images and their labels were exported in .xml format, thus forming the dataset used during the training process. A statistical analysis was then carried out on the dataset, which showed a marked imbalance between classes, as illustrated in the bar chart shown in Figure 3.2. This imbalance results from the predominance of certain digits over others, with the zero digit being particularly frequent. This distribution can be explained by the fact that meter displays typically have between four and five digits, but in residential contexts, consumption rarely reaches high values, which leads to the recurrent occurrence of zeros in the most significant positions. Despite this imbalance, and considering that the process of manually annotating images is time-consuming and demanding, it was decided to use the dataset as it was, with the aim of progressively expanding it in the future, as detailed in Section 5.2

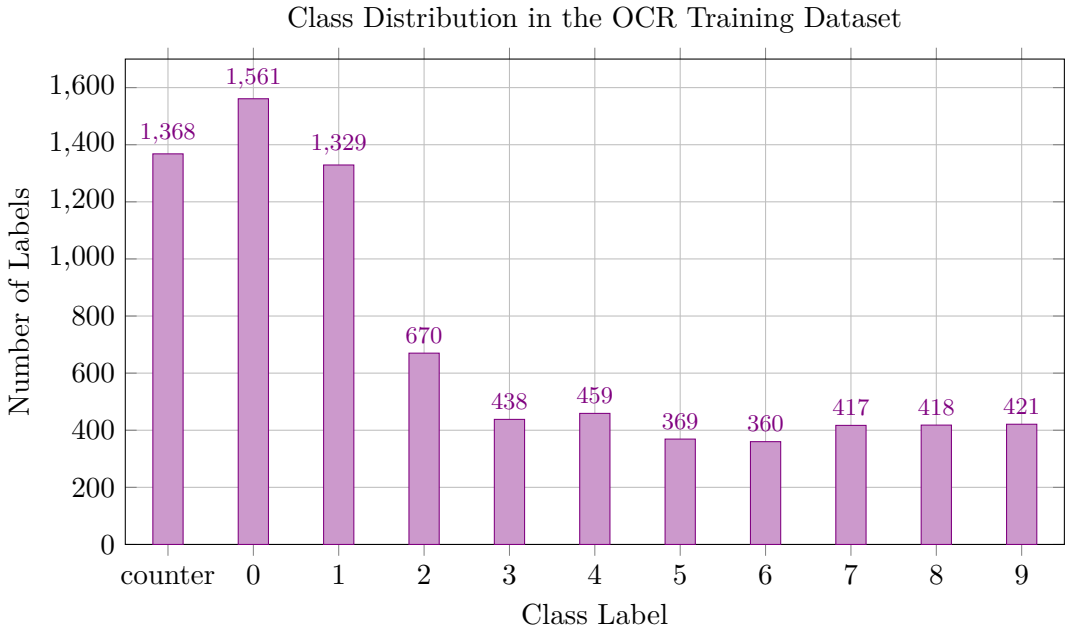


FIGURE 3.2: Bar chart showing the number of images per class (digits 0–9 and counter class) in the OCR training dataset

With regard to the development of the prediction model based on LSTM networks, the dataset used was obtained directly from the operational PostgreSQL database of the Navia system. This extraction occurred prior to the construction of the LH infrastructure and, as such, required significant manual effort in navigating the legacy system's relational structure. The database, which evolved organically over time, lacked centralized documentation or a coherent data governance strategy. Consequently, the identification of relevant information for model training involved a detailed inspection of the available tables, columns, and their relationships.

To support this process, the DataGrip tool by JetBrains was used extensively (JetBrains 2025). This integrated development environment for databases made it possible to visually explore the structure of the PostgreSQL database, including foreign key relationships and cross-schema dependencies. The visual schema diagrams generated by DataGrip were particularly useful in understanding the complex interconnections between the various modules of the system, allowing for the manual reconstruction of data flows that were not explicitly documented. These diagrams served as a critical resource for formulating the SQL queries needed to extract the dataset relevant to the prediction of operational events.

The construction of the dataset began with the formulation of a base SQL query, which aimed to retrieve intervention and request records along with their respective metadata. However, the sheer size of the database posed significant technical challenges. The volume of data made it infeasible to execute a single comprehensive query, due to limitations in available memory and query execution time. As a result, the extraction had to be split into multiple partial queries, segmented by time intervals. These query subsets were subsequently merged and consolidated into a single dataset suitable for training the sequential learning model.

An additional layer of complexity stemmed from the hierarchical structure of the data. Interventions and requests within the Navia system are linked through parent-child relationships, forming directed graphs that represent real-world workflows and dependencies. In order to reconstruct the complete event sequences required for LSTM training, a recursive traversal algorithm was implemented to navigate these graph relationships. This graph traversal process, applied to a dataset containing hundreds of thousands of nodes and edges, took over 24 hours to execute fully, highlighting the scale and complexity involved.

The final dataset, after consolidation and validation, comprised approximately 135,000 rows, placing the problem within the Deep Learning context. This justified the choice of a deep learning architecture, specifically LSTM networks, capable of modeling long-term dependencies and sequential patterns in large datasets. The effort invested in this phase not only enabled the successful training of a robust predictive model but also exposed the critical need for more structured, documented, and analytics-ready data infrastructure, further motivating the development of the LH platform described in subsequent sections.

3.1.5 Development Environment and Infrastructure

The technical infrastructure that supported the development of the CLARA solution was carefully configured to meet the computational demands associated with handling large volumes of data, training ML models and running LLMs locally.

The development environment was configured to ensure reproducibility, isolation and remote accessibility. All development was carried out via secure SSH access to the cloud machine from a private network. A Python virtual environment was created specifically for this project, encapsulating all dependencies and ensuring compatibility across runs. Development took place primarily using Visual Studio Code's remote SSH feature, which enabled direct interaction with the codebase and data from a local machine. The virtual instance was typically active between

6 to 8 hours per day, balancing computational availability with cost-efficiency throughout the project's lifecycle.

The CLARA API was executed using the FastAPI library, and the application server was managed by Uvicorn (Marcelo Trylesinski 2025). The API and other relevant interfaces, such as the MLflow control panel and the User Interface developed were exposed via SSH tunnels with port redirection, allowing remote access to local server services in a secure manner. This configuration ensured a seamless development experience without compromising data integrity or security.

This technical environment made it possible to support all stages of the dissertation, from data exploration and preparation, through model training and validation, to the final integration of the solution, ensuring robustness, flexibility and autonomy throughout the development process.

3.2 Data Protection, Security, and Ethical Considerations

The adoption of solutions based on AI raises fundamental questions regarding data protection, information security and ethical responsibility. This section describes in detail the measures adopted throughout the development of the CLARA solution to ensure compliance with applicable legal and ethical standards, with particular emphasis on data privacy, infrastructure security and the role of AI as a human support tool.

3.2.1 Legal and Ethical Framework

The development of AI systems requires a rigorous and conscious approach to the legal and ethical frameworks that regulate their use, especially when sensitive data or decisions that may directly or indirectly affect citizens' rights are involved. In the European context, this requirement is particularly relevant given the robustness of the regulatory framework in force, whose central reference is the general data protection regulation (GDPR) (European Union 2018). This legislation, which is mandatory in all European Union Member States, establishes a set of fundamental principles that guide the processing of personal data, playing a decisive role in promoting digital trust and safeguarding human dignity in the information age.

Among the structural principles established in the GDPR, the principle of data minimization stands out, which requires the collection and use of only the information strictly necessary for the defined purpose, thus promoting a conscious reduction in exposure to the risk of privacy violations (Hoofnagle, Sloot, and Zuiderveen Borgesius 2019). In addition, the regulation values mechanisms such as data anonymization and pseudonymization, which are considered effective strategies for protecting data subjects against unwanted re-identification processes. When correctly applied, anonymization makes the process virtually irreversible, allowing the legitimate use of data for research and development purposes, as in the present case, without compromising individual privacy. Data processing must also be based on consent obtained freely, informed, and unequivocally. This principle reinforces citizens' control over their own data and obliges responsible entities to ensure that any collection or processing occurs on the basis of clear and transparent authorization. At the same time, the GDPR introduces the principle of accountability, which requires organizations not only to comply effectively with the rules, but also to be able to demonstrate, in an auditable manner, that all processing operations are compatible with the legal framework in force. This guideline requires a proactive attitude on the part of entities, involving strategic planning throughout the entire data lifecycle.

Complementing this framework, the international standard ISO/IEC 27001 emerges as a fundamental reference for information security management (ISO 2025). This certification, widely

adopted globally, defines a set of best practices for establishing Information Security Management Systems (ISMS) that ensure the confidentiality, integrity, and availability of information (Culot et al. 2021). Its application is particularly relevant in projects such as the one described here, which are based on digital infrastructures and handle significant volumes of operational data. Adopting the principles of this standard allows for the systematic structuring of risk identification and mitigation, promoting an organizational culture geared towards resilience in the face of cyber threats and security incidents (Kitsios, Chatzidimitriou, and Kamariotou 2023). It is a globally recognized standard which, together with ISO 31000 (Risk Management), provides a coherent framework for identifying, assessing, and mitigating risks (Oliva 2023).

In the emerging regulatory landscape, the proposed European regulation on AI, known as the AI Act, introduces an innovative approach based on risk classification, segmenting AI systems into four categories: unacceptable, high, limited, and minimal risk (Oliva 2023). This categorization aims to ensure that the level of regulatory requirements is proportional to the potential impact of the technology on fundamental rights and the safety of individuals. Unacceptable risk systems, such as subliminal manipulation or social scoring mechanisms, are explicitly prohibited. High-risk systems, such as those involved in the management of critical infrastructure or sensitive decision-making in areas such as justice or employment, are subject to strict requirements, including compliance assessments, technical audits, and extensive documentation.

In the case of the CLARA solution, the application of AI is limited to decision support, without any degree of operational autonomy or direct control over physical systems. For this reason, the most appropriate framework is that of limited risk, a category that includes, for example, conversational assistants that interact with users and require only transparency mechanisms, such as clear identification that the interlocutor is an automated system. The automation of critical infrastructure, such as water distribution networks, is a high-risk area and, as such, has been expressly avoided in this project, ensuring that all decisions with operational impact are validated by a human.

This commitment to transparency and user self-determination is also echoed in the European Charter of Digital Rights, a guiding document that aims to strengthen fundamental rights in the digital environment (European Commission 2025). Among the principles enshrined in this charter is the right to explanation, which ensures that every citizen has the right to meaningfully understand the logic behind automated decisions that affect them (De Pasquale et al. 2022). This requirement, which is linked to the principle of explainability, aims to counteract the opacity often associated with advanced algorithmic models, promoting more understandable, auditable, and fair AI (Hoofnagle, Sloot, and Zuiderveen Borgesius 2019).

In this sense, the development of AI systems must respect not only legality but also a set of widely recognized ethical principles under the designation of Trustworthy AI (Thiebes, Lins, and Sunyaev 2021). This concept structures trust in AI around five fundamental ethical principles: the principle of beneficence, which guides the development of systems that promote human well-being and contribute to improving quality of life; the principle of non-maleficence, which requires the active prevention of harm, protecting users from physical, psychological, or social risks; the principle of autonomy, which defends the ability of individuals to make informed and free decisions, even in AI-mediated contexts; the principle of justice, which seeks to ensure that systems do not reinforce inequalities or discriminate against vulnerable groups; and, finally, the principle of explainability, which imposes the need for systems to be transparent, understandable, and auditable, so that their impacts and operating logic can be effectively scrutinized.

Alignment with these ethical and regulatory guidelines is reinforced by several international initiatives. The Ethics Guidelines for Trustworthy AI, promoted by the European Commission, establish seven fundamental requirements for the development of trustworthy AI systems, including, among others, data governance, technical robustness, and social inclusion (Smuha

2019). The Organization for Economic Co-operation and Development (OECD) Principles on AI converge in arguing that AI should promote inclusive growth, respect human rights, and function in a robust and secure manner (Corrêa et al. 2023).

In the context of this work, these principles are embodied in the decision to design all AI modules under the human-in-the-loop paradigm, ensuring that technology functions as a support tool and never as a substitute for human decisions. This approach ensures that operators always retain the ability to supervise, correct, or reject the suggestions presented by the models, thus preserving human agency and the operational integrity of the processes.

In this context, it is important to highlight a particularly relevant ethical concern in the current field of AI: the biases embedded in LLMs. These models, now widely used as pillars of NLP, have demonstrated transformative capabilities in text interpretation and generation, but their widespread adoption has highlighted substantial risks associated with the data that feeds them (Ranjan, S. Gupta, and Singh 2024). Several studies, such as the one presented by the authors in (Lin and L. Li 2025), have revealed that LLMs tend to reflect and amplify social biases present in their training corpora, including gender, racial, and cultural biases. These biases are not merely theoretical: they manifest themselves tangibly in the responses generated by the models, which can reproduce discriminatory stereotypes or reinforce historical inequalities. Their presence raises serious ethical and operational concerns, as biased outputs can negatively affect already marginalized groups, compromise the impartiality of automated processes, foster misinformation, and erode user trust.

Given these risks, it is essential to critically understand the mechanisms through which biases are established and persist in LLMs, as well as to adopt proactive strategies to mitigate them. In the context of this dissertation, where pre-trained language models play a central role in several components of the developed solution, this concern was always present. Their use was accompanied by a continuous effort of validation, human supervision, and clear delimitation of contexts of use, precisely with the aim of minimizing unwanted impacts and ensuring an ethical, transparent, and socially responsible application of AI.

3.2.2 Compliance Measures in the CLARA Solution

In light of the legal and ethical considerations discussed above, this subsection details the concrete measures implemented throughout the CLARA solution to ensure compliance with applicable standards. These measures reflect a conscious effort to translate normative principles into secure, transparent, and responsible technological practices.

All the data used in the development of this dissertation was obtained from a single client of the Navia platform, and that client gave explicit consent for it to be used for research and development purposes. It is important to note that no sensitive information from end customers was accessed or used. The data sets used relate exclusively to internal operations carried out by company employees, with no direct link to personally identifiable data. Navia already implements anonymization mechanisms in its database by default, ensuring that variables with the potential to identify individuals are anonymized. Consequently, all the data transferred to LH was already duly anonymized, guaranteeing compliance with the GDPR in all future developments resulting from the new data infrastructure.

With regard to security and privacy, one of the important points to mention is the fact that the virtual machine responsible for hosting the LH, the models and the CLARA API is protected by secure access credentials, and only two users have authorized access: the author of this dissertation and the company's information technology (IT) infrastructure manager. Access is exclusively via the SSH protocol with private key authentication, and no server port is directly exposed to the internet since all access is via controlled tunnels.

In addition, the project's code is stored in a password-protected environment and has never been published in public repositories. The management of permissions and data security has been carried out carefully, with the aim of avoiding information leaks, improper access or the risk of the system being compromised.

On the topic of information leaks, a crucial point in the development of the proposed solution was the use of the Gemini API to generate SQL queries. The use of this tool was preceded by a detailed analysis by Navia's team of administrators, with special attention to the risk associated with exposing internal database structures to external services. The JSON file containing the detailed description of the database architecture, including schema names, tables and fields, was initially considered sensitive, as it could reveal structural information about the database. This type of information, although it does not contain personal data, is a strategic asset, and its inadvertent disclosure could represent a vulnerability in terms of information security. It should be noted that platforms such as Google Gemini, like other widely used language models such as ChatGPT, often use the content of interactions with users for the purposes of retraining and continuous improvement of the models. This practice, although common, raises legitimate concerns regarding the privacy and confidentiality of the information shared with the models, especially when it involves structural knowledge of critical systems. Inadvertently sharing the structure of the database with these models, even if only for the purposes of formatting or syntactical help, could, in theory, contribute to the systems learning sensitive patterns, exposing the organization to uncontrolled risks.
For this reason, the test with Google's API only went ahead after formal approval from the company's technical management, and specific containment measures were established to mitigate the risks identified. Even with approval, it was ensured that only the database structure and never the data itself would be provided to the model, and that the results of the queries generated would never be sent or processed in the cloud. The processing, execution and interpretation of queries is done locally, using exclusively private language models, run via Ollama, guaranteeing a high degree of control and privacy at all stages of information processing.

Another of the project's main concerns was to ensure that users were fully aware that they were interacting with an AI system, and that it may have limitations. For this reason, the virtual assistant's test interface, already conceived as a skeleton of what will become the interface for future integration into the Navia software, explicitly displays the message: "CLARA is an Artificial Intelligence Model. It can make mistakes." This phrase, although simple, reinforces the principle of transparency and contributes to a more conscious and informed use of the solution.

With regard to compliance with the AI Act, the use of AI in this project will be strictly limited to functions that assist human users in accessing and analyzing information. As previously discussed, the AI Act classifies autonomous control of water infrastructures as an unacceptable risk. In alignment with this restriction, the CLARA system was designed exclusively to assist, not replace, human decision-making. All interactions and decisions derived from AI will have to be validated and approved by human operators, thus avoiding any automation that could pose risks to the security of critical infrastructures.

In line with the principles of ethical responsibility, the CLARA solution was developed with the aim of supporting the work of human operators, not replacing them. A clear example of this principle is the OCR model developed during the dissertation, which in the future will be used directly in the field to facilitate the automatic reading of water meters using the camera on the technicians' mobile device. The aim is to eliminate the need for manual reading, reducing operating time and the likelihood of human error, but always keeping the operator as the decision-making agent. This approach reinforces the vision of AI as a complementary tool, placed at the service of professionals, and never as an autonomous system with authority over critical processes.

