POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Integrated Approach to Data in Aquaponics Systems

Edna Cristina Santos Coelho

School of Management and Technology
Department of Computer Engineering
Master in Data Science

Leiria, July 2025

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Edna Cristina Santos Coelho

Supervisor: Professor Rosa Isabel Alves Cordeiro Matias
Adjunct Professor, Polytechnic of Leiria

Co-supervisor: *Professor Fernando José do Nascimento Sebastião
Adjunct Professor, Polytechnic of Leiria

**Professor Raul José Silvério Bernardino
Adjunct Professor, Polytechnic of Leiria

School of Management and Technology
Department of Computer Engineering
Master in Data Science

Project

Leiria, July 2025

***Professor Fernando José do Nascimento Sebastião**, is a member of LSRE-LCM, a research unit financially supported by *Fundação para a Ciência e a Tecnologia, I.P. /MCTES* through national funds: LSRE-LCM, UID/50020; and ALiCE, LA/P/0045/2020 (DOI: 10.54499/LA/P/0045/2020).

****Professor Raul José Silvério Bernardino**, is a member of LSRE-LCM, a research unit financially supported by *Fundação para a Ciência e a Tecnologia, I.P. /MCTES* through national funds: LSRE-LCM, UID/50020; and ALiCE, LA/P/0045/2020 (DOI: 10.54499/LA/P/0045/2020).

Integrated Approach to Data in Aquaponics Systems

Acknowledgements

To my family, for their constant support and the solid foundation they have always provided, allowing me to dedicate myself to this project. To my boyfriend, for his partnership and understanding throughout the entire process, contributing to the achievement of this goal. To my professors and advisors, for the knowledge shared, the essential guidance, and the commitment to directing this work. A special thanks to Professor Fernando José do Nascimento Sebastião, Professor Rosa Isabel Alves Cordeiro Matias, Professor Raul José Silvério Bernardino and Professor José Areia, for their guidance, support, and dedication.

I would like to express my deepest gratitude to my father, who, sadly, will not witness the conclusion of my master's degree. For 45 years, he was my greatest supporter, always providing the foundation and education that enabled me to reach this milestone. (13/07/2025).

Resumo

Este projeto consiste sobre o desenvolvimento de uma solução de Business Intelligence aplicada ao contexto de um sistema de aquaponia numa instituição de ensino superior, com o objetivo de facilitar a recolha, integração, análise e visualização de dados operacionais e ambientais. A fragmentação e a falta de padronização dos dados, quer dos ficheiros, convenções de nomenclatura ou estruturas das tabelas, dificultavam a consolidação eficiente da informação e a tomada de decisão. O objetivo principal foi desenvolver uma arquitetura baseada num Data Lakehouse para automatizar a recolha, integração, análise e visualização desses dados, favorecendo a monitorização contínua dos parâmetros e a partilha de resultados com a comunidade científica.

Para isso, concebeu-se uma arquitetura de dados baseada em paradigmas de Data Lakehouse, que integra Apache Spark para processamento distribuído e Power BI para criação de um modelo semântico e dashboards interativos. O trabalho inclui a conceção de um modelo dimensional, a implementação de um pipeline de ETL (Extração, Transformação e Carregamento) para limpeza e unificação de arquivos heterogéneos (com formatos, convenções de nomenclatura e estruturas de tabelas inconsistentes) e o desenvolvimento de relatórios visuais orientados ao desempenho do sistema aquapónico.

A principal contribuição consiste na demonstração do potencial das tecnologias analíticas e de visualização de dados na gestão sustentável de sistemas de aquaponia, mostrando como práticas de engenharia de dados aliadas a ferramentas de BI permitem superar desafios de qualidade, volumetria e escalabilidade dos dados.

Por fim, a usabilidade da solução Power BI foi avaliada, recorrendo à aplicação de um questionário, através da *Escala de Usabilidade do Sistema*, obtendo-se a classificação de Aceitável. Destacaram-se como pontos fortes a clareza do conteúdo e a facilidade de navegação, enquanto a estética do design e o desempenho foram apontados como oportunidades de aperfeiçoamento em desenvolvimentos futuros.

Palavras-Chave: Aquaponia, Business Intelligence, Data Lakehouse

Abstract

This project focuses on the development of a Business Intelligence solution applied to the context of an aquaponics system within a higher education institution. Its objective is to facilitate the collection, integration, analysis, and visualization of operational and environmental data. The fragmentation and lack of standardization of data, including files, naming conventions, and table structures, hindered the efficient consolidation of information and the decision-making process. The main goal was to develop a Data Lakehouse, based architecture to automate the collection, integration, analysis, and visualization of this data, enabling continuous monitoring of key parameters and the sharing of results with the scientific community.

To achieve this, a data architecture based on Data Lakehouse paradigms was designed, integrating Apache Spark for distributed processing and Power BI for the creation of a semantic model and interactive dashboards. The work includes the design of a dimensional model, the implementation of an ETL (Extract, Transform, Load) pipeline for cleaning and unifying heterogeneous files (with inconsistent formats, naming conventions, and table structures), and the development of visual reports focused on the performance of the aquaponic system.

The main contribution lies in demonstrating the potential of analytical and data visualization technologies in the sustainable management of aquaponics systems, showing how data engineering practices combined with BI tools can overcome challenges related to data quality, volume, and scalability.

Finally, the usability of the Power BI solution was evaluated using a questionnaire based on the *System Usability Scale*, resulting in a rating of Acceptable. Strengths included content clarity and ease of navigation, while areas for improvement identified in future developments were the visual aesthetics and performance.

Palavras-Chave: Aquaponics, Business Intelligence, Data Lakehouse

Contents

List of Figures

List of Tables

Acronyms

AAD Azure Active Directory. (*p. 83*)
ACID Atomicity Consistency Isolation Durability. (*p. 23–26, 38*)
AI Artificial Intelligence. (*p. 25, 29, 34, 35*)
AWS Amazon Web Services. (*p. 20, 27*)

BI Business Intelligence. (*p. 2, 4, 5, 15, 17–19, 22, 24, 25, 36–38, 40, 41, 52, 53, 75, 82, 104*)

CRISP-DM Cross-Industry Standard Process for Data Mining. (*p. xi, 3–5*)

DAGs Directed Acyclic Graphs. (*p. 21*)
DAX Data Analysis Expressions. (*p. 84*)
DDL Data Definition Language. (*p. 32, 33*)
DL Deep Learning. (*p. 25*)
DML Data Manipulation Language. (*p. 32, 33*)
DMVs Dynamic Management Views. (*p. 84*)
DO Dissolved Oxygen. (*p. 12, 13, 87, 92*)
DW Data Warehouse. (*p. 22, 23, 25–27, 32, 37, 38, 52, 53*)
DWC Deep Water Culture. (*p. 48, 49, 92*)

ESTG School of Technology and Management. (*p. 1, 9*)
ETL Extraction, Transformation, and Loading. (*p. 5, 7, 17, 22, 26, 31, 32, 34, 36, 37, 66–69, 71, 72, 75, 104, 105*)

GCP Google Cloud Platform. (*p. 21*)

HDFS Hadoop Distributed File System. (*p. 21*)

IaaS Infrastructure as a Service. (*p. 18*)
IDC International Data Corporation. (*p. 19, 20*)
IoT Internet of Things. (*p. 3, 18*)
IPLeiria Polytechnic Institute of Leiria. (*p. xi, xv, 1, 2, 8, 9, 11, 14, 40, 48, 51, 61, 63, 64, 67, 75, 78, 82, 85, 104*)

IT Information Technology. (*p. 17*, *18*)

JSON JavaScript Object Notation. (*p. 34*)

KQL Kusto Query Language. (*p. 68*)

LSMI Laboratory of Integrated Multitrophic Systems. (*p. xi*, *xii*, *xv*, *1–5*, *9–12*, *14*, *15*, *40*, *45*, *48*, *52*, *54–56*, *68*, *79*, *83*, *85*)

LSRE-LCM Laboratory of Separation and Reaction Engineering & Laboratory of Catalysis and Materials. (*p. 1*)

ML Machine Learning. (*p. 25*, *29*, *38*, *105*)

OLAP Online Analytical Processing. (*p. 23*, *24*, *53*)

OLTP Online Transaction Processing. (*p. 23*, *24*)

ORP Oxidation-Reduction-Potential. (*p. 13*, *87*, *92*)

PaaS Platform as a Service. (*p. 18*)

RDBMS Relational Database Management Systems. (*p. 20*)

RDDs Resilient Distributed Datasets. (*p. 21*)

SaaS Software as a Service. (*p. 18*, *19*, *31*)

SCD Slowly Change Dimension. (*p. 73*)

SD Standard Deviation. (*p. 102*)

SQL Structured Query Language. (*p. 25*, *33*, *34*, *49*, *72*, *73*, *75*)

SSMS SQL Server Management Studio. (*p. xii*, *79*)

SUS System Usability Scale. (*p. xv*, *100–103*, *133*)

TDS Total Dissolved Solids. (*p. 12*, *13*, *87*, *92*)

1 Introduction

At the Polytechnic Institute of Leiria (IPLeiria), specifically at the School of Technology and Management (ESTG), there is a greenhouse equipped with aquaponics systems, located within the Laboratory of Integrated Multitrophic Systems (LSMI). This laboratory is part of the Laboratory of Separation and Reaction Engineering & Laboratory of Catalysis and Materials (LSRE-LCM) resarch group and integrates aquaculture and hydroponics in a closed-loop environment. As part of this initiative, IPLeiria is conducting a research line focused on this aquaponics greenhouse, which connects fish and plant cultivation in a sustainable and interdependent cycle [1].

Aquaponics aims to develop sustainable food production systems that combines two techniques:(i) aquaculture which is the farming of aquatic animals like fish in tanks and (ii) hydroponics which is the cultivation of plants in water without soil, using nutrient-rich solutions [1] [2].

To further enhance the aquaponics system, it is essential to systematically monitor key environmental and operational parameters — such as temperature, humidity, pH and nutrient levels [1] — and integrate this data in order to generate actionable insights that supports the system's continuous improvement.

The system aims to promote a sustainable, ecological ecosystem where, for instance, water use is efficient and no chemical products are used. Furthermore, the closed-loop nature of aquaponics minimizes waste by reusing fish effluents as nutrients for plants, while the plants, in turn, help purify the water for the fish. This natural synergy reduces the need for external inputs, lowers operational costs, and decreases environmental impact. Additionally, aquaponics systems can be implemented in urban or resource-limited areas, contributing to local food production, reducing transportation emissions, and enhancing food security. By integrating technological monitoring and data analysis, the system's performance and efficiency can be further optimized [1].

In the context of the LSMI, this thesis aims to overcome the main challenges in data management, including the lack of standardized procedures, which has led to issues in data collection and processing.

1.1 Context

As the global population continues to expand and urbanization accelerates, the demand for efficient and sustainable food production systems has become increasingly urgent. Conventional agricultural practices often involves a substantial utilization of land, water, and chemical inputs, resulting in environmental degradation and resource depletion. In response to these challenges, integrated systems such as aquaponics, have emerged as promising alternatives due to their low environmental impact and resource efficiency [2].

Aquaponics, though promising, faces key challenges. Maintaining optimal conditions for both fish and plants highlights the continuous monitoring of environmental variables, including temperature, pH, and nutrient concentrations. Moreover, the absence of a systematic approach to data management frequently impedes the extraction of insights and the enhancement of the aquaponics system.

To address these limitations, the development of a data-driven approach can play a pivotal role. The aim of this project is the design and development of a comprehensive Business Intelligence (BI) solution enabling scalable data management and analytics to support the aquaponics project. The platforms intends to streamline data storage and enhance decision-making, allowing aquaponics systems to operate with greater intelligence and efficiency. This thesis explores the design and implementation of this approach within the context of IPLeiria's greenhouse aquaponics project.

1.2 Problem Description

The development of an analytical system is important to facilitate the acquisition of insights and promote continuous enhancement of the greenhouse ecosystem's performance and sustainability. For instance, the greenhouse currently lacks a decision support system that enables integrated and effective data exploration and visualization for each project that has been completed, is ongoing, or is planned for the future. Having a system or platform like this could be really valuable. It could help find ways to improve specific environmental factors, which would make the ecosystem more efficient overall. Also, there's no platform to present the results of the various projects developed, which limits how the data and knowledge generated within the greenhouse can be shared and accessed.

During the problem contextualization meetings with stakeholders and the analysis of reports used to register greenhouse environmental variables, several data-related chal-

lenges were identified. These issues primarily stem from the methods by which data is collected, stored, and maintained. At that time, the greenhouse staff and researchers utilized Excel files as the primary data storage system.

1.3 Objectives

In the context of the LSMI, this thesis proposes the design and implementation of a decision support system, which has the potential to significantly enhance the effectiveness of ecosystem management.

To achieve this, the first step is to overcome the main challenges outlined in **Section 1.2**, in order to prevent and eliminate inconsistencies in data collection and processing. Next, a set of strategic actions is presented to help overcome the challenges and align with the project's overall objectives:

- Design an integrated data architecture tailored to the needs of LSMI, enabling efficient and scalable data management.
- Establish standardized procedures for data collection, processing, and organization to ensure consistency and reliability.
- Develop a unified data model that consolidates heterogeneous datasets into a coherent structure, supporting cross-analysis.
- Enable systematic data analysis by transforming raw data into structured formats suitable for scientific insight and decision-making.
- Support research and monitoring activities by providing accurate, timely, and accessible information that improves operational and scientific outcomes.
- Ensure that the platform supports future developments, namely through the integration of Internet of Things (IoT) systems for the automated collection of certain parameter values, which may involve a Big Data environment.

1.4 Methodology

For this work, an adapted version of the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology was adopted. Developed in 1996 by an industry consortium, this methodology was created to be neutral in terms of industry, tools, and application domains [3]. CRISP-DM is a structured, cyclic approach for carrying out data mining and analytical projects. It is composed of six phases: *Business Understanding*, *Data Understanding*, *Data Preparation*, *modelling*, *Evaluation*, and *Deployment*, Figure 1.1. This framework provides flexibility and guidance throughout the project lifecycle, making it suitable for a wide range of data-driven initiatives [3].

The evolution of the CRISP-DM, as analyzed by [5], highlights the model's flexibility and adaptability to different business and technical contexts. In this article, published

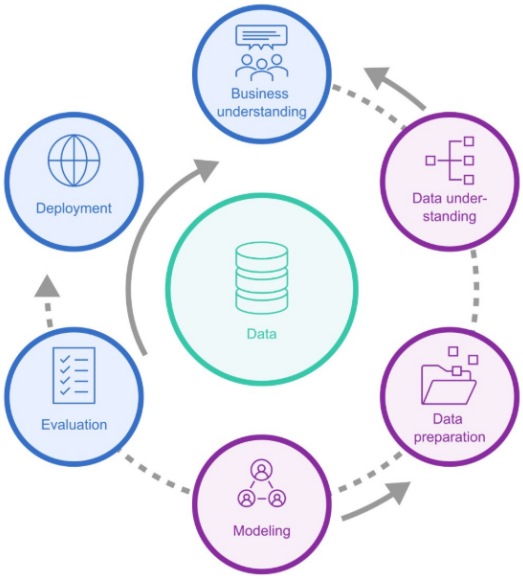


Figure 1.1: *Schema of a CRISP-DM model in* [4].

in 2024, various adaptations of the CRISP-DM methodology to various business domains were also examined, demonstrating its capacity to be tailored to specific industry needs and data-driven objectives. The authors analyzed 16 CRISP-DM derivatives, identifying common adaptation patterns such as phase addition or modification, inclusion of new features and tools, and integration with other methodologies [5]. As a result, the study proposes a theoretical guide to support the customization of CRISP-DM, reinforcing its relevance as a robust and versatile framework [5].

The CRISP-DM methodology has already been adapted in the context of BI projects. In 2020, a master's thesis from the University of Aveiro applied a CRISP-DM - based approach to the development of a Power BI solution [6]. Similarly, in 2022, a project also employed a tailored version of the CRISP-DM methodology to support the design of a Power BI solution [7].

Considering the flexibility of the CRISP-DM framework and its demonstrated effectiveness in BI projects, this thesis adopts a version of the methodology adapted to the specific requirements of the aquaponics research context at LSMI. While preserving the core structure of the original CRISP-DM model, this adaptation introduces refinements to better address the characteristics of environmental data and to support the iterative development of dashboards using Power BI desktop. This approach ensures methodological coherence with previous CRISP-DM applications, while meeting the analytical and operational demands of this particular project.

CRISP-DM Methodology Adapted to BI Solutions

Below, each phase of the adapted CRISP-DM methodology is presented in detail, highlighting how it was implemented and tailored to the context of this work.

- **Business Understanding:** Define the project's objectives in the context of the LSMI aquaponics system, identifying the need for data integration, monitoring, and analysis to support research activities.
- **Data Understanding:** Examine the structure, content, and quality of the available datasets. This phase involves identifying missing values, detecting inconsistent formats and analyzing data distributions. It also includes understanding the origin, granularity, frequency, and reliability of each data source.
- **Data Preparation:** Standardize, clean, and transform the raw data collected from various Excel files into a consistent and structured format suitable for analysis. This phase includes handling missing values, normalizing data types, and ensuring referential integrity across datasets. It also involves the implementation of an ETL process to automate the integration of data from heterogeneous sources into a unified data model. Business rules are defined and applied during this phase to ensure the accuracy and relevance of the transformed data, which is then loaded into the analytical environment to support subsequent modelling and visualization tasks.
- **Modelling:** Design a dimensional data model (star or constellation schema) tailored to the project's analytical needs. Although this phase does not involve predictive modelling, it provides the structural foundation for analysis and reporting.
- **Evaluation:** Validate the quality and completeness of the integrated data and assess whether the BI model meets the intended analytical goals. This includes reviewing dashboards, verifying key metrics, and confirming that the visualizations accurately reflect the collected responses. At this stage, the Power BI report was made available to the users involved in the project so they could validate the presented values and ensure the accuracy of the transformations and visual output. Additionally, a group of participants was involved in a survey to gather feedback regarding the usability and overall user experience of the Power BI solution.
- **Deployment:** Once the dashboards were finalized, the solution was published to the LSMI Project workspace. During this phase, access permissions were configured to ensure that stakeholders had appropriate visibility according to their roles, and automatic data refresh schedules were defined to keep the reports up to date. Additionally, documentation was prepared to describe the main results, outline the implemented features, and register lessons learned that may inform future projects and improvements.

Just like the original CRISP-DM model, the adapted methodology presented in this work also follows a bidirectional flow between certain phases, allowing for iteration and refinement throughout the process, Figure 1.1.

To ensure an organized and efficient development process throughout the project, the work was managed using **Notion**, a digital workspace tool. The thesis chapters and

subchapters were outlined and structured within the platform, thereby providing a dynamic overview of the project's progression. The classification of tasks and components was facilitated by status labels such as *Not Started*, *In Progress*, *Review*, *Done*, *For Validation* and *Waiting*, which enabled progress tracking and prioritization, Appendix A. Furthermore, a calendar, was developed to align deliverables with established milestones, thereby facilitating enhanced time management and iterative processes. This structured approach facilitated enhanced visibility regarding the workflow, thereby fostering a more agile and methodical execution of each phase of the thesis.

1.5 Structure of the Document

This document is structured to provide a comprehensive overview of the development of a data architecture to support scientific research and monitoring activities in the context of an aquaponics system. Each chapter contributes to building a clear understanding of the technical, methodological, and practical components of the project:

- **Chapter 1: Introduction** - Provides an introduction to the project, including the context, the problem being addressed, the main objectives, the methodological approach adopted, and an overview of the document structure.
- **Chapter 2: Aquaponics System Overview: Design and Challenges** - Describes the physical and operational structure of the aquaponics system and identifies the main challenges in data collection and management.
- **Chapter 3: Cloud Ecosystems for Business Intelligence and Big Data: A Technical Review** - Provides the theoretical and technological foundations for the project. It reviews key concepts related to big data, cloud computing, distributed processing, and modern data architectures, with a special focus on Lakehouse and Microsoft Fabric.
- **Chapter 4: Solution Overview and BI Arquitecture** - Details a comprehensive overview of the proposed solution and the Business Intelligence architecture designed to support it. The text explains the strategies employed to address the identified challenges, as well as the methods applied during the development of the solution.
- **Chapter 5: The Aquaponics System Design** - Details the process of locating and analyzing data sources relevant to the aquaponics system. It includes an overview of how data is collected, recorded, and governed through business rules. Describes the dimensional model used to organize the data for analytical purposes. It includes the definition of dimensions, fact tables, hierarchies, and a bus matrix, as well as a visual representation of the model. Presents the matrix that maps the fields from the original data sources to the dimensional schema, distinguishing between dimensions and fact tables.
- **Chapter 6: Lakehouse Implementation and Data Integration Project – ETL** -

Explains the architecture and implementation of the data integration pipeline, including the extraction, transformation, and loading ETL phases, as well as the orchestration strategy used to manage the process.

- **Chapter 7: Building the Visual Aquaponic Project** - Describes the practical implementation of the visual aquaponics project using Power BI. It covers the configuration of the Power BI Desktop environment, secure data connection, semantic data modelling, including table relationships, hierarchies, and DAX measures, and the design of the final reports and dashboards.
- **Chapter 8: Usability Evaluation of the Visual Aquaponics Dashboard** - Presents the usability evaluation of the developed Power BI dashboard through the application of the System Usability Scale (SUS). The chapter also includes the analysis of the results and a brief discussion of the conclusions drawn from the users' feedback.
- **Chapter 9: Conclusion** - Summarizes the main contributions and results of the thesis, reflects on its limitations, and suggests directions for future work.

2

Aquaponics System Overview: Design and Challenges

This chapter presents an overview of the aquaponics system developed at IPLeiria. It first describes the system's design and functionality, emphasizing its integration of aquaculture and hydroponics in a closed-loop environment. Next, key operational challenges, such as data acquisition, environmental monitoring, and scalability, are identified.

2.1 Introduction to Aquaponics

Aquaponics have historical roots dating back to ancient civilizations. The Aztecs developed chinampas, floating agricultural islands in shallow lakes in central Mexico, often regarded as the first aquaponics system. Similarly, in ancient China, rice paddies were integrated with aquatic organisms like fish, which entered through floodwaters [8], creating a synergistic farming system. Other aquaponics systems, such as monoculture and polyculture, also emerged, with shrimp and fish co-cultivation enhancing nutrient cycling and sustainability [8].

Aquaponics is a farming method that promises to be a good alternative against the food and environmental problem the world is facing. It is a combination between aquaculture (fish farmimg) and hydroponics (growing plants without soil), being a technique to grow plants using the aquaculture effluent [9]. This method is based on the principle of symbiosis: the metabolic waste produced by the fish, rich in nutrients, is converted into compounds usable by the plants through the action of nitrifying bacteria. In turn, the plants filter the water, returning it clean to the fish farming system. Both terrestrial plant production systems and aquatic animal production systems share a common resource: water. Plants generally consume water through transpiration, releasing it into the surrounding gaseous environment, whereas fish consume less water but, when

raised in confined environments, generate substantial wastewater flows due to the accumulation of metabolic waste [2]. This technique demonstrates high water efficiency, minimizes pesticide use, and reduces fertilizers, making it both green and sustainable. The growing interest in aquaponics highlights the challenge of ensuring its feasibility and reliability on a commercial scale [9].

Aquaponics systems can be implemented at different scales, ranging from small household units to large commercial facilities, and can be adapted to a wide variety of climates and conditions. In "On the sustainability of aquaponics," a study, show that this technology demonstrates high efficiency in the production of vegetables and fish, making it viable in both urban and rural areas [10].

The symbiotic cycle of an aquaponics system is represented in Figure 2.1, illustrating the interaction between fish, plants, and bacteria to establish a sustainable ecosystem. In this system, fish release waste in the form of ammonia (NH_3), which is toxic in high concentrations. *Nitrosomonas* bacteria convert the ammonia into nitrites (NO_2^-), which are still harmful to fish. Subsequently, *Nitrobacter* bacteria transform these nitrites into nitrates (NO_3^-), a less harmful compound that serves as a vital nutrient for plants. The plants then absorb the nitrates as fertilizer, facilitating their growth while purifying the water. Finally, the purified water is returned to the fish tank, completing the cycle. This process exemplifies the efficient use of resources in aquaponics, fostering a balanced and mutually beneficial environment for both plant and fish production [11].

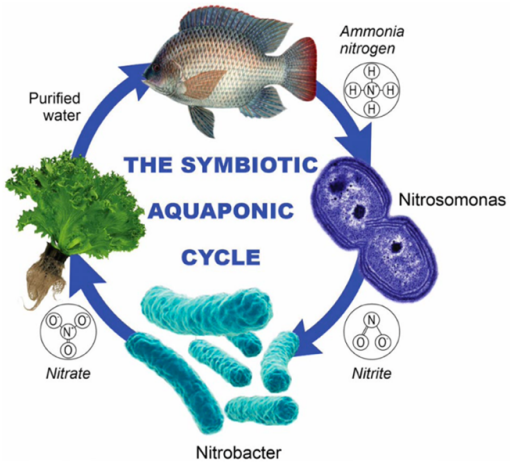


Figure 2.1: *Scheme of an aquaponics system in* [11].

2.2 The Greenhouse Aquaponics System Design

In the context of the LSMI at IPLeiria and according to the presentation document on LSMI Projects, an aquaponics system is housed in a greenhouse built in 2018. This facility, covering approximately 150m², is located on Campus 2 of the ESTG. The primary

focus is on sustainable production systems that integrate aquaculture and hydroponics, Figure 2.2 [1].

Figure 2.2: *aquaponics system is housed in a greenhouse built in 2018 in* [1].

At the moment the LSMI operates five independent aquaponics systems (Lines 1, 2, and 3, as well as the Corner System and Door System). The Lines, three independents integrated aquaponics systems, each equipped with essential components, including a 3400 L aquaculture tank, a mechanical filter, a sedimentation tank, a biofilter, a hydroponic bed, and a sump tank. These systems operate within a closed-loop water and nutrient circulation framework, ensuring continuous aeration in key components to maintain optimal performance [1], as illustrated in, Figure 2.3. The Door and Corner Systems contain the same components as the line systems, except for the Mechanical Filter and the Sump, and they also feature a different layout. The components are:

- **Aquaculture Tank (3400 L)** – Used for fish farming, where fish waste is collected and processed for plant nutrition.
- **Mechanical Filter** – Removes solid waste particles from the water before it proceeds to the biofiltration stage.
- **Sedimentation Tank (90 L)** – Separates heavier particles from the water, improving its quality.
- **Biofilter (300 L)** – Contains bioballs that promote bacterial growth, converting harmful ammonia into nitrates, which are beneficial for plant growth.
- **Hydroponic Bed (2250 L)** – Supports plant cultivation in water enriched with nutrients from the aquaculture system.
- **Sump (500 L)** – Acts as a reservoir to collect and recirculate water back into the

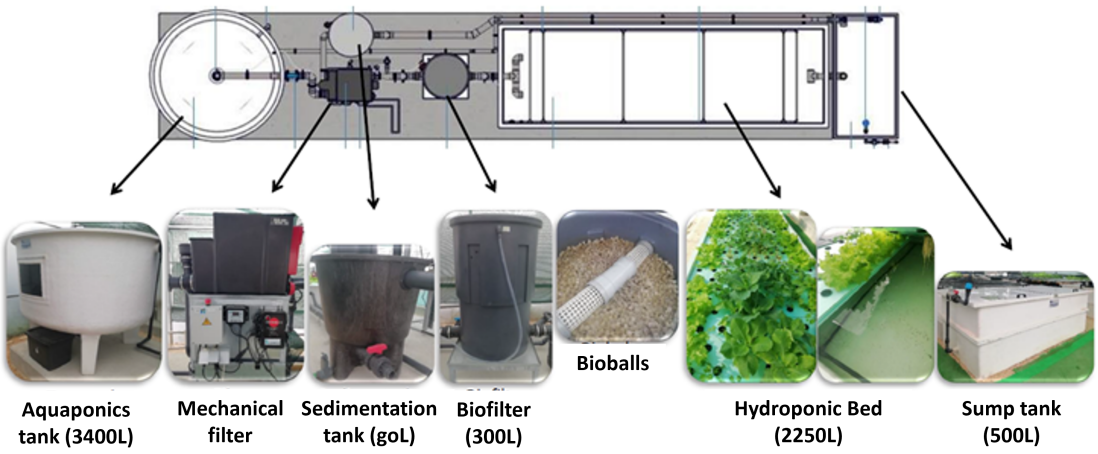


Figure 2.3: *Components of the LSMI at IPLeiria in* [1].

system, ensuring continuous operation.

The system operates as a closed loop, with green arrows illustrating the continuous flow of water and nutrients between the components. This setup highlights the symbiotic relationship between fish and plants, demonstrating a sustainable and efficient approach to integrated aquaponics. The figure is labeled as Figure 2.4.

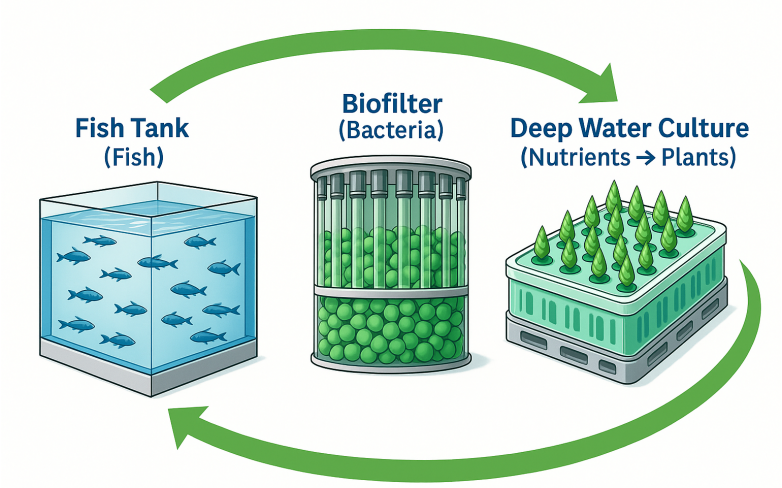


Figure 2.4: *Integrated Water Recirculation System.*

The aquaponics systems support different fish species. The greenhouse allocates some projects over time. At present, Line 1 is dedicated to a culture of *Cyprinus rubrofuscus*, typically measures between 25 and 70 cm in length, weighs between 1 and 5 kg, and thrives in water with a pH range of 6.8–8.2 and an optimal temperature between 15–25°C. Lines 2 and 3 are home to *Clarias gariepinus* (African catfish), which grow to over 100 cm in length, exceed 4 kg in weight [1].

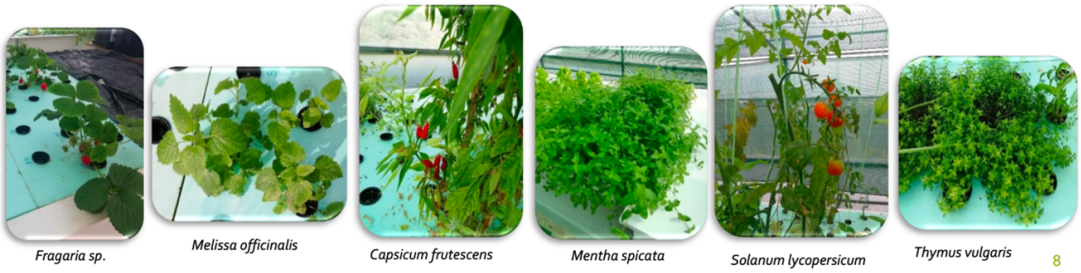


Figure 2.5: *Plants cultivated at the LSMI in* [1].

In addition to fish, the LSMI cultivates a variety of plants, Figure 2.5, including mint (*Mentha spicata*), chili (*Capsicum frutescens*), lettuce (*Lactuca sativa*), basil (*Ocimum basilicum*), tomatoes (*Solanum lycopersicum*), and many others. This diversity underscores the adaptability and efficiency of the aquaponics system in producing both plant and fish crops sustainably [1].

System maintenance is a critical aspect of the LSMI's operations. It involves monitoring physical-chemical water parameters such as pH, temperature, Dissolved Oxygen (DO), and Total Dissolved Solids (TDS) using multiparametric probes. Laboratory analyses are also conducted to assess water quality indicators, including ammonia, nitrates, nitrites, and phosphates. Furthermore, the feeding and behavioral observation of the fish, as well as the evaluation of plant growth and quality, are essential components of the system's maintenance strategy. This comprehensive approach ensures the effective functioning and sustainability of the aquaponics systems at LSMI [1].