Although the models used were not trained with sensitive data or subject to personal decisions, there is a clear concern about the possibility of bias. The solution developed will not be adopted directly into production, but will undergo a rigorous and prolonged testing phase in order to identify any limitations or inconsistencies in the models. All decisions with an operational impact will be subject to human validation, respecting the principles of fairness, responsibility and non-discrimination.

Chapter 4

A Conversational Lakehouse Architecture supported by Real-time AI

This chapter presents the solution conceived and developed as part of this dissertation, called Conversational Lakehouse Architecture supported by Real-time AI (CLARA).This solution is designed to meet multiple operational and analytical needs, including formulating answers in real time, running queries with permission control based on the user's profile, providing predictive assistance in carrying out tasks and offer software clarifications.

The CLARA's architecture is structured into three main functional blocks and several interconnected components, each responsible for a specific set of functionalities:

- **Data Infrastructure**: A data lakehouse (LH) composed of Bronze, Silver, Gold, and Models layers. Supports ingestion, cleaning, preparation, and storage of structured and unstructured data. Includes integration MLflow for model training and monitoring.
- **Application programming interface (API)**: The core orchestration layer of CLARA, which includes several interconnected components that coordinate the processing of each type of query:
  - **Classification LLM**: Analyzes the incoming question and classifies it into one of four categories: Lakehouse Query, Predictive Model, Navia Manual or General Conversation. This classification determines the processing path.
  - **Generate Query LLM**: Generates SQL queries from natural language questions. It uses the database's architecture and schema descriptions as context to ensure queries are semantically aligned.
  - **User Permission Tool**: Validates whether the user has the correct permissions (View, Create, Modify, Delete) to execute the operations in the generated query, based on Navia's transaction model.
  - **Query Execution Tool**: Executes validated queries on the structured layer of the Lakehouse and returns the result.
  - **Complete Answer LLM**: Converts the result of a database query into a natural language answer that is easy for the user to understand.
  - **Predictive Model Tool**: Uses a trained artificial intelligence (AI) model to predict the next most probable symptom in an operational sequence.
  - **Software Clarification LLM** : Answers questions about the functionality of the Navia software using the official manual as context.

  - **General Conversation LLM**: Responds to all other domain-specific questions that do not fit the predefined paths, maintaining a conversational flow aligned with Navia.

- **User Interface**: Responsible for capturing the user's question and displaying the final response. Acts as the entry and exit point of the system.

Figure 4.1 illustrates the overall structure of the system, highlighting how the three main functional blocks interact to provide secure, real-time and context-sensitive responses to user queries. In order to present CLARA's functionalities and justify the design options adopted, a detailed description of its architecture is provided throughout this chapter.

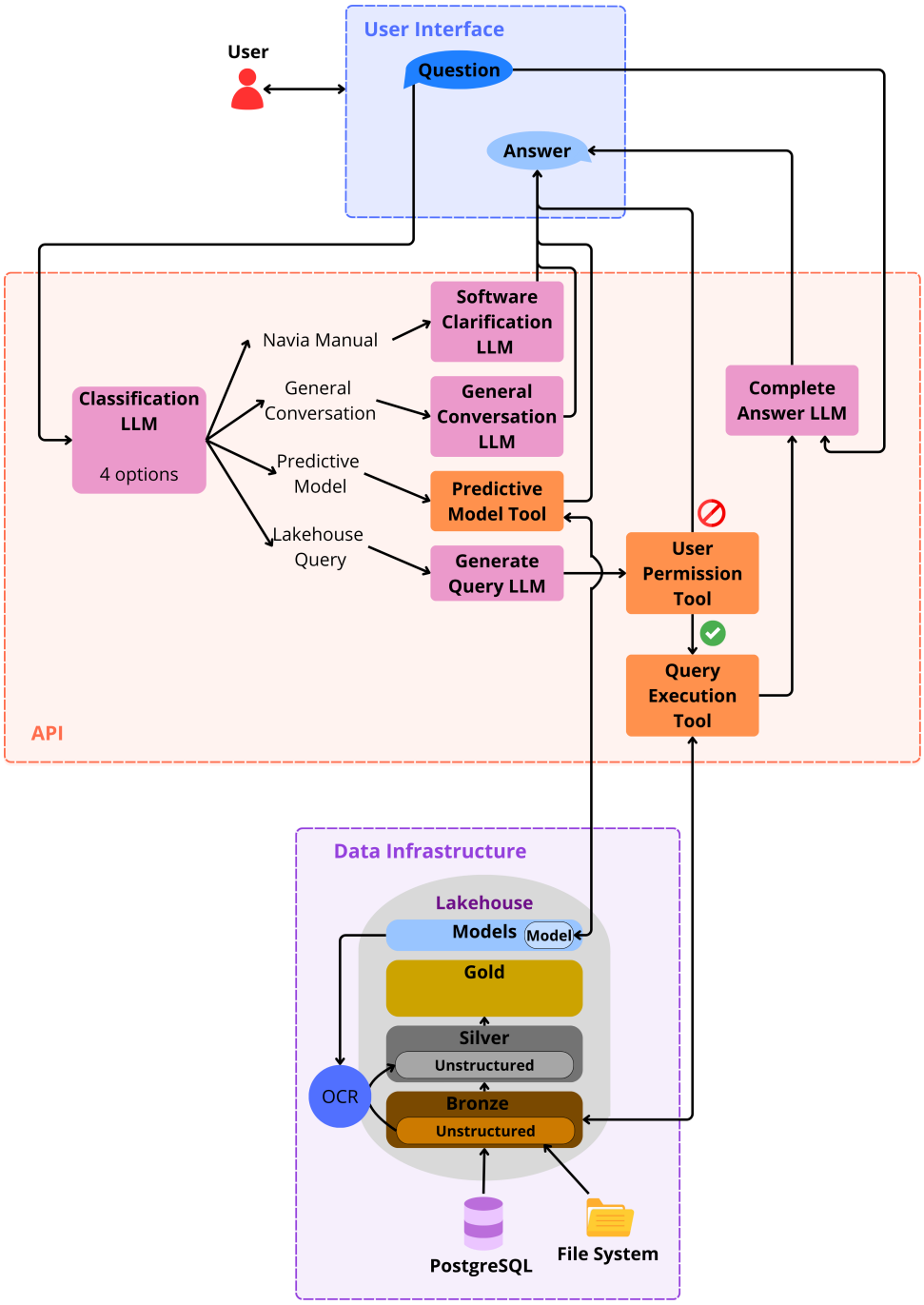


Figure 4.1: CLARA's architecture

4.1 Data Infrastructure

CLARA was developed under the foundations of a new modern data infrastructure geared towards AI workflows. The architecture adopted follows the LH paradigm, based on Delta Lake over Apache Spark, and is organized into functional layers, following the medallion architecture, also known as multi-hop architecture, which is a data design pattern used to logically organize data within the LH with the aim of incrementally and progressively improving the structure and quality of the data as it flows through each layer of the architecture - Bronze, Silver and Gold. This flow between layers starts with the ingestion of structured data from the PostgreSQL system, goes through the cleaning and refinement phases and ends with obtaining data ready for machine learning models. MLflow is also introduced in the top layer - Models - as an experiment management mechanism, contributing to the traceability and reproducibility of the models developed.

In order to build the first layer of the LH, Bronze, a data ingestion pipeline had to be built. The data present in this layer originated from a PostgreSQL database composed of 48 schemas, 1,144 tables, and 7,365 columns, totaling approximately 235 GB of historical data accumulated over ten years. In addition to structured data, 6,241 unstructured image files of photographs taken by operators in the field were loaded into the LH. In the current Navia software system, these images are segregated from the relational database and stored in a file system, accessible only to the development team via code that has to be developed for this purpose. With the proposed solution, it became possible to consolidate structured and unstructured data in a file system.

The Bronze layer is subdivided into two categories, structured and unstructured, in order to accommodate different types of data from the source system. It also allows all information from the source system to be preserved, including schemas, table and column names, data types, timestamps, etc. and serves as the single source of truth for all downstream processes. No transformations are applied at this stage; instead, the emphasis is on high-fidelity replication. The ingestion logic is designed to support full loads, performed only manually. Since the entire development of this dissertation was carried out in a cloud environment and, the virtual machine used was only accessible for a few hours a day, it was not possible, within the scope of this dissertation, to test real-time loading scenarios with continuous ingestion and change data capture (CDC) integration.

The ingestion pipeline was developed using Apache Spark and Delta Lake. A custom script iterated through the schemas and tables, inferring their structure and exporting data in chunks using the PostgreSQL JDBC driver. Full loads were used instead of CDC due to time and resource constraints. The system faced frequent `Java heap space` errors due to the volume and complexity of some tables. This was mitigated by dynamically identifying large tables and partitioning them manually, allowing memory usage to remain within acceptable limits and ensuring greater robustness in the ingestion process. In order to facilitate access to the ingested data, SQL views were created automatically for each table, allowing queries to be executed directly on top of the Bronze layer using SQL syntax compatible with Spark.

The Silver layer, also subdivided into structured and unstructured categories, is responsible for the first phase of data curation and normalization. Structured data is read from the Bronze layer and subjected to a series of cleaning operations. Completely empty tables are automatically excluded and columns with high proportions of null values (above a defined threshold of 60%) are eliminated to reduce noise. Columns with constant values, i.e. no variation between records, are also eliminated, with the exception of fields relating to dates, which are always kept due to their potential analytical value. The process also includes a slight mechanism for detecting outliers, using the interquartile range (IQR), which filters out numerical values outside

acceptable statistical limits. Each clean table is partitioned by year, when date or timestamp fields are available and suitable. The resulting datasets are written in Delta format in the Silver layer, maintaining structural coherence and significantly reducing redundancy and irrelevant data.

The Gold layer marks the final stage of the transformation pipeline and is dedicated to preparing the data for machine learning applications. The process begins by classifying the columns into numeric, boolean, categorical, free-text and temporal types. Boolean values are converted into integers and categorical fields with low cardinality are encoded using a pipeline that combines StringIndexer and OneHotEncoder. Text fields with high cardinality are tokenized and vectorized using HashingTF, after filling in the nulls with empty strings. The timestamp columns are split into their constituent parts (year, month, day and hour), providing numerical features derived from temporal patterns. All relevant fields are then assembled into a unified feature vector using VectorAssembler, forming the basis for training and inference in machine learning workflows. Tables that do not produce any valid features are ignored, ensuring that only analytically useful datasets are promoted to this final layer. Data in the new formats are again stored in Delta format in the Gold layer.

Finally, the last layer of the LH, the Models layer, differs from the previous ones because it is not directly interconnected through a logical sequential flow with the Gold layer, unlike the other layers of the architecture. Even so, this layer plays a fundamental role in supporting the development of artificial intelligence models by enabling the structured storage of all relevant elements throughout their life cycle. The Models layer is subdivided into three main subcomponents: Datasets, Artifacts and MLRuns. The Datasets sublayer is responsible for storing the datasets used to train the models. These datasets can come from either the Gold or Silver layer, depending on the type of data and the level of pre-processing applied, since processing to model-ready format has so far only been carried out on structured data. The Artifacts sub-layer stores all the trained models as well as the notebooks used for training, ensuring their preservation, versioning and subsequent reuse. The MLRuns sublayer is generated automatically by the MLflow tool, whose operation is detailed in Section 3.1.1. This layer allows all the metadata associated with the training experiments carried out to be viewed via the web interface, present on Figure 4.2, which includes the execution time, the datasets used for training and validation, the evaluation metrics obtained, the model's architecture, among other relevant parameters.

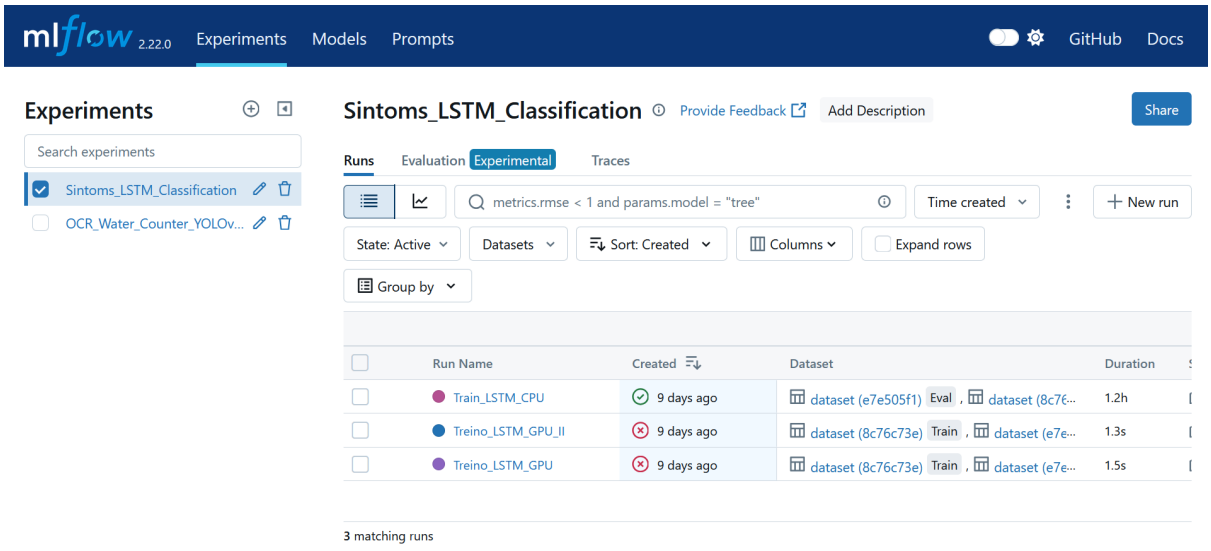


FIGURE 4.2: MLFlow Interface

It is precisely the Models layer that is the daily work of the AI developers at Navia, as it provides the necessary support for experimenting, monitoring and managing models in the context of the intelligent solutions developed in-house.

This layered implementation demonstrates the viability and efficiency of using a LH architecture to support AI development. By progressively structuring and transforming the data while respecting its operational origin, this system guarantees the quality of the data while respecting its original structure, thus keeping the information reliable throughout the process.

In addition to ingesting, organizing and making available structured and unstructured data, the new infrastructure is also designed to boost the development of artificial intelligence models, capitalizing on the resources stored in the LH. One of the most concrete examples of this integration is the implementation of an optical character recognition (OCR) model for the automatic validation of water meter readings taken in the field.

During an intervention in the field, whenever operators take meter readings, they are instructed to take a photograph of the meter and manually enter the value of the reading into the Navia software, via its mobile version. Once the intervention has been synchronized with the central system, the photographs are automatically stored in the file system, while the values entered manually are associated with the metadata of each image, both of which are integrated into the unstructured data sub-layer of the Bronze layer. The name of each file follows a specific convention: it starts with the value recorded by the operator, followed by a unique identifier, as illustrated in Figure 4.3.

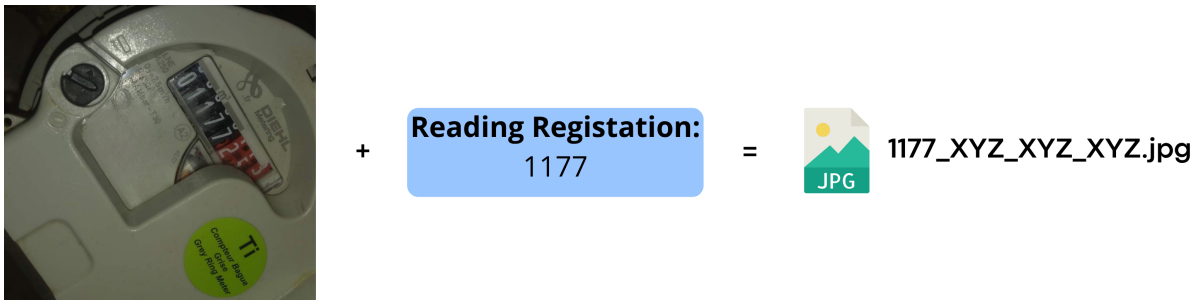


FIGURE 4.3: Naming process of field images

However, this process is naturally subject to human error, either through incorrect reading or manual input. To ensure that only correctly validated photographs are promoted to the unstructured data sub-layer of the Silver layer, an intelligent OCR model was developed, capable of autonomously identifying and extracting the visible reading on the image, and comparing it with the value entered by the operator.

To train the model, a set of 1,200 meter images from the Bronze layer was selected and manually labeled on the Computer Vision Annotation Tool (CVAT) platform. Labeling was carried out using oriented bounding boxes (OBB), which are essentially just bounding boxes with an angle, an essential approach in this context, given the nature of the devices in the field: meters are often installed vertically, in places that are difficult to access, or are simply not captured with the best framing by the operator. The dataset was then exported in XML format.

The model was trained for 100 epochs with a batch size of 16, image resolution of 640×640 pixels, and a learning rate of 0.01 using the stochastic gradient descent (SGD) optimizer. The training process was executed on the GPU-enabled virtual machine described in Chapter 3, leveraging the NVIDIA L4-90 GPU. The configuration was specified in a dedicated YAML file that defined the dataset structure, class names, and label format.

The entire process, from training to inference, was significantly simplified by the use of the Ultralytics Python library, which provides a high-level abstraction layer over the internal components of YOLO. This library enables rapid experimentation and deployment through a unified API that automatically handles data loading, preprocessing, model configuration, training, evaluation and export, requiring very little repetitive or setup code. As a result, the complexity of managing deep learning workflows is greatly reduced, allowing AI developers to focus on tuning the model and integrating it with downstream systems.

After training, the model was exported in the `.pt` format and integrated directly into the image processing pipeline of the LH. The deployment logic consisted of scanning the unstructured image repository in the Bronze layer, applying the trained model via the same Ultralytics interface, and selecting only those images for which the predicted counter value matched the one entered manually by the operator. This automatic validation mechanism ensured that only trustworthy images were promoted to the Silver layer without human intervention.

This process illustrates not only the infrastructure's ability to handle heterogeneous data (structured and unstructured), but also the ease with which it is now possible to develop machine learning models based on real data from the field. More than a storage architecture, the LH now acts as an intelligent foundation, which not only houses the data, but learns from it, actively contributing to the reliability and quality of the information available upstream.

4.2 API

When a question enters the CLARA system, it triggers a semantic analysis and computational decision process that determines the most appropriate way to resolve it. This decision logic is centered on a large language model (LLM), which performs contextual analysis and classifies the question into one of four predefined categories: database query, functional clarification based on Navia documentation, prediction of the next intervention, and general conversation.

To handle this task, the system relies on LLaMA 3.3, the most recent model made available by Meta. Its use allows the capture of semantic nuances that would not be possible through heuristic rules or traditional classifiers. The classification logic is implemented through a custom prompt specifically designed to guide the model toward a deterministic choice among the four possible paths. Notably, this prompt proved effective in multilingual scenarios, correctly classifying all tested questions even when expressed in a different language from the prompt itself. This multilingual capability is particularly relevant in production contexts, where linguistic flexibility enhances applicability across diverse user bases.

The LLaMA 3.3 model runs locally through the Ollama framework, which acts as a lightweight HTTP server that exposes large language models via a RESTful interface. This integration abstracts the complexity of deployment while ensuring that all data remains within the local infrastructure, a fundamental requirement for privacy and operational autonomy. Ollama enables fast and secure model access without relying on external APIs.

The following prompt, shown in Listing 4.1, defines the classification categories and enforces a strict response format in JSON to ensure seamless integration with the downstream routing logic of the API.

```
"""Classify the user's question into one (and only one) of the following categories:
- "lakehouse query": if the question is about operational data, interventions,
    readings, or any type of information existing in the system's databases;
- "Navia manual": if the question is about how to use the system, what a particular
    feature does, or other questions related to the use of Navia software;
- "predictive model": if the question is related to automatic predictions (e.g.,
    prediction of the next intervention, failure model, etc.);
- "general conversation": if it does not fit into any of the previous categories, but
    is still in the context of Navia.
Important: respond only in the following JSON format (without additional
    explanations):
{"name": "classify_question", "parameters": {"classification": "<write the chosen
    category here>"}}"""
```

Listing 4.1: Prompt used for LLaMa 3.3 question classification

This classification process ensures consistency, minimizes ambiguity, and guarantees compatibility between the output of the LLM and the logic responsible for routing the question to the correct processing module.

After classification, the system applies automatic routing that directs the question to the appropriate execution module. Internally, the classification function is invoked first, followed by one of the four processing paths: database query (via Gemini), manual clarification (via Mistral), next-intervention prediction (via LSTM), or general conversation (via LLaMA 3.3). These modules are orchestrated and returned through a unified interface, simplifying the logic and ensuring modularity. Each of these modules will be explored in the subsections that follow.

The CLARA system exposes a single REST endpoint through the FastAPI framework. The root POST route, `/chat`, receives a JSON object containing two required fields: `username` and `question`. The request is processed synchronously, and the system returns a JSON response with the field `answer`, which contains the final reply generated by the appropriate module.

This architectural design, centered on classification followed by modular execution, ensures that CLARA remains extensible and maintainable. The API layer is fully decoupled from the user interface and data infrastructure, allowing any component to be updated, replaced, or scaled independently.

Classification is, therefore, a critical mechanism within the architecture, ensuring that each user question follows the most efficient resolution path. It is also important to note that, although the current logic supports a fixed set of categories, the architecture was built to accommodate future extensions. The same classification mechanism can be adapted to integrate additional categories, tools, or external services, making CLARA a scalable and future-proof system.

4.2.1 Classification LLM - Navia Manual

When the user wants to ask questions about the general operation of the Navia platform, such as: "Where can I create a new team?", "What is the difference between scheduling and planning?", or "What is the request menu for?", the initial classification is "Navia manual". To answer these kinds of questions, the CLARA architecture includes a dedicated path that uses the Mistral API, a lightweight language model optimized for document comprehension tasks. The aim of this path is to transform the Navia Manual, an extensive and detailed technical document, into an interactive resource accessible through natural language.

This manual contains an exhaustive explanation of the fundamental concepts of the system, the different menus and the possible actions in each component. In addition, the document includes

screenshots, tables and examples that visually illustrate how the platform works. Unlike other sources used in the system, this manual has not undergone any prior processing or additional structuring. It was uploaded directly to Mistral's API, which is able to interpret and reason about complex documents, including their textual and visual elements. When a question is forwarded to this route, the Mistral model directly accesses the content of the document to formulate an answer based on the information contained within, as per shown in Figure 4.4. This approach eliminates the need for pre-processing and allows the document to be kept up-to-date at all times without additional integration effort.

To operationalize this functionality, a dedicated function was implemented in the API layer. This function is triggered when the classification output is "Navia manual" and performs a POST request to the Mistral API. The user's question is sent as a text field, while the full Navia Manual is attached as a binary file. This ensures that the document is treated as an integral part of the prompt, maximizing the model's ability to reason over its contents. The integration was designed to be stateless and synchronous: the document is reattached to every request, avoiding the need for persistent storage or indexing mechanisms.

After Mistral's API is called the response is parsed and returned directly to the user. In case of failure (e.g., network issues or quota limits), a fallback message is returned, maintaining the reliability of the system and providing a graceful degradation path.

The use of Mistral in this context has proved to be effective in its ability to understand functional questions, even when formulated informally or ambiguously. With the direct integration of the manual into the Navia software, it is expected that it will be possible to significantly reduce users' dependence on formal clarifications or manual consultation of the documentation, making access to information more fluid, intuitive and autonomous.

By integrating this path, the CLARA architecture reinforces its mission of intelligent operational support, guaranteeing not only access to data, but also to the functional knowledge that underpins the effective use of the platform.

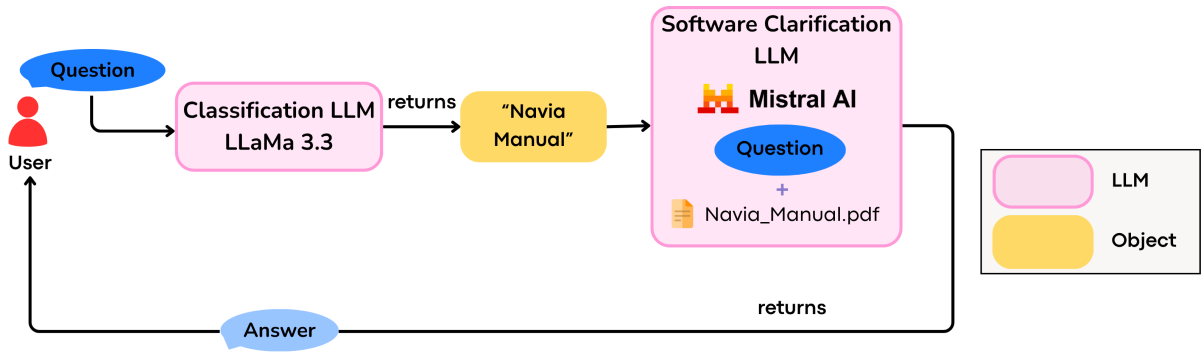


FIGURE 4.4: Navia Manual Consultation via LLM Flow

4.2.2 Classification LLM - General Conversation

In any interaction system based on natural language, users can be expected to ask questions that don't fall directly into specific or operational tasks. These interactions include generic questions such as "What is your name?", "What do you do?", or other spontaneous expressions that escape the flows previously defined in the architecture.

To deal with this type of question, a specific general conversation path was included in the CLARA solution, using the LLM LLaMA 3.3, and it was configured with a context restricted to the Navia platform domain. In this way, the model doesn't answer questions that are personal, generic or outside the scope of the system, preventing users from using it as a general assistant.

This route is activated when the question is not classified in any of the three other categories: database query, functional clarification based on the manual, or prediction of the next intervention. In these cases, the classifier assigns the "General Conversation" intention, and the question is forwarded directly to the LLaMA model, which generates a short, appropriate answer that fits within the context of the CLARA solution.

Technically, the LLaMA 3.3 model is executed locally using the Ollama runtime, which exposes the model through an HTTP server. The API layer of CLARA performs a direct POST request to this local endpoint, sending a minimal structured prompt that provides system-level context and the user's question. The system does not store or reference past messages, each interaction is treated as a standalone exchange. This stateless design guarantees that conversational drift is avoided, and the assistant maintains a consistent scope restricted to the Navia platform.

The prompt used to guide the LLM, presented in Listing 4.2 is specifically designed to reinforce this domain restriction. It clarifies that the model is part of a virtual assistant embedded in the Navia ecosystem and should never respond to questions unrelated to the software.

```
"""
You are a polite and concise virtual assistant named CLARA. You work inside the Navia
    software ecosystem, which is a management platform for infrastructure and
    utilities.

You must only respond to questions related to the Navia system. If the user asks
    about something outside the scope of Navia (e.g., personal questions, external
    topics, unrelated software), reply with something like: "Sorry, I can only help
    with topics related to Navia."

Answer briefly and in a helpful tone.
"""
```

Listing 4.2: Prompt used for LLaMA 3.3 General Conversation path

The main function of this path is to ensure that the system is able to respond politely, informatively and contextually to marginal questions, maintaining the fluidity and naturalness of the user experience, without compromising the functional limits of the application.

By including this path, represented on Figure 4.5, the CLARA architecture ensures predictable and robust behavior even in situations of ambiguity, reinforcing its communicational resilience and its positioning as a virtual assistant geared towards functional support for the Navia platform.

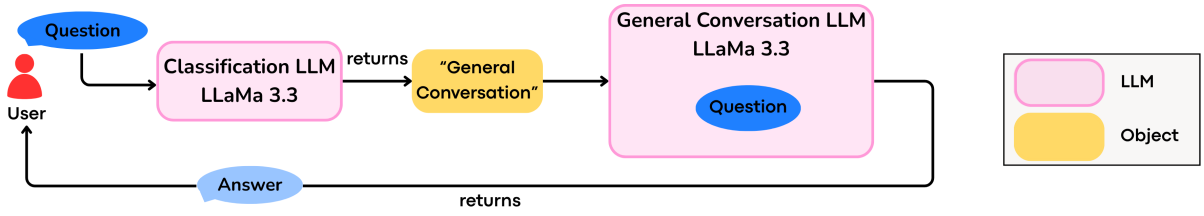


Figure 4.5: General Dialogue Management Flow

4.2.3 Classification LLM - Predictive Model

One of the features incorporated into the CLARA architecture is the intelligent prediction of the next task to be carried out by field operators, based on the interventions previously recorded in the system. This route is activated when the user's question reflects the intention to obtain

a recommendation on what to do next during an intervention, for example: "What is the next intervention I should carry out? Symptom1, Symptom2"

In the context of the Navia software, a general intervention, such as repairing a pipeline, is often made up of multiple sub-tasks, such as placing signs, opening the sidewalk, carrying out the repair, replacing the sidewalk and removing the signs. Although these patterns are common, there is no fixed or standardized sequence between teams or types of intervention. Therefore, the central aim of this route is to predict, based on real historical data, what the next "symptom" associated with the next intervention or request will be. In Navia, a symptom is the name given to the specific purpose of a task, acting as its functional identifier.

To develop this functionality, a sequential prediction model was created based on neural networks of the long short-term memory (LSTM) type, recognized for their ability to model temporal dependencies in sequential data. The construction of the dataset began with the extraction of the logical links between interventions and requests, based on the link keys in the Navia database. These are interconnected by means of parent and child ids, and through these it was possible to build the chained flows, represented in the form of ordered lists, in which each element corresponds to an entity (intervention or request) linked to the next.

Subsequently, each entity was associated with its respective symptom, resulting in sequences such as:

`Symptom1 → Symptom2 → Symptom3 → Symptom4 → SymptomN`

These sequences served as the basis for training the model. Before training, the sequences were tokenized, converting symptoms into unique numerical representations, and normalized in length using `pad_sequences` which adds zeros at the beginning to standardize the input dimensions. The dataset was then split into training (80%) and test set (20%), ensuring robust evaluation.

The model was built using TensorFlow and Keras, with a sequential architecture composed of:

- An `Embedding` layer;
- Two `LSTM` layers with 128 units each, both followed by a `Dropout` layer with rate 0.5;
- A `Dense` layer with softmax activation to predict the next symptom.

The model was compiled with the Adam optimizer (learning rate 0.001), and trained using the sparse categorical cross-entropy loss over 15 epochs with a batch size of 64. The best performing run reached over 94% accuracy on the test set. All metrics, hyperparameters and model versions were tracked using MLflow, allowing full traceability and reproducibility.

The main function of the model, in the context of the CLARA API, is to receive as input a sequence of symptoms and return the prediction of the next most likely symptom. To do this, the history of the intervention in progress is automatically extracted, transformed into the corresponding tokenized representation and fed into the model, which returns the predicted symptom that is then sent to the user.

The integration of this functionality into the CLARA system, as shown in Figure 4.6, illustrates how the infrastructure has been designed not only to consult existing data, but also to anticipate future events by exploiting operational patterns implicit in historical data. The predictive model is stored in the Models layer of the LH, together with the training data, hyperparameters and evaluation metrics.

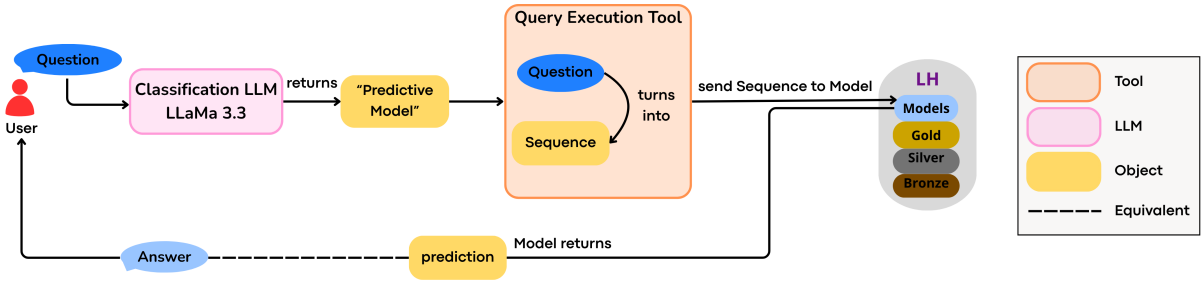


FIGURE 4.6: Access to AI models via LLM interaction Flow

It is important to note that, although the current implementation of this component supports a single predictive model, CLARA's architecture was designed to be easily extensible in order to accommodate multiple models in the future.

4.2.4 Classification LLM - Lakehouse Query

When a question is classified as belonging to the Lakehouse Query category, the aim of the system is to convert the user's intention into a valid SQL query, capable of being executed on the data stored in the LH infrastructure. This process includes three main stages: query generation, permissions checking and execution.

Query generation is carried out using the Gemini API - Gemini 2.0 Flash model, a LLM developed by Google. To make this generation possible, an input file was created in JSON format that represents the complete structure of the database: schemas, tables, columns, data types; as well as the manually generated .txt file that includes a description of the data contained within each schema in the Navia context. These files are supplied to the model along with the user's question, allowing Gemini to produce a query that respects the SQL syntax and semantics of the database structure. The prompt used for this task is showcased bellow in Listing 4.3 and it was created with the help of Google AI Studio, which allows users to select the LLM model of the API they want to use and generate the appropriate prompt to work with that specific model.

```
"""
## Instructions

I need your help generating SQL queries for PostgreSQL.  I will provide you with two
    files:

1. A TXT file named `schema_description.txt` containing general descriptions of the
    schemas in the database.
2. A JSON file named `detailed_schema.json` containing the complete definition of the
    schemas, tables, columns, and their data types.

* Use the `schema_description.txt` file to understand the purpose of each schema.
* Use the `detailed_schema.json` file to learn about the exact tables, columns, and
    data types.
* Generate **ONLY** the SQL code necessary to answer my `question` in natural
    language.
* **DO NOT include any explanations or additional text in the answer.**

question: {question}

"""
```

LISTING 4.3: Prompt used for Gemini API Lakehouse query generation

The choice of the Gemini API for the query generation task was not arbitrary but rather motivated by practical limitations related to the size of the structural representation of the database. The JSON file generated, although it contains around 43,000 lines, in practice represents a substantially higher number of tokens, given the granularity of the content and the nested structure of the JSON format. This volume of information easily exceeds the context limits allowed by most language models, including LLaMA 3.3, whose context window is limited to 128,000 tokens. Although LLaMA 3.3 has been used successfully in other components of the system, its limitation in this respect makes it impossible to use it for tasks that require extensive and detailed context.

The Gemini API, on the other hand, offers an extended context window of up to 1 million tokens, making it possible to include the entire JSON file as part of the input prompt, without the need for truncation or partial content selection. This capability is essential to ensure that the model generates queries correctly aligned with the actual structure of the database, preserving the semantic integrity of the original question.

The query generated by the Gemini model is produced using PostgreSQL's SQL dialect, a strategic decision aimed at guaranteeing the future portability of the conversational system to the Navia software, which operates directly on the relational database and not on the new LH infrastructure. This approach makes it possible, at a later stage, to decouple the dialog module from the CLARA architecture and integrate it directly into Navia's current production environment, without the need for significant re-engineering.