2.3 Identification of Measured Aquaponics Parameters

Aquaponics systems require a delicate balance to support optimal species development. Achieving this balance necessitates foundational knowledge and experience in several areas, including environmental conditions (e.g., air temperature, humidity, water temperature, pH, and DO), water quality (e.g., ammonia, nitrites, nitrates, alkalinity, heavy metal pollution, and microbial contamination), the quantity and frequency of fish feeding, and the extent of fish waste mineralization. Environmental factors such as temperature, humidity, pH, and mineral concentrations should be maintained as close as possible to the ideal conditions for species growth [11].

To document the development of work conducted in the aquaponics system at LSMI, a variety of parameters were measured. Below, all the collected parameters are presented in detail, along with relevant information.

There are parameters that are measured daily across the various tanks within the different existing systems. When a specific project is underway, measurements are taken both in the morning and afternoon. Otherwise, only one reading is performed per day,

either in the morning or in the afternoon. Throughout this study, these data will be referred to as Daily System Records, Table 2.1.

Parameters (Units)	Description	Reference
DO mg/L	Concentration of dissolved oxygen in water; essential for aquatic respiration	[11]
Temperature °C	Water temperature; influences water quality and biochemical processes	[11]
pH	Measures acidity/alkalinity (0–14 scale; <7 acidic, 7 neutral, >7 alkaline)	[12]
Electrical conductivity (σ) μS/cm	Water's ability to conduct electricity; related to dissolved ion concentration	[12]
TDS mg/L	Total dissolved substances in water	[12]
Oxidation-Reduction-Potential (ORP) mV	Oxidation-reduction potential; indicates ability to oxidize/reduce substances	[12]

Table 2.1: *Parameters of the daily system records, units, descriptions and literature references*

The Table 2.2 presents the products added to the different tanks of the existing systems. It is worth noting that fish feed is administered on a daily basis, while the addition of other products occurs sporadically.

Products (Units)	Description
Fish feed (g)	The amount of food provided to the fish
Potassium carbonate (K_2CO_3) (g)	The amount of potassium carbonate used for pH or alkalinity adjustment [12] and provide potassium, an essential macronutrient for plant growth [13]
Potassium hydroxide (KOH) (g)	The amount of potassium hydroxide used for pH or alkalinity adjustment [12] and provide potassium, an essential macronutrient for plant growth [13]
Calcium hydroxide ($Ca(OH)_2$) (g)	Used for pH or alkalinity adjustment [12]
Calcium carbonate ($CaCO_3$) (g)	Used to adjust water hardness and alkalinity [12]

Table 2.2: *Chemical and nutritional parameters used in the aquaponics system*

The air temperature and humidity were recorded daily using sensors located in Lines 1, 2, and 3. Throughout this study, these data will be referred to as Temperature and Humidity.

The water consumption in the various tanks was also recorded. This data will be referred to as Water Consumption.

The parameters assessed on a weekly basis are referred to as Weekly Laboratory Analysis. A brief description of each is provided in Table 2.3. The metals were controlled monthly, as shown in Table 2.4.

Nutrients	Description	Reference
$\mathbf{NH_4^+}$ (Ammonium)	Ion derived from ammonia ($\mathbf{NH_3}$), formed under acidic conditions. Essential for plants, but toxic to aquatic organisms at high concentrations.	[12]
$\mathbf{PO_4^{3-}}$ (Phosphate)	Phosphorus-containing compound commonly used as fertilizer. Promotes plant and algae growth; excessive amounts may cause eutrophication.	[12]
$\mathbf{NO_3^-}$ (Nitrate – UV-C Method)	Measurement of nitrate levels using UV spectroscopy. Nutrient for plants, but harmful at high levels for health and the environment.	[12]
$\mathbf{NO_2^-}$ (Nitrite)	Intermediate in nitrification/denitrification. Highly toxic to fish and aquatic organisms even at low concentrations.	[12]

Table 2.3: *Description of nutrients relevant to aquaponics*

Metals	Description	Reference
Na590 (Sodium – 590 nm)	Refers to the spectral line at 590 nm associated with sodium's emission/absorption. Sodium is essential in biological systems and present in various aquatic solutions.	[12]
K676 (Potassium – 676 nm)	Potassium is a vital macronutrient for both plant growth and animal health. The value 676 refers to its characteristic spectral line.	[12]
Ca423 (Calcium – 423 nm)	Calcium plays a key role in biological processes such as bone formation and cellular regulation. The 423 nm value corresponds to its spectral analysis.	[12]
Mg285 (Magnesium – 285 nm)	Magnesium is essential for plants and organisms, particularly as a central component of chlorophyll. The 285 nm wavelength is used in its spectrophotometric measurement.	[12]

Table 2.4: *Description of metals relevant to aquaponics analysis*

The number and type of fish introduced into the various tanks were documented. These data are referred to as Fish. Information regarding the plants and fruits produced in the various systems was also recorded. Since 2019, three main projects have been carried out, the details of which are described in the Table 2.5.

Project Number	Project Name	Start Date Project	End Date Project
Project1	Arugula & Lamb's Lettuce	20191023	20191211
Project2	Mealworms	20210427	20210608
Project3	Papaya Trees	20220502	20230504

Table 2.5: *LSMI projects at IPLeiria*

2.4 Greenhouse Data: A Closer Look at the Challenges

The main objective of the project is the development of a BI solutions using the parameters collected at the greenhouse to monitor them regarding several projects. Additionally, to visualize and explore the parameters thought dashboards sharing results efficiently with the scientific community.

However, this process faces a series of technical challenges related to data quality [14]. In the data records of the LSMI greenhouse several challenges were identified. One of the primary issues was related to data structure, particularly the fragmentation of data across multiple files without clear criteria, requiring significant effort for consolidation. Additionally, some datasets were dispersed across different files, as seen in the case of the fish records, necessitating a unification process to ensure consistency.

Another major challenge was the lack of standardization in file naming conventions, which hindered proper data organization and mapping. The improper use of merged cells further complicated the process, making it difficult for programmatic recognition of headers or specific values. Furthermore, inconsistent table placement across different files introduced additional complexity in data extraction. The column order varied from file to file, even when representing the same fields, leading to inconsistencies in data analysis. Moreover, some files contained a different number of columns, deviating from the established standard and further complicating integration.

To address these challenges, it was necessary to reorganize the data. In metting with stakeholders standardized templates were created, defining clear formatting rules to ensure uniform data entry. The greenhouse researchers were invited to use these templates when filling in the data and saving the files in the corresponding folders. This structured approach ensured that the files could later be ingested efficiently and reducing inconsistencies the data processing workflow.

One of the main challenges was the absence of data on certain dates, as seen, for example, in the **Weekly Laboratory Analysis** entries where some weeks had no records. In addition, duplicate values and discrepancies were found, resulting from human errors, which compromised the integrity of the database.

Another relevant issue was the presence of different data types within the same column, for example, textual and numerical data which adversely affected both the analysis and automated processing. Numerical data containing symbols, such as the percentage sign (%) or an asterisk (*), were also identified in certain cells, making it difficult to convert these values into a usable numerical format. The inconsistent use of decimal separators further complicated matters, as the mixed use of comma (,) and period (.) as decimal separators led to confusion and incorrect interpretations of numerical values. For instance, “1,000” could be interpreted as one thousand or one, depending on the context. Finally, another challenge was the use of inconsistent date formats, as

dates were represented in different ways, such as DD/MM/YYYY and MM/DD/YYYY, which hindered the unification and temporal analysis of the data.

Challenges related to the volume and scalability of the data were also an important concern. The continuous growth of the files over time, as new data is added, demands constant updates to the database. Moreover, periodic updates to existing data are carried out, especially when errors are identified in previously entered records, which increases the complexity of version management.

3

Cloud Ecosystems for BI and Big Data: A Technical Review

This chapter presents and develops a technical overview of cloud ecosystems in the context of BI and Big Data, along with essential definitions and concepts that support the foundation of this work.

Given the wide range of technologies currently available in this domain, a particular emphasis will be placed on the utilization of Microsoft Fabric as the designated platform for the implementation of the ETL process. Therefore, a more in-depth analysis of Fabric will be conducted. This analysis will explore the key features, and integration within modern business workflows.

3.1 Introduction to BI

As stated by [15], BI refers to a set of processes, technologies, and tools that enable organizations to collect, integrate, analyze, and present data to support strategic and operational decision making [15]. In their conventional manifestation, BI systems were generally centralized and constructed upon structured data warehouses devised using dimensional modelling methodologies, such as the star and snowflake schemas put forth by Kimball [14]. These environments were characterized by a significant reliance on rigid Information Technology (IT) processes and were predominantly managed by IT departments, exhibiting minimal flexibility for end-users. However, over the past decade, the field has undergone a significant transformation driven by advancements in cloud computing, data democratization, and real-time analytics. The contemporary paradigm of BI places significant emphasis on the capabilities of self-service, the utilization of interactive data visualizations, and the integration of a diverse array of data sources, which are often characterized by a lack of structure. This assertion is supported by the seminal work of [16] in *What is modern BI? How is it different from traditional BI?*. The objective is to empower business users to autonomously explore data

and extract insights with minimal reliance on technical teams.

The rapid growth in the volume and granularity of data generated by organizations, driven by sources such as social media, the Internet of Things (IoT), and multimedia content, has resulted in an unprecedented influx of both structured and unstructured data. This accelerated pace of data generation, commonly referred to as Big Data, has become a prominent and widely acknowledged trend [17].

3.2 An Introduction to Cloud Business Intelligencee

Big Data refers to extremely large and complex datasets that traditional data processing systems are not equipped to handle efficiently. The concept is often characterized by five key dimensions, commonly known as the 5 Vs: Volume, the massive scale of data generated continuously by users, systems, and devices; Velocity, the high speed at which data is produced and needs to be processed in real-time or near real-time; Variety, the wide range of data formats—structured, semi-structured, and unstructured—originating from diverse sources such as sensors, social media, and transactional systems; Veracity, the degree of accuracy and reliability of data, which can be affected by inconsistencies and noise; and Value, the potential insights and business impact that can be extracted from data through proper analysis. These dimensions highlight the challenges and opportunities that Big Data presents, particularly in the context of cloud computing and BI, where scalable infrastructure and advanced analytics techniques are essential to extract meaningful insights [18] [19].

Cloud computing has emerged as a transformative technology in recent years, significantly reshaping the landscape of IT service delivery [20]. It has transformed the way organizations manage, process, and analyze data, offering scalable, on-demand access to computational resources, enhanced security, parallel processing capabilities, and elastic data storage. In the context of BI, cloud platforms provide flexible environments that support data-driven decision-making at scale [17], [20].

The convergence of BI and cloud computing arises from the increasing volume, velocity, and variety of data, characteristics typically associated with Big Data, that demand more dynamic and scalable infrastructure. Traditional on-premises BI solutions often face limitations in terms of scalability, maintenance overhead, and cost efficiency [20]. Cloud-based BI platforms address these challenges by leveraging distributed computing, elastic resources, and service-oriented architectures [17], [21].

Moreover, the adoption of cloud services supports a variety of deployment and service models, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), each offering different levels of abstraction and control over the computing environment [22]. These models are particularly relevant for BI applications, as they enable organizations to tailor their analytics pipelines to specific

business needs and technical capabilities.

In the SaaS model, providers deliver business applications over the internet, eliminating the need for companies to host and manage these solutions internally [20]. This shift enables quicker implementation cycles, easier access to updates, and greater focus on data analysis rather than infrastructure management.

Recent advancements in cloud-native storage architectures have also introduced new paradigms, such as Data Lakes and Lakehouses, that are increasingly relevant for modern BI initiatives. Data lakes allow for the centralized storage of vast amounts of raw, heterogeneous data at a relatively low cost, accommodating both structured and unstructured formats [23]. However, the lack of structure and governance in data lakes can lead to challenges related to data quality and consistency [23]. To address these issues, the Lakehouse architecture has emerged as a hybrid solution that combines the scalability and flexibility of data lakes with the reliability and schema enforcement of traditional data warehouses [24]. This architecture supports robust, high-performance analytics while maintaining the agility required for modern cloud BI systems [24]. The paradigms will be more deeply analyzed in **Section 3.4**.

In the evolving landscape of BI, three prominent trends have emerged: cloud-native BI, self-service analytics, and embedded BI [16], [25]. Cloud-native BI leverages the scalability and flexibility of cloud platforms, enabling organizations to process and analyze vast datasets efficiently, a necessity highlighted by the increasing demand for real-time insights [16]. Self-service analytics empowers non-technical users to access and interpret data independently, fostering a data-driven culture and reducing reliance on IT departments [16]. Embedded BI integrates analytical capabilities directly into business applications, allowing users to access insights within their regular workflows, thereby enhancing decision-making processes [25]. Collectively, these trends are transforming BI into a more accessible, integrated, and agile tool for modern enterprises [25].

3.3 Big Data Processing Paradigms

3.3.1 Difference Between Traditional Processing vs Distributed Processing

According to [26], the term Big Data first appeared in 1997 when NASA scientists reported challenges in visualizing large data sets. Later, the consultancy firm McKinsey formalized the broader concept, emphasizing its growing importance. The processing life cycle of big data includes acquisition, preprocessing, storage, security, analysis and visualization. These challenges and processes have become central to modern data science [26].

According to International Data Corporation (IDC), it is expected that the world data will grow at a compound annual rate of 61% that is from 33 Zettabytes in 2018 to 175

Zettabytes by 2025 (Figure 3.1) [19].

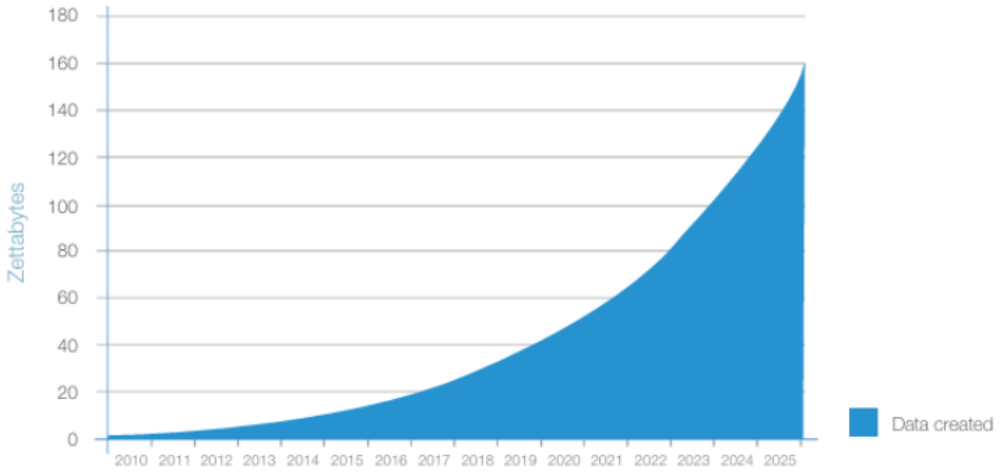


Figure 3.1: *Evolution of Big Data in* [27]

Traditional relational databases and data warehouses were designed for structured data with fixed schemas [28]. Relational Database Management Systems (RDBMS) were widely used for data storage but began facing limitations with the rise of big data. The massive growth in data volume, often reaching terabytes and petabytes, surpassed the capacity of traditional systems. Upgrades in hardware to cope with this growth led to higher costs. Additionally, RDBMS struggled with handling semi-structured and unstructured data, which now make up most of data types. They also proved inefficient in processing high-velocity data streams, prompting the shift toward big data technologies [27]. In response, Yahoo developed Hadoop in 2006 as an open-source Apache project to support distributed processing of big data across clusters [28].

IDC categorizes the evolution of data usage into three main platforms, Figure 3.2. The 1st Platform (before 1980) centralized data and processing within mainframes located in dedicated datacenters, primarily for business purposes. The 2nd Platform (1980–2000) introduced personal computers and digital entertainment, with datacenters distributing data across emerging networks to personal devices. The 3rd Platform (2000 to present) is defined by high-speed networks and cloud computing, enabling data access from any device, such as smartphones, wearables, and gaming consoles, reducing the need for local storage and transforming data usage across business and social environments [27].

3.3.2 Scalable Distributed Computing with Hadoop and Spark

Distributed data processing relies on dividing large, data-intensive tasks into smaller subtasks that are executed in parallel across cloud-based infrastructures. These infrastructures consist of containerized systems, virtual machines, or serverless computing platforms provided by major cloud vendors like Amazon Web Services (AWS), Azure,

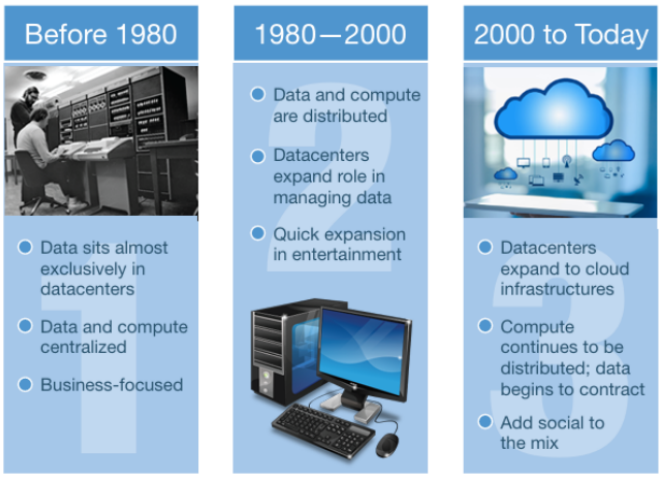


Figure 3.2: *Evolution of Computing in* [27]

and Google Cloud Platform (GCP). This distributed architecture enhances processing speed, resource efficiency, and fault tolerance. Core components such as data partitioning, parallel computing, and dynamic service allocation allow for scalable and cost-effective analytics. Frameworks like Apache Spark further support the design and execution of distributed workflows, empowering organizations to harness the full potential of cloud computing for large-scale data processing [29].

One of the foundational technologies in this domain is Apache Hadoop, which introduced the MapReduce programming paradigm [30]. MapReduce structures computations into two main phases: the Map phase processes input data in parallel across nodes, while the Reduce phase aggregates intermediate results. This model provides fault tolerance and scalability through distributed storage via Hadoop Distributed File System (HDFS) and computation. Despite its robustness, Hadoop's disk-based intermediate steps can be inefficient for iterative or real-time tasks [30].

To overcome these limitations, Apache Spark emerged as a high-performance alternative designed for in-memory processing. At its core, Spark uses Resilient Distributed Datasets (RDDs), which are immutable, distributed collections of objects that support parallel operations. Computations on RDDs are represented as Directed Acyclic Graphs (DAGs), enabling the system to optimize execution plans and reduce unnecessary data shuffling. Additionally, Spark employs lazy evaluation, meaning transformations are not immediately executed but are only triggered when an action is invoked. This allows Spark to group operations for more efficient execution [31], [32].

Cloud integration has significantly enhanced the accessibility and scalability of these frameworks. Apache Spark on Microsoft and Google Cloud allow users to provision and manage Spark clusters in a cloud-native environment. Databricks, a unified analytics platform founded by the creators of Spark, offers managed Spark clusters with collaborative notebooks, optimized performance, and seamless integration with cloud

storage, making it a popular choice for enterprise-scale data analytics [24], [32].

Overall, frameworks like Hadoop and Spark, especially when integrated with cloud-native tools, form the backbone of scalable, efficient, and resilient data analytics infrastructures.

3.4 Modern Data Architectures for Analytics

In the contemporary landscape of analytics, there is an increasing demand for data architectures that exhibit flexibility, scalability, and efficiency. These architectures must evolve beyond the conventional Data Warehouse (DW) model. Over the past decade, there has been a significant evolution from rigid, schema-on-write DW to more adaptable, schema-on-read Data Lakes, and more recently, to the hybrid model of Lakehouses. These architectures are designed to manage the increasing volume, velocity, and variety of data generated by modern applications [24].

3.4.1 From Data Warehouse to Data Lake and Lakehouse

Recent studies suggest that the traditional DW architecture is expected to decline in relevance over the coming years, being gradually replaced by a more modern and flexible architectural paradigm known as the Lakehouse. This emerging model is distinguished by its utilization of open, direct-access file formats, including Apache Parquet [24].

As illustrated in Figure 3.3, the evolution of data platform architectures has undergone three distinct transitions over time. The first model, which represents first-generation platforms, is predicated on a conventional centralized DW. In this architecture, the data is extracted, transformed, and loaded into a centralized warehouse optimized for analytical queries. The utilization of data warehouses is predominantly oriented towards the facilitation of tools and reporting systems that are of a more immediate nature. While this model is characterized by its robustness and reliability, it is important to note its limitations in processing unstructured data. Furthermore, as data volumes increase, this model faces scalability and cost challenges [24] [33].

As data became increasingly diverse and unstructured, a two-tier architecture emerged, as illustrated in the second model. In this model, raw data is first ingested into a data lake, which can store structured, semi-structured, and unstructured data formats, including images, videos, and documents. The Data Lake has the capacity to process and transfer information to a DW via ETL for traditional BI consumption. This approach offers increased flexibility and lower storage costs; however, it introduces architectural complexity and often leads to data duplication across systems and pipelines [24] [33].

The third model represents the most recent development: the Lakehouse architecture.

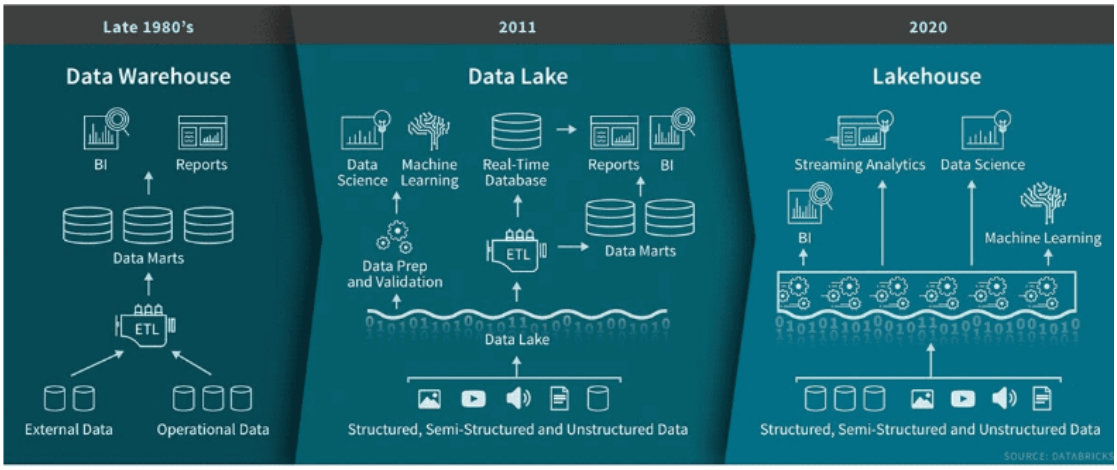


Figure 3.3: *Adapted illustration of the evolution of data platform architectures in* [34]*, based on the original figure from article:"Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics"* [24].

The objective of Lakehouse platforms is to integrate the governance, reliability, and performance characteristics of data warehouses with the scalability and flexibility of data lakes within a cohesive architecture [24] [35]. The data persists in the data lake, augmented by layers that facilitate metadata management, indexing, and caching, thereby enabling high-performance analytical queries [35]. This architecture facilitates the concurrent execution of big data, reporting, data science, and machine learning workloads on the same data, eliminating the necessity for replication. Consequently, the Lakehouse model has been shown to simplify data infrastructure, reduce operational overhead, and enhance consistency and accessibility across analytical environments [24].

Before presenting the comparison between DW, Data Lake, and Lakehouse architectures, it is important to clarify the concept of ACID. This acronym stands for Atomicity, Consistency, Isolation, and Durability, four fundamental properties that ensure data integrity within transactional systems [36].

- **Atomicity** guarantees that each transaction is executed fully or not at all, preventing partial updates and data corruption in case of failure.
- **Consistency** ensures that transactions transition the database from one valid state to another, preserving data integrity.
- **Isolation** protects concurrent transactions from interfering with each other, allowing them to behave as if executed sequentially.
- **Durability** ensures that once a transaction is successfully committed, its changes persist permanently, even after system failures.

OLTP systems prioritize fast, frequent transactions, while OLAP systems handle complex analytical queries over large datasets. Traditionally, OLAP emphasized query performance over transactional integrity [37]. However, with the addition of data ingestion (batch and streaming) and concurrent querying, ACID compliance has become crucial to ensure data reliability and correctness in modern OLAP systems [24].

As mentioned earlier, the data lake presents several issues, particularly the lack of essential management capabilities, such as ACID transaction and efficient access mechanisms like indexing, which are necessary to match the performance of traditional data warehouses [24].

ACID transactions are essential for data reliability, consistency, and integrity in data systems. Traditional data lakes lacked these guarantees, prompting the evolution of the Lakehouse architecture. By adding transactional capabilities over scalable storage using open table formats, Lakehouses unify the flexibility of data lakes with the reliability of databases, enabling consistent analytics, real-time ingestion, and concurrent operations [24].

To support a clear understanding of the distinctions between traditional Data Warehouses, Data Lakes, and the Lakehouse paradigm, Table 3.1 presents a comparative overview of their key characteristics. Given the flexibility, scalability, analytical capabilities, and BI potential of both Data Lakes and Data Warehouses, Lakehouse data management is rapidly emerging as an industry standard particularly appealing to growing, cost-conscious organizations [33]. The table highlights the unique strengths of each architecture across several relevant criteria, including supported data types, access mechanisms, performance, governance, scalability, cost, and quality. It also considers compliance with ACID properties to ensure transactional reliability, customization flexibility for developers, and the usability of currently compatible tools and storage systems. This comparison offers a comprehensive foundation for selecting the most appropriate solution based on specific analytical and operational requirements [33].

3.4.2 The Medallion Architecture: Bronze, Silver, and Gold Layers

The Medallion Architecture is a contemporary data design pattern that has gained significant traction on platforms such as Databricks, where it has been shown to enhance the efficiency of data transformation pipelines. The architecture is structured into three distinct layers: Bronze, Silver, and Gold. This structural design promotes clarity, traceability, and scalability in data processing [40] [41]. The system under consideration has been demonstrated to offer support for both Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) systems [40].

Although Databricks popularized the Medallion Architecture [40], the foundational idea of organizing data into layers had already been explored by experts such as Ralph Kimball and Bill Inmon in *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* in the year 2013 [14]. Kimball also introduced dimensional modelling via a bottom-up approach based on star schemas with fact and dimension tables effectively embodying the progressive refinement of raw data for analysis, long before the "bronze," "silver," and "gold" terminology emerged [42].

Criteria | Source	Data Warehouse	Data Lake	Lakehouse
Importance [33]	Data analytics and business intelligence	Machine Learning (ML) and artificial intelligence	Both data analytics and machine learning
Data [24]	Relational data from transactional systems, operational databases, and business applications	All data including structured, semi-structured, and unstructured	Query all kinds of data, including image, audio, video, and others
Data type [33] [38]	Structured	Semi-structured and unstructured	Structured, semi-structured, and unstructured
Usability [33]	Users can easily access and report data	Analyzing vast amounts of raw data without tools that classify and catalog the data can be arduous	Combines the structure and simplicity of a DW
Data Access [33]	SQL only	Open API*, SQL, Python	Open API, SQL, Python
ACID Conformity [33]	Guarantees the greatest levels of integrity; data is recorded in an ACID-compliant way	Updates and deletes are difficult procedures that need non-ACID compliance	ACID-compliant to ensure consistency when several parties read or write data simultaneously
Cost [33] [38]	Expensive and time-consuming	Inexpensive, quick, and adaptable	Inexpensive, quick, and adaptable
Usability [33]	Users can easily access and report data	Complex to analyze vast amounts of raw data without classification and cataloging tools	Combines the structure and simplicity of a DW with the broader use cases of a Deep Learning (DL)
Quality [33]	High	Low	High
Scaling [33]	Vertical scaling	Horizontally scalable	Horizontally scalable
Data Governance [38]	Built-in governance features for data quality and integrity	Requires additional tools for effective governance	Hybrid approach, leverages data warehouse features for better governance
Use Cases	BI, regulatory reporting, performance dashboards, structured trend analysis [39]	AI, ML and data science [39]	Combines the structure and simplicity of a DW with the broader use cases of a DL [38]

Table 3.1: *Comparison of Data Warehouse, Data Lake, and Lakehouse Architectures*

Conversely, Inmon proposed a top-down strategy focused on a centralized, normalized enterprise DW [43]. Despite their structural differences, both approaches introduced key ideas such as data staging, transformation, and presentation layers. These concepts closely align with the tiered design of the Medallion Architecture.

Considering the aforementioned advantages, the Medallion Architecture is a prevalent choice within the Lakehouse paradigm. It functions as a logical design pattern intended to incrementally enhance the structure and quality of data as it progresses through successive layers (Figure 3.4), which organizes data into incrementally refined layers to support data quality, reusability, and traceability [41].

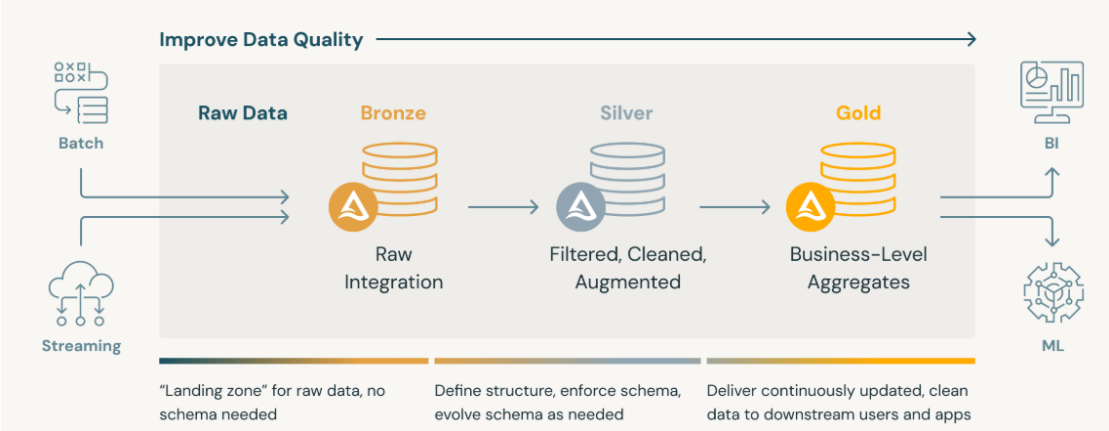


Figure 3.4: *Medallion Architecture in* [41]

Below are the descriptions of the core layers (Bronze, Silver, and Gold) commonly used in data Lakehouse architectures. Each layer plays a distinct role in transforming raw data into structured, analytics-ready information [41].

- **Bronze Layer:** Contains raw, ingested data in its original format, directly from source systems. This layer ensures full data lineage and auditability [41].
- **Silver Layer:** Holds cleaned, filtered, and structured data, often integrating data from multiple sources. It is optimized for standard reporting and business logic [41].
- **Gold Layer:** Provides curated, aggregated, and analytics-ready data tailored for consumption by business intelligence tools or data science workflows [41].

The Lakehouse architecture offers several key benefits, including a simple and intuitive data model that is easy to implement, support for incremental ETL processes, the ability to recreate tables from raw data at any time, and robust data management capabilities such as ACID transactions and time travel [41].

3.4.3 Key Technologies and Benefits

In the context of a Lakehouse architecture, a table format is the layer that organizes raw data from the data lake into table-like formats, making it easy to find what you're looking for. It adds features such as ACID transactions, schema evolution, and time travel, allowing the Lakehouse to combine the flexibility of a data lake with the performance and governance of a DW [44].

The three leading Lakehouse table formats are: Delta Lake, Apache Hudi, and Apache Iceberg [35].

- **Delta Lake** is an open-source storage format that enhances Parquet files with a transaction log, providing reliability and robustness to data lakes. Provides ACID transactions, scalable metadata handling, and unifies batch and streaming data processing on top of existing Data Lake storage [45].
- **Apache Iceberg** is an open-source table format that treats metadata as first-class data, storing file listings, schema versions, and snapshots in immutable manifests. This design enables fast metadata queries, incremental streaming ingestion, zero-downtime schema evolution, and rapid partition pruning—even at multi-billion–object scale—dramatically improving performance and manageability in large data lakes. [46].
- **Apache Hudi** is a framework designed to facilitate fast updates and deletes as well as incremental processing on top of data file systems, just like Delta Lake and Apache Iceberg [47].