However, to enable immediate execution of the query generated in the LH, it became necessary to automatically convert the SQL statement from PostgreSQL format to a dialect compatible with the underlying execution engine. For this purpose, the sqlglot library was used, which offers a feature called transpose, capable of performing the translation between different SQL dialects in an automated and reliable way.

This step guarantees the frictionless portability of SQL statements between different environments, avoiding manual dependencies and increasing the system's flexibility, both for local execution and for future integration with other platforms.

Before the query is executed, the system triggers a permissions checking module called the User Permissions Tool. This module will be explored in detail below, but it essentially cross-checks the intentions extracted from the query with the user's access rights, previously defined in Navia. The check includes the type of operation (read, write, update or delete), the tables involved and the permissions profile assigned to the user in question. If the requested operation is not authorized, execution is blocked and an informative response is sent to the user.

If the permission is successfully validated, the query is forwarded to the Query Execution Tool, which communicates with the structured Bronze layer of the LH, where SQL views have previously been created to allow data to be read directly from the file system.

Finally, the result of the query is returned to the central system, where it is processed by another LLaMa 3.3 that converts it into a natural, understandable and useful response for the end user. The prompt used for this task is present below in Listing 4.4. It is also important to note that the language used by the user in the initial question is detected, using the `langdetect` library, so that it can be sent to the LLM at this stage to ensure that the user receives a response in the same language in which they asked the question. This process is only performed when the classification is "Lakehouse query" and served only as a guarantee that the final response would be in the same language. However, as the author of this dissertation was able to ascertain, all LLMs used in the CLARA solution are capable of recognizing the user's language and responding in the same language, ignoring the language in which the prompt was written, in this case, English.

```
""" You are a helpful, polite and friendly assistant. Your job is to rewrite an
    existing answer in a warm, clear and supportive tone...
- Answer strictly in this language: **{final_language}**
- Never use another language.
- Do not ask questions or for clarification.
- Be warm, friendly and confident.
Now here is the input to rewrite:
Original user question:
{user_question}
**Correct answer (rewrite this in a friendly manner):**
{intermediate_answer}  """
```

Listing 4.4: Prompt used for LLaMa 3.3 complete answer

This flow, ilustrated on Figure 4.7 , demonstrates how it is possible to combine generative models with security and technical compatibility mechanisms, making large volumes of operational data accessible through simple natural language interactions, one of the main objectives of the CLARA architecture.

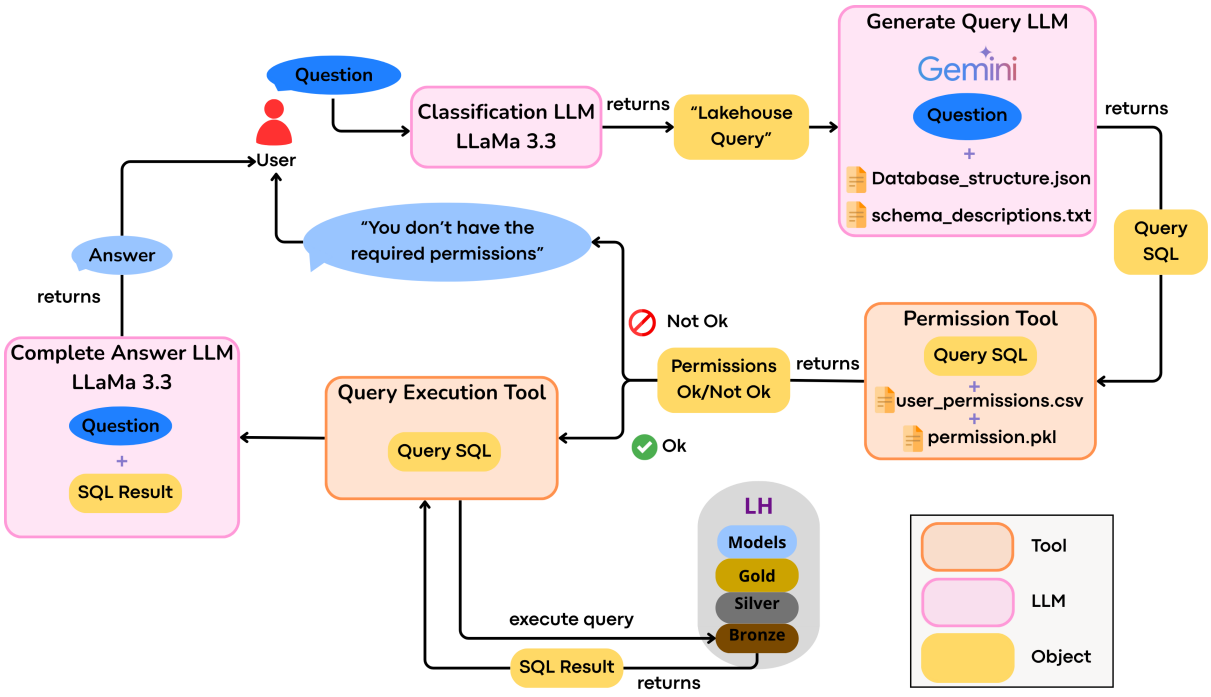


Figure 4.7: Query Generation and Execution Flow

One of the fundamental pillars of the CLARA architecture is its ability to guarantee that only duly authorized users can access the data and functionalities provided by the infrastructure. To ensure this control, a permissions verification system has been implemented that is strongly based on the native structure of the Navia software, which uses a transaction-centered access control model.

In Navia, each user can be associated with one or more virtual teams. It is these teams that indirectly determine the set of transactions to which the user has access, where each transaction represents a concrete functional action in the system - such as “Create User” or “Intervention Query” - and is always associated with a functional module, such as “Users”, ‘Requests’ or “Reports”. Different operation permissions can also be defined for each transaction, represented by action masks: View, Create, Modify or Delete.

To enable the automatic verification of permissions in the context of the LH infrastructure, a process was developed to extract and consolidate permissions per user. This process starts by identifying the virtual teams associated with each user, followed by extracting the authorized transactions for each team. The resulting permissions are unified and organized by user and stored in individual .csv files, which indicate the available transactions, the associated masks, the functional modules and the corresponding URL in the Navia interface.

In order to semantically enrich the system's understanding of the functional meaning of each transaction, the official Transactions Manual document, available in .docx format, was used. This document organizes the transactions hierarchically, grouped by module, and provides a detailed textual description of how each feature works. A script was developed that automatically went through the document, identifying the modules and transactions, and extracting a complete description for each one, comprising all the paragraphs up to the next transaction. The information collected was structured in a dataset with columns representing the module, the short description and the expanded functional description.

Based on this data, an enriched dataset of transactions was built, which served as the basis for a semantic matching process with the database structure. To do this, the JSON file was also analyzed, containing all the schemas, tables and columns of the relational database. Each table was converted into a simplified textual representation, describing its name and columns. From here, a phrase embeddings model (all-mpnet-base-v2) was used, along with the cosine similarity metric which measures the semantic proximity between each transaction and the database tables. For each transaction, the three most semantically close tables were identified, and the result was stored in a Python dictionary listing transactions, tables and available permissions.

This dictionary was then persisted in .pkl files and used in real time by the CLARA API backend when executing natural language queries. The verification process begins when the classifier identifies a question as a "database query". Once the query has been generated by LLM (Gemini), the list of tables involved in the SQL statement is extracted through syntactic analysis. At the same time, the type of access required is inferred - "View" in the case of a SELECT, "Create" in the case of an INSERT, 'Modify' in the case of an UPDATE and "Delete" in the case of a DELETE.

Next, the user-specific permissions file is loaded, which contains all the authorized transactions. For each of these transactions, the system consults the semantic dictionary to check whether there is a match between the tables referred to in the query and the tables associated with the transaction, as well as whether the type of operation required is compatible with the permission masks assigned. Only if a complete match is found between the tables and the access rights is the query authorized and executed. Otherwise, the system blocks the operation and informs the user that they do not have sufficient permissions to carry out the requested action.

This approach offers multiple advantages. On the one hand, it guarantees a high level of security by ensuring that no query is executed without explicit validation of the user's permissions. On the other hand, it allows for clear traceability, since permissions are organized by file, by user, and can be easily audited. In addition, the solution is scalable: new users or changes in permissions can be accommodated with simple updates to the permissions files, without the need for internal reconfiguration of the verification logic. Finally, the use of the Transaction Manual has allowed the system to be enriched with a deeper semantic understanding of how Navia works, making the verification mechanism more robust, even in cases of overlapping transaction names between different modules.

Permission verification, as designed, is thus transparently integrated into CLARA's conversational flow, functioning as a critical layer of logical control that ensures that the system's intelligence always respects the limits of human authorization.

4.3 User Interface

As part of the development of the CLARA solution, a dedicated user interface was implemented with the specific purpose of validating the integrated operation of all system components in a controlled prototype context, as shown in Figure 4.8. It is important to note that this interface does not aim to replicate or replace the Navia software interface, nor is it intended for deployment in its final version.

Within the Navia software development process, no functionality is integrated directly into the product without going through formal product management processes such as backlog prioritization, sprint planning, and validation via user stories. Any modification to the official interface must be previously analyzed, discussed, and validated by the Product Management team, following internal workflows and timelines. Given the limited time frame available for the development of this dissertation, direct integration of CLARA into the Navia interface was not feasible.

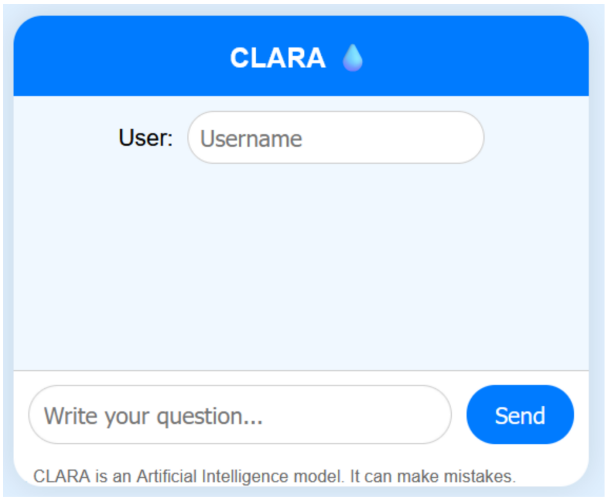


FIGURE 4.8: CLARA's User Interface

For this reason, a lightweight, standalone HTML interface was created as a proof of concept to simulate user interaction with the CLARA system. This interface serves both as a visual skeleton of what may be incorporated into the final product and as a testing environment to support development and iterative debugging of the backend components. It also reinforces the robustness of the CLARA solution as a whole, given that the user interface is the primary point of contact between users and the underlying system composed of the API and LH infrastructure.

Technically, the interface was implemented as a single HTML file with embedded JavaScript, without the use of any frameworks or external libraries. The layout is minimal and functional, allowing the user to input their name and pose questions in natural language. Each question is sent via an HTTP POST request to the unified `/chat` route exposed by the CLARA API. The response is then displayed in real time in the conversation window, simulating a chat-like interaction. The interface also includes a basic loading animation to improve the experience and simulate production-level feedback.

This interface enabled end-to-end testing of multiple features: intent classification, permission validation, database query generation and execution, document-based clarification via LLM, predictive model invocation, and multilingual support. Its design ensured that each interaction could be tested independently and reliably, accelerating the development cycle and contributing to functional coherence across all system layers. The multilingual capabilities of CLARA are shown in Figure 4.9.

Furthermore, by developing this layer in complete separation from the backend logic, the architecture of CLARA reinforces the principle of modularity and system decoupling. This ensures that, when the time comes to integrate the assistant into the actual Navia product, the backend components will already be ready to connect to a new frontend layer, without requiring any architectural refactoring.

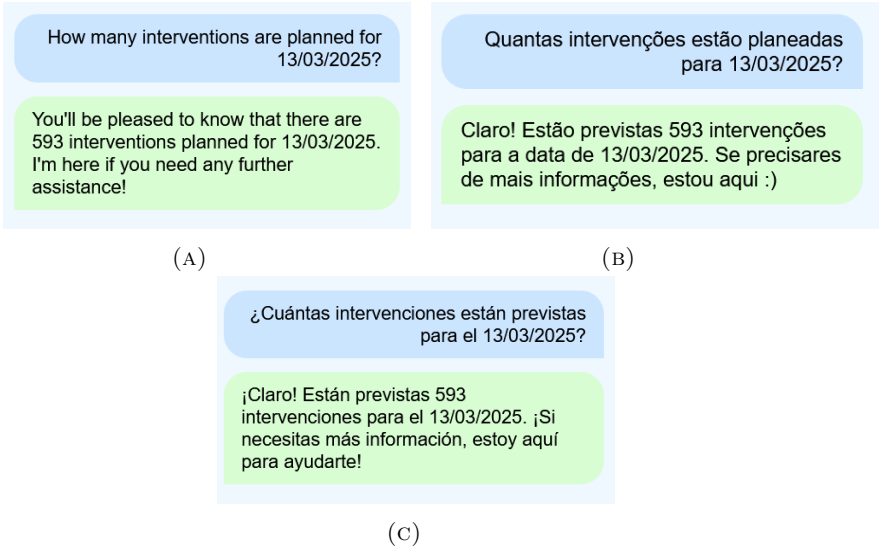


FIGURE 4.9: CLARA's multilingual support (A) English (B) Portuguese (C) Spanish

In summary, the interface developed in the context of this dissertation is not a final product but a functional testbed that validates the feasibility and consistency of the CLARA system, simulating real-world usage scenarios in a controlled and modular way.

Chapter 5

Case Studies

To demonstrate the practical applicability of the CLARA architecture, four case studies were developed using real data, each focusing on a key component of the proposed solution. These case studies aim to validate the system's capabilities in real-world contexts and to showcase how each module contributes to solving distinct operational challenges. Table 5.1 provides a summary of the objectives and scope of each case.

Table 5.1: Description of the case studies

Case Study	Objective	Description
Data Lakehouse for AI pipelines	Demonstrate the creation of a modern Lakehouse for AI and analytics	A Delta Lake-based data lakehouse was implemented with a layered architecture, supporting raw ingestion, cleansing, ML pre-processing and model storage.
Vision and predictive models for operational support	Demonstration of OCR and predictive modeling for continuous automation and validation of tasks	An optical character recognition (OCR) model based on YOLOv8 validates meter readings entered by operators, while a predictive model predicts the next likely tasks, both integrated and executed in the ML-ready layer.
Conversational intelligence framework	Validate the orchestration of multiple LLMs for domain-specific interaction	Test Gemini for SQL query generation, the Mistral API for software explanations and a local LLM for contextual and resource dialog in the conversational intelligence framework.
End-to-end orchestration and integration	To validate the overall integration and responsiveness of the intelligent system	This case demonstrates the complete solution flow, from question classification to proper LLM invocation, SQL execution and predictive response, including a test interface and dynamic routing logic.

5.1 Data Lakehouse for AI pipelines: a data architecture case study

This first case study aims to demonstrate the effectiveness of the data lakehouse (LH) architecture developed, not only as a functional structure, but above all as an optimized solution for ingesting, organizing and rationalizing large-scale data. Its implementation responds to Objective O1 of the dissertation, validating the infrastructure's ability to support artificial intelligence pipelines in a structured, scalable and traceable way.

To fully understand the rationale behind the adoption of a LH architecture, it is crucial to analyze the condition of the original relational database that served as the source for this transformation. Over the years, the PostgreSQL database of the Navia system has evolved organically, without a defined data governance strategy or rigorous documentation practices. As a result, it currently contains over a thousand interrelated tables, many of which are deprecated, redundant, or completely empty. Additionally, the naming conventions used throughout the database are inconsistent and often ambiguous, further complicating the task of data interpretation.

Figure 5.1 illustrates the structure of a single schema from the PostgreSQL database used by Navia, obtained with the DataGrip software. Although it represents only a subset of the system's data architecture, the figure already reveals a high degree of complexity. Each node corresponds to a table and each edge represents a foreign key or dependency, often spanning across multiple schemas. This visual density exposes not only the tight coupling between modules, but also the absence of a clear modular or semantic separation, making the schema extremely difficult to interpret, evolve or query. Such structural opacity significantly hinders analytical workflows, onboarding of new team members and the development of intelligent components that depend on consistent and clean data access.

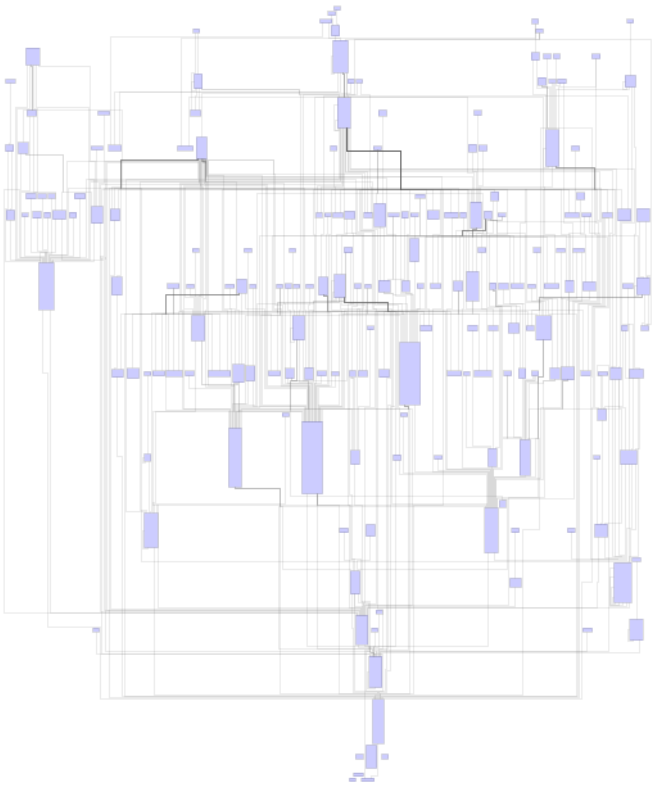


FIGURE 5.1: Visualization of the PostgreSQL database schema used by Navia

This structural "entropy" of the legacy system strongly motivated the development of the LH architecture and, after the initial ingestion, the data transformation and cleaning process began, in order to create the Silver layer, where empty tables, redundant columns and inconsistent data were removed. The gains obtained were significant: while the Bronze layer occupies 3.01 GB, with the same 1145 tables and 6662 columns, the Silver layer occupies just 2.51 GB, with 647 tables and 3102 columns. This reduction not only represents a saving of around 50% in the number of columns and 43% in the number of tables, but also shows a substantial reduction in the storage space required, which proved to be less than 5% of the space originally occupied in the relational database. Figure 5.2 visually illustrates this comparison, highlighting

the efficiency obtained with the transition to the Delta Lake format and the application of systematic data filtering, transformation and organization processes.