The Figure 3.5 represents a key feature comparison between Delta Lake, Apache Iceberg and Apache Hudi. Among the platforms that leverage modern table formats to deliver robust Lakehouse solutions, Databricks stands out as a clear industry leader. This is evidenced by its recognition in the Forrester Wave report.

Databricks named a Leader in the 2024 Forrester Wave for Data Lakehouses in Q2 2024, Figure 3.6. Databricks is the top platform in the Lakehouse ecosystem. The Forrester Wave chart looks at different vendors in two ways: how good their current services are and how strong their plans are. Databricks is the best in the Leaders quadrant, with the most functionality and strategic vision [49].

Other platforms, such as Google with BigQuery, and Snowflake, are also considered leaders in the field, but they seem to trail Databricks in at least one of the two areas. Microsoft and AWS are classified as Strong Performers. This means they have solid capabilities and are widely used, but they are less specialized in Lakehouse-specific functionalities. Vendors like Oracle, Teradata, and Salesforce are considered strong competitors, while platforms such as IBM, SAP, and Alibaba Cloud are seen as challengers, offering a more limited range of strategies and features in this area.

This evaluation shows that Databricks is a key player in the Lakehouse paradigm, es-

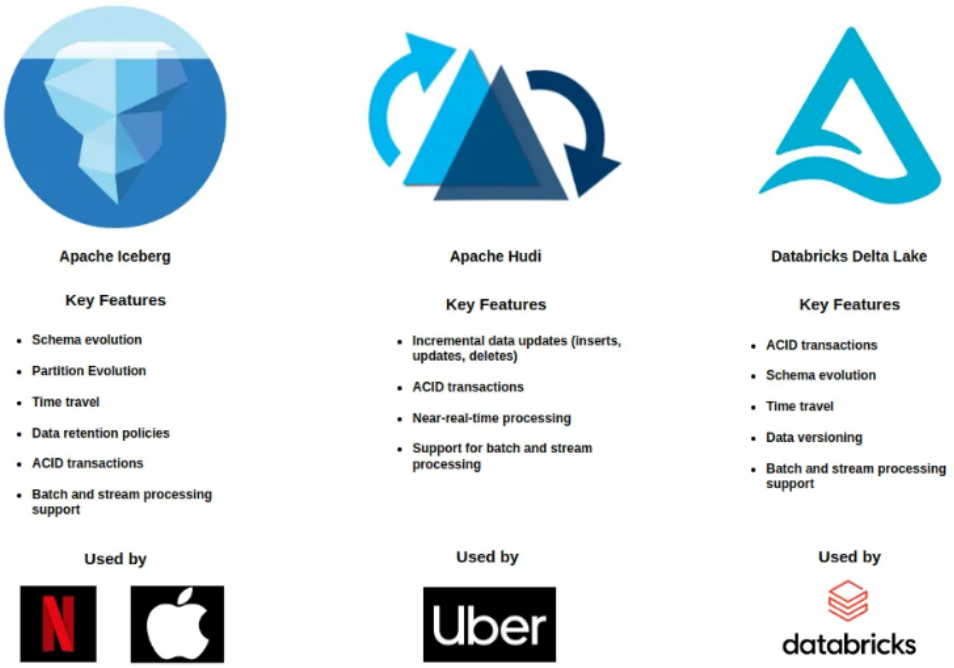


Figure 3.5: *Key feature comparison in article* Data lake Table formats: Apache Iceberg vs Apache Hudi vs Delta lake [*48*]

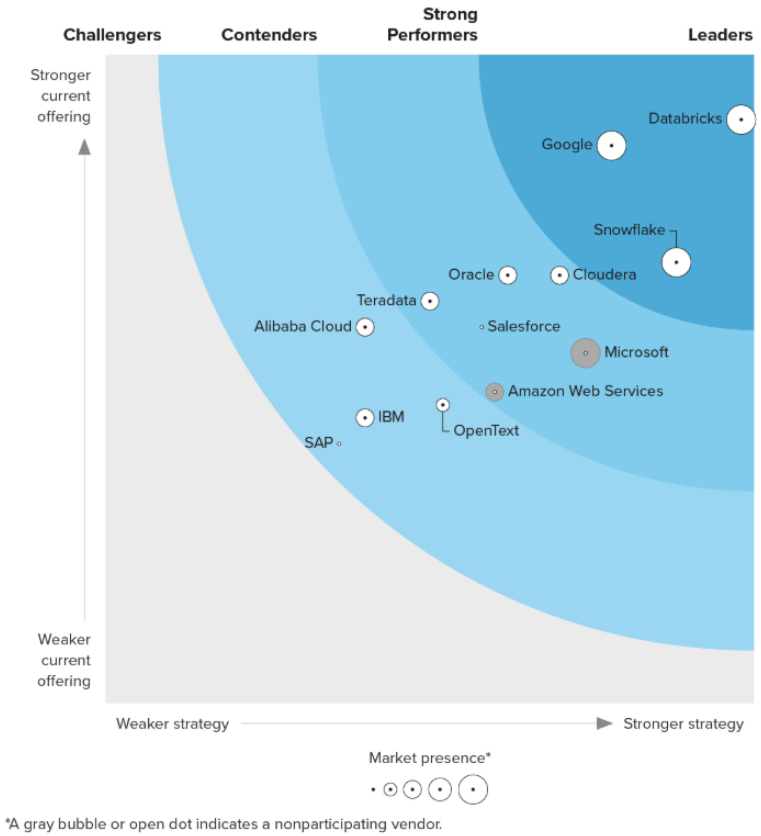


Figure 3.6: *Results of Forrester Wave for Data Lakehouses in Q2 2024 in* [*49*]

pecially because it supports Delta Lake, uses the Medallion Architecture, and works well with AI and ML. Its leadership position shows its commitment to innovation, performance, and scalability in modern data platforms.

Based on the information above, we will now look at the main differences between Databricks, Microsoft Fabric, Snowflake, and Google Cloud BigQuery. We will focus on what they can do and how they work together in the modern data ecosystem. Microsoft Fabric didn't take part in the official review of the Forrester Wave for Data Lakehouses report, and it's considered less specialized in Lakehouse-specific features. However, it's still considered in this analysis Figure 3.6 because it's a major player in the market. The size of its representation in the report's visual (even though it wasn't assessed in detail) seems about the same as that of Google and Databricks. This suggests that it has a similar level of industry relevance Figure 3.6.

The Table 3.2, provides a comparative analysis of four leading data platforms (Google Cloud BigQuery, Databricks, Snowflake and Microsoft Fabric) focusing on their architectural and operational characteristics. Key aspects such as billing models, scalability, data sharing, integration capabilities, and governance mechanisms are explored to highlight each platform's strengths and positioning within the Lakehouse ecosystem.

Modern data architectures bring numerous benefits, including:

- **Data Reliability and Consistency:** Technologies like Delta Lake and Apache Iceberg implement ACID transactions, ensuring data consistency across operations. This provides a solid foundation for building trustworthy analytics pipelines [24], [45].
- **Time Travel and Data Versioning:** Delta Lake and Iceberg enable time travel capabilities, allowing users to access historical versions of data. This is particularly useful for debugging, auditing, and reproducibility in machine learning and reporting tasks [45].
- **Scalability and Engine Interoperability:** Modern data lake technologies are designed to scale efficiently with increasing data volumes and support distributed processing across a wide range of execution engines. This allows organizations to decouple storage from compute, choose the best processing engine for each use case, and adopt a modular architecture that grows with evolving business and analytical needs [24].
- **Schema Evolution and Flexibility:** All three technologies support schema evolution, enabling changes to data structure over time without compromising data integrity or requiring costly migrations [24].

These capabilities are essential for building enterprise-scale analytics systems that are both agile and resilient in the face of rapidly evolving data landscapes.

Feature	Databricks	Snowflake	Microsoft Fabric	Google Cloud: BigQuery
Billing Model	Consumption-based pricing using Databricks Units for compute resources; separate charges for storage; offers reserved capacity for cost savings. [50]	Compute credits (virtual warehouses) and storage usage; options for pre-purchasing credits are offered at discounted rates. [50]	Capacity-based (pay-as-you-go) model. [51]	Pay-as-you-go model; billed by usage and resource consumption. [52], [53]
Primary Users	Data scientists, data engineers, and analysts proficient in Python, Scala, or R. [50]	Data analysts, BI professionals, engineers familiar with SQL. [50]	Data scientists, data engineers, and analysts are proficient in Python, Scala, or R; BI professionals; includes BI professionals. [51]	Data scientists, data engineers, and analysts are proficient in Python, Scala. [54], [55]
Scalability	Auto-scaling clusters for distributed workloads with Spark. [50]	Automatic scaling by resizing virtual warehouses (scale up) and adding clusters (scale out) for concurrency; designed for effortless scalability. [50]	Elastic scaling with capacity pricing. [56]	Scales without managing infrastructure; integrated BI. [54]
Data Structure Support	Structured, semi-structured, unstructured; Lakehouse optimized. [50]	Structured and semi-structured, some unstructured via external stages. [50]	Supports structured and unstructured data. [56]	Blended pipelines with structured/unstructured data. [53], [54]
Data Sharing	Delta Sharing, an open protocol for secure data sharing across platforms, and provides Databricks Marketplace for data exchange and collaboration. [50]	Secure Data Sharing to share live data between Snowflake accounts without copying; offers Snowflake Data Marketplace for third-party data access. [50]	Centralized data discovery that simplifies governance, sharing, and access. [51]	Secure data sharing without copying or moving the underlying data, allowing partners to collaborate directly within the platform. [54]
Governance & Security	Role-based access, strong encryption. [56]	Advanced security and compliance features. [56]	Built-in Purview for governance. [56]	Access Management, Cloud Security Command Center and Cloud Key Management Service. [53]
Integration	Open-source and 3rd-party integration. [56]	Native connectors, supports Kafka, NiFi, Fivetran. [53]	Automated data workflows with Data Pipelines. [51]	Data Fusion visual interface for pipelines. [53]
BI & Reporting	Requires external BI tools. [56]	Supports third-party BI. [56]	Native Power BI integration. [56]	Supports third-party BI. [54]
Table Format	Delta Lake and Apache Iceberg [57] [58]	Apache Iceberg. [59]	Delta Lake and Apache Iceberg (preview). [60] [61]	Delta Lake, Apache Hudi and Apache Iceberg. [54]

Table 3.2: *Comparison between Databricks, Snowflake, Microsoft Fabric, and Google Cloud BigQuery based on architectural and operational features*

Microsoft Fabric was selected to implement the ETL process for the aquaponics system. The following section provides a detailed overview of this platform, and the selection rationale is discussed in **Chapter 6**.

3.5 Microsoft Fabric as an Analytical Platform

Microsoft Fabric is an analytics platform made for businesses offering end-to-end capabilities such as data ingestion, processing, transformation, and reporting. It brings together services like Data Engineering, Real-Time Intelligence, and Data Science in a unified SaaS environment. By centralizing storage with OneLake and embedding AI features, Fabric simplifies the data lifecycle, from raw data to insights, without requiring complex integration efforts [51].

3.5.1 OneLake: The Unification of Lakehouses

Microsoft Fabric is built on a unified architecture that integrates the Lakehouse model with OneLake, its centralized data lake. OneLake serves as the foundation for all Fabric workloads, providing a single, tenant-wide storage system based on Azure Data Lake Storage Gen2. It simplifies data access and governance by removing the complexity of traditional cloud infrastructure and eliminating the need for an Azure account. Designed to prevent data silos, OneLake ensures consistent policy enforcement and facilitates easy data discovery and sharing. Its hierarchical structure allows data to be organized across tenants, workspaces, and containers, supporting efficient and scalable management across users, regions, and environments [51].

3.5.2 Items of Microsoft Fabric

Microsoft Fabric includes a range of items, each designed to support specific user roles and address distinct tasks. These items are functionally grouped based on the main stages of the data engineering lifecycle within Microsoft Fabric. Specifically, they are organized into 8 categories, Table 3.3 [62]. Since this thesis involves the implementation of an ETL process, the following subsection will focus specifically on the relevant items of Microsoft Fabric.

3.5.3 Key Microsoft Fabric Items for the Scope of this Thesis

Get data: Dataflows Gen2

The Get Data function includes the item Dataflows Gen2, which are used to ingest and transform data from multiple sources.

The cleansed data is then loaded into a target destination. These dataflows can be integrated into data pipelines for more complex orchestration and can also serve as

Name	Description
Get Data	Ingest batch and real-time data into a single location within your Fabric workspace.
Store Data	Organize, query, and store your ingested data in an easily retrievable format.
Prepare Data	Clean, transform, extract, and load your data for analysis and modelling tasks.
Analyze and Train Data	Propose hypotheses, train models, and explore your data to make decisions and predictions.
Track Data	Monitor your streaming or nearly real-time operational data, and make decisions based on gained insights.
Data Engineering	Based on Apache Spark, it supports the creation, management, and optimization of data pipelines. Integrates with Data Factory to schedule notebooks and Spark jobs.
Visualize Data	Present your data as rich visualizations and insights that can be shared with others.
Develop Data	Create and build your software, applications, and data solutions.
Others	Find unique or third-party provided functionality that builds on Fabric's core capabilities.

Table 3.3: *Fabric Capabilities and Their Descriptions*

data sources in Power BI. Dataflows Gen2 are cloud-based ETL tools that enable users to extract, transform, and load data using a visual interface like Power Query Online [63].

Store Data: Warehouse

Warehouse in Microsoft Fabric, is a modern lake-centric solution rather than a traditional DW. It includes two key components: the Fabric Warehouse and the SQL Analytics Endpoint, both designed to offer high performance with lower costs and simplified management. The Fabric Warehouse specifically provides full transactional capabilities (supporting Data Definition Language (DDL) and Data Manipulation Language (DML)) and is identified by the "Warehouse" label in the workspace [64].

Store Data: Lakehouse

Microsoft Fabric Lakehouse, is a unified platform that combines the scalability of a data lake with the structure and performance of a DW. It enables the storage, management, and analysis of both structured and unstructured data in a single location, integrating with data engineering and analytics tools to deliver a comprehensive and scalable solution [65].

To guide the selection between Lakehouse and Warehouse functionalities within Microsoft Fabric, a decision-making algorithm, Figure 3.7, is provided based on three main criteria: development approach, warehousing requirements, and data complexity [66].

Firstly, the choice depends on the preferred development method. Users who opt for Spark-based development are better served by the Lakehouse model, which is optimized for this environment. Conversely, those who prefer to use T-SQL for querying

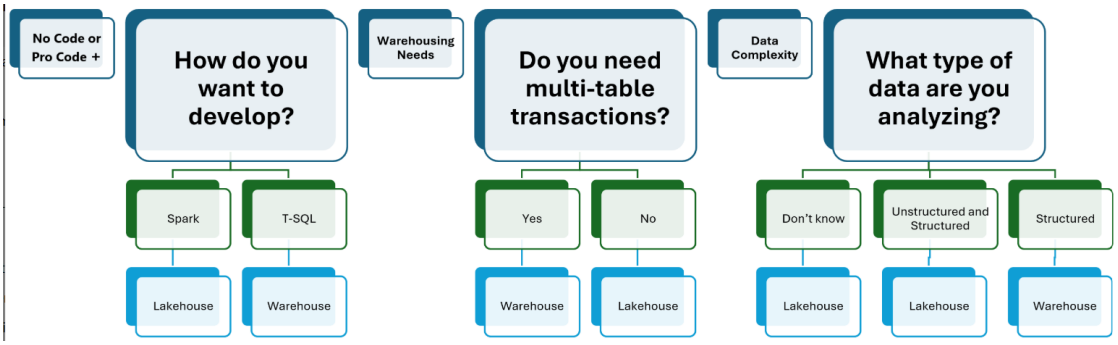


Figure 3.7: *Guidelines for Choosing Between Lakehouse and Warehouse in Microsoft Fabric in* [66]

and development should choose the Warehouse option, which provides full SQL support and transactional capabilities [66]. Secondly, the need for multi-table transactions if the use case requires transactions across multiple tables, the Warehouse is the appropriate choice. If this requirement does not exist, the Lakehouse offers a more flexible alternative for data processing [66]. Finally, if data includes unstructured formats or is not clearly defined, Lakehouse is recommended for its flexibility. For strictly structured data, Warehouse is preferred due to its performance and advanced SQL capabilities [66].

The Table 3.4, compares the Warehouse to the SQL analytics endpoint of the Lakehouse.

Feature	Warehouse	SQL Analytics Endpoint - Lakehouse
Primary Capabilities	ACID compliant, full T-SQL transaction support.	Read-only endpoint for T-SQL querying over Lakehouse.
Developer Profile	SQL Developers or citizen developers.	Data Engineers or SQL Developers.
Data Loading	SQL, pipelines, dataflows.	Spark, pipelines, dataflows, shortcuts.
Delta Table Support	Reads and writes Delta tables.	Reads Delta tables.
Storage Layer	Open Data Format - Delta.	Open Data Format - Delta.
Recommended Use Case	Enterprise data warehousing and advanced BI scenarios.	Departmental/self-service warehousing, medallion architecture support.
Development Experience	Full T-SQL ingestion, modelling, development, querying. Full tool support.	UI for modelling/querying. Limited tool support.
T-SQL Capabilities	Full DML, and DDL with full transaction support.	Full no DML, limited DDL (views).

Table 3.4: *Comparison of Microsoft Fabric Warehouse and SQL Analytics Endpoint*

Prepare data: Notebook

The Microsoft Fabric notebook, is a central tool for developing Apache Spark jobs and machine learning solutions.

It provides a web-based, interactive environment designed for data engineers and scientists to perform data ingestion, transformation, and analysis. Notebooks support multiple languages, including T-SQL, PySpark, Spark SQL, Spark R and Scala. Enabling flexibility in data processing and analytics tasks. With no setup required, users benefit from built-in data visualizations, enterprise-level security, and support for var-

ious file formats such as CSV, JavaScript Object Notation (JSON), Parquet, and Delta Lake. This makes notebooks an efficient and versatile environment for both data engineering and data science workloads [67].

Prepare data: Data Pipeline

Data pipelines, in Microsoft Fabric provide robust workflow capabilities at cloud scale, enabling the construction of complex ETL and data factory workflows.

These pipelines can orchestrate various tasks, such as refreshing dataflows, moving petabyte-scale data, and executing logic using built-in control flow features like loops and conditionals. They support both low-code and code-first approaches, allowing users to combine configuration-driven copy activities with dataflow refreshes, and to integrate Spark notebooks, SQL scripts, and stored procedures within a single, end-to-end ETL process, Figure 3.8 [68].

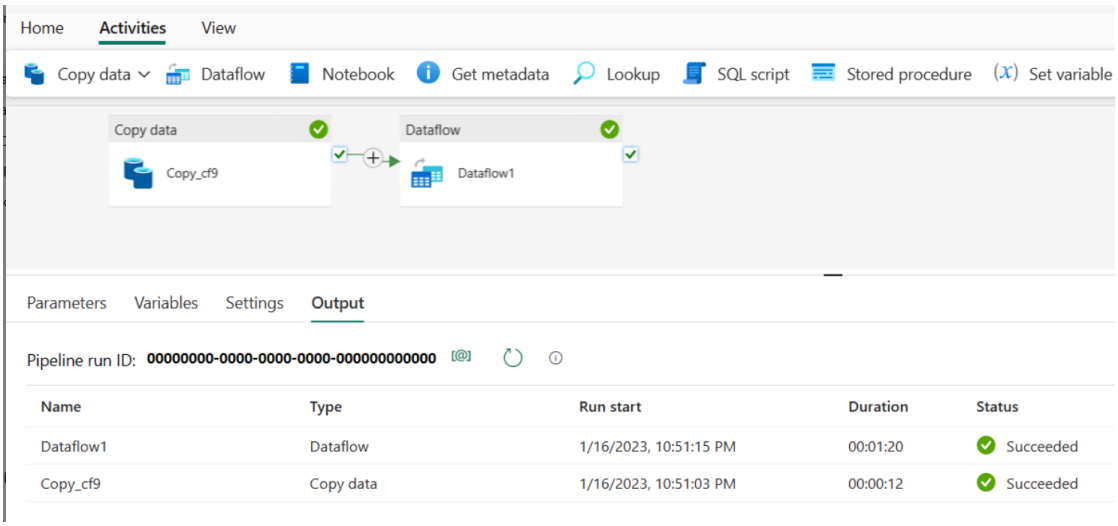


Figure 3.8: *Data Pipeline Interface in* [62]

3.6 Real-World Impact: Business Applications

The Lakehouse architecture is transforming how industries harness data by enabling scalable analytics, real-time insights, and AI-driven decision-making across diverse sectors. The following applications will now be presented:

- Healthcare and Life Sciences
- Finance and Banking
- Public Sector and Smart Cities
- Education and Research
- Retail and E-commerce
- Environmental Sciences, Biology, and Aquaponics

In the healthcare sector, the Lakehouse architecture is used for genomic data analysis, patient monitoring, and predictive diagnostics. Databricks introduced the Lakehouse for Healthcare and Life Sciences, aiming to improve health outcomes through data collaboration and AI - driven insights [69].

Financial institutions are adopting the Lakehouse architecture for fraud detection, risk management, and regulatory compliance. Databricks launched the Lakehouse for Financial Services to accelerate data-driven innovation across the industry [70].

Governments and municipalities use data platforms for traffic monitoring, public safety, and open data initiatives. Opendatasoft discusses how open data can help build smarter and more connected communities [71].

In education and research, Data Lakes store large academic datasets, enabling reproducible experiments and interdisciplinary collaboration. Nature presents SciSciNet, a large-scale open data lake designed for science of science research [72].

Retailers and e-commerce platforms are leveraging Lakehouse architectures to unify customer, sales, inventory, and marketing data in a single platform. This enables advanced customer analytics, real-time personalized recommendations, dynamic pricing, and accurate demand forecasting. For example, companies like H&M have used Lakehouse solutions to improve customer engagement, optimize inventory levels, and streamline supply chain operations through AI-powered insights and predictive models [73].

In the field of scientific research, leveraging Lakehouse architectures offers a powerful solution to the growing need for scalable, integrated data management. Natural sciences, such as environmental monitoring, biology, and aquaponics, routinely generate vast volumes of heterogeneous data, including experimental results, sensor outputs (e.g., temperature, humidity, water quality), genomic sequences, and unstructured field notes. The ability to store, manage, and analyze all of this information within a unified platform is critical for accelerating research, improving reproducibility, and enabling interdisciplinary collaboration.

Lakehouse architectures provide a framework for real-time data collection, long-term ecological data storage, and advanced analytics aimed at optimizing biological systems, such as plant and fish growth in aquaponic environments. By eliminating traditional data silos and supporting both structured and unstructured data, Lakehouses empower researchers to derive insights more efficiently and make data-driven decisions that enhance scientific reasoning and discovery.

4

Solution Overview and BI Arquitecture

4.1 Solution Overview and BI Architecture

Based on the challenges outlined in **Section 2.4** the following solutions have been proposed to address these issues and ensure data integrity, consistency, and scalability within a BI environment. First it is necessary to develop and implement a robust process for consolidating data scattered across multiple Excel files into a unified, structured format. This process will involve establishing clear criteria for file organization, enforcing standardized naming conventions, and ensuring consistent table structures and column orders throughout all files, as well as establishing validation and governance rules to guarantee data quality and consistency.

In addition, a comprehensive data cleansing strategy will be designed and executed to tackle issues such as missing records, duplicate entries, discrepancies, and the coexistence of mixed data types within the same column. Specific actions under this objective include converting numerical data containing extraneous symbols (e.g., %, *) into an usable format, standardizing decimal separators, and unifying date formats to facilitate accurate temporal analysis.

Performance optimization and scalability are also critical. Accordingly, an efficient ETL process will be developed to manage large volumes of data while minimizing performance bottlenecks and memory consumption. This objective further includes developing mechanisms for regular updates and effective version management as new data are added or existing records corrected. To streamline the overall ingestion process, automated data processing solutions will be implemented to reduce manual intervention and minimize the risk of human error. By leveraging modern scripting and data processing tools.

Moreover, the adoption of modern BI technologies and best practices is essential for constructing a robust data architecture that supports advanced data analysis and visualization, thereby enabling robust analyses and data-driven decision-making.

Finally, the goal is to empower researchers by developing a dashboard that facilitates the analysis and monitoring of critical parameters within the aquaponics system. This tool will integrate data from DW, enabling researchers to extract actionable insights and promptly identify any deviations or emerging trends, thereby contributing significantly to the advancement of aquaponics research.

Together, these solution form a comprehensive framework designed to overcome the identified challenges and ensure a high level of performance and data quality in the BI environment.

4.2 Strategies to Overcome Challenges

To effectively address the challenges outlined above, a comprehensive set of strategies has been developed that focuses on automation, standardization, validation, consolidation, scalability, and governance.

Automation of File Renaming and Organization:

- Utilize scripts to standardize file names and ensure structural consistency.
- Validate the presence and order of columns based on a predefined schema.

Standardization of Types and Formats:

- Implement routines to standardize numerical and date formats during data ingestion.
- Use regular expressions to clean data (e.g., removing symbols such as %, $).
- Enforce data type conversion throughout the ETL process.

Data Cleansing and Deduplication:

- Detect and address inconsistencies
- Implement deduplication based on unique keys.

Consolidation and Mapping:

- Develop pipelines that unify data from multiple worksheets into a single table.
- Create logic to identify relationships between different files and records.

Scalability and Performance:

- Employ optimized libraries for processing large volumes of data, such as Apache Spark.
- Adopt cloud-based ingestion solutions, such as Pipelines of Microsoft Fabric.

Documentation and Data Governance:

- Document the format requirements and validation steps to reduce reliance on implicit knowledge.
- Ensure that future files comply with established standards by using pre-approved templates.

With these measures in place, it is possible to mitigate the challenges related to Excel data ingestion, ensuring that the process remains efficient and scalable.

4.3 Methods

The Lakehouse architecture was chosen for this thesis leveraging modern data management and analysis technologies to build reports for data analysis and visualization.

The selection of the Lakehouse architecture for this project is grounded in a comparative analysis of the key characteristics of traditional DW, Data Lakes, and Lakehouse systems, as presented in Table 3.1 in **Section 3.4.1**.

One of the most compelling reasons for adopting the Lakehouse architecture is its versatility in terms of performance and usability, the Lakehouse offers faster and deeper insights without requiring data movement, by combining the structured nature of DWs with the low-cost, scalable storage of Data Lakes. Users benefit from the simplicity of a DW interface and structure while maintaining the flexibility and accessibility typical of Data Lake systems. This makes the Lakehouse especially suitable for dynamic environments where rapid analysis and decision-making are crucial.

From a cost and scalability perspective, the Lakehouse aligns with the advantages of Data Lakes offering low-cost scaling regardless of data type, while avoiding the expensive and rigid infrastructure typically associated with DWs. This economic efficiency, combined with adaptability, makes the Lakehouse ideal for organizations aiming to manage growing volumes of data without compromising on performance or governance.

Moreover, the Lakehouse architecture is ACID compliant, ensuring data consistency and integrity even in concurrent operations, a significant improvement over traditional Data Lakes, which lack robust transactional guarantees. This characteristic is essential for enterprise, grade data solutions where data reliability is critical.

Finally, the Lakehouse supports a broad range of use cases, including BI, ML, and Data Science. This multi-purpose capability reduces the need for multiple data platforms and tools, promoting architectural simplicity and operational efficiency. Although the scope of this work focuses on BI, future developments may extend into the field of Data Science, for instance, to predict optimal conditions for the aquaponics system, such as

the appropriate type and quantity of fish feed, as well as ideal temperature, humidity, and other environmental parameters.

Additionally, data quality validation techniques will be applied to ensure the consistency and integrity of the data at each stage of the process, from collection to visualization. The effectiveness of the architecture will be evaluated based on criteria such as processing efficiency, system scalability, and ease of data access and analysis.

In summary, the Lakehouse architecture was chosen for its ability to deliver a unified, scalable, and cost-effective data platform that meets the analytical and operational requirements of this project.

5

The Aquaponics System Design

5.1 Data Source Profiling and Business Rules

5.1.1 Location, Identification, and Analysis of Data Sources Supporting the Project

This section is dedicated to the location, identification, and analysis of data sources that support the project. The process involves the identification of relevant data sources, the assessment of their reliability, and the determination of their contribution to the system's monitoring and management.

Location of source tables

The data from the LSMI aquaponics system at IPLeiria was initially stored on IPLeiria's OneDrive. As stated in **Section 2.4** of this work, a series of structural inconsistencies were identified during the review of the Excel Workbooks and CSV files. These inconsistencies complicated their integration into a BI environment. File names lacked any standard conventions, merged cells impeded reliable detection of headers and values, and tables were placed in varying locations across different files. In addition, the order and number of columns often differed even when they represented the same data fields, resulting in further discrepancies that would have undermined automated extraction and consolidated analysis.

To overcome these obstacles, we first consolidated and cleaned every workbook before relocating them to a dedicated repository designed for data integration. At the same time, we introduced rigorous guidelines to ensure consistency going forward: all files now follow a uniform naming scheme, merged cells have been eliminated to simplify header recognition, and column order is standardized across every dataset. We also developed reusable templates for data entry, guaranteeing that future records adhere to the same structure and quality standards.

These measures laid the groundwork for a smooth and dependable integration process. By unifying the data's format and location, we were able to load the cleansed datasets into the BI system without further manual intervention.

A Teams channel was created in the BI tenant to store all the files related to the project, and access was granted to all project participants. The name of the channel is **Tese LSMI**.

In the channel, under the *Files* tab, there are several folders containing the files, organized by data collection themes, as we can see in the image Figure 5.1:

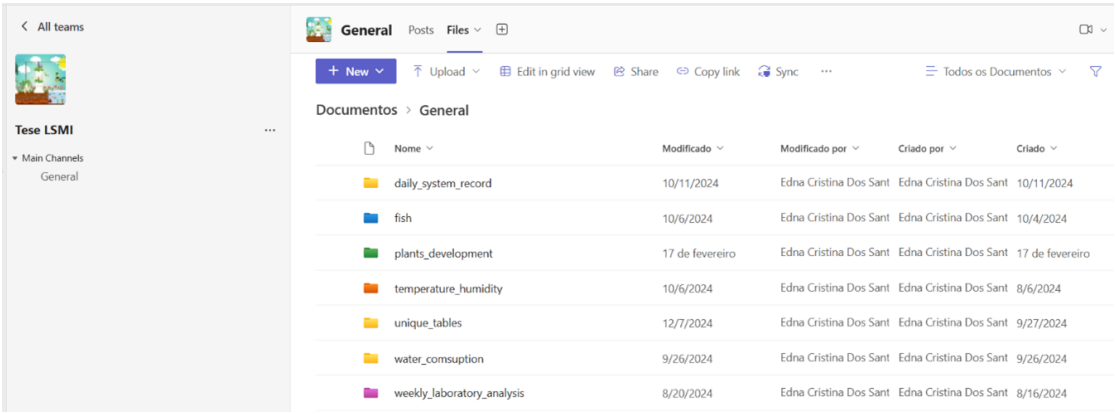


Figure 5.1: *Folders in the Teams channel*

5.1.2 Data Collection and Recording

The **Section 5.1.1** shows that all aquaponics data are stored as Excel Workbooks and CSV files organized into distinct folders based on the type of information and its recording frequency. The following provides a brief overview of each folder's contents:

- **Daily System Records**: Multiple Excel files, each corresponding to the daily records for a specific month.
- **Temperature and Humidity**: Multiple CSV files, where each document may cover variable time periods (ranging from one to five months). The recorded months are not always complete.
- **Water Consumption**: Multiple Excel files containing data related to water consumption.
- **Weekly Laboratory Analysis**: Multiple Excel files, each containing data for one week of analyses, some files report nutrient parameters (e.g., NH_3, PO_4, NO_2 B, NO_3), while others provide weekly metal concentration measurements (e.g., Mg, K, Fe).
- **Fish**: Multiple Excel files containing information about the number and types of fish in each tank, along with descriptions of the fish.
- **Unique Tables**: Multiple Excel files containing prospective dimension tables (Measures, Lines, FishTanks, Projects) and fact tables (Papaya Measurements and Pa-

paya Radiation).

- **Plant Development**: Excel files with measurements related to arugula, lamb's lettuce, mealworms, and papaya tree projects.

Following this overview, we introduce and illustrate the templates employed for structuring and storing the information.

Daily System Record

The names of these Excel Workbook follow a standardized format: they all start with *daily_system_record*, followed by the year, the month (represented by three letters in Portuguese), and end with *_v4*, resulting in the format *daily_system_record_yyyy _mmmpt_v4*, as shown in Figure 5.2. The first columns serve as the key, with the first indicating the location, followed by the year, month, day, and time. Subsequently, various measured parameters are listed, each explained in **Section 2.1**. The file concludes with the *Observation* column, where researchers record relevant notes.