In addition to storage gains, this reorganization of information helps eliminate functional redundancy, facilitating analytical exploration, model training and data quality management. Separation into layers also makes it possible to isolate raw data from clean data, and to reserve specific layers for data ready for modeling and for storing models, as can be seen in the Gold and Models layers, respectively.

This case study clearly demonstrates that the LH infrastructure not only makes it possible to consolidate and simplify access to data, but also promotes more efficient use of computing resources and greater operational sustainability. Its adoption in the Navia context represents a significant evolution from the traditional approach based on monolithic relational databases, paving the way for the fluid integration of intelligent components such as those described in the other case studies.

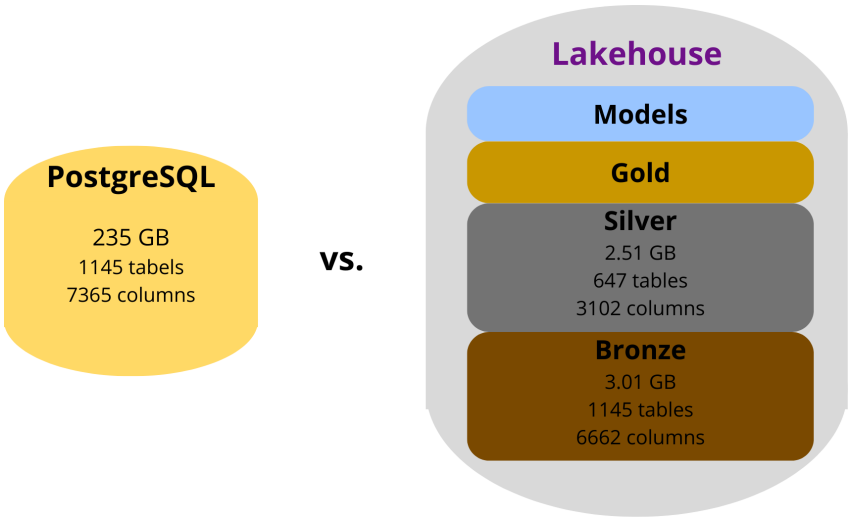


FIGURE 5.2: Comparison between PostreSQL and Data Lakehouse

5.2 Vision and predictive models for operational support: a dual-purpose AI module case study

The second case study aims to demonstrate the practical integration of AI models into the CLARA architecture, validating its ability to support models developed with structured and unstructured data, ensuring the organized storage of their artifacts, the recording of their training data and the monitoring of their evolution over time. To this end, two different models were designed and integrated: an optical character recognition (OCR) model, applied to water meter images captured during interventions in the field, and a sequential prediction model, based on LSTM networks, designed to anticipate the next intervention based on historical patterns of recorded interventions. Both models were trained and versioned using the MLflow platform and are now duly integrated into the Models layer of the LH infrastructure.

The first solution developed sought to mitigate the risks associated with human error in the process of recording meter readings. In the field, operators must photograph the meters and manually enter the respective values into the Navia software mobile application. Once synchronized, both the image and the value are stored in the system. However, these readings can be affected by interpretation or typing errors. To automatically validate the readings entered, an OCR model was trained based on the YOLOv8 OBB architecture, a variant specialized in

detecting objects with variable orientation, ideal for scenarios in which the meters are in vertical positions or at irregular angles.
The train model will be used as per shown in Figure 5.3.b where it is possible to verify the class "counter" as the big red rectangle drawn around the display area, as well as all the different digits detected by the model. It is also possible to validate the models capacity to understand the numbers even in cases where the angle is highly accentuated and the lighting conditions are not ideal.

(a) Water Meter

(b) Oriented Bounding Boxes on Water Meter

Figure 5.3: Image and the Labeling made by the Model

The training process was conducted entirely with the support of the MLflow platform, which made it possible to store and version the trained model in the Artifacts sublayer of the LH, record the dataset and hyperparameters used, as well as continuously monitor the model's evaluation metrics, such as precision and recall. All the information on the different tests was automatically stored in the MLRuns sublayer and can be viewed and analyzed centrally via the MLflow web interface. In addition, the Ultralytics library also automatically generates graphs of evaluation metrics, like the normalized confusion matrix, present in Figure 5.4, which greatly facilitated the analysis of the results obtained by this model.

The normalized confusion matrix shows that the model has a solid overall performance, with most classes showing high levels of recognition rates. The "counter" class has a remarkable rate of 97%, which reflects the robustness of the model in distinguishing between digits and background. Classes '0', '1' and '6' also performed consistently, with rates of over 80%. However, there are increased difficulties in distinguishing more visually similar digits, such as '3', '4' and '5', whose recognition rates range between 55% and 66%, with significant dispersion between neighboring classes. These errors suggest that the model is particularly sensitive to ambiguous or poorly represented visual patterns in the training set. This was already highlighted in Chapter 3 when analyzing the dataset characteristics, specially the discrepancy between number of labels per class.

Even so, this case study fulfills its main purpose: to demonstrate that a LH architecture is capable of hosting AI models, facilitating their development, management and continuous evolution. In fact, the model's current performance precisely reinforces the relevance of this type of infrastructure, since it will be necessary to carry out new training with a more comprehensive

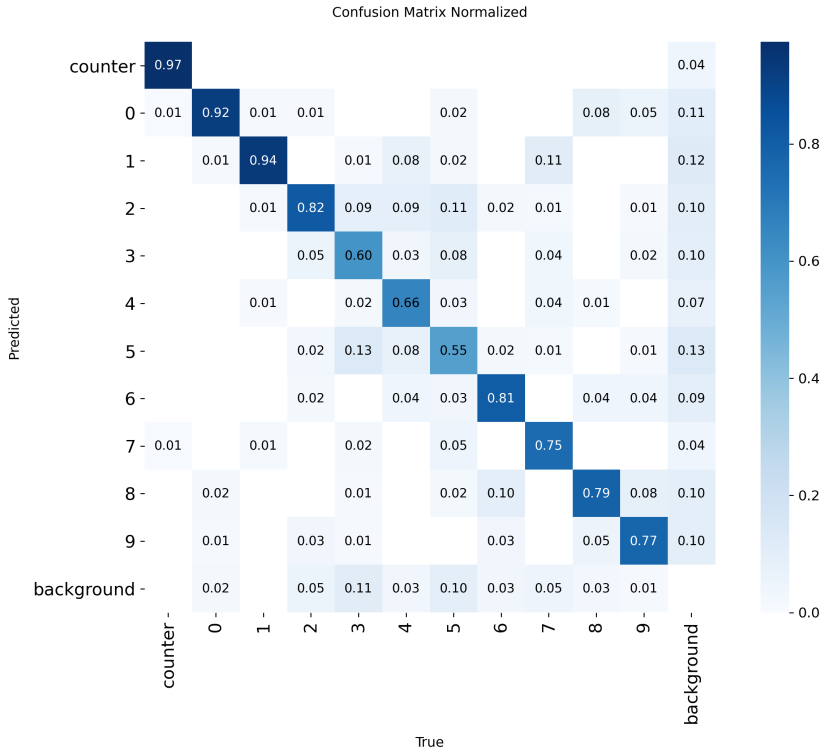


FIGURE 5.4: Confusion Matrix of the OCR Model

dataset and balanced classes, a process that can be significantly facilitated when supported by a LH.

Even though the results were not ideal, the model was still applied to the complete set of images stored in the Bronze layer. Only the photographs whose estimated reading matched the value entered by the operator were promoted to the Silver layer, helping to improve the reliability of the unstructured data that is now available for analytical exploration or later use in AI flows.

The images stored in this layer were not restricted to photographs of water meters, since operators in the field often capture images related to the various tasks they perform. Given that the aim of this case study is to assess the ability of this LH infrastructure to promote and host AI models, this model was used to detect all the photographs that contained detections of the "counter" class, which, as described above, has a particularly high recognition rate.

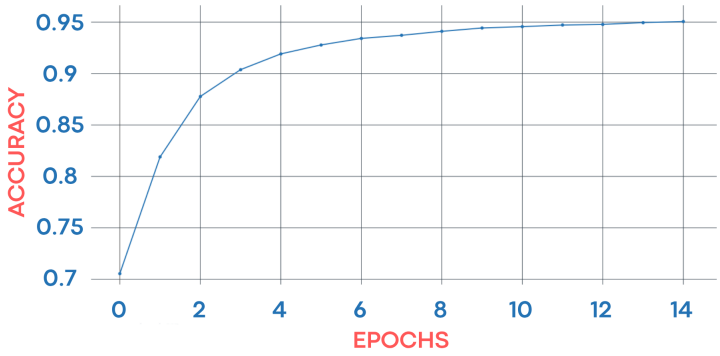
Initially, this layer contained a total of 6241 images of the operators' daily work. After applying the model, it was possible to reduce this number to 2483 images containing only water meters. This process proved to be particularly relevant given that, before the existence of this model, building datasets suitable for training OCR models on water meters required significant manual effort in selecting the images where the meters were visible.

The 2483 images identified as containing counters were then subjected to a new stage of analysis, with the aim of automatically validating the reading made in the field. This validation consisted of comparing the value read by the OCR model with the value included in the name of each image file, as described in Chapter 4. Only the images that successfully passed this check were promoted from the Bronze layer to the Silver layer.

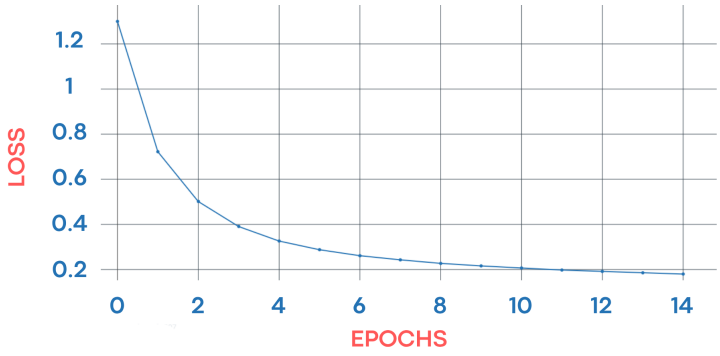
In total, 669 images met the defined criteria and were promoted. Although this number may reflect limitations in the performance of the OCR model, the results obtained show the ability of

an LH to incorporate intelligence directly into the infrastructure, making it possible to automate validation tasks and promote the development and application of AI models on heterogeneous and unstructured data.

The second component of this case study aimed to support operators in the process of creating work orders by automatically suggesting the next most likely intervention based on previous occurrences. For this a LSTM model was used since they have a particularly effective architecture for modeling temporal dependencies in sequential data.
Although this model had previously been trained outside the LH infrastructure, a new iteration of training was carried out on this architecture, with the aim of ensuring that the entire life cycle of the model, from data to metrics, was properly recorded in MLflow, as shown on Figure 5.5 which presents the graphs generated during the model's training, illustrating the evolution of both accuracy, Figure 5.5a, and loss, Figure 5.5b over time. With this new iteration, the model is now fully integrated in the Models layer, with all the essential elements such as datasets, artifacts and experimental records, properly versioned and accessible for future reuse.



(A) Accuracy During Training



(B) Loss During Training

FIGURE 5.5: Predicting the Next Task Model - Training Metrics obtained with MLFlow

The analysis of the performance curves shown above reveals a behavior that is consistent and in line with the expected patterns of a successful learning process. The accuracy curve shows a clear and monotonic progression throughout the iterations. The model starts from an initial accuracy of around 70%, quickly reaching values above 90% after just a few epochs. From the 6th epoch onwards, there is an asymptotic approach to the performance plateau, with progressive marginal improvements until stabilizing at around 95%. This pattern is indicative of a good generalization capacity, suggesting that the model is learning to effectively capture the sequential dependencies of the data without overfitting.
At the same time, the loss curve shows a rapid and sharp decrease in the first iterations, going

from a value above 1.2 to values below 0.4 in just a few epochs. Thereafter, the decline becomes more gradual, stabilizing at around 0.18. This gently decreasing trajectory of the loss function reinforces the conclusion that the model is effectively optimizing its parameters, reducing the predictive error without signs of instability. The convergent behaviour of both curves, a steady increase in accuracy and a consistent decrease in loss, shows that the training process was successful, with the cost function being steadily minimized and the predictive ability increasing with each iteration.

To validate the model's performance in practical contexts, a test code was developed that simulated real life situations. This code used the following steps:

1. **Randomization of Dataset Rows:** Several rows from the dataset were randomly selected to create test cases.
2. **Random Splitting of Sequences:** The selected sequences were split into different positions, also randomly, generating input and expected output subsequences for the tests.
3. **Testing the Model:** The input sequences were fed into the model and the predictions made were compared with the expected symptoms.

This testing method made it possible to assess the model's ability to generalize to different flows and sequence lengths, ensuring the system's robustness. In these randomization tests, the model was successful in 1652 cases and failed in 348 cases, which shows a high success rate in what would be real contexts.

This case study unequivocally validates the integration of AI models into the CLARA architecture, demonstrating how it is possible, from real data collected in the field, to develop models with practical use and incorporate them into the system's functional paths. The use of MLflow proved essential to guarantee the traceability, reproducibility and maintenance of models over time, allowing the Model layer to establish itself as a central point for model governance within the LH infrastructure. The success of this case study reinforces CLARA's vision of an intelligent, modular solution prepared to evolve with the support of AI models applied to heterogeneous data.

5.3 Conversational intelligence framework: a modular query and assistance system case study

This third case study aims to validate the technical feasibility of the four functional paths that make up the conversational basis of the CLARA architecture, as outlined in Chapter 4. These paths represent different types of questions that users can ask the system, and correspond to four different forms of processing and response, each supported by a specialized module. The main objective was to ensure that each individual path was functional, coherent and capable of generating appropriate responses within its domain. The four paths tested were:

1. Generation of queries
Using the Gemini API, the ability to convert natural language questions into SQL queries was tested. These queries were generated based on a JSON file containing the structure of the PostgreSQL database and a .txt file with descriptions of the contents of the schemas. At this stage, the queries generated were being produced in the PostgreSQL dialect and validated manually, being tested directly on the original database in order to confirm the syntactic and semantic correctness of the instructions. This approach made it possible to validate the Gemini model's potential for understanding technical questions and generating queries appropriate to the existing data structure.

2. Functional clarification based on the Navia manual
This route was supported by the Mistral API, which was provided with the Navia.docx manual document. This manual, which is functional in nature, describes the general operation of the platform, the menus available and the actions possible in each component. Since the document contains knowledge represented in different ways, such as tables, images, etc. the use of a multimodal tool was essential. The questions posed were of the type "Where can I consult requests?" or "How do I create a team?", and the Mistral model was able to respond in a clear and reasoned manner based on the content of the document, demonstrating a good ability to reason about unstructured technical documentation.

3. Predicting the next task based on historical symptoms
This course tested the LSTM model developed to predict the next symptom of an intervention or request. Symptom sequences extracted from real data were used as input, and the prediction results were analyzed manually to check whether the responses were consistent with the context. The integration between the model, the dataset and the test interface was successful, validating this path as an autonomous predictive component.

4. General conversation in the context of Navia
The behavior of the local LLaMA 3.3 model, configured to answer generic questions but restricted to Navia's functional domain, was also tested. The questions tested included "What is your name?" or "What is this assistant for?", and the model responded in an appropriate and controlled manner, without leaving the scope of the system. This behavior validated the conversational fallback path as a robustness and interaction control mechanism.

These tests showed that the four central paths of the CLARA system could function independently, each responding appropriately to its type of input. Although there is not yet a unified classification and routing flow, this phase was essential to ensure that each module, in isolation, was ready to be integrated into a more complete architecture. The results obtained in this phase served as the basis for the system's subsequent evolution, where all the paths were interconnected, allowing for a fluid and automated conversational experience.

5.4 End-to-end orchestration and integration: a complete system evaluation case study

The fourth and final case study developed as part of this dissertation aimed to demonstrate the complete integration of all the components of the CLARA architecture, validating their functional articulation through a centralized API and a dedicated user interface. This phase represented the culmination of the development process, in which the different modules built throughout the work were orchestrated in a cohesive manner, allowing the solution to function end-to-end, from receiving questions in natural language to generating informative and secure contextual responses.