Line	FishTank_DWC	Year	Month	Day	Time	DO mg/L	Temp. °C	pH	σ µS/cm	TDS mg/L	ORP	Fish Feed g	K_2CO_3 g	KOH	$CaOH_2$	$CaCO_3$	Larvae	Observation
sistema_canto	FishTank T3	2024	8	1	9:20 AM	7.83	23.7	5.41	332	166.2							10	T3
sistema_canto	FishTank T4	2024	8	1	9:20 AM	7.25	23.9	5.45	334	166.8			50				20	T4/ Adição de 50g K2CO3
sistema_canto	FishTank T3	2024	8	2	9:50 AM	8.23	22.8	7.00	368	184.6							10	T3
sistema_canto	FishTank T4	2024	8	2	9:50 AM	7.51	22.9	6.89	372	185.8							20	T4
sistema_canto	FishTank T3	2024	8	3	10:00 AM												10	T3 +T4
sistema_canto	FishTank T3	2024	8	3	6:00 PM												10	T3 +T4
sistema_canto	FishTank T3	2024	8	4	9:45 AM												10	T3 +T4
sistema_canto	FishTank T3	2024	8	4	7:10 PM												10	T3 +T4
sistema_canto	FishTank T3	2024	8	5	10:21 AM	7.74	24.6	5.86	356	178							10	T3
sistema_canto	FishTank T4	2024	8	5	10:23 AM	7.74	24.8	5.56	358	180			50				20	T4/ Adição de 50g K2CO3
sistema_canto	FishTank T3	2024	8	6	9:52 AM	7.85	24.4	7.17	385	192							10	T3
sistema_canto	FishTank T4	2024	8	6	9:53 AM	7.33	24.5	7.07	387	194							20	T4
sistema_canto	FishTank T3	2024	8	7	9:45 AM	7.69	24.3	6.78	382	191							10	T3
sistema_canto	FishTank T4	2024	8	7	9:47 AM	5.28	24.4	6.54	384	194							20	T4
sistema_canto	FishTank T3	2024	8	8	9:47 AM	7.53	24.1	6.86	400	200							10	T3
sistema_canto	FishTank T4	2024	8	8	9:49 AM	7.29	24.3	6.76	407	202							20	T4
sistema_canto	FishTank T3	2024	8	9	9:45 AM	7.24	24.6	5.85	412	205							10	T3
sistema_canto	FishTank T4	2024	8	9	9:47 AM	6.24	24.7	5.65	416	208							20	T4

Figure 5.2: *Example of a daily records file*

Temperature Humidity

The most recent files containing air temperature and humidity measurements recorded at 30-minute intervals, with each row listing a sequential reading number (Order), a timestamp in *DD/MM/YYYY HH:MM* format (Time), ambient temperature in °C (Celsius), relative humidity in %RH (Humidity), dew point in °C (Dew Point), and the sensor's serial number when available, thereby documenting the system's thermal and humidity profile of systems on Lines 1, 2, and 3 for environmental variation analysis, Figure 5.3.

Since these files are generated automatically, older exports retained their original structure. Standardisation was applied during ingestion into *Dataflow Gen2* to simplify processing of these large CSV datasets.

Water Comsuption

Water consumption data are stored in Excel Workbooks named according to the pattern *water_consumption_yyyymm_yyyymm*, where the first *yyyymm* denotes the year

Order;Time;Celsius(∞C);Humidity(%rh);Dew Point(∞C);Serial Number
1;18/11/2024 18:16;17.0;84.5;14.4;67013654
2;18/11/2024 18:46;16.0;84.5;13.4;
3;18/11/2024 19:16;15.5;86.5;13.3;
4;18/11/2024 19:46;15.0;86.5;12.8;
5;18/11/2024 20:16;15.0;88.0;13.0;
6;18/11/2024 20:46;15.0;89.0;13.2;
7;18/11/2024 21:16;15.5;89.0;13.7;
8;18/11/2024 21:46;16.0;89.0;14.2;
9;18/11/2024 22:16;16.0;89.5;14.3;
10;18/11/2024 22:46;15.5;90.5;14.0;

Figure 5.3: *Example of a temperature and humidity records file*

and month of the file's earliest record and the second *yyyymm* the year and month of its latest record. Each workbook comprises three sheets, one for each sensor line (*Lines 1, 2* and *3*). The column headers for each table are shown in the accompanying Figure 5.4.

Linha	Ano	Mês	Dia	Dados Tanque Altura(cm) hi	Dados Tanque Altura(cm) hf	Dados Sedim. Volume(L)	Dados Outros Volume(L)	Dados Repos. Volume(L)	Cálculos Tanque	E cont Leitura do Contador (L)	S dia Gasto diário (L)	E dia Consumo diário (L)	Bdia Balanço diário (L)	B acum Balanço diário acumulado (L)
Line3	2021	4	29	6.4	23				521.5	74308	521.5	497.0	-24.5	6674.8
Line3	2021	4	30	8	34				816.8	75293	816.8	985.0	168.2	6843.0
Line3	2021	5	3	7	17.5				329.9	75293	329.9	0.0	-329.9	6513.1
Line3	2021	5	4			5.0	1.5		0.0	75293	6.5	0.0	-6.5	6506.6
Line3	2021	5	5				2.0		0.0	75293	2.0	0.0	-2.0	6504.6
Line3	2021	5	6	7	13				188.5	75797	188.5	504.0	315.5	6820.1
Line3	2021	5	7						0.0	75797	0.0	0.0	0.0	6820.1
Line3	2021	5	10				65.0		0.0	75847	65.0	50.0	-15.0	6805.1
Line3	2021	5	11						0.0	75847	0.0	0.0	0.0	6805.1
Line3	2021	5	12			5.0	3.5		0.0	75847	8.5	0.0	-8.5	6796.6

Figure 5.4: *Example of water consumption records file*

Weekly Laboratory Analysis

In the *Weekly Laboratory Analysis* folder, annual subfolders (2019 to 2025) contain two file series:

- ***weekly_lab_analysis_YYYYMMDD***: daily reports
- ***monthly_lab_metals_analysis_YYYYMM***: monthly metal profiles

Next, a detailed overview of each file type and its contents follows.

Laboratory analysis results are archived in Excel Workbooks named using the pattern *weekly_lab_analysis_YYYYMMDD* (for example, *weekly_lab_analysis_20190723*). Each workbook contains four sheets (NH_3, PO_4, NO_2 B, NO_3, or others parameters) all of which follow an identical four-column structure. The Measure column specifies the analyte, Line identifies the line, FishTank_DWC denotes the sampling location (e.g.

FishTank L3, *DWC L3* or *Sump L3*) and *Value* records the measured concentration, Figure 5.5. The date is derived from the file name.

Measure	Line	FishTank_DWC	Value
PO4	Line3	FishTank L3	13.552
PO4	Line3	DWC L3	11.987
PO4	Line3	Sump L3	14.238

Figure 5.5: *Example of weekly analysis records file*

The file names of the metals-analysis workbook follow the pattern *monthly_lab_metals_analysis_YYYYMM*, where *YYYYMM* indicates the year and month of sampling. Each metals-analysis workbook contains a separate sheet for every analyte (e.g., Na590, K767, Ca423), and sheets prefixed with an underscore (e.g., _Na590, _K767) represent the original raw data files which, as previously described, were restructured and standardized to enable seamless data integration, Figure 5.6.

Line	FishTank_DWC	Date	Abs	Abs-Br	C(mgNa/L)	Va(mL)	Dil(%)	Value	C(mgNa/L)3	Measure
Line1	FishTank L1	20240101	0.2456	0.2317	2.8136	0.5	5	56	59.6	C(mgNa/L)
Line1	FishTank L1	20240101	0.2722	0.2583	3.1438	0.5	5	63		C(mgNa/L)
Line1	FishTank L1	20240201	0.2850	0.2711	3.3027	0.5	5	66	61.8	C(mgNa/L)
Line1	FishTank L1	20240201	0.2511	0.2372	2.8819	0.5	5	58		C(mgNa/L)
Line1	FishTank L1	20240301	0.2793	0.2654	3.2320	0.25	2.5	129	124.2	C(mgNa/L)
Line1	FishTank L1	20240301	0.2589	0.2450	2.9787	0.25	2.5	119		C(mgNa/L)
Line1	FishTank L1	20240401	0.3010	0.2871	3.5014	0.25	2.5	140	134.9	C(mgNa/L)

Na590 K767 Ca423 Mg285 _K767 _Na590 _Mg285 Lists +

Figure 5.6: *Example of a monthly metals analysis file*

All sheets adhere to a standardized column structure: *Line* (system identifier), *FishTank_DWC* (sampling location), *YYYYMMDD*, raw readings (*Abs* and *Abs-Br*), calculated concentrations (e.g., $C(mg/L)$), analyzed volume (*Va*), dilution factor (*Dil*), and final Value, Figure 5.6.

Fish

The Fish folder contains two files: *dim_fish*, which functions as a dimension table, Figure 5.7, holding all relevant attributes for each fish, and *fish_data*, which records the details of every fish's entry into and exit from each tank, Figure 5.8.

fish_type	species	scientific_name	pH_range	optimal_temperature	average_weight_kg	average_length_cm
Carpa Comum	Carpa Comum	Cyprinus carpio	6.5 - 9.0	15 - 20 °C	2 a 5	30 a 60
Barbos	Barbo Comum	Barbus barbus	6.0 - 8.0	10 - 24°C	1 a 3	30 a 50
Clarias Gariepinus	Peixe Gato Africano	Clarias gariepinus	6.5 - 8.0	25 - 30 °C	2 a 10	40 a 100
Carpa Koi	Carpa Koi	Cyprinus rubrofuscus	6.8 - 8.2	15 - 25 °C	1 a 5	25 a 70

Figure 5.7: *Example of fish data characteristics records file*

File	DateRegister	DateEvent	FishType	FishTank	Doubt	NumberOfFish	Observation
Sistema Porta (Barbos)	20240910	20190921	Barbos	FishTank L1		9	Colocação de 9 barbos provenientes do rio Lis no tanque
Sistema Porta (Barbos)	20240910	20190922	Barbos	FishTank L1		23	Colocação de mais 23 barbos provenientes do rio Lis no tanque
Sistema Porta (Barbos)	20240910	20191202	Barbos	FishTank L1		-2	Morte de 2 peixes
Sistema Porta (Barbos)	20240910	20191203	Barbos	FishTank L1		-2	Morte de 2 peixes
Sistema Porta (Barbos)	20240910	20191204	Barbos	FishTank L1		-1	Morte de 1 peixes
Sistema Porta (Barbos)	20240910	20191205	Barbos	FishTank L1		-3	Morte de 3 peixes
Sistema Porta (Barbos)	20240910	20191206	Barbos	FishTank L1		-3	Morte de 3 peixes
Sistema Porta (Barbos)	20240910	20191207	Barbos	FishTank L1		-5	Morte de 5 peixes
Sistema Porta (Barbos)	20240910	20191208	Barbos	FishTank L1		-2	Morte de 2 peixes
Sistema Porta (Barbos)	20240910	20191210	Barbos	FishTank L1		-2	Morte de 2 peixes

Figure 5.8: *Example of fish data records file*

Unique Tables

The Unique Tables folder contains standalone Excel Workbooks. Unlike the daily records or water analysis files, there are no multiple versions, each workbook exists as a single, unique file, the files are:

- ***dim_lines_and_fishtank***: Information on fish tanks (Figure 5.9) and lines (Figure 5.10) within the LSMI, including unique identifiers, tank capacities, fish species, and physical locations to support spatial and operational mapping.
- ***dim_measures***: Definitions and metadata for all measured parameters used across tables and projects (Figure 5.11), specifying measurement units, analytical methods, and acceptable value ranges to ensure data quality.
- ***papaya_fruits_measures***: Papaya fruit development measurements from Project 3 (Figure 5.12), capturing attributes such as fruit weight, diameter, seed color, fruit skin roughness and if is tasty or not.
- ***project***: Project-level metadata, including project identifiers, project names, start and end dates, and the lines involved (Figure 5.13).
- ***radiation_papaya***: Radiation readings linked to papaya cultivation (Figure 5.14), by line and date.

FishTank_DWC	Line	Localization	Observation	area	water_volume
Control	Control	IPLeiria			
Control20	Control	IPLeiria			
FishTank T3	CornerSystem	IPLeiria			
FishTank T4	CornerSystem	IPLeiria			
FishTank T3 + T4	CornerSystem	IPLeiria			
FishTank T1	DoorSystem	IPLeiria			
FishTank T2	DoorSystem	IPLeiria			
FishTank T1 + T2	DoorSystem	IPLeiria			

Figure 5.9: *Fish Tank Example File*

Plants Development

In the LSMI , plant performance is also tracked through periodic measurements. Since 2019, three projects have been carried out:

- Project 1 - Arugula & Lamb's Lettuce - Figure 5.15
- Project 2 - Mealworms - Figure 5.16
- Project 3 - Papaya Trees - Figure 5.17

line	localization	Latitude	Longitude	observation
Control	IPLeiria	39°4403700 N	8°4802500 W	
CornerSystem	IPLeiria	39°4403700 N	8°4802500 W	
DoorSystem	IPLeiria	39°4403700 N	8°4802500 W	
Line1	IPLeiria	39°4403700 N	8°4802500 W	
Line2	IPLeiria	39°4403700 N	8°4802500 W	
Line3	IPLeiria	39°4403700 N	8°4802500 W	
NoLine	IPLeiria	39°4403700 N	8°4802500 W	só usado nas tabelas relacionadas com os peixes
OldSystem	IPLeiria	39°4403700 N	8°4802500 W	só usado nas tabelas relacionadas com os peixes

Figure 5.10: *Line Example File*

measure	measure_group	units	measure_type	fact_table	min_value	max_value	description
Ca423	Ca	mg/L	metals	fact_measurements_nutrients_metals			Calcium 423
CaCO3_acd	CaCO3	mg CaCO3/L	metals	fact_measurements_nutrients_metals			Calcium Carbonate in an Acidic Medium
CaCO3_alc	CaCO3	mg CaCO3/L	metals	fact_measurements_nutrients_metals			Calcium Carbonate in an Alkaline Medium
DO	DO	mg/L		fact_daily_system	4		Dissolved Oxygen
Fe	Fe	mg/L	metals	fact_measurements_nutrients_metals			Iron
Fe_fenantrolina	Fe	mg/L	metals	fact_measurements_nutrients_metals			Iron with Phenanthroline
Fe248	Fe	mg/L	metals	fact_measurements_nutrients_metals			Iron 248
K	K	mg/L	metals	fact_measurements_nutrients_metals			Potassium

Figure 5.11: *Measures Example File*

N.º da papaia	Data	N.º da Linha ou Sistema	N.º da papaieira	Diâmetro da papaia (cm)	Comprimento da papaia (cm)	Peso da papaia (g)	Pele (Lisa /Rugosa)	Sementes (Sim/Não)	Cor das Sementes	Saborosa (Sim/Não)	N.º Foto fruto inteiro	N.º Foto fruto partido em dois
1	31-Mar	Linha 3	25	58	27.5	3488.02	Rugosa	Sim	Brancas; Pretas; Castanho	Sim	L30001	L30002
2	31-Mar	Linha 3	21	53.5	17.5	1684.56	Rugosa	Não	-	Sim	L30003	L30004
3	31-Mar	Linha 3	20	49.5	16	1133.12	Rugosa	Não	-	Sim	L30005	L30006
4	31-Mar	Linha 3	20	33	8.5	345.31	Rugosa	Não	-	Sim	L30007	-
5	31-Mar	Sist. Canto	2	46	18	1213.44	Lisa	Não	-	Sim	SC0001	SC0002
6	31-Mar	Sist. Canto	2	50	19	1642.08	Lisa	Não	-	Sim	SC0003	SC0004
7	31-Mar	Sist. Canto	2	46	17	1198.03	Lisa	Não	-	Sim	SC0005	SC0006

Original | Linha 2 - Filtro & Média | Linha 3 - Filtro & Média | Linha2-Estatisticas | Linha3-Estatisticas

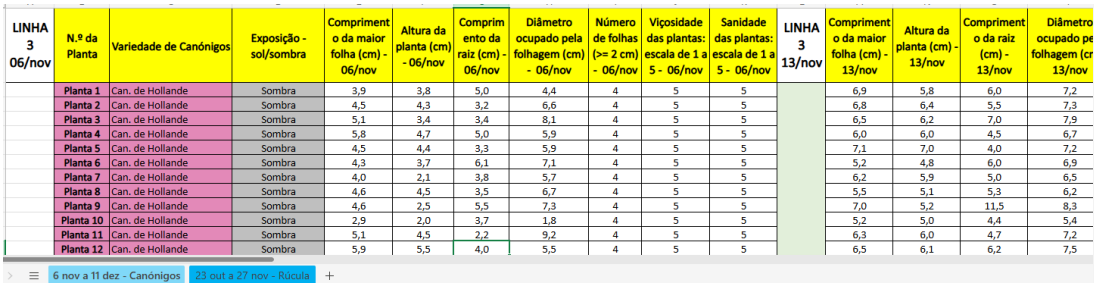
Figure 5.12: *Papaya Fruit Measures Example File*

project_name	project_name_eng	project_number	start_date	end_date	duration	table	line
Rúcula & Canónigos	Arugula & Lamb's Lettuce	Project1	20191023	20191211	7 weeks	all	Line3
Tenébrios	Mealworms	Project2	20210427	20210608	6 weeks	all	Line1 + Line2 + Line3
Papaieiras	Papaya Trees	Project3	20220502	20230501	53 weeks	all	Line2 + Line3

Figure 5.13: *Project Example File*

Laboratório de Sistemas Multitróficos Integrados: Medição da radiação com sonda específica LI-250 (µmol)				
Data	Hora	Linha 1	Linha 2	Linha 3
	10:48		231.6	614.8
	14:31	252.4	197.54	236.9
5/11/2022	18:12	135.36	54.25	67.81
	10:30	196.51	184.99	559.8
	14:35	235.6	228	236.2
5/13/2022	17:10	246.4	90.06	124.06
	10:20	122.27	125.81	202.4
	14:00	114.15	96.33	106.34
5/16/2022	17:53	349.8	205	142.32

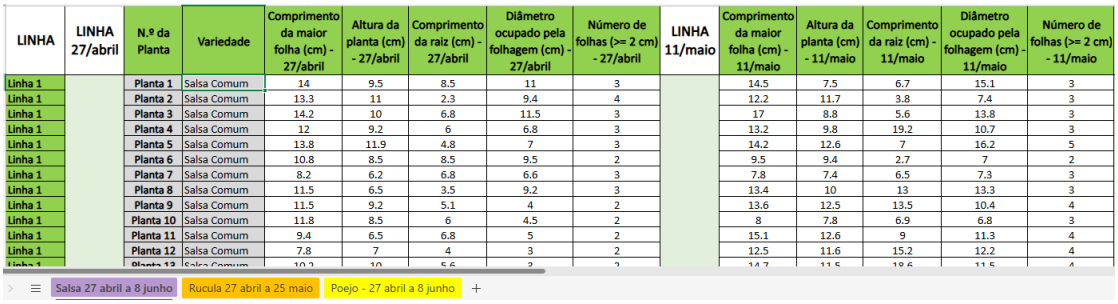
Figure 5.14: *Radiation Papaya Example File*



LINHA 3 06/nov	N.º da Planta	Variedade de Canónigos	Exposição - sol/sombra	Comprimento da maior folha (cm) - 06/nov	Altura da planta (cm) - 06/nov	Comprimento da raiz (cm) - 06/nov	Diâmetro ocupado pela folhagem (cm) - 06/nov	Número de folhas (>= 2 cm) - 06/nov	Viçosidade das plantas: escala de 1 a 5 - 06/nov	Sanidade das plantas: escala de 1 a 5 - 06/nov	LINHA 3 13/nov	Comprimento da maior folha (cm) - 13/nov	Altura da planta (cm) - 13/nov	Comprimento da raiz (cm) - 13/nov	Diâmetro ocupado pe folhagem (cr 13/nov
	Planta 1	Can. de Hollande	Sombra	3,9	3,8	5,0	4,4	4	5	5		6,9	5,8	6,0	7,2
	Planta 2	Can. de Hollande	Sombra	4,5	4,3	3,2	6,6	4	5	5		6,8	6,4	5,5	7,3
	Planta 3	Can. de Hollande	Sombra	5,1	3,4	3,4	8,1	4	5	5		6,5	6,2	7,0	7,9
	Planta 4	Can. de Hollande	Sombra	5,8	4,7	5,0	5,9	4	5	5		6,0	6,0	4,5	6,7
	Planta 5	Can. de Hollande	Sombra	4,5	4,4	3,3	5,9	4	5	5		7,1	7,0	4,0	7,2
	Planta 6	Can. de Hollande	Sombra	4,3	3,7	6,1	7,1	4	5	5		5,2	4,8	6,0	6,9
	Planta 7	Can. de Hollande	Sombra	4,0	2,1	3,8	5,7	4	5	5		6,2	5,9	5,0	6,5
	Planta 8	Can. de Hollande	Sombra	4,6	4,5	3,5	6,7	4	5	5		5,5	5,1	5,3	6,2
	Planta 9	Can. de Hollande	Sombra	4,6	2,5	5,5	7,3	4	5	5		7,0	5,2	11,5	8,3
	Planta 10	Can. de Hollande	Sombra	2,9	2,0	3,7	1,8	4	5	5		5,2	5,0	4,4	5,4
	Planta 11	Can. de Hollande	Sombra	5,1	4,5	2,2	9,2	4	5	5		6,3	6,0	4,7	7,2
	Planta 12	Can. de Hollande	Sombra	5,9	5,5	4,0	5,5	4	5	5		6,5	6,1	6,2	7,5

6 nov a 11 dez - Canónigos | 23 out a 27 nov - Rúcula

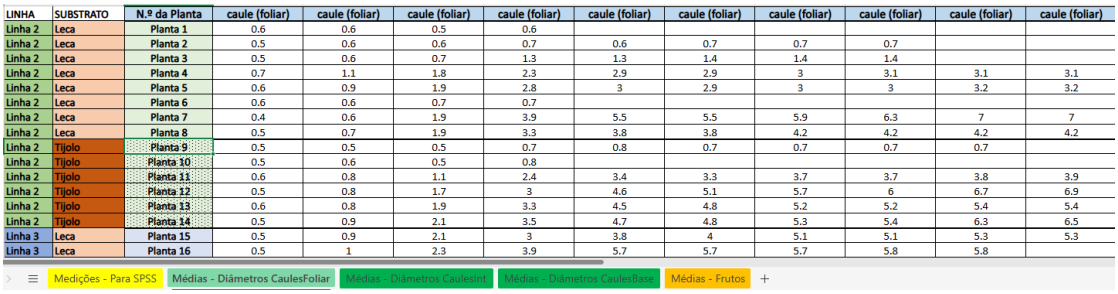
Figure 5.15: *Plant Development Example (Project 1 – Arugula & Lamb's Lettuce)*



LINHA	LINHA 27/abril	N.º da Planta	Variedade	Comprimento da maior folha (cm) - 27/abril	Altura da planta (cm) - 27/abril	Comprimento da raiz (cm) - 27/abril	Diâmetro ocupado pela folhagem (cm) - 27/abril	Número de folhas (>= 2 cm) - 27/abril	LINHA 11/maio	Comprimento da maior folha (cm) - 11/maio	Altura da planta (cm) - 11/maio	Comprimento da raiz (cm) - 11/maio	Diâmetro ocupado pela folhagem (cm) - 11/maio	Número de folhas (>= 2 cm) - 11/maio
Linha 1		Planta 1	Salsa Comum	14	9.5	8.5	11	3		14.5	7.5	6.7	15.1	3
Linha 1		Planta 2	Salsa Comum	13.3	11	2.3	9.4	4		12.2	11.7	3.8	7.4	3
Linha 1		Planta 3	Salsa Comum	14.2	10	6.8	11.5	3		17	8.8	5.6	13.8	3
Linha 1		Planta 4	Salsa Comum	12	9.2	6	6.8	3		13.2	9.8	19.2	10.7	3
Linha 1		Planta 5	Salsa Comum	13.8	11.9	4.8	7	3		14.2	12.6	7	16.2	5
Linha 1		Planta 6	Salsa Comum	10.8	8.5	8.5	9.5	2		9.5	9.4	2.7	7	2
Linha 1		Planta 7	Salsa Comum	8.2	6.2	6.8	6.6	3		7.8	7.4	6.5	7.3	3
Linha 1		Planta 8	Salsa Comum	11.5	6.5	3.5	9.2	3		13.4	10	13	13.3	3
Linha 1		Planta 9	Salsa Comum	11.5	9.2	5.1	4	2		13.6	12.5	13.5	10.4	4
Linha 1		Planta 10	Salsa Comum	11.8	8.5	6	4.5	2		8	7.8	6.9	6.8	3
Linha 1		Planta 11	Salsa Comum	9.4	6.5	6.8	5	2		15.1	12.6	9	11.3	4
Linha 1		Planta 12	Salsa Comum	7.8	7	4	3	2		12.5	11.6	15.2	12.2	4

Salsa 27 abril a 8 junho | Rucula 27 abril a 25 maio | Poejo - 27 abril a 8 junho

Figure 5.16: *Plant Development Example (Project 2 – Mealworms)*



LINHA	SUBSTRATO	N.º da Planta	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)	caule (foliar)
Linha 2	Leca	Planta 1	0.6	0.6	0.5	0.6						
Linha 2	Leca	Planta 2	0.5	0.6	0.6	0.7	0.6	0.7	0.7	0.7		
Linha 2	Leca	Planta 3	0.5	0.6	0.7	1.3	1.3	1.4	1.4	1.4		
Linha 2	Leca	Planta 4	0.7	1.1	1.8	2.3	2.9	2.9	3	3.1	3.1	3.1
Linha 2	Leca	Planta 5	0.6	0.9	1.9	2.8	3	2.9	3	3	3.2	3.2
Linha 2	Leca	Planta 6	0.6	0.6	0.7	0.7						
Linha 2	Leca	Planta 7	0.4	0.6	1.9	3.9	5.5	5.5	5.9	6.3	7	7
Linha 2	Leca	Planta 8	0.5	0.7	1.9	3.3	3.8	3.8	4.2	4.2	4.2	4.2
Linha 2	Tijolo	Planta 9	0.5	0.5	0.5	0.7	0.8	0.7	0.7	0.7	0.7	
Linha 2	Tijolo	Planta 10	0.5	0.6	0.5	0.8						
Linha 2	Tijolo	Planta 11	0.6	0.8	1.1	2.4	3.4	3.3	3.7	3.7	3.8	3.9
Linha 2	Tijolo	Planta 12	0.5	0.8	1.7	3	4.6	5.1	5.7	6	6.7	6.9
Linha 2	Tijolo	Planta 13	0.6	0.8	1.9	3.3	4.5	4.8	5.2	5.2	5.4	5.4
Linha 2	Tijolo	Planta 14	0.5	0.9	2.1	3.5	4.7	4.8	5.3	5.4	6.3	6.5
Linha 3	Leca	Planta 15	0.5	0.9	2.1	3	3.8	4	5.1	5.1	5.3	5.3
Linha 3	Leca	Planta 16	0.5	1	2.3	3.9	5.7	5.7	5.7	5.8	5.8	

Medições - Para SPSS | Médias - Diâmetros CaulesFoliar | Médias - Diâmetros CaulesInt | Médias - Diâmetros CaulesBase | Médias - Frutos

Figure 5.17: *Plant Development Example (Project 3 – Papaya Trees)*

The Plants folder contains three Workbooks named according to the convention *plants_projectX_project_name* (where X is the project number). Each project targets a different plant species and records distinct traits, so the internal layout of each workbook varies. Since these projects are now closed and no further data will be added, schema harmonization was performed during the data ingestion phase, and the original files have been preserved unchanged.

It is worth noting that, wherever applicable, data entry fields in these files are constrained to the appropriate data types (string, integer, or date) with defined minimum and maximum ranges for dates and numeric values. Furthermore, columns such as *Line* and *FishTank* are restricted to predefined lists of valid entries to minimize data entry errors.

5.1.3 Business Rules

A business rule is a concise statement that defines, constrains, or controls an aspect of a business, ensuring consistency and logical decision-making. It is expressed in clear, unambiguous terms that are easily understood by stakeholders such as business owners, analysts, and architects. Business rules serve as constraints, establishing what must or must not happen, and are fundamental to defining business logic. They help structure and influence business behavior. While a complete business system may involve thousands of rules, individual rules are usually straightforward and facilitate processes such as discovery, definition, and maintenance [74].

According to Ronak Ravjibhai Pansara, the relationship between master data quality and business rules plays a crucial role in organizational performance. This multifaceted strategy not only introduces new perspectives but also reinforces previous findings on the subject [75].

Therefore, the following list of rules has been compiled to enhance the understanding of data within the LSMI :

- A row in the LSMI at IPLeiria can contain for example fish tank or a DWC.
- Each plant is linked to a specific row, a project, a unique number, and a name, as well as whether it is exposed to sunlight or shade. Additionally, a plant may bear multiple fruits or none.
- On any given date, a fish belongs to a specific fish tank and is characterized by various attributes, including optimal pH and temperature, geographical distribution, average weight, and length.
- A measurement is associated with a specific type, its maximum and minimum values, and the fact table where it is recorded.
- Projects are linked to fact tables, as well as the project's start and end dates.
- Atmospheric conditions, such as temperature, humidity, and radiation, are recorded for a specific row, measurement type, day, hour, and minute.

- The daily system is related to either a tank or a grow bed, recorded in the morning or afternoon of a specific day, and linked to a particular measurement.
- Fish events are registered based on the date and the specific fish tank where they occur.
- Nutrient and metal measurements are recorded for a specific day, associated with either a tank or a DWC, and linked to a particular measurement type.
- Developing plants are tracked in relation to a project, a row, a specific plant, and its assigned location.
- Water consumption data is recorded based on the row, a specific day, and the corresponding measurement.

After analyzing the previous files, the dimensional model was designed, as shown in the following chapter.

5.2 Dimensional Data Model

After analyzing the provided Excel files, it was determined that a structured, standardized data model was necessary to support analytical processing. The raw data is dispersed across multiple files with heterogeneous formats and variable temporal coverage, reinforcing the need for a systematic integration approach.

The process revealed the need to formally represent descriptive entities through dimension tables, as well as the need to design fact tables to store quantitative metrics and event-driven data.

This structure enables multidimensional analysis, allowing metrics to be contextualized by various characteristics. The following subsections present the dimension and fact tables resulting from the transformation of the original Excel sources.

5.2.1 Dimensions and their Attributes

In the dimensional model, there are dimension and fact tables. The Table 5.1 lists all dimension tables along with their contextualization. The Table 5.2 displays all columns, their descriptions, data types, and three example values for each column in the dimension table measures.

All other dimension tables can be accessed directly in the SQL Server database, specifically in the *lsmi_gold._metadata_dimension_tables* table, Figure 5.18.

It is important to note that the table mentioned above was created using a script executed within a notebook, utilizing both SQL Spark and PySpark. The corresponding code can be found in the following notebook: *nb_lkh_lsmi_metadata.ipynb*.

Dimensions	Table Name	Contextualization
Plants	dim_plants	The plants dimension includes data on aquaponics plants, such as name, number, and variety.
Fruits	dim_fruit	The fruit dimension stores aquaponics fruit data, including seed presence, diameter, taste, and roughness.
Fishtank DWC	dim_fishtank_dwc	The fishtank DWC dimension stores data related to tanks and DWC systems, including information on location, area, and the corresponding line.
Lines	dim_lines	The lines dimension contains information about the aquaponics system lines and their respective locations.
Fish Attributes	dim_fish_attributes	The fish attributes dimension stores data related to fish, such as pH range, optimal temperature, and other characteristics.
Date	dim_date	The date dimension stores information such as the date, day, month, year, quarter, whether it is a weekend, and other attributes.
Time	dim_time	The time dimension stores information such as the time, hour, minute.
AM PM	dim_am_pm	The AM_PM dimension stores data related to the distinction between morning and afternoon.
Projects	dim_projects	The projects dimension stores information related to projects, including their start and end dates, as well as the associated fact tables.
Measures	dim_measures	The measures dimension stores information related to the studied measures, such as measure type, units, description, and other attributes.

Table 5.1: *Dimension tables and their contextualization*

Table	Column name	Description	Datatype	Top3 examples
dim_measures	measure_sk	Integers generated by SQL Spark	int	31, 53, 34
dim_measures	measure	Quantitative value recorded and analyzed in a fact table	string	K, FishFeed, CaCO3
dim_measures	units	Unit of measurement of the measure	string	scale_1_5, units, g
dim_measures	measure_group	Group to which the measure belongs	string	K, Ca, FishFeed
dim_measures	measure_type	Type of measure, example: parameter, metal, nutrient	string	nutrients, water, metals
dim_measures	fact_table	Fact table to which the measure belongs	string	fact_daily_system, fact_temp_humidity, fact_water_consumption
dim_measures	min_value	Minimum value that the measure can have	decimal(18,2)	0.00, 100.00, 1.26
dim_measures	max_value	Maximum value that the measure can have	decimal(18,2)	100.00, 0.57, 184.41
dim_measures	description	Key description	string	Iron with Phenanthroline, Nitrate ES, Calcium Carbonate
dim_measures	end_date	Date when the record became inactive	timestamp	31/12/9999 00:00
dim_measures	start_date	Date when the record became active	timestamp	01/01/2019 00:00
dim_measures	ingest_date	Date when the data was ingested	timestamp	2025-02-21 21:17:36.997450

Table 5.2: *Metadata of the dimension measures table*

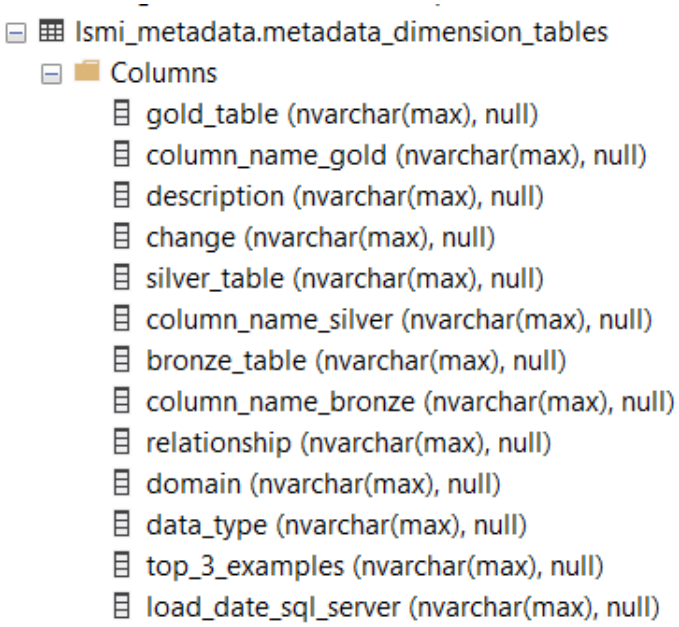


Figure 5.18: *Table lsmi_gold.metadata_dimensions_tables*

5.2.2 Fact Tables and Measures

The Table 5.3 presents the measures from the fact tables, along with their descriptions and data types. It is important to note that these measures maintain the granularity of the table's primary keys. For instance, the value in the *fact_daily_system* table represents a measurement recorded for a specific morning or afternoon in a given tank.

gold_table	column_name_gold	description	data_type
fact_atmospheric_conditions	value	Values of metrics or parameters	Decimal
fact_daily_system	value	Values of metrics or parameters	Decimal
fact_fish_events	fish_out_in	This field indicates whether a fish has entered or exited a tank	String
	number_fish	Number of fish	Int
fact_measurements_nutrients_metals	value	Values of metrics or parameters	Decimal
fact_plants_development	value	Values of metrics or parameters	Decimal
fact_water_consumption	value	Values of metrics or parameters	Decimal

Table 5.3: *Measures from the fact tables with data type and description*

As with the previous table, the ***lsmi_gold.metadata_fact_tables*** table is also accessible in the database hosted on the IPLeiria server, and it was generated using the script provided in the following notebook: *nb_lkh_lsmi_metadata.ipynb*.

5.2.3 Identification of Hierarchies

In dimensional modelling, a hierarchy is a structured arrangement of a dimension's attributes into ordered levels, such as Year → Quarter → Month → Day in a date dimension or Continent → Country → State → City in a Geography dimension, that enables users to navigate data from coarse to fine granularity (drill-down) or aggregate from detailed to summary views (roll-up). By defining these level-based (or, where applicable, parent-child) relationships, hierarchies facilitate intuitive exploration in BI tools, support consistent and efficient aggregation across fact tables, improve query performance through pre-defined paths, and ensure uniform analytics by enforcing the same organizational structure for all related measures [14].

In the context of this work, hierarchies were defined according to analysis requirements:

- **Date Dimension**: Year → Month → Day
- **Time Dimension**: Hour → Minute → Time
- **Fishtank_DWC Dimension**: Line → Fishtank_DWC
- **Plants Dimension**: Plant Name → Variety → Plant Number

5.2.4 The Bus Matrix

Kimball's Bus Matrix is a conceptual framework that organizes fact tables and shared dimensions in a data warehouse. Each row in the Bus Matrix corresponds to a business process (fact), while each column corresponds to a dimension [14].

The business process identifies the operational event or transaction under analysis. The grain defines the level of detail, such as daily transactions or monthly aggregates. Measures are the numeric values recorded in fact tables, capturing quantitative information about the process. Conformed dimensions are dimension tables whose structure, attributes, and meaning remain identical across multiple fact tables; by sharing the same keys and definitions, they enable consistent filtering, grouping, and reporting across the enterprise [14].

The image below, Figure 5.19, represents the Bus Matrix of the LSMI, where business processes, granularity, measures, and their relationships with the dimensions are depicted. It is important to note that there is a Dimension Fruits table linked to the Dimension Plants table, indicating a form of denormalization that occurs in a snowflake model. As the details may not be fully discernible here, the full-size figure is also provided in Appendix B: Bus Matrix for clarity.

5.2.5 Dimensional Data Model

The Dimensional Model is a data modelling approach widely used in DW and BI systems. Popularized by Ralph Kimball, it organizes and represents data to facilitate anal-

Bussiness Process	Grain	Measures	Dimension Date	Dim Time	Dim AM PM	Dim Measures	Dim Fish Attributes	Dim Fishtank DWC	Dim Lines	Dim Plants	Dim Projects	Dimensions Counts
Fact Fish Events	One row for each fish in a tank on a given day that either moved to another tank or died	Number Fish	✓				✓	✓				3
Fact Daily System	One row represents a record of a parameter value at a specific AM or PM in a tank	Value	✓		✓	✓		✓				4
Fact Water Consumption	One row represents a record of a parameter value at a specific day in a line	Value	✓			✓			✓			3
Fact Atmospheric Condition	One row represents a record of a parameter value at a specific time in a line	Value	✓	✓		✓			✓			4
Fact Measurements Nutrients Metals	One row represents a record of a parameter value at a specific day in a tank	Value	✓			✓		✓				3
Fact Plants Development	One row represents a record of a parameter value for a specific day in a line that contains a plant or fruit.	Value	✓			✓			✓	✓ Fruit	✓	5

Figure 5.19: *Bus Matrix*

ysis and reporting. This model is structured around facts and dimensions, providing a multidimensional view of data that is essential for Online Analytical Processing (OLAP) [14].

Kimball proposed the dimensional model as the best way to design databases for decision support since it provides an intuitive and efficient design for analytical queries [14]. The model is based on key fundamental concepts. The fact table contains quantitative metrics and numerical data representing business events. Examples include sales, banking transactions, or customer interactions [42]. Dimension tables store descriptive and categorical data that provide context for facts. Examples of dimensions include time, product, customer, and location [14]. Two common schema designs define the relationship between fact and dimension tables. The star schema is the most common structure in dimensional modelling, where a single fact table is surrounded by directly related dimension tables [42]. A variation of this model is the snowflake schema, in which dimensions are normalized to reduce redundancy [14].

The presentation of the diagram of the dimensional model plays a crucial role in understanding and implementing a DW. The diagram helps visualize the data structure and facilitate interpretation by various stakeholders, such as BI analysts, data scientists, and developers [42]. The benefits of this presentation include a clearer understanding of the relationships between facts and dimensions, optimization of analytical queries and reporting processes [42], and ease of integration and expansion of the model [14]. Kimball's approach emphasizes that a DW should be designed with a focus on business analysis needs, ensuring an accessible, intuitive, and high-performance data model [14].

The Star Schema is the simplest and most popular structure for data warehouses. It is organized around a central fact table that stores quantitative data, such as sales amounts or quantities sold, which are known as measures or metrics. Surrounding the fact table are dimension tables that contain descriptive attributes, providing context to the facts. Each dimension table is linked to the fact table through foreign key relation-

ships. The star schema is favored for its simplicity and efficiency, making it well-suited for business intelligence applications where fast query performance is crucial [42].

In the Snowflake Schema dimension tables are further normalized into multiple related tables, referred to as sub-dimensions. This reduces data redundancy and saves storage space, but it also makes queries more complex due to the need for additional joins between tables. Although it may require more processing time for queries, it can simplify data maintenance and improve data consistency [76].

The Constellation Schema, also known as the Galaxy Schema, extends beyond the star schema by incorporating multiple fact tables that share common dimension tables. This approach is particularly useful in complex business scenarios where different processes or events need to be analyzed collectively. Unlike the star schema, which is centered around a single fact table, the constellation schema allows for a more flexible and interconnected data structure [14].

The most appropriate dimensional model for the case under study is the Constellation Schema, because there are multiple fact tables, as illustrated in Figure 5.20, it can also be found in the appendix. Appendix C, it includes multiple fact tables interconnected by common dimensions. This schema allows for the analysis of different processes, providing a comprehensive and integrated view of the system's operations. The constellation schema is particularly advantageous in scenarios where multiple related events must be analyzed simultaneously, enhancing the system's analytical capabilities.

The schema represents all the fact tables and their connections to the dimensional tables. It is important to note that the *dim_plants* table had to be denormalized, resulting in a partial snowflake model (*dim_plants* and *dim_fruits*).

The LSMI includes the following fact tables, each storing measurable events:

- ***fact_daily_system:*** Stores daily system data related to key operational metrics.
- ***fact_fish_events:*** Captures fish-related events, including movements in and out of the system, as well as the number of fish in specific conditions.
- ***fact_measurements_nutrients_metals:*** Records measurements of various nutrients and metal concentrations in the system over time.
- ***fact_atmospheric_conditions:*** Contains data related to temperature and humidity measurements, crucial for monitoring environmental conditions.
- ***fact_water_consumption:*** Tracks water consumption metrics, essential for assessing resource usage efficiency.
- ***fact_plants_development:*** Stores records of the development of various plants cultivated in the aquaponics system, linking each record to the date, measurements, lines, projects, and plants dimensions.

Each of these fact tables contains foreign keys linking them to common dimension

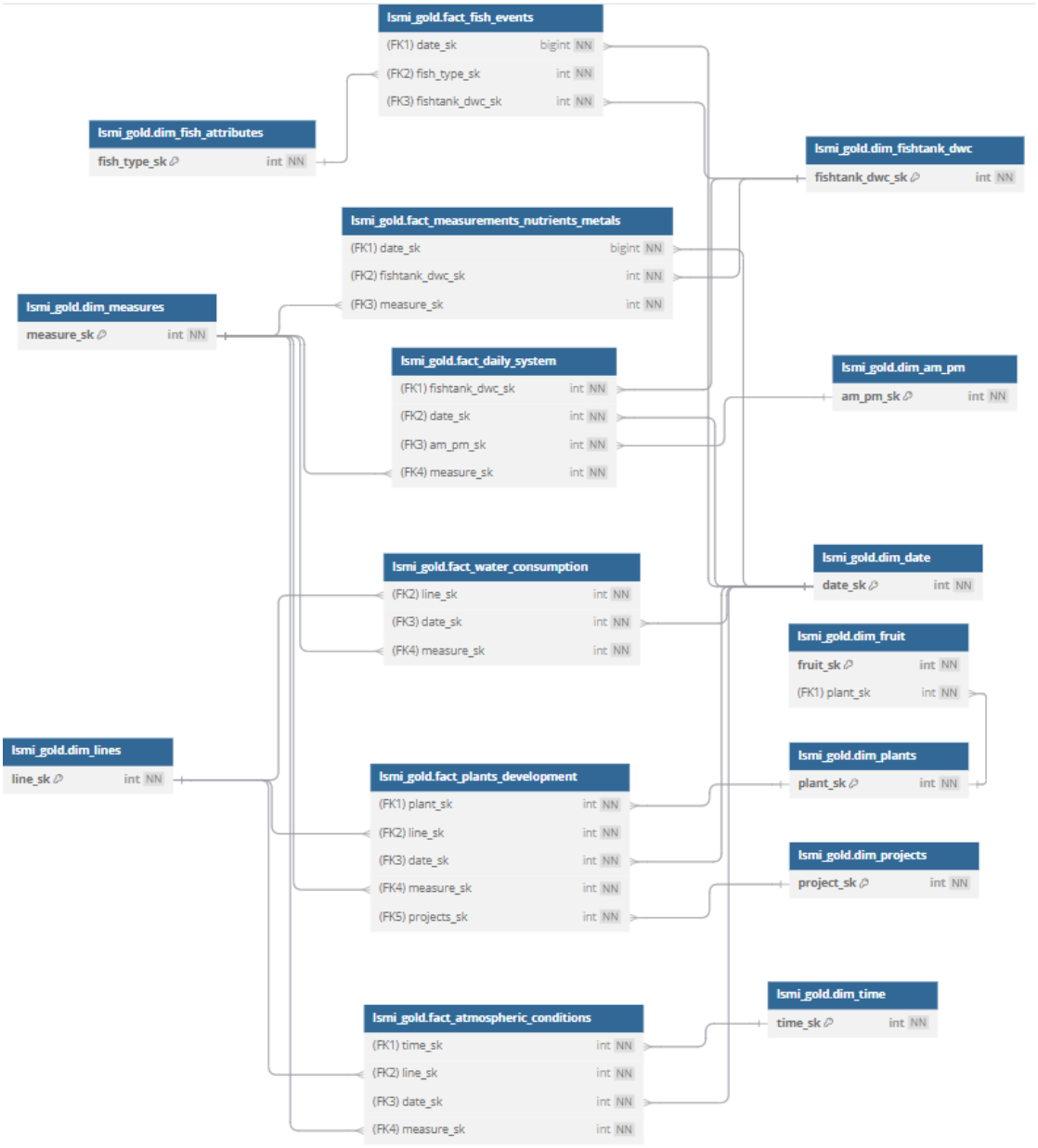


Figure 5.20: *Diagram with the dimensional data model of LSMI*

tables, ensuring data consistency, efficient querying, and analytical flexibility. The dimension tables provide descriptive attributes that give context to the measured facts, facilitating in-depth analysis and reporting. The LSMI 's dimensional model consists of the following dimensions:

- ***dim_date:*** Stores date information, enabling temporal analysis of events.
- ***dim_time:*** Represents specific time information, allowing for precise time-based event analysis.
- ***dim_am_pm:*** Categorizes time-based data into AM and PM periods.
- ***dim_fish_attributes:*** Contains descriptive attributes related to fish types and classifications.
- ***dim_fishtank_dwc:*** Identifies and categorizes different fish tanks used in the system.
- ***dim_lines:*** Defines line-related categorizations within the system.
- ***dim_measures:*** Stores different types of measurements that are tracked across various fact tables.
- ***dim_projects:*** Lists projects that are referenced in plant development records.
- ***dim_plants:*** Contains information about plants.
- ***dim_fruit:*** Contains information about fruits.

Given that the dimensional model consists of six fact tables that share some dimensional tables, the model will be presented by fact table. The dimension tables only display the linking columns (foreign keys in the fact table). This structure is typical in a star schema, where the fact table holds the core business metrics (e.g., value in the fact table), while the dimension tables provide descriptive context for those metrics.

Daily System Record:

The daily records are stored in the table *lsmi_gold.fact_daily_system_record*, Figure 5.21. This diagram represents a star schema in which the central fact table, is connected to four-dimension tables:

- *lsmi_gold.dim_fishtank_dwc*
- *lsmi_gold.dim_date*
- *lsmi_gold.dim_am_pm*
- *lsmi_gold.dim_measures*

Consumption Water:

The diagram, Figure 5.22, represents a star schema in which the central fact table, *lsmi_gold.fact_water_consumption*, is connected to three-dimension tables:

- *lsmi_gold.dim_lines*
- *lsmi_gold.dim_date*

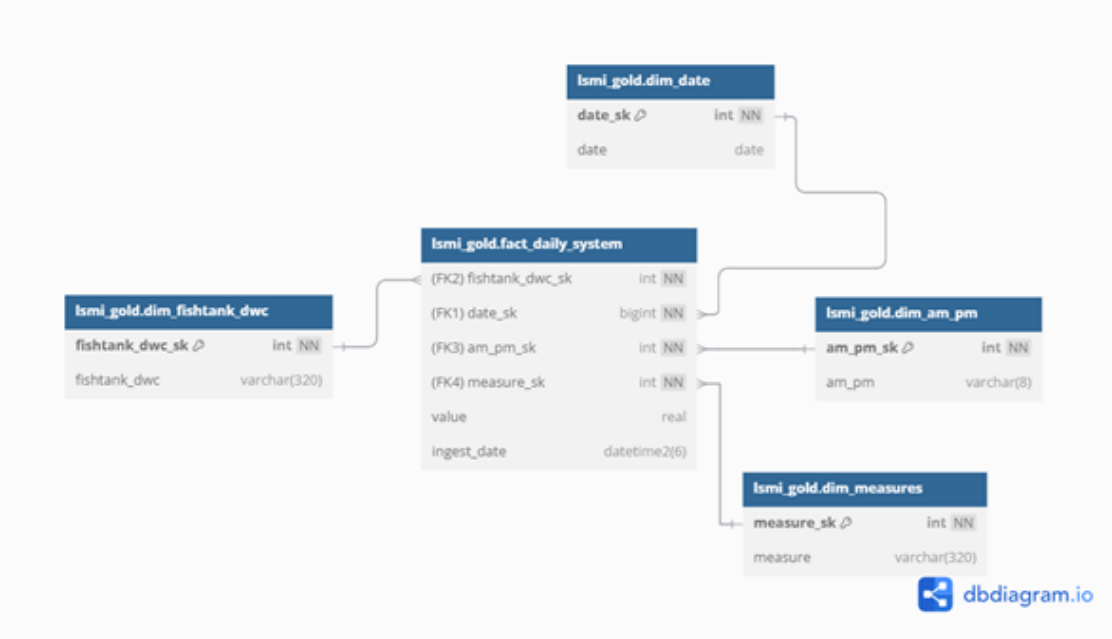


Figure 5.21: *Star schema with fact table, lsmi_gold.fact_daily_system*

- *lsmi_gold.dim_measures*

The fact table contains foreign keys linking it to the dimension tables.

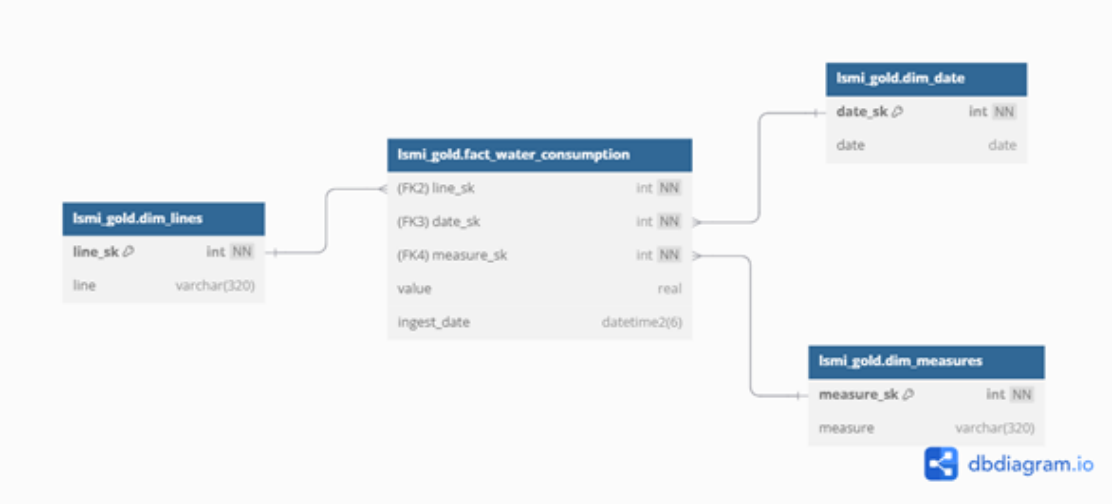


Figure 5.22: *Star schema with fact table, lsmi_gold.fact_water_consumption*

Atmospheric Conditions:

The diagram illustrates a star schema where the central fact table, *lsmi_gold.fact_atmospheric_conditions*, is connected to four-dimension tables:

- *lsmi_gold.dim_lines*
- *lsmi_gold.dim_date*
- *lsmi_gold.dim_time*

- *lsmi_gold.dim_measures*

In this schema, the fact table stores key transactional data, such as the value representing the temperature, humidity and radiation measurements, Figure 5.23.

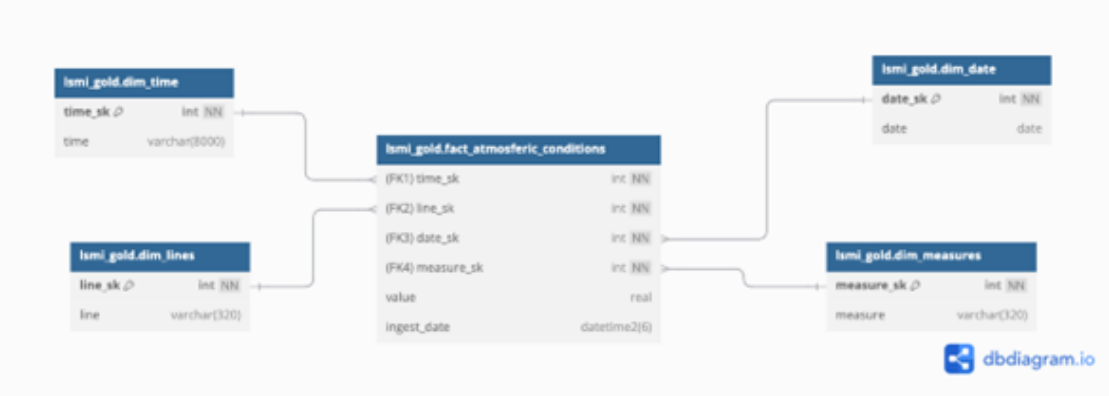


Figure 5.23: *Star schema with fact table, lsmi_gold.fact_atmospheric_conditions*

Measurements of Nutrients and Metals:

The diagram, Figure 5.24, illustrates a star schema where the central fact table, *lsmi_gold.fact_measurements_nutrients_metals*, is connected to three-dimension tables:

- *lsmi_gold.dim_date*
- *lsmi_gold.dim_fishtank_dwc*
- *lsmi_gold.dim_measures*

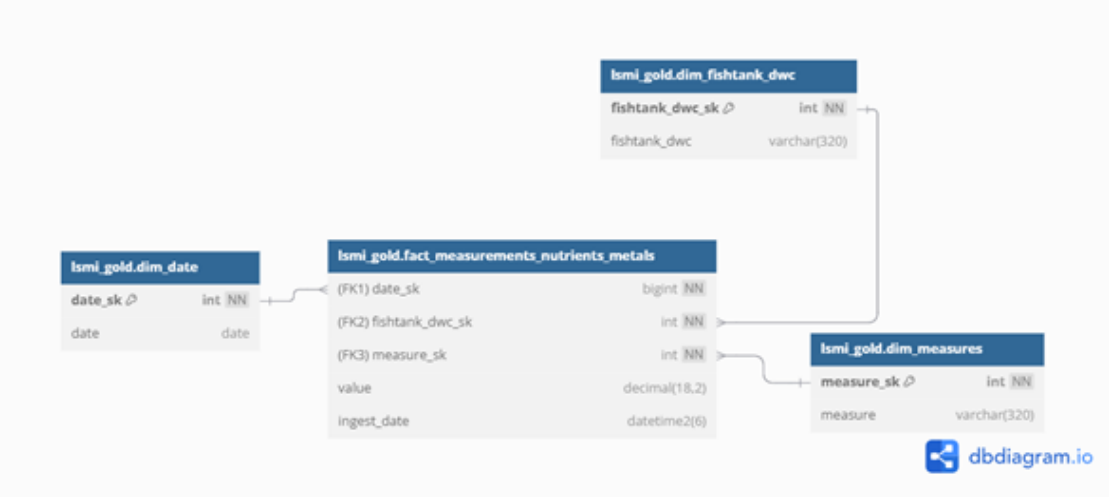


Figure 5.24: *Star schema with fact table, lsmi_gold.fact_measurements_nutrients_metals*

Fish Events:

The diagram illustrates, Figure 5.25, a star schema, in which the central fact table, designated as *lsmi_gold.fact_fish_events*, is connected to three-dimension tables.

- *lsmi_gold.dim_date*
- *lsmi_gold.dim_fishtank_dwc*
- *lsmi_gold.dim_fish_attributes*

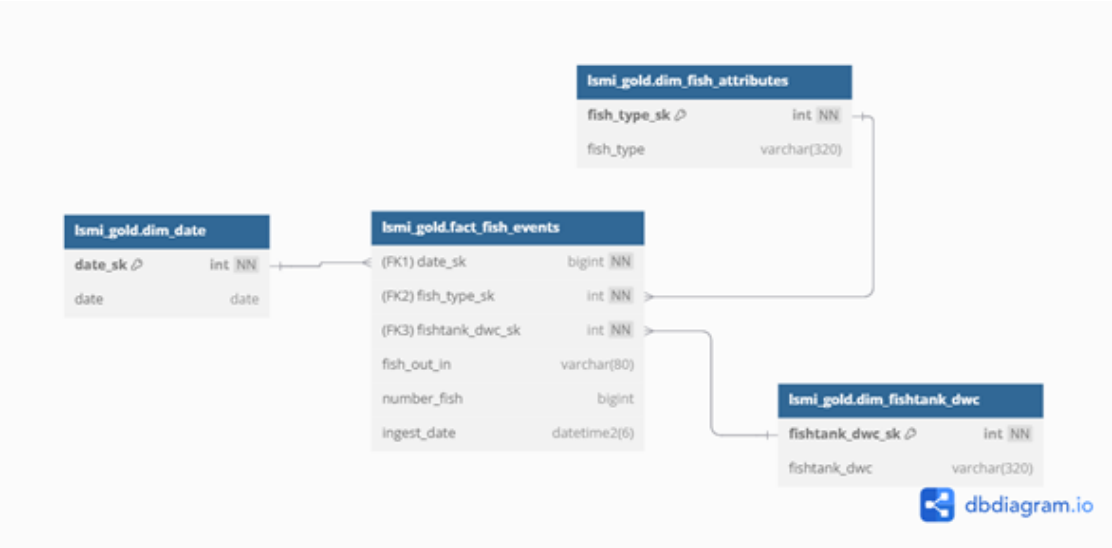


Figure 5.25: *Star schema with fact table, lsmi_gold.fact_fish_events*

Plants Development:

The Figure 5.26 illustrates a star schema centered on the fact table: *lsmi_gold.fact_plants_development*. Each record captures a single development measurement, such as fruit weight, tasty, or skin roughness, for a specific plant within a project, on a given date and production line, along with the ingestion timestamp. The fact table is linked via one-to-many relationships to the following dimensions:

- *lsmi_gold.dim_plants*
- *lsmi_gold.dim_measures*
- *lsmi_gold.dim_projects*
- *lsmi_gold.dim_lines*
- *lsmi_gold.dim_date*
- *lsmi_gold.dim_fruit*

All relationships between the fact tables and their corresponding dimension tables are defined as one-to-many, with each dimension record linking to multiple fact records.

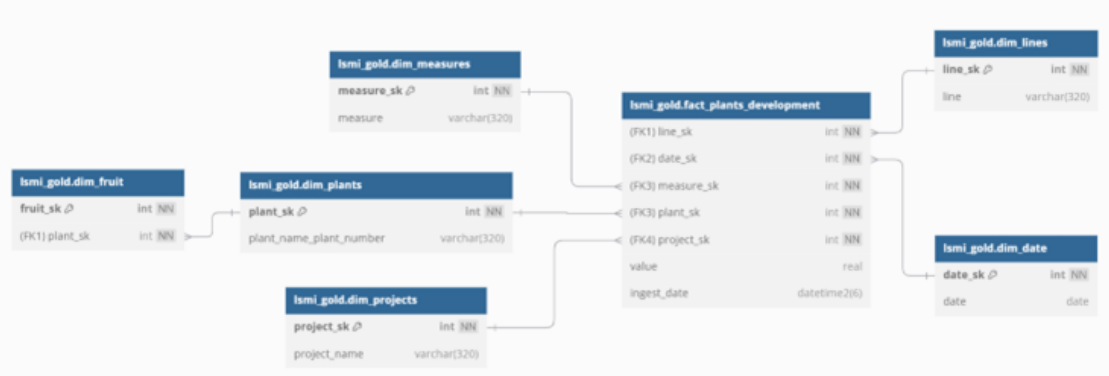


Figure 5.26: *Star schema with fact table, lsmi_gold.fact_plants_development*

5.3 Logical Data Mapping

5.3.1 Dimensions Tables

This section presents a table listing all dimension tables along with relevant metadata. The matrix maps the source (Bronze), Silver and target (Gold) layers, providing key attributes for each dimension. For the sake of clarity and conciseness, only the *dim_measures*, Table 5.4 and Table 5.5, will be included in this report.

gold_table	column_name_gold	description	change	silver_table
dim_measures	measure_sk	Integers generated by SQL Spark	Yes	measures
dim_measures	measure	Quantitative value that is recorded and analyzed in a fact table	Yes	measures
dim_measures	units	Unit of measurement of the measure	No	measures
dim_measures	measure_group	Group to which the measure belongs	Yes	measures
dim_measures	measure_type	Type of measure, example: parameter, metal, nutrient	No	measures
dim_measures	fact_table	Fact table to which the measure belongs	Yes	measures
dim_measures	min_value	Minimum value that the measure can have	Yes	measures
dim_measures	max_value	Maximum value that the measure can have	Yes	measures
dim_measures	description	Key description	Yes	measures
dim_measures	star_date	Date when the record became available	Yes	measures
dim_measures	end_date	Date when the record became available	Yes	measures
dim_measures	ingest_date	Date when the data was ingested	Yes	measures

Table 5.4: *Metadata of the dimension table dim measures*

column_name_silver	bronze_table	column_name_bronze	relationship	domain
measure_sk	measure_shpt	measure_sk	NULL	Integers generated by SQL Spark
measure	measure_shpt	measure	NULL	Measure name param pH, Temperature, TDS, ORP
units	measure_shpt	units	NULL	Text
measure_group	measure_shpt	measure_group	NULL	Text
measure_type	measure_shpt	measure_type	NULL	Text
fact_table	measure_shpt	fact_table	NULL	Text
min_value	measure_shpt	min_value	NULL	Decimal numbers
max_value	measure_shpt	max_value	NULL	Decimal numbers
description	measure_shpt	description	NULL	Text
start_date	measure_shpt	start_date	NULL	Timestamp with format: 2025-02-21 08:02:16.854267
end_date	measure_shpt	end_date	NULL	Timestamp with format: 2025-02-21 08:02:16.854267
ingest_date	measure_shpt	ingest_date	NULL	Timestamp with format: 2025-02-21 08:02:16.854267

Table 5.5: *Metadata of the dimension table dim measures (continuation)*

The *lsmi_gold.metadata_dimension_tables* table contains information about other dimension tables. These tables were generated in a notebook using Spark SQL and PySpark. The full code is available in the following notebook: *nb_lkh_lsmi_metadata.ipynb*.

However, since the structure of the dimension tables remains identical in both the Bronze and Gold layers, the *dim_date*, *dim_time*, and *dim_am_pm* tables were not included in the table.

5.3.2 Fact Tables

In Appendix D, two tables are presented that map the measure columns of the fact tables between the source and the target in the dimensional model (Table D.1 and Table D.2).

It is also important to demonstrate the relationship between the foreign keys in the fact tables and the primary keys in the dimension tables, Table 5.6.