To enable the different modules developed during case study 3 to be orchestrated, a large language model (LLM) LLama3.3 was used to classify the intention of the user's question. This LLM was developed with the aim of semantically analyzing the user's question and determining, based on its content, which functional path is most suitable for resolving it. Possible categories include: consultation of the database, functional clarification based on the Navia manual, prediction of the next intervention and generic dialog in the context of Navia. This classification, carried out automatically, proved to be highly reliable even in situations of linguistic ambiguity or multilingualism, allowing each question to be correctly routed to the respective API module. To test whether the classification mechanism would be effective across a variety of scenarios and languages, a dataset of 50 questions was manually constructed. These questions were written by the author with the deliberate intention of simulating different types of user intent, taking into

account variations in phrasing and multilingual input. Although this dataset does not cover the full spectrum of linguistic styles such as aggressive, overly formal, or particularly polite formulations for example, it still provides a representative and diverse set of realistic queries. Each question was associated with its correct category and then individually submitted to the LLM. The predicted categories were compared with the predefined ground truth, and the results, as shown in Table 5.2, were highly promising, demonstrating the model's robustness even in the presence of ambiguity and subtle variations in expression.

Table 5.2: User Intention Classification Tests

Nº of Questions	Correctly Classified	Accuracy
50	50	100%

After classification, the path corresponding to the nature of the question begins. In the context of queries to the database, a fundamental structural change was implemented: unlike the previous phases of development, in which the query was generated and tested manually on the PostgreSQL database, in this phase the execution was carried out directly on the LH infrastructure, more specifically on the structured Bronze layer, where views were previously created in Delta Lake to allow them to be queried. To do this, the query generated by Gemini was cleaned, due to the presence of unwanted characters, and it was then converted using the sqlglot library, whose transpose function allows translation between different SQL dialects automatically and robustly, guaranteeing the portability of instructions between different systems without the need for manual rewriting. This function was necessary in order to convert the query into the format needed to run it in the views created in LH. This transformation process is presented in Figure 5.6.

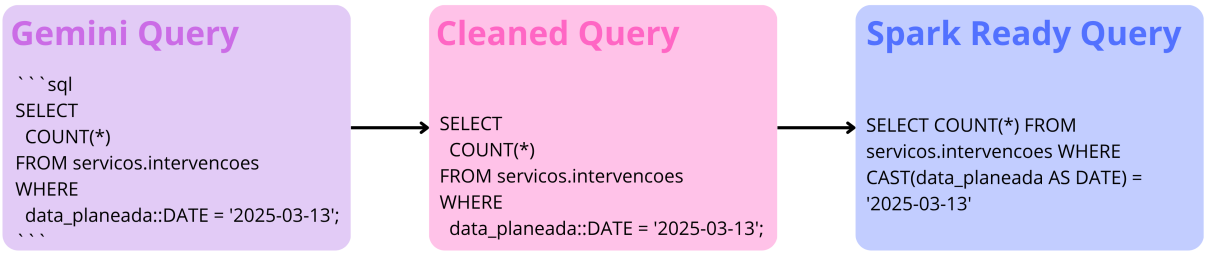


Figure 5.6: Query transformation Process

Before the query is executed, the permissions verification module, introduced in this case study is activated, specifically developed to ensure that users only access information for which they have explicit authorization. This module performs a rigorous compliance check between the requested operation (e.g. read or write), the tables involved in the query and the access rights defined for the user, which result from Navia's transaction-based permissions structure. A series of tests were carried out in order to attest this module ability to comprehend the user's permissions. The module showed a 100% accuracy by blocking all the querys that shouldn't be executed and allowing all of the ones the user had permissions to do. Some examples of these tests are presented in table 5.3. Once the query was validated and executed, the result is forwarded to a new LLM LLaMA 3.3, configured to produce a complete answer based on the question and the result of the query.

All the components described have been integrated into a modular and extensible API, which acts as the core orchestrator of the solution. This API, designed from scratch, includes the classification logic, the modules corresponding to each functional path, the validation of permissions, the conversion of queries and the generation of the final response. For testing and

Table 5.3: User Permission Tool Test

User	Action	Module	Required	Actual	API Behaviour	Expected Behaviour
user1	SELECT	Variables	View	Delete, Create, Modify, View	Allowed	Query executed
user2	DELETE	Placard	Delete	Delete, Create, Modify, View	Allowed	Query executed
user3	UPDATE	Registers	Modify	Delete, Create, Modify, View	Allowed	Query executed
user4	UPDATE	Indicators	Modify	View	Blocked	Query blocked
user5	DELETE	Flows	Delete	None	Blocked	Query blocked
user6	DELETE	Admin	Delete	Create, View	Blocked	Query blocked

functional demonstration purposes, a simple HTML interface was also developed which simulates the real behavior of the system in a controlled environment. This interface made it possible to validate the robustness of the architecture, test the end-to-end flows and prove the logical separation between the presentation layer and the solution's internal components.

This case study validates CLARA's ability to combine multiple semantic classification, permission checking, query execution, response generation and user interface modules into a single orchestrated, functional and adaptable flow. By uniting all the components in a testable operating system, it is confirmed that the architecture developed is not just theoretical, but feasible, efficient and ready for integration into real production environments.

Chapter 6

Conclusion

This chapter presents the main conclusions of the dissertation and outlines future work directions. Firstly, it summarizes the main outcomes achieved, relating each objective to the corresponding research questions and demonstrating how the proposed solution was validated through empirical case studies. Then, it discusses potential areas for improvement and expansion of the system, highlighting both technical and scientific opportunities. Together, these sections provide a comprehensive synthesis of the dissertation's contributions and its potential for continued development.

6.1 Main Conclusions

The aim of this dissertation was to explore the potential of integrating modern data architectures and artificial intelligence (AI) systems, applying this knowledge to the specific context of infrastructure management in the water sector. Through the conception of the CLARA solution — Conversational Lakehouse Architecture supported by Real-time AI — it was possible to develop, test, and validate a modular architecture composed of four main layers: data infrastructure, AI models, dialogue engine, and integration layer. With the completion of the work described in this dissertation, it is possible to state that all the defined objectives have been achieved and all research questions have been properly answered.

Subquestion SQ1 — What are the main differences between data lakes, data warehouses, and data lakehouses, and how do these infrastructures influence data analysis processes and the application of artificial intelligence? — was addressed through a detailed analysis of the main data management paradigms, presented in Section 2.1. This comparative analysis identified the main benefits and limitations of data warehouse (DW), data lake (DL), and data lakehouse (LH) architectures, leading to the conclusion that the LH model represents the most balanced approach to the challenges of the Big Data era. The LH combines the structure, governance, and organization of DWs with the scalability, flexibility, and support for data heterogeneity typical of DLs, making it particularly suitable for the integration and operationalization of AI-based solutions. This reflection supported the choice of the base architecture for the CLARA solution and allowed the achievement of objective 1 (O1 - Explore and evaluate different modern data infrastructures - data lake, data warehouse, and data lakehouse - to identify their characteristics, benefits, and limitations in supporting advanced analyses and the application of artificial intelligence).

Subquestion SQ2 — What are the challenges and best practices in designing and implementing modern data infrastructures to integrate data from relational and non-relational sources in order to support AI technologies? — was answered based on empirical developments conducted in Case Studies 1 and 2, described in Sections 5.1 and 5.2, respectively. In Case Study 1, a Delta Lake-based data infrastructure was designed and implemented, with Bronze, Silver, Gold,

and Models layers, which integrated relational data (235 GB of history in PostgreSQL) and unstructured data (6241 field images). The ingestion and transformation pipeline developed, in accordance with the ELT approach, not only preserved the structure and metadata of the original database but also achieved significant gains in storage optimization with a reduction to less than 5 GB after cleaning and restructuring. These developments achieve objectives 2 (O2 - Design and implement a modern data infrastructure solution to store, organize, and integrate data from relational and non-relational sources, ensuring scalability, flexibility, and compatibility with artificial intelligence technologies) and 3 (O3 - Develop an ELT pipeline that operates seamlessly between relational databases and modern data infrastructures, enabling ingestion and organization of data in different formats, with a focus on reuse for analysis). In Case Study 2, AI models were integrated into the LH's "Models" layer, with support from the MLFlow tool, which ensures training traceability, model versioning, and metrics monitoring. This environment enabled the training of a sequential predictive model to anticipate maintenance symptoms and the development of an OCR model based on YOLOv8-OBB for automatic validation of meter readings. The integration of these models proved that the developed infrastructure is effectively prepared to support and enhance AI applications, validating objective 4 (O4 - Develop and test artificial intelligence models to validate their integration and execution within the data infrastructure) and providing a complete answer to SQ2.

Subquestion SQ3 — What are the capabilities and limitations of virtual assistants for analyzing data in real-time through conversational interfaces? — was addressed both from a theoretical point of view, based on the literature review in Section 2.4, and in a practical way, through Case Study 3 in Section 5.3. The state-of-the-art analysis of conversational assistants, including prompt engineering techniques, retrieval-augmented generation strategies, and fallback mechanisms, supported the technical decisions made in developing the CLARA solution's virtual assistant. On a practical level, four distinct instances of large language models (LLMs) were implemented and tested, each with a specific role: SQL query generation (via Gemini), technical documentation interpretation (via Mistral), sequential prediction (via internal model), and general conversation in the context of Navia (via local LLaMA). These components were integrated into a multilingual pipeline with automatic language detection and adaptive flow based on user intent. The tests demonstrated that it is possible to build a reliable, fluid, and informative virtual assistant capable of interacting with structured data, technical documents, and AI models in natural language. This work allowed the achievement of objective 5 (O5 - Develop and test a virtual assistant capable of using different large language models adapted to different purposes, including information retrieval, direct access to artificial intelligence models, clarification of questions based on technical documentation, and a conversational fallback mechanism restricted to a specific domain), demonstrating the current maturity of these technologies and responding affirmatively to SQ3.

Subquestion SQ4 — How can virtual assistants manage multiple large language models with different purposes, and how can real-time integration with business data infrastructures support this orchestration effectively? — was explored and validated in Case Study 4, described in Section 5.4. This case aimed to orchestrate the different flows developed in O5, integrating them with the data infrastructure built. A modular API was designed to allow the coordinated management of user intent classification, permission verification, query generation and execution, AI model invocation, and multilingual response. This orchestration mechanism proved to be effective, modular, and extensible, enabling cooperation between specialized components to generate the best possible response in real time. The fluid integration between the virtual assistant and the data infrastructure ensures the consistency and robustness of the system in real operational scenarios, fulfilling objective 6 (O6 - Integrate the virtual assistant with the modern data infrastructure, using methods that guarantee consistent data retrieval and consultation, without jeopardizing the robustness of responses) and fully responding to SQ4.

Since all research sub-questions (SQ1 to SQ4) were answered based on concrete developments, validated through case studies, and aligned with the respective operational objectives, the validation process for the main research question RQ1 — How can modern data management infrastructures be used to support advanced analyses and artificial intelligence applications, while providing an intelligent virtual assistant for business aid and real-time analyses? This final validation was reinforced by testing and demonstrating the CLARA solution in a real context, with operational data from the water sector, including the development of a user interface that allowed real flows to be simulated and the overall behavior of the solution to be validated. Thus, objective 7 (O7 - Test and validate the integrated solution - modern data infrastructure, AI models, and virtual assistant - in real scenarios in the water sector, assessing its applicability in analyzing data and supporting business decision-making) was also achieved, confirming the applicability of CLARA as an integrated, scalable, and replicable solution for decision support and intelligent real-time data analysis.

Thus, it can be concluded that the solution developed throughout this dissertation not only provides a clear and well-founded answer to the proposed research question, but also constitutes a relevant scientific and technological contribution by demonstrating in practice how modern data architectures, AI models, and conversational interfaces can be integrated synergistically to solve complex challenges in a critical sector such as water management.

6.2 Future Work

Despite the promising results achieved, there are several directions that can be explored in future work, both to deepen the technical robustness of the solution and to expand its practical and scientific impact.

One of the most immediate steps will be to integrate the conversational interface directly into the Navia software, replacing the HTML test application with a fully functional production interface. This process will require collaboration with the company's product management team and should follow internal feature prioritization and planning processes. In addition, it will be essential to conduct tests with real users in a production environment in order to evaluate the user experience, the usefulness of the responses generated, and the effectiveness of predictions and automatisms.

From an architectural point of view, there are opportunities to strengthen the adaptive reasoning component. The current classification by categories may, in the future, be replaced or complemented by more flexible and contextually sensitive mechanisms, such as multi-agent architectures or dynamic decision flows based on continuous learning. At the same time, the generalization of the models developed, namely OCR and predictive, will require the collection of additional data and its revalidation in other operational contexts, thus ensuring its applicability to different Navia customers.

Although the system already incorporates semantic matching mechanisms based on embeddings, the integration of ontology-based mechanisms should be explored. This approach would allow terms used by users to be mapped to formal concepts in the database, overcoming linguistic and semantic disparities that currently limit the automatic interpretation of queries. The use of equivalence dictionaries, controlled vocabularies, or even automatic alignment with domain ontologies could substantially improve the accuracy and flexibility of the system.

The limitation of context in local LLMs led to the adoption of the Gemini API for query generation, given the need to process the JSON file of the database structure. Future iterations

may explore the use of models with greater contextual capacity or more optimized retrieval-augmented generation (RAG) strategies, with the aim of ensuring greater accuracy and coverage, even with extensive contexts or complex documents.

It is also important to mention two objectives initially outlined for this dissertation but which were not implemented due to time constraints. The first relates to the development of mechanisms that allow the virtual assistant to identify changes in stored data, automatically adapting its behavior to these changes. This type of functionality is essential to ensure that the responses generated remain consistent, even in dynamic data environments. The second objective to be achieved involves the creation of a system for validating the responses generated by the virtual assistant, capable of monitoring the quality of responses, detecting possible hallucinations, and ensuring the reliability of the information made available to end users.

Finally, given the practical relevance of the developed solution and the solid results obtained, a technical article based on this dissertation is currently being prepared for presentation at ENEG 2025 — one of the most important national events in the water and sanitation sector. A summary of the article has already been submitted for acceptance. The goal is to share the CLARA solution with professionals, researchers, and decision-makers in the field, contributing to the dissemination of innovative approaches that combine artificial intelligence with modern data architectures. This line of work is also expected to continue through the supervision of future dissertations within the scope of the STREAM and AQUASHIELD projects, reinforcing the collaboration between the company and the University of Minho and paving the way for the development of a reference hub in AI for critical infrastructure management.

References

Adamopoulou, Eleni and Lefteris Moussiades (2020). "An Overview of Chatbot Technology". en. In: *Artificial Intelligence Applications and Innovations*. Ed. by Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis. Cham: Springer International Publishing, pp. 373–383. ISBN: 978-3-030-49186-4. DOI: 10.1007/978-3-030-49186-4_31.

Affolter, Katrin, Kurt Stockinger, and Abraham Bernstein (Oct. 2019). "A comparative survey of recent natural language interfaces for databases". en. In: *The VLDB Journal* 28.5, pp. 793–819. ISSN: 0949-877X. DOI: 10.1007/s00778-019-00567-8. URL: https://doi.org/10.1007/s00778-019-00567-8 (visited on 06/20/2025).

Aggarwal, Jyoti (May 2025). "Building an AI-Ready Data Strategy Using Lakehouse Technology". en. In: *Journal of Computer Science and Technology Studies* 7.3. Number: 3, pp. 663–676. ISSN: 2709-104X. DOI: 10.32996/jcsts.2025.7.3.76. URL: https://al-kindipublishers.org/index.php/jcsts/article/view/9422 (visited on 06/23/2025).

Ali, Nafez (2023). "Influence of Data-Driven Digital Marketing Strategies on Organizational Marketing Performance: Mediating Role of IT Infrastructure". en. In: *Cutting-Edge Business Technologies in the Big Data Era*. Ed. by Saad G. Yaseen. Cham: Springer Nature Switzerland, pp. 337–347. ISBN: 978-3-031-42463-2. DOI: 10.1007/978-3-031-42463-2_31.

Al-Amin, Md et al. (Feb. 2024). *History of generative Artificial Intelligence (AI) chatbots: past, present, and future development*. arXiv:2402.05122 [cs]. DOI: 10.48550/arXiv.2402.05122. URL: http://arxiv.org/abs/2402.05122 (visited on 12/29/2024).

Apache Spark (2025). *Apache Spark™ - Unified Engine for large-scale data analytics*. URL: https://spark.apache.org/ (visited on 06/16/2025).

Armbrust Michael et al. (2021). "Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics". en. In.

Aryan Gupta and Meera Patel (Sept. 2021). "Designing Resilient AI-Driven Data Pipelines for Real-Time Analytics in High-Volume Environments". en. In: *Innovative Computer Sciences Journal* 7.1. Number: 1. ISSN: 3007-6471. URL: https://innovatesci-publishers.com/index.php/ICSJ/article/view/412 (visited on 12/29/2024).

Associação Portuguesa de Distribuição e Drenagem de Águas (July 2024). *Inteligência Artificial, Inovação & Pessoas - Os novos horizontes dos serviços de águas*. URL: https://www.apda.pt/noticia/5105/inteligencia-artificial-inovacao-pessoas-os-novos-horizontes-dos-servicos-de-aguas.

Berntsson Svensson, Richard and Maryam Taghavianfar (2020). "Toward Becoming a Data-Driven Organization: Challenges and Benefits". en. In: *Research Challenges in Information Science*. Ed. by Fabiano Dalpiaz, Jelena Zdravkovic, and Pericles Loucopoulos. Cham: Springer International Publishing, pp. 3–19. ISBN: 978-3-030-50316-1. DOI: 10.1007/978-3-030-50316-1_1.