Both tables can be accessed in the *metadata_fact_tables* and *metadata_fact_dimensions_relationship* tables, respectively, on the server of IPLeiria, Figure 5.27.

gold_table	column_name_gold	relationship
fact_atmospheric_conditions	line_sk	dim_line.line_sk
fact_atmospheric_conditions	date_sk	dim_date.date_sk
fact_atmospheric_conditions	measure_sk	dim_measure.measure_sk
fact_daily_system	line_sk	dim_line.line_sk
fact_daily_system	date_sk	dim_date.date_sk
fact_daily_system	measure_sk	dim_measure.measure_sk
fact_fish_events	fish_type_sk	dim_fish_type.fish_type_sk
fact_fish_events	line_sk	dim_line.line_sk
fact_fish_events	date_sk	dim_date.date_sk
fact_measurements_nutrients_metals	fishtank_dwc_sk	dim_fishtank_dwc.fishtank_dwc_sk
fact_measurements_nutrients_metals	line_sk	dim_line.line_sk
fact_measurements_nutrients_metals	measure_sk	dim_measure.measure_sk
fact_plants_development	line_sk	dim_line.line_sk
fact_plants_development	date_sk	dim_date.date_sk
fact_plants_development	project_sk	dim_project.project_sk
fact_water_consumption	line_sk	dim_line.line_sk
fact_water_consumption	measure_sk	dim_measure.measure_sk
fact_water_consumption	am_pm_sk	dim_am_pm.am_pm_sk

Table 5.6: *Relationship between the foreign keys and primary keys*

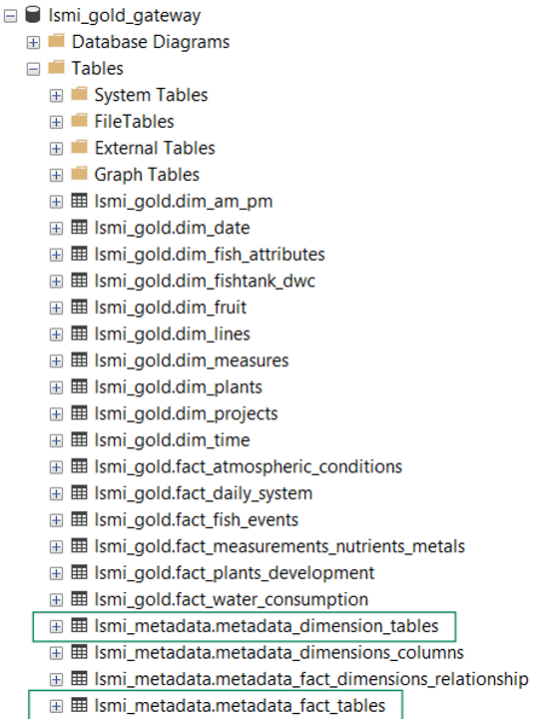


Figure 5.27: *Metadata tables in the lsmi_gold_gateway database*

6

Lakehouse Implementation and Data Integration Project – ETL

6.1 Data Solution Architecture

One of the objectives of this project was to adopt a cloud-based platform capable of supporting a modern data architecture aligned with the requirements of the aquaponics monitoring system. Data Warehouses, Data Lakes and Lakehouses play essential roles, each offering distinct advantages and limitations. Their respective characteristics, benefits, and trade-offs have been previously discussed in **Chapter 3**. Based on this analysis, the Lakehouse architecture was selected for the implementation of the solution. The justification for the choice is provided in the **Subsection 4.3**.

To implement the Lakehouse architecture, the comparative overview presented in Table 3.2 in **Subsection 3.4.3**, which evaluates Snowflake, Google Cloud BigQuery, Databricks, and Microsoft Fabric, was examined. Figure 3.6 was also analyzed to inform the selection of the most suitable Lakehouse platform.

Microsoft Fabric and Databricks were preselected based on their strategic fit and practical benefits for this project. Both platforms provide comprehensive trial offerings and are widely recognized for their Lakehouse, oriented capabilities in modern data management. Familiarity with their respective ecosystems further smoothed adoption, accelerating development and minimizing the learning curve. Additionally, they share similar underlying technologies, such as distributed computing paradigms listed in Table 3.2. Databricks excels with, auto-scaling, while Microsoft Fabric delivers elastic scalability without infrastructure management.

After careful consideration, Microsoft Fabric was selected for this project. All students and teachers at IPLeiria already possess Microsoft accounts and can access the Microsoft Fabric trial without creating additional credentials, whereas on Databricks would require every participant to register for a separate Databricks account, making

Fabric the more convenient choice. Moreover, although the Databricks Free Edition allows sharing individual notebooks within a workspace via the *Share* button [77], it lacks advanced production features consequently, there is no mechanism for sharing pipelines or an entire Lakehouse with fine-grained access controls [78]. Full asset management and pipeline collaboration on Databricks necessitate a paid subscription. [78].

Another major advantage of Microsoft Fabric lies in its fully integrated and unified environment access [79] [51]. It brings together all the essential components for modern data solutions, such as Data Warehousing, Lakehouse architecture, Pipelines, Notebooks, and Dataflow Gen2 within a single platform [51]. This seamless integration simplifies the development process, reduces architectural complexity, and enhances maintainability throughout the data lifecycle [51].

Databricks requires external BI tools for reporting, whereas Microsoft Fabric provides native Power BI integration. Microsoft Fabric scalability and flexibility make it particularly well suited for iterative development and the evolving needs of a project like the aquaponics monitoring system. Additionally, Fabric's integrated environment enables rapid experimentation and validation of new data models and dashboards, satisfying the curiosity about system behavior and performance.

In the analysis and visualization phase, Power BI Desktop (rather than the Fabric - embedded service) will be employed to create interactive reports and dashboards. This decision reflects our reliance on a Microsoft trial subscription, which automatically renews every two months but may expire without notice. To guard against potential data loss, the Silver and Gold layers have been periodically copied to an on-premises SQL Server at IPLeiria. Power BI Desktop will connect to this SQL Server, leveraging its ability to handle large volumes of data and generate customized reports for various objectives.

Thus, the solution architecture, illustrated in Figure 6.1, was designed to gather and process various data sources through Microsoft Fabric. The source files, primarily in Excel format, are stored in Microsoft Teams as raw data.

To structure the data transformation process efficiently, a Medallion Architecture was adopted, comprising the Bronze, Silver, and Gold layers. As discussed in **Chapter 3**, this approach promotes data quality, reusability, and performance optimization. Raw data is first ingested into the Bronze layer, then cleaned and enriched in the Silver layer, and finally aggregated and modeled in the Gold layer to support analytical workloads. This layered structure ensures traceability, modularity, and a clear separation of concerns across the data lifecycle.

There are several options for data ingestion, including Dataflow Gen2 and Copy Activity in a Data Pipeline. However, since the source files are stored in SharePoint folders, Dataflow Gen2 is the only viable option in this case Figure 6.2. It provides a direct

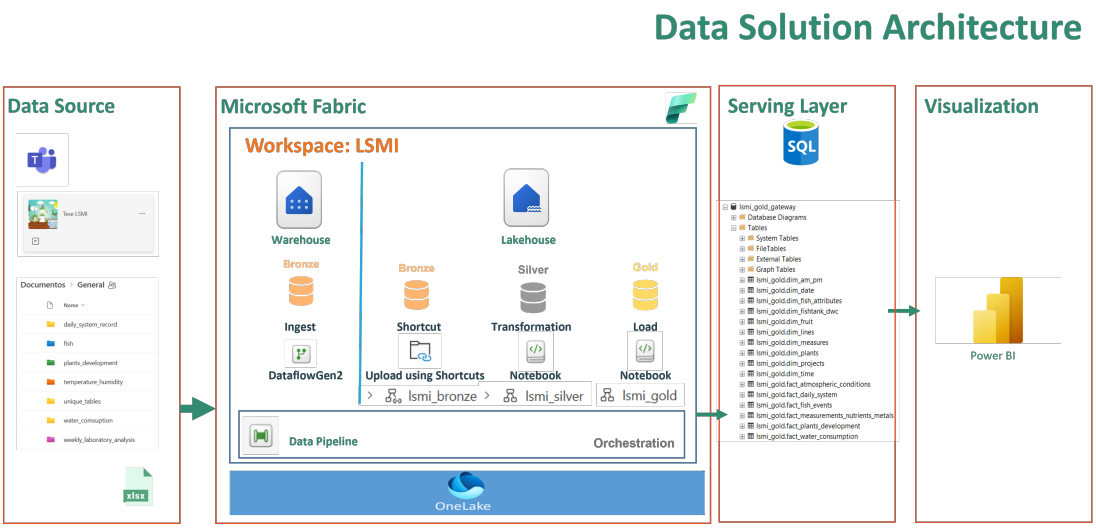


Figure 6.1: *Data Solution Architecture*

connector for SharePoint folders, whereas Data Factory in Microsoft Fabric does not currently support SharePoint folders in pipelines [80].

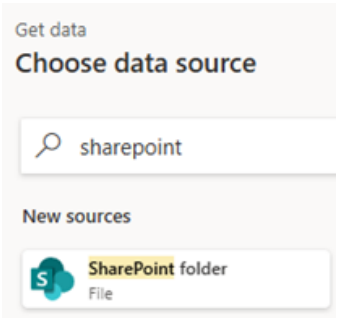


Figure 6.2: *Dataflow Gen2 SharePoint folder connector*

Building on the previous discussion, these files are ingested into the Fabric environment using Dataflow Gen2 and then copied to a Warehouse for further processing.

It was possible to write the data processed by Dataflow Gen2 directly into the Lakehouse or Warehouse, the decision depends on the specific needs of the organization. The Warehouse was chosen. The rationale for this decision lies in the subsequent data quality processes, particularly the analysis of null values and duplicates. Within the data ingestion Pipeline, these analyses require querying the data, which can be accomplished using either the *script* or *lookup* activities of the Pipelines, as illustrated in Figure 6.3. It is important to note that the *script* activity supports connections exclusively to SQL Databases or Warehouses, while the *lookup* activity is also capable of connecting to the Lakehouse. The *script* activity was chosen due to its lower latency when interfacing with a Warehouse. This process will be described in greater detail in **Section 6.3**.

It's important to note that the SQL analytics endpoint in a lakehouse is designed to

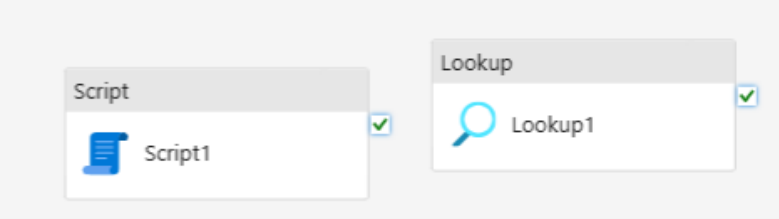


Figure 6.3: *'Script' or 'Lookup' activities of the Pipelines*

provide high-performance, low-latency SQL queries, utilizing the same engine as the Warehouse. However, under certain conditions, such as when there are numerous small-sized parquet files or high cardinality in partition columns, there can be delays in syncing changes between the Lakehouse and its associated SQL analytics endpoint. These factors can impact on the performance and latency of queries executed against the Lakehouse, [81].

Therefore, in scenarios where immediate data consistency and minimal latency are critical, connecting directly to a Warehouse may offer more predictable performance. This consideration influenced the decision to utilize the *script* activity with a warehouse connection in the data ingestion pipeline.

Subsequently, the data is replicated from the Bronze layer Warehouse to the Bronze schema of the Lakehouse using a *shortcut*. This approach was chosen instead of applying transformations directly to the Bronze layer within the Warehouse and subsequently writing the results to the Lakehouse.

Shortcuts in a Lakehouse enable users to reference data without duplication, allowing seamless integration across multiple sources, including other Lakehouses, Eventhouses, Workspaces, and external storage such as ADLS Gen2 or AWS S3. This approach provides fast, local access to large datasets without the latency of physical data movement [82].

In this project, shortcuts were used to create virtual references from the Warehouse to the Lakehouse, improving storage efficiency and ensuring that both layers remain synchronized. This method minimizes redundancy and reduces the risk of inconsistencies during the ETL process. Moreover, it enables Silver layer transformations to be performed directly on the Lakehouse while leveraging the structured schema defined in the Warehouse, resulting in better performance and simplified orchestration by avoiding unnecessary data transfers.

Overall, the use of shortcuts supports better performance, easier maintenance, and cost-effective resource usage, aligning with the architectural principles defined in the Medallion model.

As shown in the architecture **Figure 6.1: Data Solution Architecture**, within the Lake-

house environment, data is organized into Bronze, Silver and Gold layers. To write tables in the Silver and Gold layers, were used notebooks to take advantage of distributed computing technology and parallel processing, enabling efficient data transformations. This approach allowed for the use of both SQL Spark and Pyspark [83].

Theoretically, to fully leverage the benefits of the Lakehouse architecture, the Gold layer should serve as the foundation for a semantic model, which would then be used as the data source for Power BI reports. The optimal approach would involve establishing a Direct Lake Mode connection between Power BI and the semantic model.

Direct Lake Mode is an innovative engine capability that enables Power BI to analyze large datasets efficiently by consuming parquet-formatted files directly from a data lake. This eliminates the need to query a Warehouse or a SQL analytics endpoint, as well as the necessity of importing or duplicating data into a Power BI semantic model. By allowing Power BI to access data directly from the data lake, Direct Lake Mode provides a high-performance query and reporting experience, ensuring faster data retrieval and analysis [84].

However, this approach was not implemented in this study due to the limitations of the trial environment. As previously mentioned the Microsoft Fabric trial has been renewed every two months, but its expiration date remains uncertain, to avoid potential loss of data and ensure the continuity of the report, an alternative solution was adopted. Instead of relying entirely on Microsoft Fabric, a copy of the Gold layer was stored on a server at IPLeiria. This dataset was then connected to Power BI, ensuring data persistence and uninterrupted access to reporting capabilities.

6.2 Description of the ETL Process

According to Kimball, ETL is a data integration process that involves three main stages. The first stage, extraction, involves collecting data from various sources such as APIs, databases and files. The second stage, transformation, includes cleaning, enriching, and transforming the data to make it useful for analysis. The final stage, loading, involves inserting the transformed data into a storage system, such as a data warehouse or data lake. This process is essential for ensuring that data is ready for analysis and decision-making [14].

As previously mentioned, the ETL process will be implemented in Microsoft Fabric. A dedicated workspace was created for this purpose, containing all the created items organized into folders, with each folder holding items of the same type.

Each created item was assigned a name following a specific naming convention: *item _workspace_table*. For example, the DataflowGen2 for the *daily_system* table is named *df_lsmi_daily_system*. The following initials are used for other items:

- DataflowGen2 - df
- Notebook - nbk
- Pipeline - pp
- Lakehouse - lkh
- Warehouse - dw
- PowerBI - pbi

It is important to note that in the following subsections the ETL process will be explained; however, only the process for the *daily system* table will be detailed, since the remaining tables were processed in the same way.

6.2.1 Extraction

In the LSMI data integration project, it began with the extraction process. As previously mentioned, the source files are stored in various folders in Microsoft Teams, in the **Tese LSMI** Team.

Dataflow Gen2 in Microsoft Fabric is a cloud-based data preparation technology that enables users to connect, transform, and combine data from multiple sources in a low-code environment using the Power Query Online interface [85]. It offers over 300 data and AI-based transformations, making it easier to process and load data into destinations such as Lakehouse, Warehouse, or Kusto Query Language (KQL) Database within Microsoft Fabric [86].

Data was ingested from the source and loaded into the Bronze layer of the Warehouse using overwrite mode, with only new or modified files processed since the last ingestion. The source files presented several challenges that hindered a fully automated ingestion, requiring some manual adjustments to facilitate the process. Due to the large volume of data, not all standardization transformations were performed in Excel; instead, these were executed within Dataflow Gen2. Although some transformations were applied during ingestion, according to best practices, the ideal approach would have been to load the files directly into the Bronze layer through Dataflow Gen2 without applying transformations at this stage.

The process of Data Extraction is detailed below.

- Dataflow Gen2 connects to SharePoint Files using *SharePoint.Files* (link, [ApiVersion = 15]), Figure 6.4.

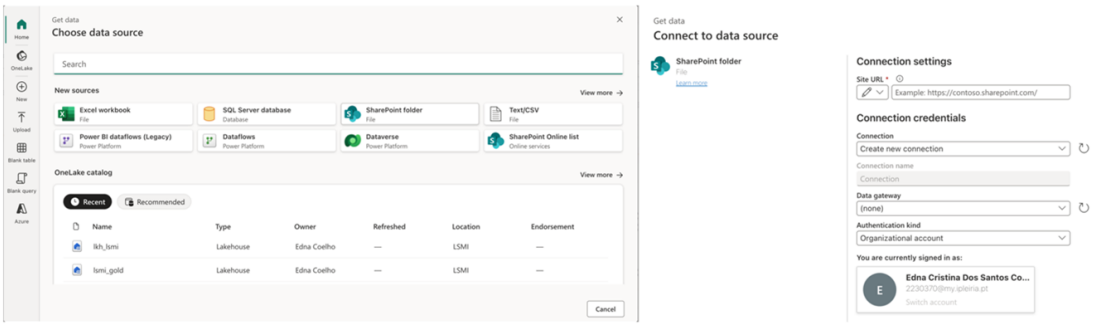


Figure 6.4: *Get data from source Sharepoint folder and connection*

- Filter the files to include only those containing *v4* in the filename.
- Select relevant columns (*Content*, *Name*, *Date modified*, *Folder Path*).
- Remove hidden files.
- Add an ingestion date (*ingest_date*) using a *lookup* from the *silver_daily_system* table or assign a default date. Filter files based on modification date, loading only those where the ingestion date is earlier to the file's modification date (*ingest_date* $\leq$ *date_modified*). Ensures incremental ingestion by processing only files that have been added or modified.
- Expand the data structure by applying a custom transformation function to extract relevant fields.
- Remove unnecessary columns to optimize the dataset.
- Standardize column data types to ensure proper formatting.
- Rename columns to follow a structured naming convention.
- Filter out header rows that contain *Line* as a value.
- Replace specific values in the *Line* column (e.g., *linha_1* $\rightarrow$ *Line1*, *sistema_porta* $\rightarrow$ *DoorSystem*).
- Drop redundant columns (e.g., NH_3 mg/L, NO_2 mg/L, NO_3 mg/L) that are not needed for further processing.
- Format numeric and categorical fields to ensure consistency.
- Create a unique identifier (*daily_system_key*) by concatenating *line*, *fishtank_dwc*, *year*, *month*, *day* and *time*.
- Format the key as a string and remove any empty records.
- Add a timestamp (*ingest_date*) to track when the data was ingested.
- Remove unnecessary columns before loading the final dataset.
- This structured ETL process ensures that the ingested data is clean, structured, and ready for analysis within the Microsoft Fabric ecosystem (see Fig. 6.5).

Finally, the destination is added, which in this case is the table *lsmi_bronze_daily_system_shpt*, previously created in the *dw_lsmi* Warehouse, with overwrite data, Figure 6.6.

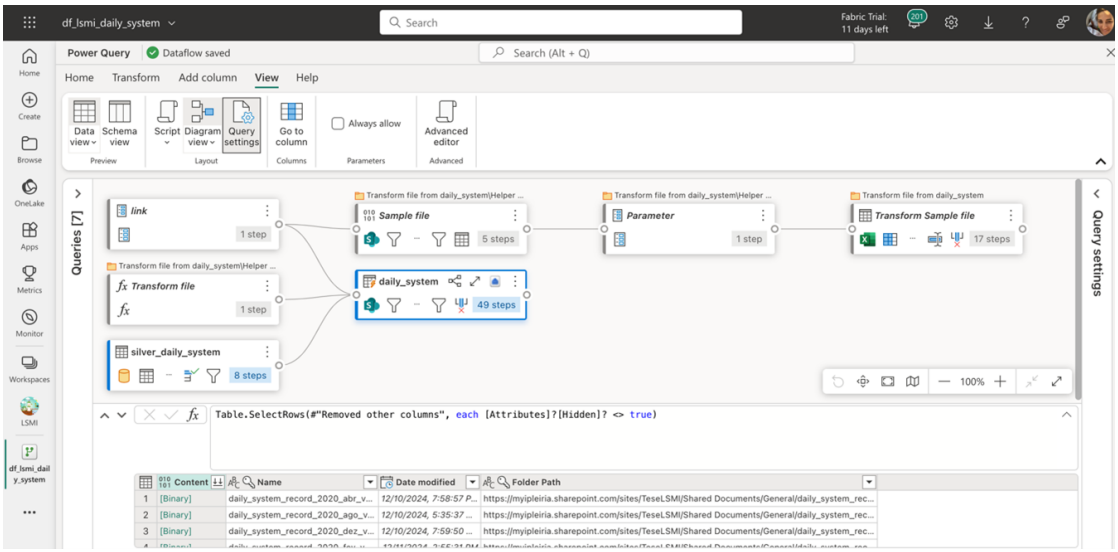


Figure 6.5: *Diagram view from DataFlowGen2 - df_lsmi_daily_system.png*

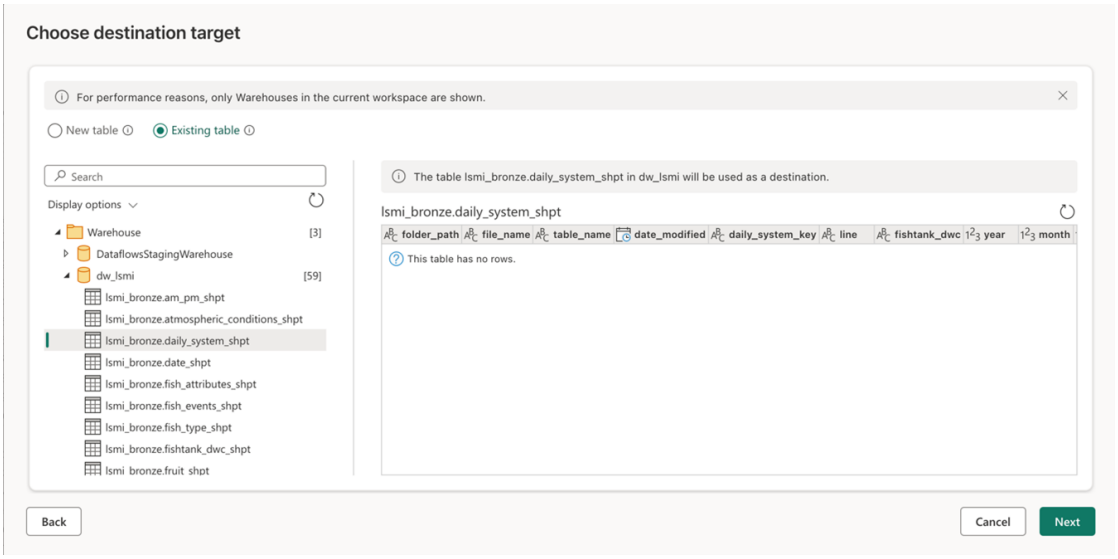


Figure 6.6: *Add a destination*

6.2.2 Transformation

After the data Extraction and loading into the Bronze layer, the next phase in the ETL process is data Transformation, where the raw data is refined and structured for analytical use. This transformation occurs in the Silver layer, where data is cleansed, enriched, and standardized to ensure consistency and usability. In this section, the key transformation steps applied to the daily system dataset will be described in detail, including:

- **Data Cleaning**: Handling missing values, removing duplicates, and standardize formats.
- **Data Structuring**: Converting raw attributes into well-defined entities.
- **Business Rule Application**: Implementing transformations based on predefined logic.

These transformations are essential for ensuring that the data is reliable, structured, and ready for aggregation and reporting in the Gold layer. The following subsections will provide a detailed breakdown of each transformation applied in the Silver layer of the Lakehouse.

This phase of the process is carried out in the Lakehouse *lkh_lsmi*. In the Lakehouse, a shortcut to the Bronze layer of *lsmi_dw* is created, Figure 6.7.

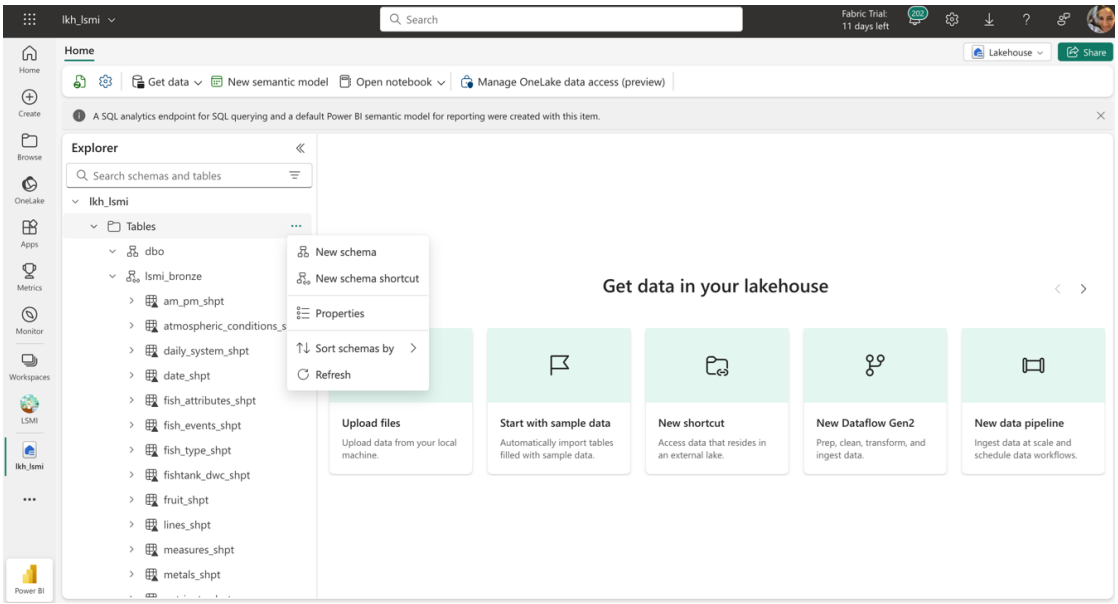


Figure 6.7: *Shortcut from dw_lsmi to lkh_lsmi in lsmi_bronze*

In the notebook *_scripts_silver_gold_tables*, the Silver and Gold layer tables are created, including both dimensional and fact tables. It is important to note that all Silver tables have the following common columns:

- folder_path

- file_name
- date_modified
- ingest_date

Below is an example of the SQL Spark code used to create the table *lsmi_silver.daily_system*:

```
CREATE TABLE lsmi_silver.daily_system(
    folder_path varchar(8000),
    file_name varchar(100),
    table_name varchar(100),
    date_modified timestamp,
    daily_system_key varchar(1000),
    fishtank_dwc varchar(20),
    fishtank_dwc_sk int,
    date_sk bigint,
    time varchar(20),
    am_pm_sk int,
    measure varchar(20),
    measure_sk int,
    value float,
    ingest_date timestamp
)
using DELTA
```

The notebook *nbk_lkh_lsmi_daily_system_bronze_silver* performs ETL transformations from the Bronze layer to the Silver layer for the *daily_system dataset* within Lakehouse *lkh_lsmi*. The key steps involved are:

Data Validation & Cleaning

- Identifies duplicates using a Common Table Expression (CTE) and filters records where *daily_system_key* appears more than 10 times.
- Handles missing values by replacing empty or whitespace values with *NULL*.
- Removes non-breaking spaces (`\u00A0`) from numerical fields to ensure proper parsing.

Data Type Standardization

- Ensures that critical numerical columns like DO mg/L, temperature (°C), and pH are formatted correctly.
- Uses *REGEXP_REPLACE()* to clean data inconsistencies in text-based fields.

Key Transformations

- Extracts relevant columns from the Bronze layer (*lsmi_bronze.daily_system_shpt*).

- Standardizes column names and applies formatting improvements.
- Ensures that the dataset is structured correctly for further aggregation.

Data Loading into the Silver Layer

- Write the transformed data into *lsmi_silver.daily_system*.
- Uses optimized queries for handling large datasets, with SQL Spark: *MERGE INTO*.
- Prepares the dataset for the Gold layer and further analytical processing.
- This structured transformation process ensures that data in the Silver layer is clean, reliable, and formatted for efficient analysis within Microsoft Fabric's Lakehouse architecture.
- These types of transformations and updates for the Silver table were applied in the same way to all tables that will, in the Gold layer, become a fact table, as is the case with *daily_system*.

The code used to perform all these transformations can be found in: **Access to notebooks**.

In the dimension tables, two approaches were used: Slowly Change Dimension (SCD) Type 1 and SCD Type 2. The SCD Type 1 dimension tables (fruit, plants, and project) were created because there was no need to retain historical data. The remaining dimension tables were implemented as SCD Type 2, as it was necessary to store historical changes.

Another particularity concerns 3 dimensions tables: date, hour, and am_pm. Since these tables do not change over time, they do not have a Silver layer. Instead, they are directly loaded from the Bronze layer to the Gold layer using an *INSERT OVERWRITE* operation.

It is important to highlight that a comprehensive data quality process was implemented at the Silver layer to ensure data integrity. As part of this process, records containing null or duplicate values in the dimension tables are not ingested into the Silver layer. Instead, queries are executed to retrieve relevant metadata for further analysis. For null values, the query extracts the following fields:

- folder_path
- file_name
- date_modified
- modified_by
- column_key

For duplicate values, the query retrieves:

- folder_path

- file_name
- composite_key
- date_modified
- ingest_date
- The count of duplicate records

These queries are executed within the dedicated notebooks: *nbk_lsmi_dynamic_nulls* – for handling null values and *nbk_lsmi_dynamic_duplicates* – for identifying duplicate records.

Additionally, to obtain the *modified_by* attribute when dealing with null values, a separate query is required. This query references a metadata table *metadata_tables*, specifically created for this purpose, within the *lsmi_metadata* schema, Figure 6.8.

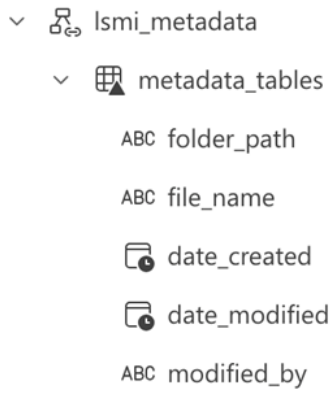


Figure 6.8: *lsmi_metadata schema, containing metadata tables and columns*

Once these records are identified, an email notification is sent to the data engineers and the personnel responsible managing the Excel files in Microsoft Teams. This allows them to review the affected records and take corrective action as needed.

Similarly, in the fact tables, a separate data quality check is performed. If a fact table record does not have a corresponding foreign key match in the related dimension table, it is stored in a dedicated table for further analysis. An example of such a table is:

- *lsmi_data_quality.aux_daily_system_key_not_dim_table*, Figure 6.9.

These records are also communicated via email to ensure that any data integrity issues are addressed promptly.

6.2.3 Load

In this phase, the Silver tables are loaded into the Gold layer, where the final structure is optimized for analytical processing. The fact tables retain only the columns containing measurable values and the foreign keys that link to the respective dimension tables. Meanwhile, the dimension tables include a primary key column along with several

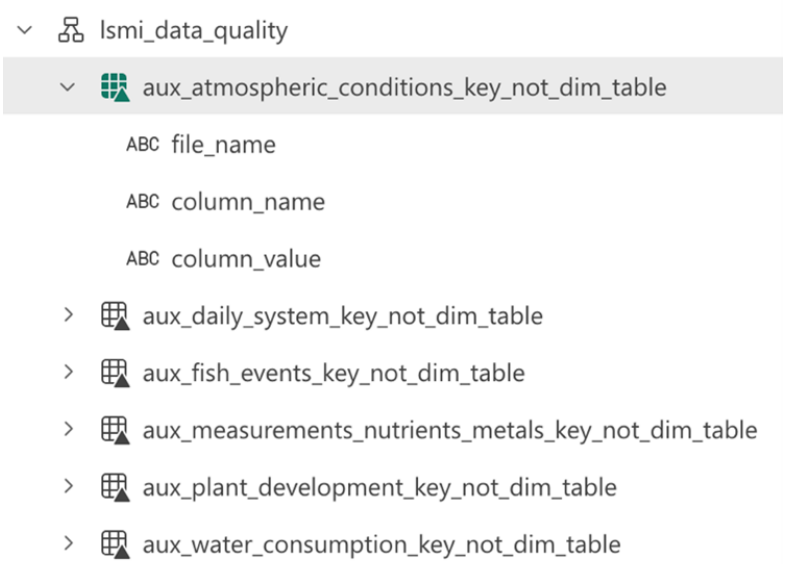


Figure 6.9: *lsmi_data_quality schema in lkh_lsmi*

attribute columns that describe the entity. This structured approach ensures that the Gold layer is well-organized for efficient querying, reporting, and BI applications. The process is executed in notebooks, and for *daily_system*, it is performed in notebook: *nbk_lkh_lsmi_daily_system_silver_gold.ipynb*.

To ensure data persistence and prevent potential loss, these Gold layer tables are replicated in a SQL Server database hosted on an IPLeiria server. Since this project is being developed using a Microsoft Fabric trial account, there is a risk that the data processed and stored within the Fabric environment could be lost once the trial period expires. By maintaining a backup in SQL Server, the transformed and structured data remains accessible for future use, ensuring the continuity and reproducibility of the work.

6.3 Orchestration

The orchestration of the entire ETL process was executed using Data Pipelines. The following section provides a detailed explanation of the Data Pipelines process. Dynamic pipelines were utilized whenever possible to enhance reusability, as the implementation process was identical for all tables. To achieve this, parameters were used within the pipelines Figure 6.10. Therefore, the explanation will focus on a single table, the *daily_system*.

Data Pipeline: pp_lsmi_bronze_layer_table_name

The first pipeline is *pp_lsmi_bronze_layer_table_name*, in this example *pp_lsmi_bronze_layer_daily_system*, is an automated pipeline for data ingestion and validation. This pipeline was developed to ensure the integrity of processed data and the detection of inconsistencies, such as null values and duplicate records, through a structured se-

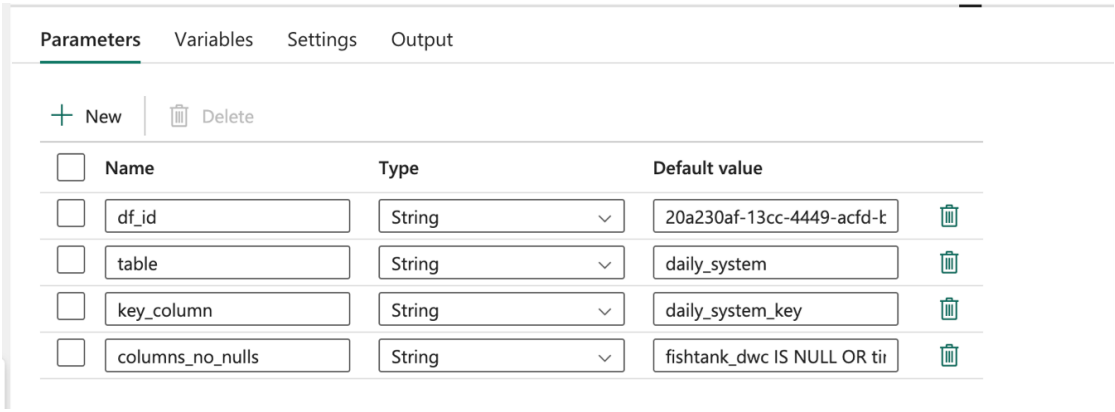


Figure 6.10: *Example of parameters of data pipelines*

quence of activities, Figure 6.11.

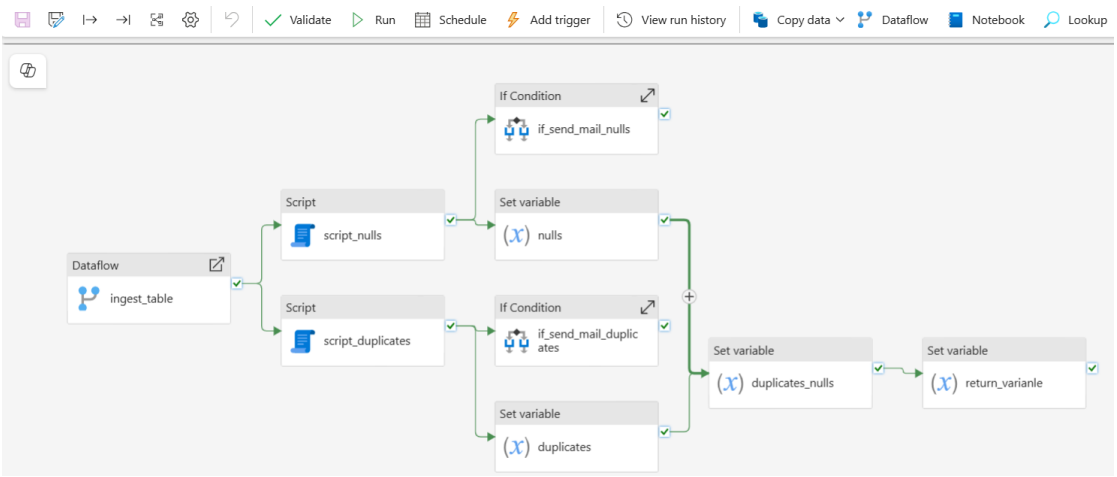


Figure 6.11: *Data Pipeline pp_lsmi_bronze_layer_daily_system*

If the number of duplicates exceeds two, the *if_send_mail_duplicates* activity is triggered by sending an email notification. If records with null values are found, the *if_send_mail_nulls* activity triggers an alert email. In this case, as previously mentioned, the maximum time granularity is divided into morning and afternoon. Multiple measurements for the same parameter may be recorded within each period to minimize errors. However, there will never be more than ten measurements for the same parameter in each period. If more than ten values are detected, duplicates will be identified, and an email will be sent for further analysis.

In addition to these checks, the pipeline manages variables to store duplicate and null count values. The *count_nulls* and *count_duplicates* variables record these counts, which are subsequently consolidated in the *duplicates_nulls* variable, storing the results in JSON format. The final step of the pipeline defines the *pipelineReturnValue* variable, encapsulating the results for future reference.

Whenever a pipeline is created in Microsoft Fabric, a corresponding JSON file is automatically generated. This JSON definition fully describes the pipeline's structure,

activities, parameters, and connections. The files can be found in **Access to Pipelines JSONs**. All pipelines follow similar structural logic. These JSON files are valuable not only for documentation purposes but also for operational flexibility. For instance, a pipeline can be replicated or migrated to a different workspace simply by importing the JSON file. This process ensures data integrity by automatically detecting and notifying any identified issues, contributing to a more reliable and efficient data flow in analytical and operational environments.

Data Pipeline: *pp_lsmi_exe_(dim_fact)_table_name*

The *pp_lsmi_exe_fact_daily_system* pipeline, Figure 6.12, is an automated process designed for data ingestion, validation, and transformation. The process begins by activating the *pp_lsmi_bronze_layer_table_name* pipeline, followed by verifying whether the Bronze table contains new records and ensuring that the number of null or duplicate values remains within an acceptable threshold. This verification is performed using the *if_bronze_table_have_rows* condition, which evaluates the outputs of the *number_rows_bronze_table* and *bronze_layer* activities. If no new data is detected or if the level of inconsistencies exceeds the threshold, an email notification is sent via *mail_pipeline_not_run* to indicate that the pipeline was not executed.

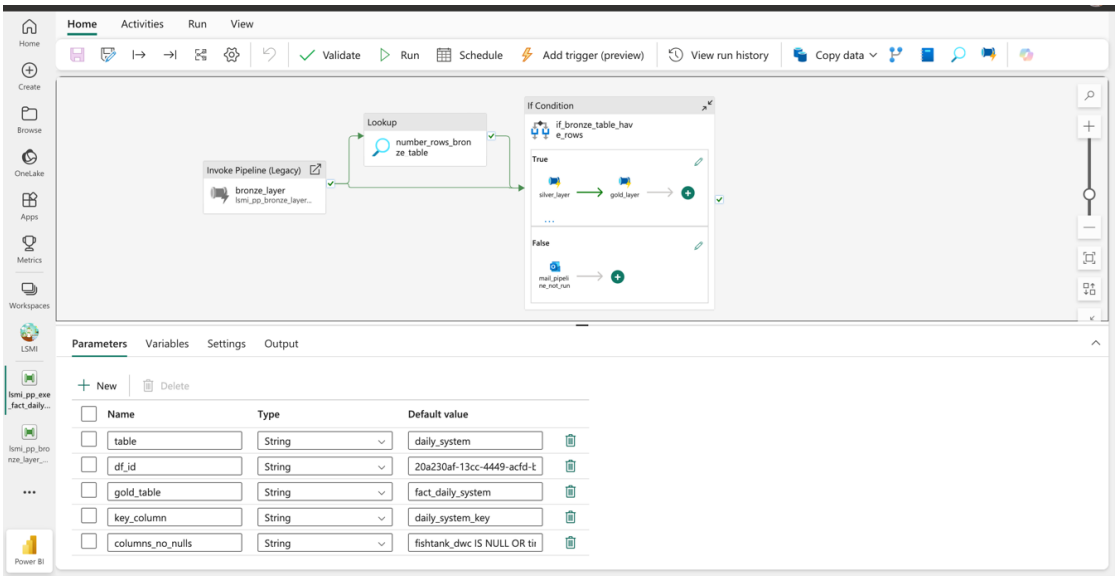


Figure 6.12: *pp_lsmi_exe_fact_daily_system pipeline*

If the conditions are met, the pipeline proceeds with data transformation through multiple layers. The *silver_layer* activity is triggered first, followed by the *gold_layer* process, ensuring a structured and refined data processing approach. Additionally, a validation step, *send_mail_keys_not_dimension_tables*, ensures that key values are correctly aligned with dimensional tables.

The pipeline monitors key data quality metrics, such as *count_nulls* and *count_duplicates*, to determine whether processing can continue. A lookup operation, *number_rows_bronze*

_*table*, counts the available records before advancing to higher processing layers. Processing in the Silver and Gold layers proceeds only if the Bronze table contains at least one record and has no more than five null values or duplicates, as shown in the following script. If these conditions are met, processing continues; otherwise, the pipeline execution is halted. The code used is from the data pipeline build expression, Figure 6.13.

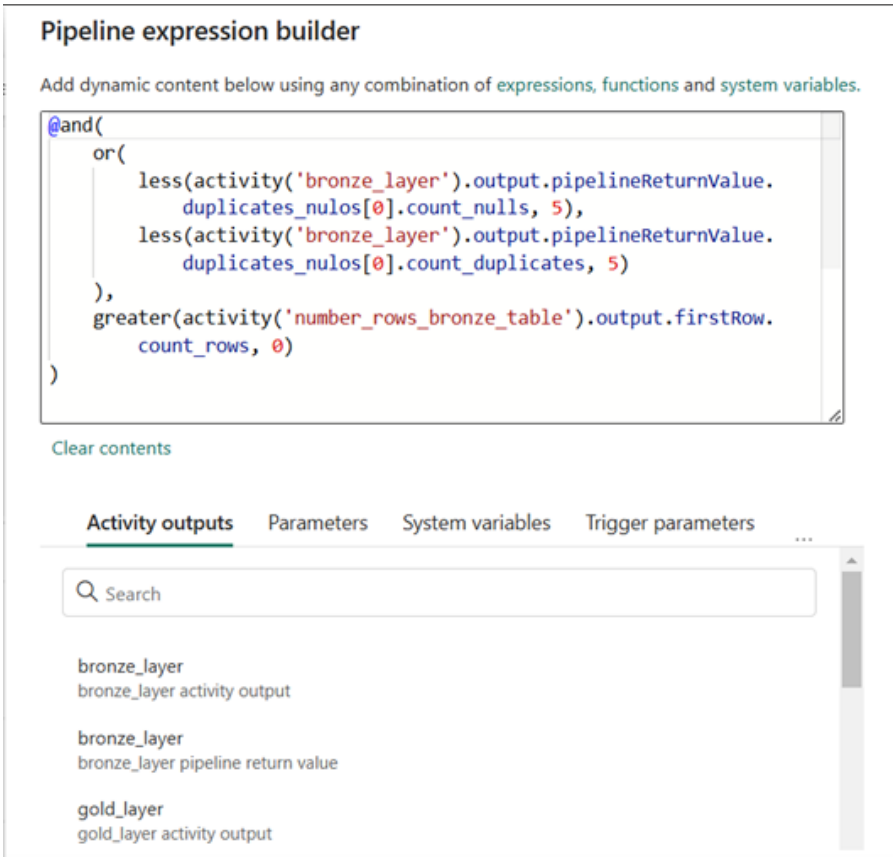


Figure 6.13: *Pipeline Expression for Data Quality Validation*

By leveraging dynamic parameters, the pipeline enables efficient reuse across different tables while maintaining a standardized processing workflow. This ensures automation, reduces manual intervention, and enhances the reliability of the data transformation process, contributing to more accurate analytical outcomes.

As previously mentioned, this entire process is applied to both dimension and fact tables. Each pipeline may have its specific characteristics depending on whether it processes a dimension table or a fact table.

Data Pipeline: ***pp_lsmi_gold_gateway_dynamic***

The final load of the Gold layer, Figure 6.14, was executed to an on-premises IPLeiria server, Figure 6.15, to Database: *lsmi_gold_gateway*.

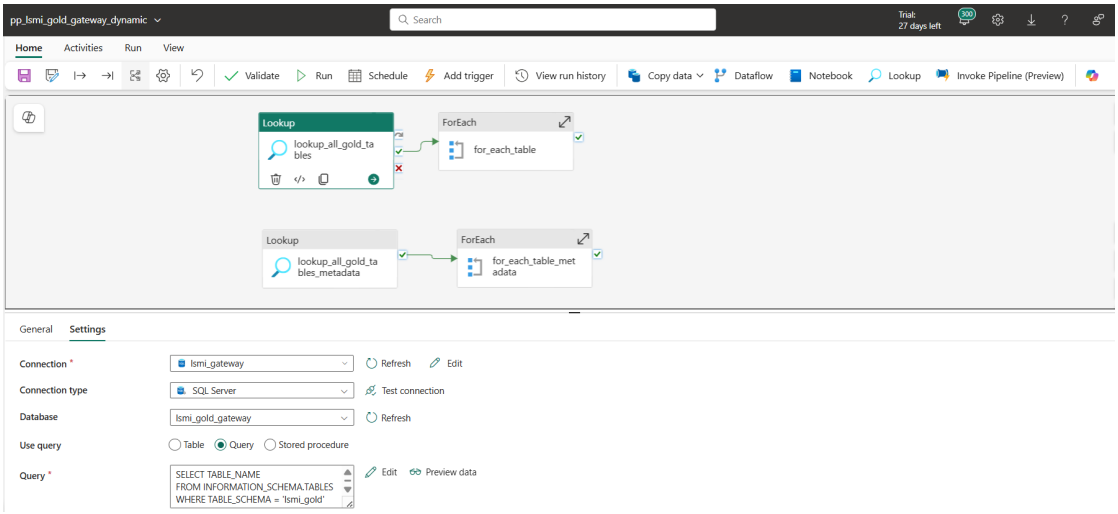


Figure 6.14: *Pipeline: pp_lsmi_gold_gateway_dynamic*

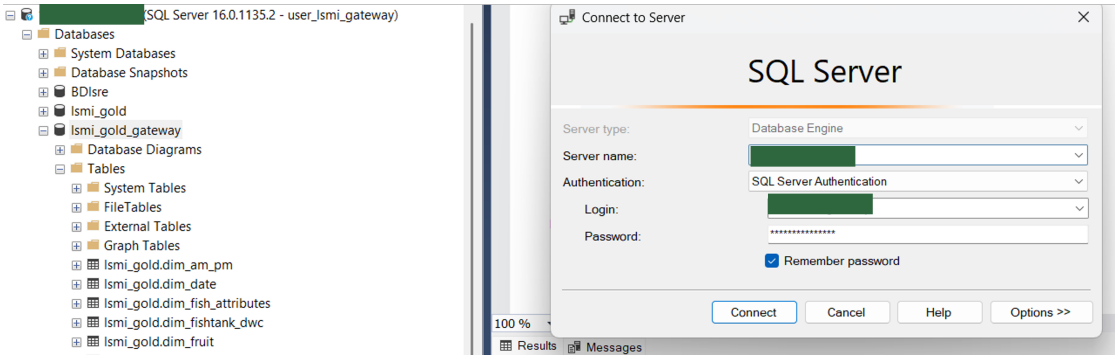


Figure 6.15: *SQL Server Management Studio (SSMS) Object Explorer Showing the LSMI Gateway Instance*

The Figure 6.16 illustrates the configuration of the Copy Data activity within the *pp_lsmi_gold_gateway_dynamic* pipeline, which is responsible for dynamically copying data to the *lsmi_gold_gateway* SQL Server. This activity is executed inside a *for_each_table* loop, allowing iteration over multiple tables. Prior to the copy operation, a script activity named *delete_table* is executed, likely to clear the destination table before inserting new data. In the "Destination" tab, the connection used is *lsmi_gateway*, with the connection type set to *SQL Server* and the target database specified as *lsmi_gold_gateway*. The option *Auto create table* is selected, allowing the system to automatically create the destination table if it does not already exist. The table name is assigned dynamically using the expression *@item(). TABLE_NAME*, enabling a flexible and reusable configuration across multiple tables within the pipeline.

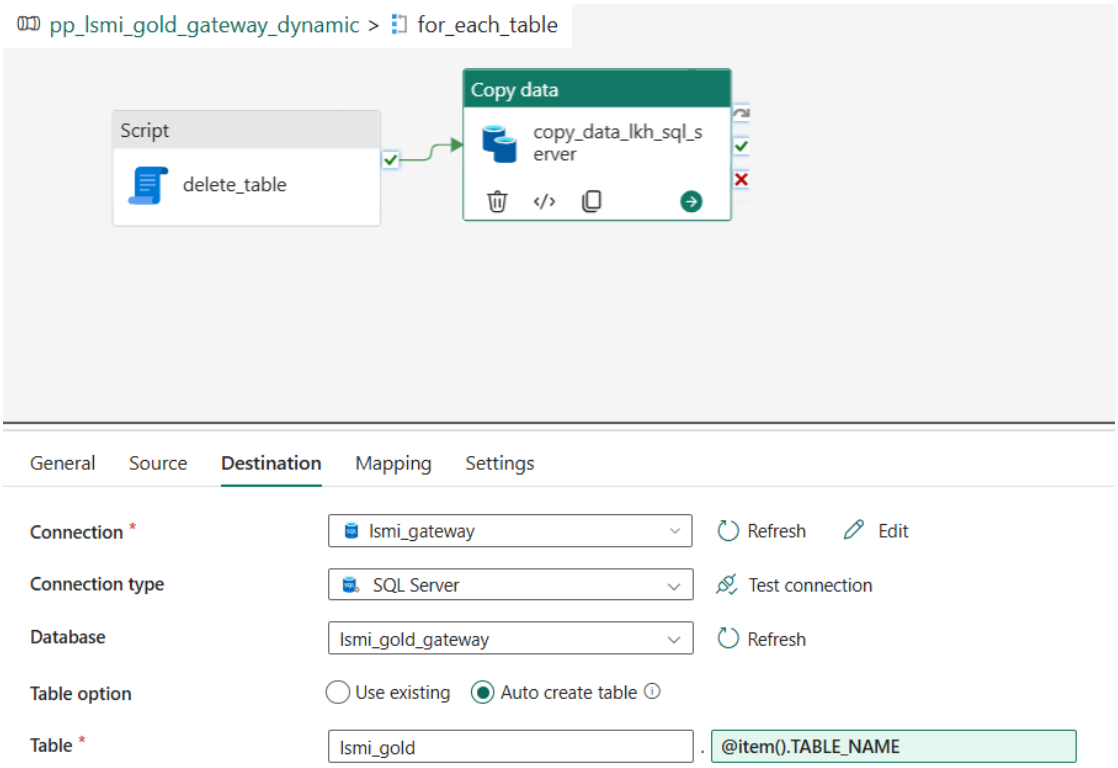


Figure 6.16: *Configuration of the connection to the lsmi_gold_gateway SQL Server in the pp_lsmi_gold_gateway_dynamic pipeline's Copy Data activity*

Data Pipeline: *pp_lsmi_master*

A master pipeline *pp_lsmi_master*, Figure 6.17, was created to orchestrate the execution of multiple data processing pipelines in a structured sequence.

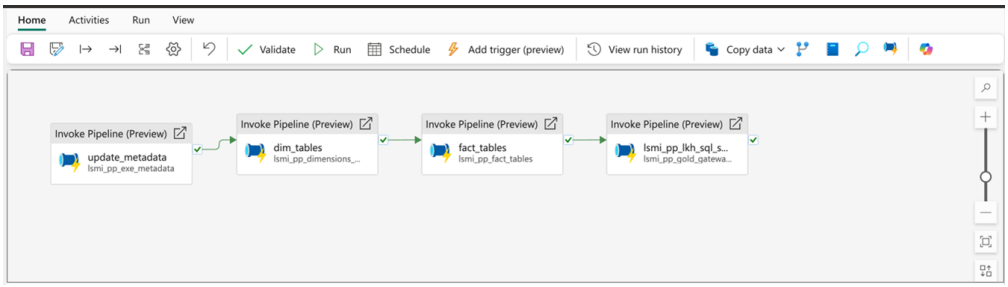


Figure 6.17: *pp_lsmi_master pipeline*

The process begins with the *update_metadata* activity, which updates relevant metadata before initiating subsequent operations. Once the metadata update is completed, the *dim_tables* pipeline is triggered to process dimension tables, ensuring their data integrity and readiness for analytical use. Following the successful execution of the dimension tables pipeline, the *fact_tables* pipeline is invoked to process fact tables, integrating and structuring transactional data for reporting and analysis. After the fact tables are processed, the final step in the pipeline execution is *lsmi_pp_lkh_sql_server*,

which likely ensures that all processed data is correctly transferred and stored in the designated database environment.

Each step in the sequence depends on the successful completion of the previous stage, ensuring a controlled and systematic data processing workflow. This structured orchestration enhances efficiency, maintains data integrity, and ensures seamless execution of data transformations across different tables.

A weekly trigger will be added to this pipeline, as shown in the configuration below, Figure 6.18. While most files are added or updated monthly at the source, occasional changes can occur at any time, making a weekly trigger the optimal choice.

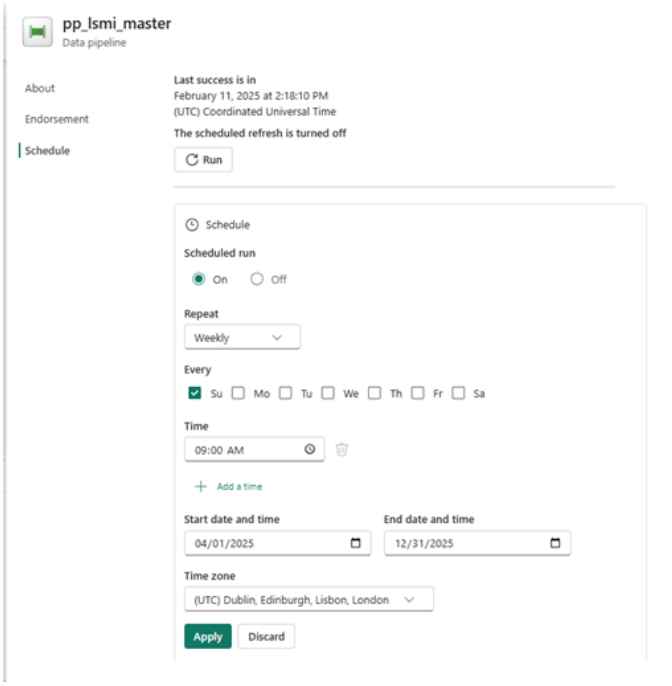


Figure 6.18: *Master Pipeline Trigger Configuration*

7

Building the Visual Aquaponic Project

Power BI serves as the front-end for exploring and visualizing the insights generated by the *lsmi_gold_gateway* database.

This Power BI solution was developed based on the specific requirements provided by the aquaponics research team at IPLeiria. The purpose of this chapter is not to present a storytelling approach, but rather to provide a detailed explanation of the configuration of Power BI Desktop to connect with the on-premises SQL Server (Gold layer), the sharing and publishing the settings that ensure stakeholders across IPLeiria have reliable access to up-to-date information, the design of the semantic model, and the structure of the interactive reports and dashboards. Additionally, this chapter outlines how users can navigate through the different pages of the Power BI report in a user-friendly and efficient manner.

7.1 Power BI Desktop Configuration and Secure Data Connection

In Power BI Desktop, the connection to the Gold layer tables on the on-premises IPLeiria server is configured using specific connection settings, Figure 7.1. For security reasons, the connection parameters such as machine address, database name, login, and password are not disclosed in detail here, but they follow the standard access credentials defined for the internal infrastructure.

Stakeholders can currently connect directly to the on-premises database, either to build their own dashboards or to access the Power BI solution developed for this thesis. Because anyone possessing these credentials gains full database access, it is recommended that passwords be rotated every three months (or according to IPLeiria's security policy). Embedding connection strings in this manner is not considered best

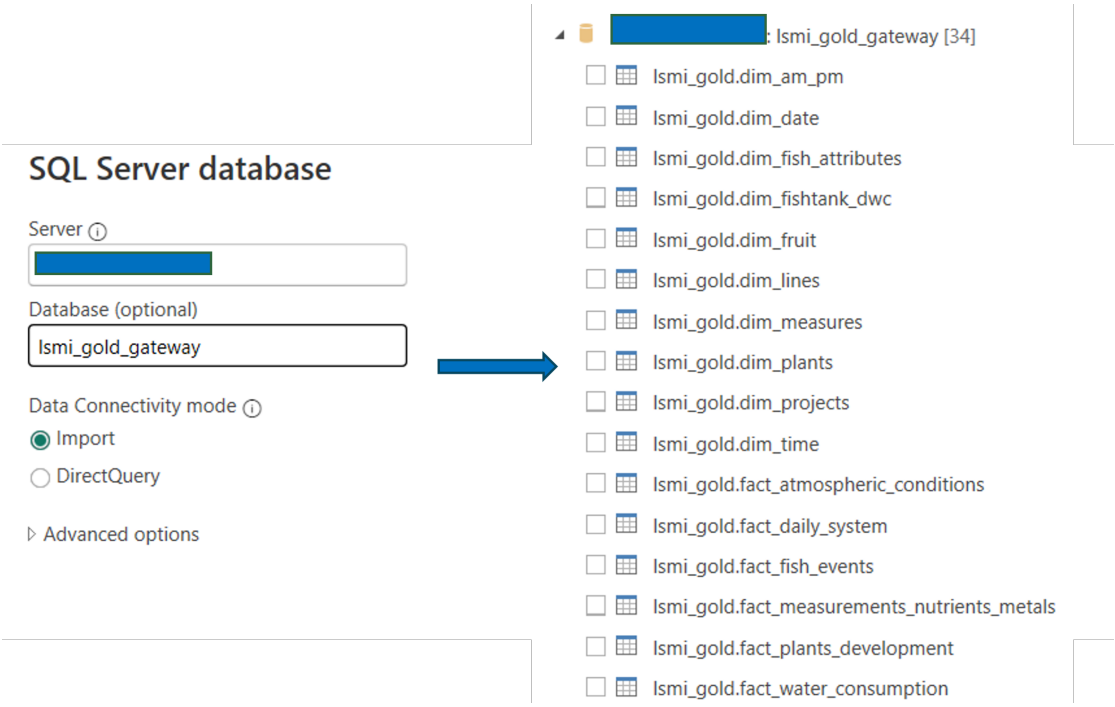


Figure 7.1: *Power BI Desktop SQL Server database Connection Dialog and tables to import*

practice. Although storing credentials in Azure Key Vault and granting access only to a designated Azure Active Directory (AAD) security group would enable centralized credential management and minimize exposure, this method was not adopted due to Azure Key Vault being a paid service under current budget constraints.

Users can access the dashboards either through Power BI Desktop or via the LSMI Project Workspace in Microsoft Fabric. To access the latter, it is necessary for the workspace owner to grant at least Viewer privileges to the intended users. This ensures controlled access while allowing stakeholders to consult the published reports directly through the Power BI service.

7.2 Semantic Data modelling in Power BI

7.2.1 Table Relationships and Cardinality

In Power BI's model, relationships define how tables are linked and how filters and aggregations propagate across the dataset. Each relationship is characterized by its cardinality, most commonly one-to-many or many-to-one in a star schema, which determines whether a single dimension record can relate to multiple fact records or vice versa [14]. Correctly specifying cardinality is essential for accurate calculations, optimized query performance, and predictable filter behavior.

In the LSMI model, all dimension-to-fact links use one-to-many cardinality, ensuring that each dimension entry (for example, a given date or fishtank) can be associ-

ated with multiple measure or development records without ambiguity. This foundation enables seamless drill-down, roll-up, and cross-filtering across reports and dashboards. The model view can be found in Appendix E: Model View in Power BI.

7.2.2 Hierarchies

As mentioned in **Section 5.2.3** defining hierarchies in Power BI improves both data navigation and user experience by enabling intuitive drill-down and roll-up operations. Organizing related fields, such as e.g: State > City > Store, allows users to explore data across multiple levels without manual selection. Hierarchies also enhance report interactivity and optimize query performance for faster analysis [87].

The figure Figure 7.2 illustrates the hierarchies defined in Power BI.

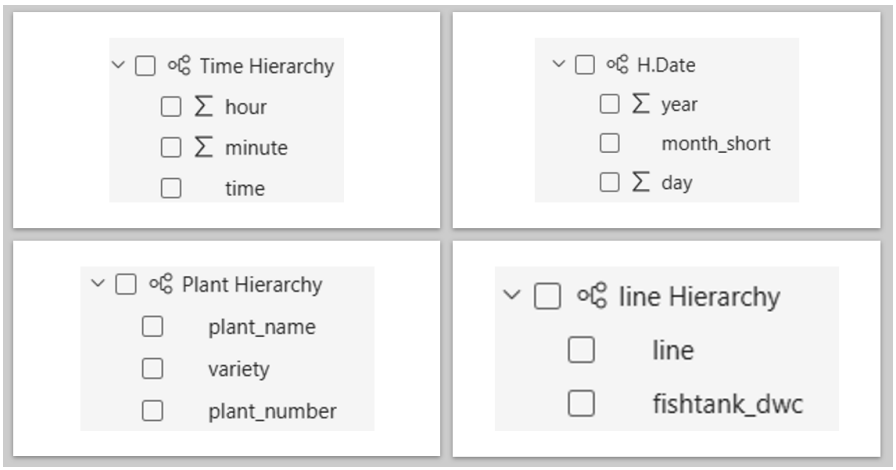


Figure 7.2: *Hierarchies in Power BI*

7.2.3 DAX Measures

In Power BI, measures and calculated columns are defined using the Data Analysis Expressions (DAX) language to enrich the semantic model with business logic. Measures compute dynamic aggregations at query time, such as sums, averages, and more [88]. To ensure transparency and reproducibility, all DAX measures defined in this model were exported and documented using DAX Studio, providing a complete catalog of expressions and performance metrics, Appendix F: DAX Measures.

In DAX Studio, Analysis Services Dynamic Management Views (DMVs) were used to list all measures from the model. By connecting to the .pbix file and executing one simple queries, Figure 7.3, it was possible to dynamically extract this information. A dynamic Excel file containing the exported data is included in: Download DAX Studio Measures Excel File.

These measures were developed with the specific purpose of underpinning the creation

```
1 select MEASUREGROUP_NAME as measuregroup_name, MEASURE_UNIQUE_NAME as measure_name, EXPRESSION as expressions
2 from $SYSTEM.MDSCHEMA_MEASURES
```

Log **Results** History

measuregroup_name	measure_name	expressions
fact_daily_system	[Measures].[daily_value_dynamic]	SWITCH(TRUE(), -- Se a medida for CaOH2, CaCO3 ou Tenebrios SELECTEDVALUE('dim_measures'[measure]) IN { "CaOH2", "CaCO3", "Tenebrios"}, SUM('fact_daily_system'[value]), -- Caso contrário (medidas restantes) AVERAGE('fact_daily_system'[value]))
fact_daily_system	[Measures].[axis_title]	SELECTEDVALUE('dim_measures'[measure], "Parâmetro")

fact_daily_system	[Measures].[daily_max_value_dynamic]	SWITCH(
		TRUE(),
		SELECTEDVALUE('dim_measures'[measure]) IN { "CaOH2", "CaCO3", "Tenebrios" }, 5, //tenho que ajust...
		SELECTEDVALUE('dim_measures'[measure]) = "FishFeed", 200,
		AVERAGE('fact_daily_system'[max_value_fds]) // Valor padrão
		)

Figure 7.3: *DAX Studio Query*

of interactive reports and dashboards, thereby delivering consistent, reusable metrics that streamline report construction and enhance end-user self-service analysis.

7.3 Report and Dashboard Design

The Power BI report presents an overview of LSMI at IPLeiria. It was developed using Microsoft Power BI Desktop and it was published in the workspace of Microsoft Fabric named PowerBI LSMI, where it can be accessed and shared with relevant stakeholders. The report is organized into several pages, the first ones are general (Open, Main), and the remaining ones are dedicated to a system project (Arugula and Lamb's Lettuce,Mealworms and Papaya Trees).