Bilal Khan et al. (2024). "An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing". en. In: *Journal on Big Data* 6. Publisher: Tech Science Press, pp. 1–20. ISSN: 2579-0048, 2579-0056. DOI: 10.32604/jbd.2023.046223. URL: https://www.techscience.com/jbd/v6n1/55252 (visited on 12/29/2024).

Chad, Felix (2025). "Semantic Interoperability in Heterogeneous Data Integration". en. In.

Chuangtao Ma and Bálint Molnár (2022). "Ontology Learning from Relational Database: Opportunities for Semantic Information Integration". en. In: *Vietnam Journal of Computer Science* 9.1. URL: https://www.worldscientific.com/doi/epdf/10.1142/S219688882150024X (visited on 06/09/2025).

Corrêa, Nicholas Kluge et al. (Oct. 2023). "Worldwide AI ethics: A review of 200 guidelines and recommendations for AI governance". English. In: *Patterns* 4.10. Publisher: Elsevier. ISSN: 2666-3899. DOI: 10.1016/j.patter.2023.100857. URL: https://www.cell.com/patterns/abstract/S2666-3899(23)00241-6 (visited on 06/24/2025).

Culot, Giovanna et al. (Mar. 2021). "The ISO/IEC 27001 information security management standard: literature review and theory-based research agenda". en. In: *The TQM Journal* 33.7. Publisher: Emerald Publishing Limited, pp. 76–105. ISSN: 1754-2731. DOI: 10.1108/TQM-09-2020-0202. URL: https://www.emerald.com/insight/content/doi/10.1108/tqm-09-2020-0202/full/html (visited on 06/24/2025).

Databricks (Oct. 2023). *Databricks: Leading Data and AI Solutions for Enterprises*. en-US. URL: https://www.databricks.com/ (visited on 06/16/2025).

de Assis Vilela Flávio et al. (May 2023). "A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments". In: *Heliyon* 9.5, e15728. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2023.e15728. URL: https://www.sciencedirect.com/science/article/pii/S2405844023029353 (visited on 12/29/2024).

De Pasquale, Patrizia et al. (2022). "Is the european union thinking about a charter of (fundamental) digital rights?" In: *European review of digital administration & law* 1, pp. 125–129.

Dulam, Naresh and Karthik Allam (Aug. 2024). "Data Lakehouses: Merging Real-Time Analytics and Big Data Processing". en. In: *Australian Journal of Machine Learning Research & Applications* 4.2. Number: 2, pp. 170–193. ISSN: 2457-0982. URL: https://sydneyacademics.com/index.php/ajmlra/article/view/213 (visited on 12/25/2024).

Dulam, Naresh, Karthik Allam, and Kishore Reddy Gade (Oct. 2021). "Data Lakehouse Architecture: Merging Data Lakes and Data Warehouses". en. In: *Journal of AI-Assisted Scientific Discovery* 1.2. Number: 2, pp. 282–303. ISSN: 2394-3750. URL: https://scienceacadpress.com/index.php/jaasd/article/view/226 (visited on 12/23/2024).

European Commission (2025). *Princípios e Direitos Digitais Europeus | Shaping Europe's digital future*. pt-pt. URL: https://digital-strategy.ec.europa.eu/pt/policies/digital-principles (visited on 06/24/2025).

European Union (2018). *Regulamento geral sobre a proteção de dados (RGPD)*. URL: https://www.pgdlisboa.pt/leis/lei_mostra_articulado.php?nid=2961&tabela=leis.

Franciscatto, Maria Helena et al. (2022). "Querying multidimensional big data through a chatbot system". In: *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. SAC '22. New York, NY, USA: Association for Computing Machinery, pp. 381–384. ISBN: 978-1-4503-8713-2. DOI: 10.1145/3477314.3507692. URL: https://dl.acm.org/doi/10.1145/3477314.3507692 (visited on 12/23/2024).

Frederik Möller et al. (2021). "What is a Data-Driven Organization?" en. In: *Americas Conference on Information Systems* 27.

Gade, Kishore Reddy (Jan. 2022). "Data Lakehouses: Combining the Best of Data Lakes and Data Warehouses". en. In: *Journal of Computational Innovation* 2.1. Number: 1. URL: https://researchworkx.com/index.php/jci/article/view/3 (visited on 12/25/2024).

Garrido-Baserba, Manel et al. (Apr. 2020). "The Fourth-Revolution in the Water Sector Encounters the Digital Revolution". In: *Environmental Science & Technology* 54.8. Publisher: American Chemical Society, pp. 4698–4705. ISSN: 0013-936X. DOI: 10.1021/acs.est.9b04251. URL: https://doi.org/10.1021/acs.est.9b04251 (visited on 12/18/2024).

George, Jobin (Oct. 2022). *Optimizing hybrid and multi-cloud architectures for real-time data streaming and analytics: Strategies for scalability and integration*. en. SSRN Scholarly Paper. Rochester, NY. URL: https://papers.ssrn.com/abstract=4963389 (visited on 12/29/2024).

Google (2025). *Modelos do Gemini | Gemini API*. pt-BR-x-mtfrom-en. URL: https://ai.google.dev/gemini-api/docs/models?hl=pt-br (visited on 06/16/2025).

Guo, Xuan et al. (May 2025). "A Natural Language-Based Automatic Identification System Trajectory Query Approach Using Large Language Models". en. In: *ISPRS International Journal of Geo-Information* 14.5. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 204. ISSN: 2220-9964. DOI: 10.3390/ijgi14050204. URL: https://www.mdpi.com/2220-9964/14/5/204 (visited on 06/23/2025).

Gupta, Nikhil and Jason Yip (2024). "Generative AI with Databricks". en. In: *Databricks Data Intelligence Platform: Unlocking the GenAI Revolution*. Ed. by Nikhil Gupta and Jason Yip. Berkeley, CA: Apress, pp. 219–253. ISBN: 979-8-8688-0444-1. DOI: 10.1007/979-8-8688-0444-1_10. URL: https://doi.org/10.1007/979-8-8688-0444-1_10 (visited on 12/23/2024).

Hai, Rihan et al. (Dec. 2023). "Data Lakes: A Survey of Functions and Systems". In: *IEEE Transactions on Knowledge and Data Engineering* 35.12. Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 12571–12590. ISSN: 1558-2191. DOI: 10.1109/TKDE.2023.3270101. URL: https://ieeexplore.ieee.org/abstract/document/10107808 (visited on 12/16/2024).

Hambardzumyan, Sasun et al. (Dec. 2022). *Deep Lake: a Lakehouse for Deep Learning*. arXiv:2209.10785 [cs]. DOI: 10.48550/arXiv.2209.10785. URL: http://arxiv.org/abs/2209.10785 (visited on 12/23/2024).

Harby, Ahmed A. and Farhana Zulkernine (Jan. 2025). "Data Lakehouse: A survey and experimental study". In: *Information Systems* 127, p. 102460. ISSN: 0306-4379. DOI: 10.1016/j.is.2024.102460. URL: https://www.sciencedirect.com/science/article/pii/S0306437924001182 (visited on 12/09/2024).

Hees, Levi van (2024). "Exploring the utility of large language models for achieving semantic interoperability in data ecosystems". en. PhD thesis. Universiteit Utrecht.

Hermanus, Davy Ronald et al. (Sept. 2024). "Robust SmartCityAI Lakehouse: IKN (New Capital City of Indonesia) Case Study". In: *2024 International Conference on ICT for Smart Society (ICISS)*, pp. 1–7. DOI: 10.1109/ICISS62896.2024.10751179. URL: https://ieeexplore.ieee.org/abstract/document/10751179 (visited on 12/29/2024).

Hiran, Kamal Kant et al. (2021). *Machine learning: Master supervised and unsupervised learning algorithms with real examples (english edition)*. BPB Publications.

Hoofnagle, Chris Jay, Bart van der Sloot, and Frederik Zuiderveen Borgesius (Jan. 2019). "The European Union general data protection regulation: what it is and what it means*". In: *Information & Communications Technology Law* 28.1. Publisher: Routledge _eprint: https://doi.org/10.1080/13600834.2019.1573501, pp. 65–98. ISSN: 1360-0834. DOI: 10.1080/13600834.2019.1573501. URL: https://doi.org/10.1080/13600834.2019.1573501 (visited on 06/24/2025).

Houlsby, Neil et al. (May 2019). "Parameter-Efficient Transfer Learning for NLP". en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, pp. 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html (visited on 06/23/2025).

ISO (2025). *ISO/IEC 27001:2022*. en. URL: https://www.iso.org/standard/27001 (visited on 06/24/2025).

JetBrains (2025). *DataGrip: The Cross-Platform IDE for Databases & SQL by JetBrains*. en. URL: https://www.jetbrains.com/datagrip/ (visited on 06/24/2025).

Jha, Arnav, Naman Anand, and H. Karthikeyan (2025). "Conversion of natural language text to SQL queries using generative AI". In: *Hybrid and Advanced Technologies*. Num Pages: 8. CRC Press. ISBN: 978-1-003-55913-9.

Jiang, Jinhao et al. (Oct. 2023). *StructGPT: A General Framework for Large Language Model to Reason over Structured Data*. arXiv:2305.09645 [cs]. DOI: 10.48550/arXiv.2305.09645. URL: http://arxiv.org/abs/2305.09645 (visited on 12/29/2024).

Justin Levandoski et al. (June 2024). "BigLake: BigQuery's Evolution toward a Multi-Cloud Lakehouse". In: *Companion of the 2024 International Conference on Management of Data*. SIGMOD/PODS '24. New York, NY, USA: Association for Computing Machinery, pp. 334–346. ISBN: 979-8-4007-0422-2. DOI: 10.1145/3626246.3653388. URL: https://dl.acm.org/doi/10.1145/3626246.3653388 (visited on 12/29/2024).

Kamyab, Hesam et al. (Dec. 2023). "The latest innovative avenues for the utilization of artificial Intelligence and big data analytics in water resource management". In: *Results in Engineering* 20, p. 101566. ISSN: 2590-1230. DOI: 10.1016/j.rineng.2023.101566. URL: https://www.sciencedirect.com/science/article/pii/S259012302300693X (visited on 12/17/2024).

Kannan Nova (June 2023). "AI-Enabled Water Management Systems: An Analysis of System Components and Interdependencies for Water Conservation". en. In: *Eigenpub Review of Science and Technology* 7.1. Number: 1, pp. 105–124. URL: https://studies.eigenpub.com/index.php/erst/article/view/12 (visited on 12/29/2024).

Kauffmann, Jacob et al. (Mar. 2025). "Explainable AI reveals Clever Hans effects in unsupervised learning models". en. In: *Nature Machine Intelligence* 7.3. Publisher: Nature Publishing Group, pp. 412–422. ISSN: 2522-5839. DOI: 10.1038/s42256-025-01000-2. URL: https://www.nature.com/articles/s42256-025-01000-2 (visited on 06/22/2025).

Kavaz, Ecem, Anna Puig, and Inmaculada Rodríguez (Jan. 2023). "Chatbot-Based Natural Language Interfaces for Data Visualisation: A Scoping Review". en. In: *Applied Sciences* 13.12. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 7025. ISSN: 2076-3417. DOI: 10.3390/app13127025. URL: https://www.mdpi.com/2076-3417/13/12/7025 (visited on 06/20/2025).

Kimiya Keyvan and Jimmy Xiangji Huang (2022). "How to Approach Ambiguous Queries in Conversational Search: A Survey of Techniques, Approaches, Tools, and Challenges". In: *ACM Comput. Surv.* 55.6, 129:1–129:40. ISSN: 0360-0300. DOI: 10.1145/3534965. URL: https://dl.acm.org/doi/10.1145/3534965 (visited on 12/29/2024).

Kitsios, Fotis, Elpiniki Chatzidimitriou, and Maria Kamariotou (Jan. 2023). "The ISO/IEC 27001 Information Security Management Standard: How to Extract Value from Data in the IT Sector". en. In: *Sustainability* 15.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 5828. ISSN: 2071-1050. DOI: 10.3390/su15075828. URL: https://www.mdpi.com/2071-1050/15/7/5828 (visited on 06/24/2025).

Kumar, Ajay Prasantha (2025). "Semantic Reconciliation Techniques in Multidomain MDM Frameworks for Heterogeneous Data Sources". en. In.

Lake, Delta (2025). *Build lakehouses with delta lake*. URL: https://delta.io/.

Le, Ngan et al. (Apr. 2022). "Deep reinforcement learning in computer vision: a comprehensive survey". en. In: *Artificial Intelligence Review* 55.4, pp. 2733–2819. ISSN: 1573-7462. DOI: 10.1007/s10462-021-10061-9. URL: https://doi.org/10.1007/s10462-021-10061-9 (visited on 06/22/2025).

Li, Tianxu et al. (2022). "Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey". In: *IEEE Communications Surveys & Tutorials* 24.2, pp. 1240–1279. ISSN: 1553-877X. DOI: 10.1109/COMST.2022.3160697. URL: https://ieeexplore.ieee.org/abstract/document/9738819 (visited on 06/22/2025).

Li Ruichen, Choy Murphy, and Ma Nang Laik (Sept. 2022). "A cloud data lakehouse-based AI diagnostic solution for small and medium-sized health facilities". en. In: vol. 12. ISSN: 2169-8767 Issue: 6. IEOM Society. ISBN: 978-1-7923-9162-0. DOI: 10.46254/AP03.20220190.

URL: https://index.ieomsociety.org/index.cfm/article/view/ID/11718 (visited on 12/29/2024).

Lin, Xinru and Luyang Li (Mar. 2025). *Implicit Bias in LLMs: A Survey*. arXiv:2503.02776 [cs]. DOI: 10.48550/arXiv.2503.02776. URL: http://arxiv.org/abs/2503.02776 (visited on 06/24/2025).

Liu, Kaihong, William R. Hogan, and Rebecca S. Crowley (Feb. 2011). "Natural Language Processing methods and systems for biomedical ontology learning". In: *Journal of Biomedical Informatics*. Ontologies for Clinical and Translational Research 44.1, pp. 163–179. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2010.07.006. URL: https://www.sciencedirect.com/science/article/pii/S153204641000105X (visited on 06/25/2025).

Mahdi, Nibras M. et al. (Oct. 2024). "Leak detection and localization in water distribution systems using advanced feature analysis and an Artificial Neural Network". In: *Desalination and Water Treatment* 320, p. 100685. ISSN: 1944-3986. DOI: 10.1016/j.dwt.2024.100685. URL: https://www.sciencedirect.com/science/article/pii/S1944398624107059 (visited on 12/29/2024).

Majeed, Abdul and Seong Oun Hwang (Jan. 2024). "A Data-Centric AI Paradigm for Socio-Industrial and Global Challenges". en. In: *Electronics* 13.11, p. 2156. ISSN: 2079-9292. DOI: 10.3390/electronics13112156. URL: https://www.mdpi.com/2079-9292/13/11/2156 (visited on 06/11/2025).

Marcelo Trylesinski (2025). *Uvicorn*. URL: https://www.uvicorn.org/ (visited on 06/17/2025).

Marcu Ovidiu-Cristian and Bouvry Pascal (Sept. 2024). *Big data stream processing*. Tech. rep. University of Luxembourg. URL: https://hal.science/hal-04687320 (visited on 12/29/2024).

Martín, Cristian et al. (Jan. 2022). "Kafka-ML: Connecting the data stream with ML/AI frameworks". In: *Future Generation Computer Systems* 126, pp. 15–33. ISSN: 0167-739X. DOI: 10.1016/j.future.2021.07.037. URL: https://www.sciencedirect.com/science/article/pii/S0167739X21002995 (visited on 12/29/2024).

Matthew Lowe, Ruwen Qin, and Xinwei Mao (2024). *A Review on Machine Learning, Artificial Intelligence, and Smart Technology in Water Treatment and Monitoring*. URL: https://www.mdpi.com/2073-4441/14/9/1384 (visited on 12/29/2024).

Mazumdar, Dipankar, Jason Hughes, and J. B. Onofre (Oct. 2023). *The Data Lakehouse: Data Warehousing and More*. arXiv:2310.08697 [cs]. DOI: 10.48550/arXiv.2310.08697. URL: http://arxiv.org/abs/2310.08697 (visited on 12/18/2024).

Meta (2025). *Llama 3.3 | Model Cards and Prompt formats*. en. URL: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/ (visited on 06/16/2025).

Michal Danilk (2025). *langdetect: Language detection library ported from Google's language-detection*. URL: https://github.com/Mimino666/langdetect (visited on 06/16/2025).

Mistral AI (2025). *Frontier AI LLMs, assistants, agents, services | Mistral AI*. en. URL: https://mistral.ai/ (visited on 06/16/2025).

MLFlow (2025). *MLflow*. en. URL: http://mlflow.org/ (visited on 06/16/2025).

Morse, Mohamed Hamdy, Al-Azhar Information, and Karam Gouda (2025). "Exploring Ontology-Driven Approaches to Data Integration: Methods and Challenges". en. In: 23.1.

Moujahid, Abdelmalik and Fadi Dornaika (May 2025). "Advanced unsupervised learning: a comprehensive overview of multi-view clustering techniques". en. In: *Artificial Intelligence Review* 58.8, p. 234. ISSN: 1573-7462. DOI: 10.1007/s10462-025-11240-8. URL: https://doi.org/10.1007/s10462-025-11240-8 (visited on 06/22/2025).

Munikoti, Sai et al. (Nov. 2024). "Challenges and Opportunities in Deep Reinforcement Learning With Graph Neural Networks: A Comprehensive Review of Algorithms and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 35.11, pp. 15051–15071. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2023.3283523. URL: https://ieeexplore.ieee.org/abstract/document/10161704 (visited on 06/22/2025).

Naeem, Muhammad et al. (2022). "Trends and Future Perspective Challenges in Big Data". en. In: *Advances in Intelligent Data Analysis and Applications*. Ed. by Jeng-Shyang Pan, Valentina Emilia Balas, and Chien-Ming Chen. Singapore: Springer, pp. 309–325. ISBN: 978-981-16-5036-9. DOI: 10.1007/978-981-16-5036-9_30.

Nathalie Janssen et al. (Dec. 2024). "The evolution of data storage architectures: examining the secure value of the Data Lakehouse". en. In: *Journal of Data, Information and Management* 6.4, pp. 309–334. ISSN: 2524-6364. DOI: 10.1007/s42488-024-00132-1. URL: https://doi.org/10.1007/s42488-024-00132-1 (visited on 01/20/2025).

NumPy community (2025). *NumPy - The fundamental package for scientific computing with Python*. manual. URL: https://numpy.org/.

Oliva, David (2023). "Readiness assessment for the Artificial Intelligence Act: with a requirements catalogue in the field of critical infrastructure". phd. Technische Universität Wien.

Ollama (2025). *Ollama*. URL: https://ollama.com (visited on 06/16/2025).

Oreščanin, Dražen and Tomislav Hlupić (Sept. 2021). "Data Lakehouse - a Novel Step in Analytics Architecture". In: *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. ISSN: 2623-8764, pp. 1242–1246. DOI: 10.23919/MIPRO52101.2021.9597091. URL: https://ieeexplore.ieee.org/abstract/document/9597091 (visited on 12/17/2024).

Ouyang, Long et al. (Dec. 2022). "Training language models to follow instructions with human feedback". en. In: *Advances in Neural Information Processing Systems* 35, pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html (visited on 06/23/2025).

Őzcan, Fatma et al. (2020). "State of the Art and Open Challenges in Natural Language Interfaces to Data". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD '20. New York, NY, USA: Association for Computing Machinery, pp. 2629–2636. ISBN: 978-1-4503-6735-6. DOI: 10.1145/3318464.3383128. URL: https://dl.acm.org/doi/10.1145/3318464.3383128 (visited on 06/20/2025).

pandas development team (2025). *pandas - python data analysis library*. manual. URL: https://pandas.pydata.org/.

Paras Jain et al. (2023). "Analyzing and comparing lakehouse storage systems." In: *CIDR*.

Pateria, Shubham et al. (June 2021). "Hierarchical Reinforcement Learning: A Comprehensive Survey". In: *ACM Comput. Surv.* 54.5, 109:1–109:35. ISSN: 0360-0300. DOI: 10.1145/3453160. URL: https://dl.acm.org/doi/10.1145/3453160 (visited on 06/22/2025).

Patil, Sujal Dilip, Rupali Atul Mahajan, and Nitin Sakhare (2024). "Advancements in Data-Centric AI Foundations, Ethics, and Emerging Technology". In: *Data-Centric Artificial Intelligence for Multidisciplinary Applications*. Chapman and Hall/CRC. ISBN: 978-1-003-46150-0.

Philip Salqvist (2024). *A comparative study of the Data Warehouse and Data Lakehouse architecture*. eng. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-345638 (visited on 01/20/2025).

Putrama, I Made and Péter Martinek (Oct. 2024). "Heterogeneous data integration: Challenges and opportunities". In: *Data in Brief* 56, p. 110853. ISSN: 2352-3409. DOI: 10.1016/j.dib.2024.110853. URL: https://www.sciencedirect.com/science/article/pii/S2352340924008175 (visited on 12/29/2024).

Python (June 2025a). *Welcome to Python.org*. en. URL: https://www.python.org/ (visited on 06/16/2025).

– (2025b). *pickle — Python object serialization*. en. URL: https://docs.python.org/3/library/pickle.html (visited on 06/16/2025).

Ramírez, Sebastián (2023). *FastAPI: Modern, fast (high-performance), web framework for python*. manual. URL: https://fastapi.tiangolo.com/.

Rane, Atharvaa et al. (Dec. 2022). "AI driven Chatbot and its Evolution". In: *2022 5th International Conference on Advances in Science and Technology (ICAST)*, pp. 170–173. DOI:

10.1109/ICAST55766.2022.10039515. URL: https://ieeexplore.ieee.org/abstract/document/10039515 (visited on 06/23/2025).

Ranjan, Rajesh, Shailja Gupta, and Surya Narayan Singh (Sept. 2024). *A Comprehensive Survey of Bias in LLMs: Current Landscape and Future Directions*. arXiv:2409.16430 [cs]. DOI: 10.48550/arXiv.2409.16430. URL: http://arxiv.org/abs/2409.16430 (visited on 06/24/2025).

Ranpara, Ripal (Mar. 2025). "A semantic and ontology-based framework for enhancing interoperability and automation in IoT systems". en. In: *Discover Internet of Things* 5.1, p. 22. ISSN: 2730-7239. DOI: 10.1007/s43926-025-00122-8. URL: https://doi.org/10.1007/s43926-025-00122-8 (visited on 06/09/2025).

Ray Saikat Sinha et al. (Jan. 2024). "Leveraging ChatGPT and Bard: What does it convey for water treatment/desalination and harvesting sectors?" In: *Desalination* 570, p. 117085. ISSN: 0011-9164. DOI: 10.1016/j.desal.2023.117085. URL: https://www.sciencedirect.com/science/article/pii/S0011916423007178 (visited on 12/28/2024).

Ruder, Sebastian et al. (June 2019). "Transfer Learning in Natural Language Processing". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Ed. by Anoop Sarkar and Michael Strube. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 15–18. DOI: 10.18653/v1/N19-5004. URL: https://aclanthology.org/N19-5004/ (visited on 06/23/2025).

Sadhu, Srestha, Ayusha Burman, and Lopa Mandal (2022). "A Systematic Survey of the Chatbot Evolution". en. In: *Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing*. Ed. by Lopa Mandal, Joao Manuel R. S. Tavares, and Valentina E. Balas. Singapore: Springer Nature, pp. 299–308. ISBN: 978-981-19-1657-1. DOI: 10.1007/978-981-19-1657-1_25.

Salim, Soja, Jayasudha J. S., and Soniya B. (Sept. 2024). "Ensuring Ethical AI: Unpacking the Significance of Risk Analysis Under the European Union's Artificial Intelligence Act". In: *2024 IEEE Region 10 Symposium (TENSYMP)*. ISSN: 2642-6102, pp. 1–6. DOI: 10.1109/TENSYMP61132.2024.10752133. URL: https://ieeexplore.ieee.org/document/10752133 (visited on 12/23/2024).

Santos, Gabriel et al. (2021). "Semantic Interoperability for Multiagent Simulation and Decision Support in Power Systems". en. In: *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection*. Ed. by Fernando De La Prieta et al. Cham: Springer International Publishing, pp. 215–226. ISBN: 978-3-030-85710-3. DOI: 10.1007/978-3-030-85710-3_18.

Schneider, Jan et al. (Apr. 2024). "The Lakehouse: State of the Art on Concepts and Technologies". en. In: *SN Computer Science* 5.5, p. 449. ISSN: 2661-8907. DOI: 10.1007/s42979-024-02737-0. URL: https://doi.org/10.1007/s42979-024-02737-0 (visited on 12/29/2024).

Sela, Lina et al. (Mar. 2025). "Making waves: The potential of generative AI in water utility operations". In: *Water Research* 272, p. 122935. ISSN: 0043-1354. DOI: 10.1016/j.watres.2024.122935. URL: https://www.sciencedirect.com/science/article/pii/S0043135424018359 (visited on 12/18/2024).

Sezgin, Anıl (June 2025). "Natural Language Interfaces for Structured Query Generation in IoD Platforms". en. In: *Drones* 9.6. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 444. ISSN: 2504-446X. DOI: 10.3390/drones9060444. URL: https://www.mdpi.com/2504-446X/9/6/444 (visited on 06/23/2025).

Shakya, Ashish Kumar, Gopinatha Pillai, and Sohom Chakrabarty (Nov. 2023). "Reinforcement learning algorithms: A brief survey". In: *Expert Systems with Applications* 231, p. 120495. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.120495. URL: https://www.sciencedirect.com/science/article/pii/S0957417423009971 (visited on 06/22/2025).

Shanahan, Murray (Jan. 2024). "Talking about Large Language Models". In: *Commun. ACM* 67.2, pp. 68–79. ISSN: 0001-0782. DOI: 10.1145/3624724. URL: https://dl.acm.org/doi/10.1145/3624724 (visited on 06/20/2025).

Sharma, Kumar, Ujjal Marjit, and Utpal Biswas (Sept. 2018). "Efficiently Processing and Storing Library Linked Data using Apache Spark and Parquet". en. In: *Information Technology and Libraries* 37.3, pp. 29–49. ISSN: 2163-5226. DOI: 10.6017/ital.v37i3.10177. URL: https://ital.corejournals.org/index.php/ital/article/view/10177 (visited on 06/16/2025).

Simitsis, Alkis, Spiros Skiadopoulos, and Panos Vassiliadis (2023). "The History, Present, and Future of ETL Technology". en. In: *DOLAP*, pp. 3–12.

Singhal, Bharat and Alok Aggarwal (Dec. 2022). "ETL, ELT and Reverse ETL: A business case Study". In: *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, pp. 1–4. DOI: 10.1109/ICATIECE56365.2022.10046997. URL: https://ieeexplore.ieee.org/abstract/document/10046997 (visited on 06/21/2025).

Sivabalan, S and R I Minu (Nov. 2021). "Heterogeneous Data Integration with ELT and Analytical MPP Database for Data Analysis Application". In: *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, pp. 1–5. DOI: 10.1109/i-PACT52855.2021.9696841. URL: https://ieeexplore.ieee.org/abstract/document/9696841 (visited on 06/21/2025).

Smaili, N. and A. Kabbaj (2025). "Enabling semantic interoperability for smart farming". en. In: 564.65 KB. ISSN: 2228-4907. DOI: 10.15159/AR.25.021. URL: https://dspace.emu.ee/items/16b046e6-014e-4cb6-95ba-9d6d6607690a (visited on 06/09/2025).

SMARTBEAR (2025). *REST API Documentation Tool | Swagger UI*. URL: https://swagger.io/tools/swagger-ui/ (visited on 06/25/2025).

Smuha, Nathalie A. (2019). *The EU Approach to Ethics Guidelines for Trustworthy Artificial Intelligence*. en. SSRN Scholarly Paper. Rochester, NY. URL: https://papers.ssrn.com/abstract=3443537 (visited on 06/24/2025).

Tamim Mahmud Al-Hasan et al. (Apr. 2024). "From Traditional Recommender Systems to GPT-Based Chatbots: A Survey of Recent Developments and Future Directions". en. In: *Big Data and Cognitive Computing* 8.4. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 36. ISSN: 2504-2289. DOI: 10.3390/bdcc8040036. URL: https://www.mdpi.com/2504-2289/8/4/36 (visited on 12/29/2024).

Team, Keras (2025). *Keras documentation: Getting started with Keras*. en. URL: https://keras.io/getting_started/ (visited on 06/17/2025).

Thiebes, Scott, Sebastian Lins, and Ali Sunyaev (June 2021). "Trustworthy artificial intelligence". en. In: *Electronic Markets* 31.2, pp. 447–464. ISSN: 1422-8890. DOI: 10.1007/s12525-020-00441-4. URL: https://doi.org/10.1007/s12525-020-00441-4 (visited on 06/24/2025).

Tiwari, Ashish (Jan. 2022). "Chapter 2 - Supervised learning: From theory to applications". In: *Artificial Intelligence and Machine Learning for EDGE Computing*. Ed. by Rajiv Pandey et al. Academic Press, pp. 23–32. ISBN: 978-0-12-824054-0. DOI: 10.1016/B978-0-12-824054-0.00026-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128240540000265 (visited on 06/21/2025).

Toby Mao (2025). *sqlglot API documentation*. URL: https://sqlglot.com/sqlglot.html (visited on 06/16/2025).

Tom Aarsen (2025). *SentenceTransformers Documentation — Sentence Transformers documentation*. URL: https://www.sbert.net/ (visited on 06/16/2025).

Ultralytics (2025). *Ultralytics | Revolutionizing the World of Vision AI*. pt. URL: https://www.ultralytics.com/pt (visited on 06/17/2025).

Valkenborg, Dirk et al. (June 2023a). "Unsupervised learning". English. In: *American Journal of Orthodontics and Dentofacial Orthopedics* 163.6. Publisher: Elsevier, pp. 877–882. ISSN: 0889-5406, 1097-6752. DOI: 10.1016/j.ajodo.2023.04.001. URL: https://www.ajodo.org/article/S0889-5406(23)00193-2/fulltext (visited on 06/22/2025).

– (June 2023b). "Unsupervised learning". English. In: *American Journal of Orthodontics and Dentofacial Orthopedics* 163.6. Publisher: Elsevier, pp. 877–882. ISSN: 0889-5406, 1097-6752. DOI: 10.1016/j.ajodo.2023.04.001. URL: https://www.ajodo.org/article/S0889-5406(23)00193-2/fulltext (visited on 06/23/2025).

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (visited on 06/23/2025).

Vekaria, Darshan and Sunil Sinha (Apr. 2024). "aiWATERS: an artificial intelligence framework for the water sector". en. In: *AI in Civil Engineering* 3.1, p. 6. ISSN: 2730-5392. DOI: 10.1007/s43503-024-00025-7. URL: https://doi.org/10.1007/s43503-024-00025-7 (visited on 12/19/2024).

Vishal, M. and H. Vishalakshi Prabhu (2023). "A Comprehensive Review of Conversational AI-Based Chatbots: Types, Applications, and Future Trends". en. In: *Internet of Things (IoT): Key Digital Trends Shaping the Future*. Ed. by Rajiv Misra et al. Singapore: Springer Nature, pp. 293–303. ISBN: 978-981-19-9719-8. DOI: 10.1007/978-981-19-9719-8_24.

Wang, Xu et al. (Apr. 2024). "Deep Reinforcement Learning: A Survey". In: *IEEE Transactions on Neural Networks and Learning Systems* 35.4, pp. 5064–5078. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2022.3207346. URL: https://ieeexplore.ieee.org/abstract/document/9904958 (visited on 06/22/2025).

Whang Steven Euijong et al. (July 2023). "Data collection and quality challenges in deep learning: a data-centric AI perspective". en. In: *The VLDB Journal* 32.4, pp. 791–813. ISSN: 0949-877X. DOI: 10.1007/s00778-022-00775-9. URL: https://doi.org/10.1007/s00778-022-00775-9 (visited on 12/17/2024).

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Qun Liu and David Schlangen. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6/ (visited on 06/23/2025).

Wrembel, Robert (2024). "Optimizing data integration processes with the support of machine learning-is it really possible?" In: *DOLAP*, pp. 91–94.

Yang, Xiangli et al. (Sept. 2023). "A Survey on Deep Semi-Supervised Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 35.9, pp. 8934–8954. ISSN: 1558-2191. DOI: 10.1109/TKDE.2022.3220219. URL: https://ieeexplore.ieee.org/abstract/document/9941371 (visited on 06/21/2025).

Zha, Daochen et al. (Jan. 2023). "Data-centric AI: Perspectives and Challenges". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, pp. 945–948. URL: https://epubs.siam.org/doi/10.1137/1.9781611977653.ch106 (visited on 06/11/2025).

Zhang, Duzhen et al. (May 2024). *MM-LLMs: Recent Advances in MultiModal Large Language Models*. arXiv:2401.13601 [cs]. DOI: 10.48550/arXiv.2401.13601. URL: http://arxiv.org/abs/2401.13601 (visited on 06/23/2025).

Zhao, Zehui et al. (May 2024). "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations". In: *Expert Systems with Applications* 242, p. 122807. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.122807. URL: https://www.sciencedirect.com/science/article/pii/S0957417423033092 (visited on 06/22/2025).

Zhizhong Wu (Oct. 2024). "Large Language Model Based Semantic Parsing for Intelligent Database Query Engine". en. In: *Journal of Computer and Communications* 12.10. Number: 10 Publisher: Scientific Research Publishing, pp. 1–13. DOI: 10.4236/jcc.2024.1210001. URL: https://www.scirp.org/journal/paperinformation?paperid=136477 (visited on 12/29/2024).

Zhou, Jiwei and Jorge D. Camba (July 2025). "The status, evolution, and future challenges of multimodal large language models (LLMs) in parametric CAD". In: *Expert Systems with Applications* 282, p. 127520. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2025.127520. URL: https://www.sciencedirect.com/science/article/pii/S095741742501142X (visited on 06/23/2025).

Zhu, Jiachen et al. (June 2025). *Evolutionary Perspectives on the Evaluation of LLM-Based AI Agents: A Comprehensive Survey*. arXiv:2506.11102 [cs]. DOI: 10.48550/arXiv.2506.11102. URL: http://arxiv.org/abs/2506.11102 (visited on 06/23/2025).

Zong, Mingyu et al. (Jan. 2025). "Integrating large language models with internet of things: applications". en. In: *Discover Internet of Things* 5.1, p. 2. ISSN: 2730-7239. DOI: 10.1007/s43926-024-00083-4. URL: https://doi.org/10.1007/s43926-024-00083-4 (visited on 06/23/2025).

DECLARAÇÃO DE INTEGRIDADE

DECLARAÇÃO DE INTEGRIDADE

Declaro ter conduzido este trabalho académico com integridade. Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Declaro que o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

Joana Rodrigues Figueiredo

ISEP, Porto, 30 de junho de 2025