Open

The Open page serves as an introduction to the laboratory, outlining its geographical context and structural organization. It provides the location, coordinates, altitude, and climate classification, offering essential environmental data. The page also features a card that functions as a button, red square, Figure 7.4, by clicking on it, users are redirected to another page, Figure 7.5, containing detailed information about key system variables such as temperature, seasonal characteristics, and the dimensions of the aquaponics system. Navigation icons at the top of the page represent the different modules and allow users to access detailed insights into each biological element managed within the integrated system.

Main

The Main page of the Power BI report offers an overview of the different five integrated aquaponic systems and allows users to interactively navigate through all available system lines and fishtanks, selecting any desired time period between 2019 and 2025. This central dashboard aggregates environmental and operational data in a visual and dy-

Figure 7.4: *Power BI - Open - Page 1*

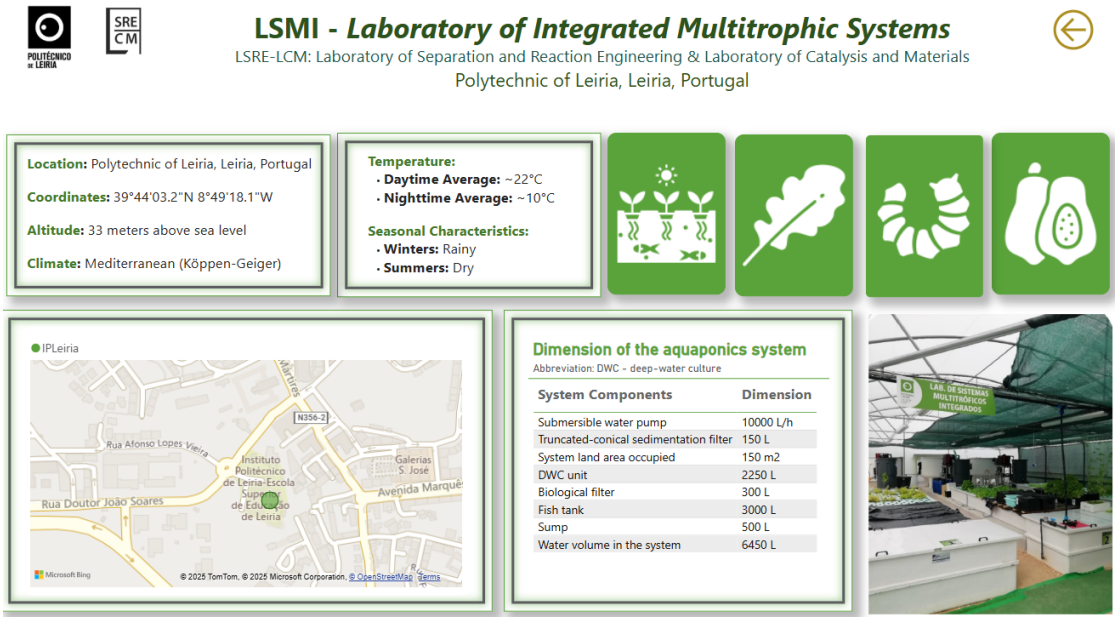


Figure 7.5: *Power BI - Open - Page 1a*

namic format, Figure 7.6.

Below the indicators, the page features visualizations of accumulated water balance per year and fish events, as well as a detailed time series chart showing various parameters of the daily system such as DO, fish feed quantity, ORP, pH, TDS, water temperature, and electrical conductivity. These charts allow an in-depth analysis of trends and behavior of each system over time.

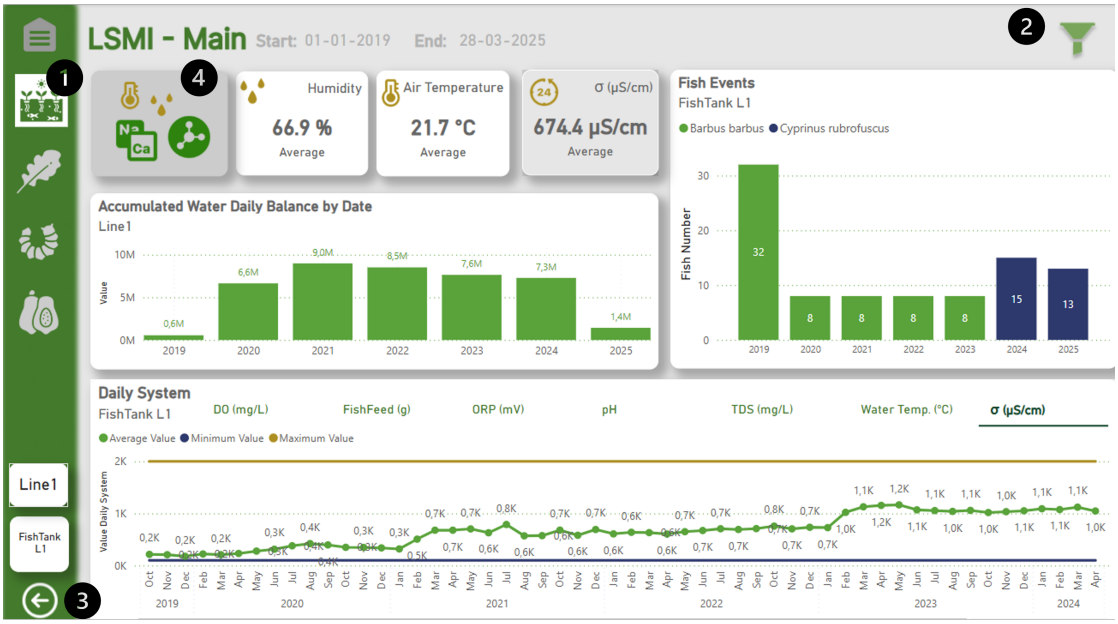


Figure 7.6: *Power BI - Main - Page 2*

The following outlines key interactive components of the dashboard:

1. **Navigation Menu:** Users have buttons available in the sidebar. The first button (home icon) redirects to the *Open* page. The second, third, and fourth buttons link to the respective project sections: Arugula & Lamb's Lettuce, Mealworms, and Papaya Trees.
2. **Filter Symbol:** A dynamic date slicer allows the user to define a custom time range for which data will be displayed throughout the dashboard. It also enables the selection of a specific line and the corresponding Fishtank or DWC system. An important feature of the report is the expandable filter panel, Figure 7.7, which can be opened to refine the data being visualized and subsequently closed by clicking on the "X" icon. This interactive behavior was implemented using Power BI's *Bookmarks* and the *Show/Hide* functionality for objects, combined with button actions. Through this method, different report elements are revealed or hidden based on user interaction, allowing for a more dynamic and user-friendly experience without cluttering the visual space Figure 7.7.
3. **Back:** The button with an image of a left-pointing arrow redirects the user to the previously viewed page.

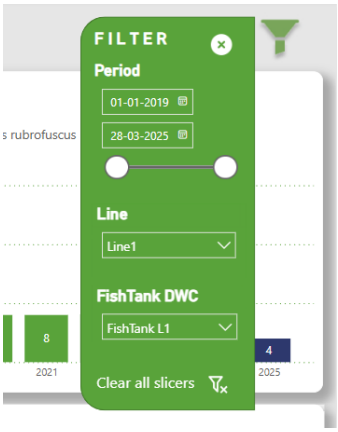


Figure 7.7: *Power BI - Main - Page 2 - Slicer*

4. **Key Environmental Indicators:** The card function as buttons that redirect the user to the second page of the Main section, provides a detailed analysis of environmental and water quality parameters. It focuses on Line2 and FishTank L2, displaying average values of humidity, temperature of the air, and NO_3^- (nitrate) concentration. Interactive filters allow users to switch between different types of nutrients (e.g., NO_2^-, NH_4^+, PO_4^{3-}) and metals. Visualizations include a table and line chart showing the yearly evolution of nitrate levels and a comparative graph of humidity and temperature of the air over time. This page enables a deeper understanding of nutrient dynamics and environmental conditions within the aquaponic system, Figure 7.8.

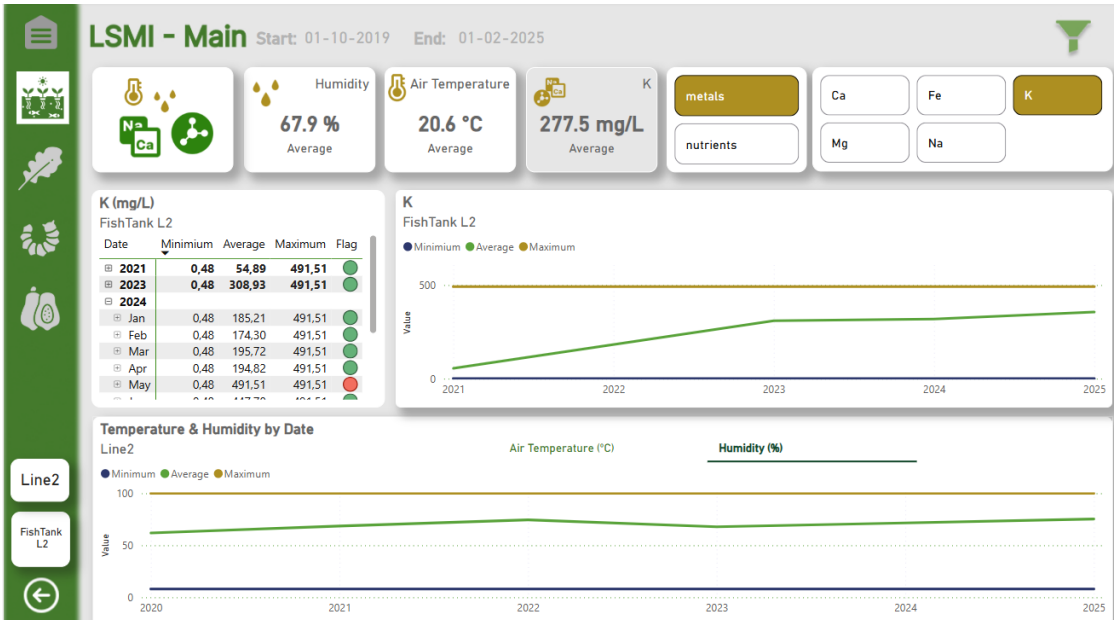


Figure 7.8: *Power BI - Main - Page 2a*

A brief overview of each project (Table 7.1) and the functionality of its Power BI pages is presented below.

Project Number	Project Name	System (Line)	Start Date	End Date
Project1	Arugula & Lamb's Lettuce	Line3	2019-10-23	2019-12-11
Project2	Mealworms	Line1 + Line2 + Line3	2021-04-27	2021-06-08
Project3	Papaya Trees	Line2 + Line3	2022-05-02	2023-05-01

Table 7.1: *Summary of the projects carried out, including associated systems and execution periods*

First, the Arugula & Lamb's Lettuce project evaluated a catfish-based aquaponics system's capacity to grow two cultivars each of lamb's lettuce (*Valerianella locusta* var. Favor and var. de Hollande) and arugula (*Eruca vesicaria* var. sativa and *E. sativa*) under varying light intensities. During the growth period, plants were monitored for root length, height, leaf count, foliage diameter, and largest-leaf length, then assessed for biomass, greenness, and overall health. Lamb's lettuce—particularly var. de Hollande, maintained high quality across light treatments, whereas arugula showed reduced greenness and vigor under full sunlight. Increased light intensity led to higher antioxidant and phenolic contents, which correlated with improved viability in a Caco-2 cell model. This study underscores the need for species, and cultivar-specific acclimation conditions for successful commercial aquaponics.

Next, the Mealworms trial involved feeding African catfish (*Clarias gariepinus*) a diet with 30 % yellow mealworm (*Tenebrio molitor*) substitution to cultivate parsley, arugula, and pennyroyal over six weeks. Water quality and environmental conditions matched those of a standard fish-meal system, but the mealworm diet yielded reduced plant biomass—lower height, foliage spread, leaf count, and root length—as well as diminished leaf greenness and overall health. *Tenebrio molitor* proved high in protein and fiber yet comparatively deficient in key minerals versus conventional fish meal.

Finally, the Papaya Trees project spanned 13 months and examined papaya (*Carica papaya*) growth and fruiting viability in Line 2 and Line 3 aquaponic systems with African catfish (*Clarias gariepinus*). Morphological and developmental parameters were monitored and compared under two substrates—brick waste and Leca—to evaluate their effects on tree performance and fruit production.

Arugula and Lamb's Lettuce

The page Arugula and Lamb's Lettuce presents monitoring data for the Arugula & Lamb's Lettuce project conducted between October 27 and December 11, 2019, using the specific system designed Line3. It provides an overview of water usage, fish count, and key water quality parameters relevant to the aquaponic system, Figure 7.9.

Additional visuals include:

- A bar chart showing the Accumulated Water Daily Balance over time.
- A fish count bar indicating the number of *Clarias gariepinus* present.
- A summary table of water quality parameters (DO, pH, TDS, temperature, and

conductivity) with minimum, maximum, and average $\pm$ standard deviation.

- A detailed time series chart at the bottom displaying daily variations of system parameters such as DO, fish feed, ORP, pH, TDS, water temperature, and conductivity.

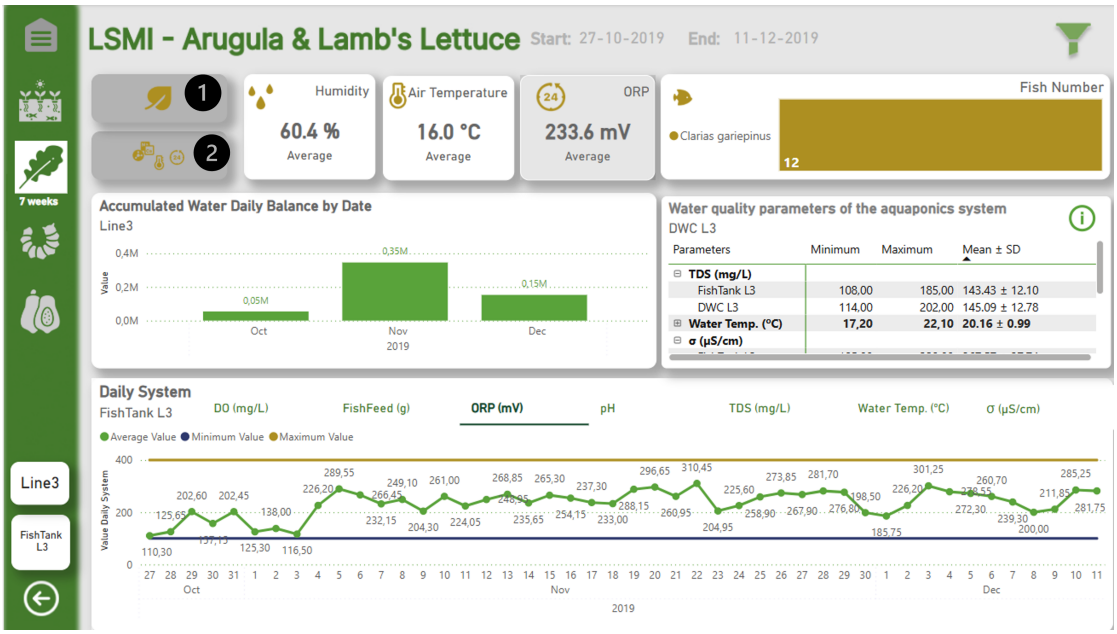


Figure 7.9: *Power BI - Arugula and Lamb's Lettuce - Page 3*

1. **Leaf Icon Button (Figure 7.9):** Directs the user to a dedicated page containing detailed information on the morphological dimensions of the plants, Figure 7.10, specifically *Arugula* and *Lamb's Lettuce*. This page presents various morphological parameters such as *biggest leaf length*, *foliage diameter*, *leaf number*, *plant height*, and *root length*, with mean and standard deviation values plotted over time and across light intensity conditions (shade and sun).

   Within this dimension page, there is a second button labeled **Health**, which redirects the user to another page focusing on the greenness and health status of the plant specimens, Figure 7.11. This section provides tabular and graphical data showing the distribution of plants by greenness level (high/low) and health status (strong/weak), with classification by variety, light exposure, and plant type.

   This hierarchical navigation allows for a structured exploration of plant growth and health, contributing to a deeper understanding of how different environmental factors influence biomass and quality in aquaponic systems.
2. **Parameters Icon Button (Figure 7.9):** This button redirects the user to a detailed monitoring page dedicated to environmental and nutrient parameters within the Arugula & Lamb's Lettuce project, Figure 7.12.
   The page provides information on the concentration of ammonium (NH_4^+) in **FishTank L3**, presenting minimum, average, and maximum values over the project

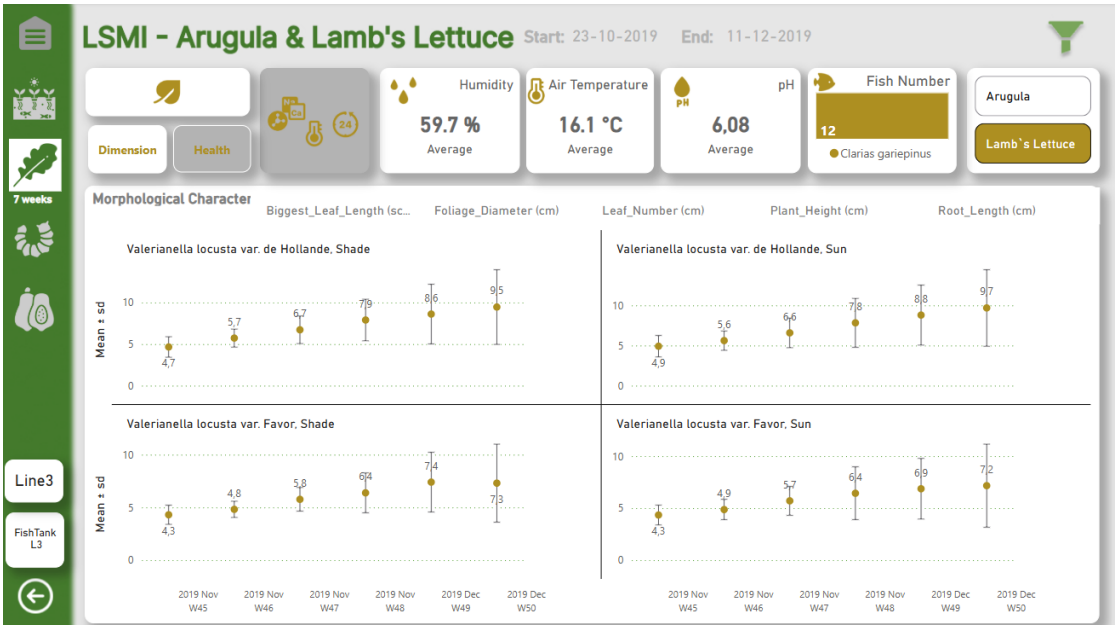


Figure 7.10: *Power BI - Arugula and Lamb's Lettuce - Page 3b*

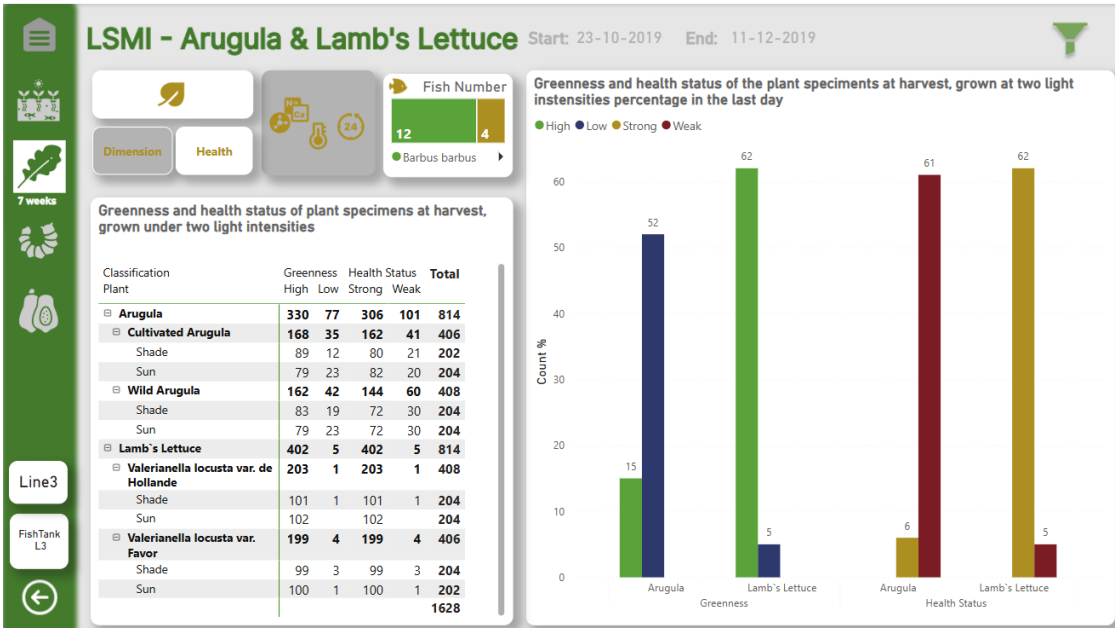


Figure 7.11: *Power BI - Arugula and Lamb's Lettuce - Page 3c*

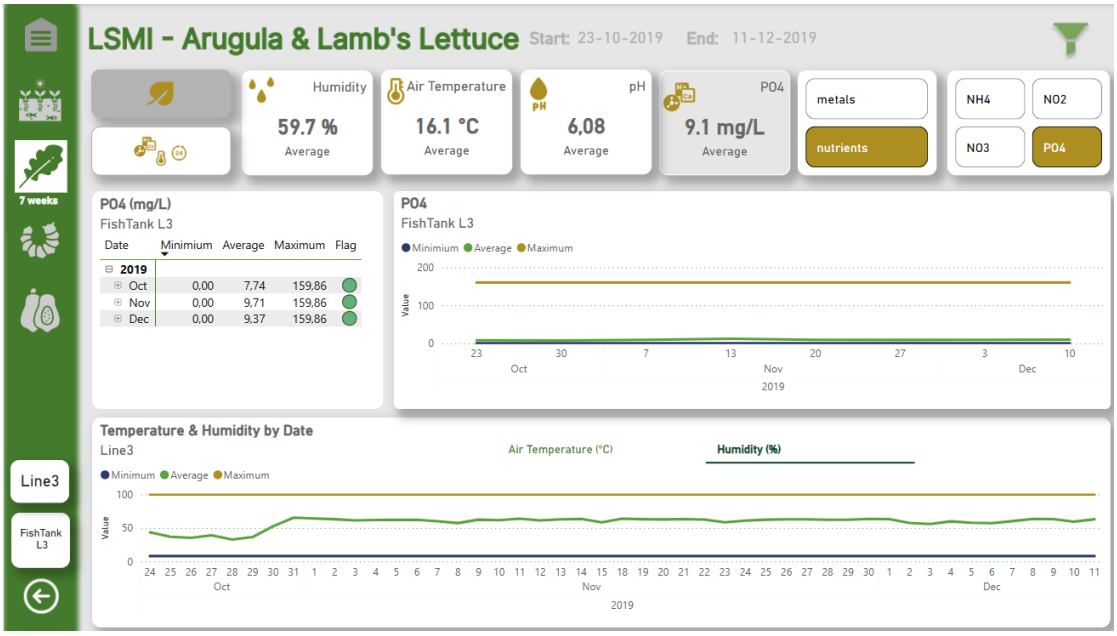


Figure 7.12: *Power BI - Arugula and Lamb's Lettuce - Page 3a*

period. Data are visualized in both tabular and line chart formats to facilitate temporal trend analysis. Additionally, the page includes a chart showing the evolution of **temperature** and **humidity** by date for **Line3**, which helps assess environmental stability and its influence on nutrient behavior. Interactive buttons allow users to switch between other nutrients (e.g., NO_2^-, NH_4^+, PO_4^{3-}) and metals for further analysis.

Mealworms

The following pages relate to the Mealworms project, conducted from April 27 to June 8, 2021, across three different systems: Line1, Line2, and Line3. This section provides an overview of key environmental indicators, including an average humidity of 67.3% and an average temperature of 22.1 °C, as well as the presence of 7 *Clarias gariepinus* specimens in this period in Line3, Figure 7.13.

The page includes a bar chart showing the **Accumulated Water Daily Balance by Date**, reflecting water usage trends across the different months. Additionally, a detailed table summarizes the main **Water Quality Parameters** such as DO, pH, TDS, water temperature, and conductivity, with minimum, maximum, and mean ± standard deviation values for both the DWC and FishTank components.

At the bottom of the page, a line chart tracks daily values of multiple operational parameters (DO, fish feed, ORP, pH, TDS, temperature, and conductivity), enabling a deeper analysis of system dynamics.

The button layout and navigation logic applied on this page follow the same interactive

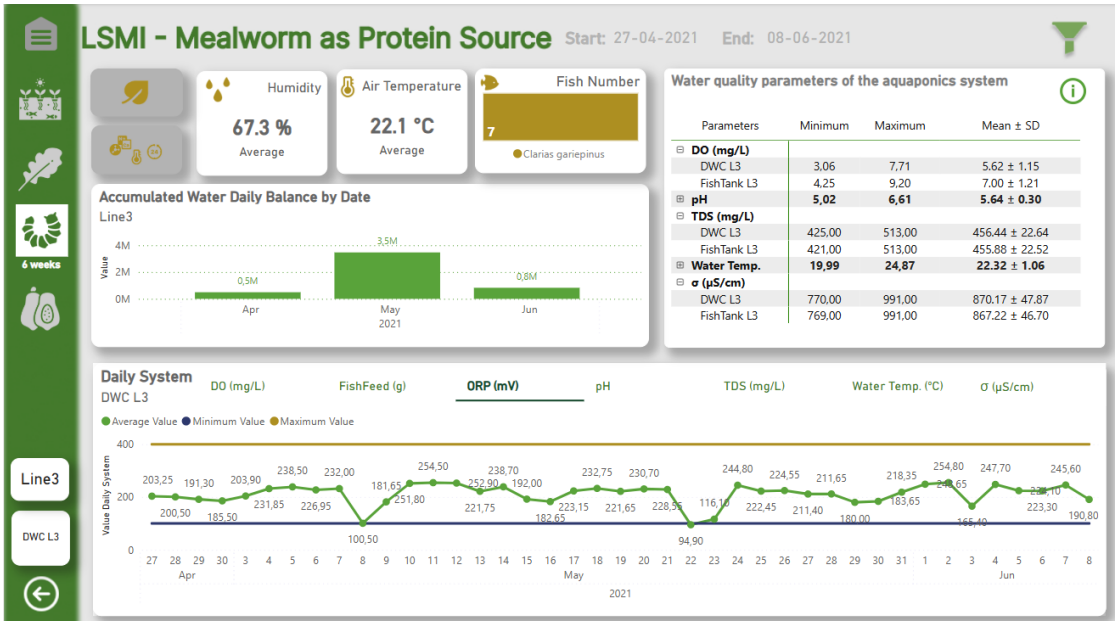


Figure 7.13: *Power BI - Mealworms - Page 4*

structure as in the **Arugula & Lamb's Lettuce** project, employing *bookmarks*, object visibility toggles, and button actions to ensure a seamless and intuitive user experience.

It is also important to note that when hovering the mouse over an icon that functions as a button, a label appears indicating the destination page that will be opened if the button is clicked. This tooltip name was defined using the *Format Image* pane, as illustrated in the image below, Figure 7.14.

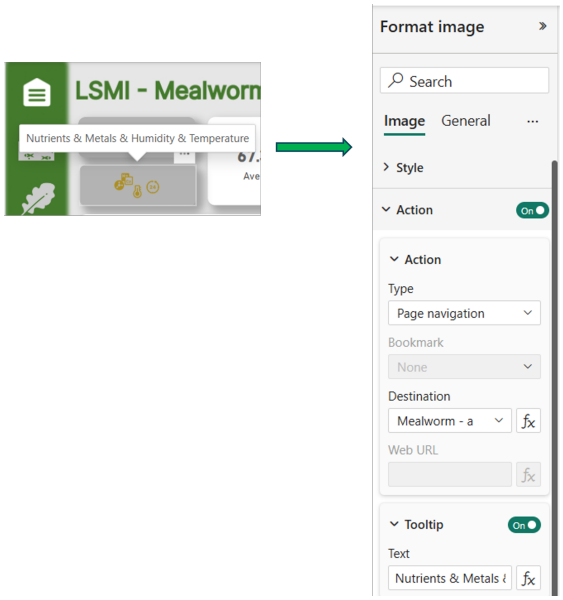


Figure 7.14: *Power BI - Buttom - Mealworms - Nutrients & Metals & Humidity & Temperature*

Clicking on the **Mealworms - Nutrients & Metals & Humidity & Temperature** button redirects the user to the page shown above, Figure 7.15.

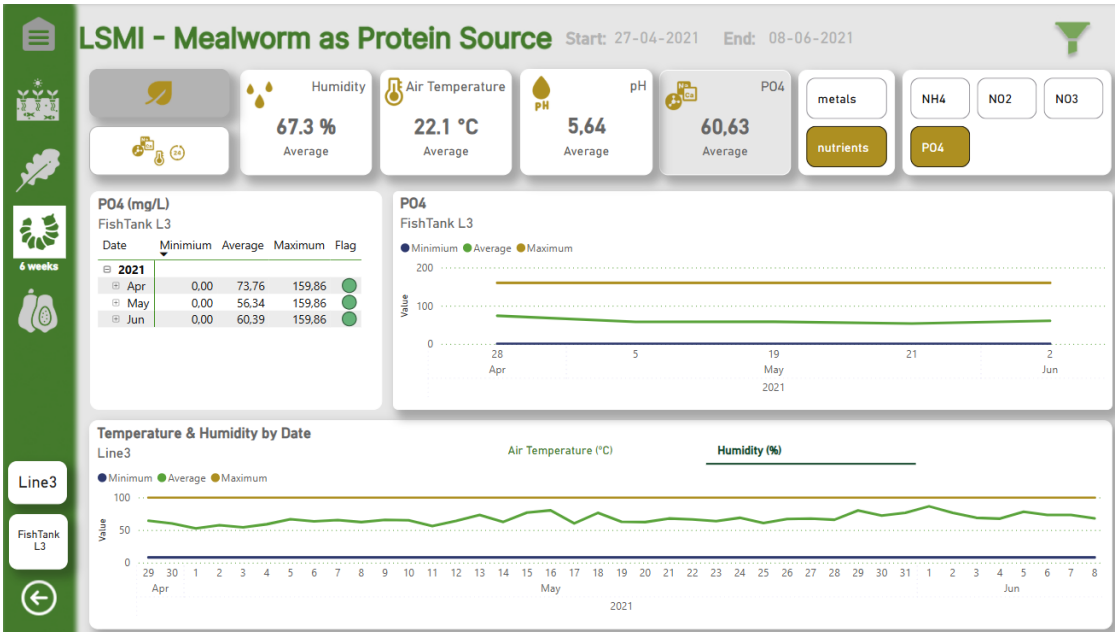


Figure 7.15: *Power BI - Mealworms - Page 4a*

Which provides detailed monitoring data for the concentration of phosphate (PO_4^-) in **FishTank L1**, as part of the Mealworms project. The table presents minimum, average, and maximum values by year, while the adjacent line chart visualizes the evolution of (PO_4^-) levels over time. In addition to nutrient monitoring, the page includes a line chart that displays the variation of temperature and humidity in **Line1**, helping assess environmental conditions that may influence nutrient behavior. The interface maintains the same interactive design logic as in previous modules, allowing users to switch between nutrient types using dynamic buttons and to navigate seamlessly between related data views.

On the page **Figure 7.13 – Power BI - Mealworms - Page 4**, clicking the *Mealworms - Dimension* button (Figure 7.16) redirects the user to the page shown in Figure 7.17 – Power BI - Mealworms - Page 4b. Which displays the morphological development of plants (Parsley) associated with the mealworms project.

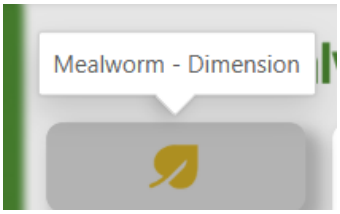


Figure 7.16: *Power BI - Button - Dimension*

This page presents key morphological parameters, including *biggest leaf length*, *foliage diameter*, *leaf number*, *plant height*, and *root length* (*cm*). These indicators are visualized using mean values accompanied by standard deviation, plotted over time and across

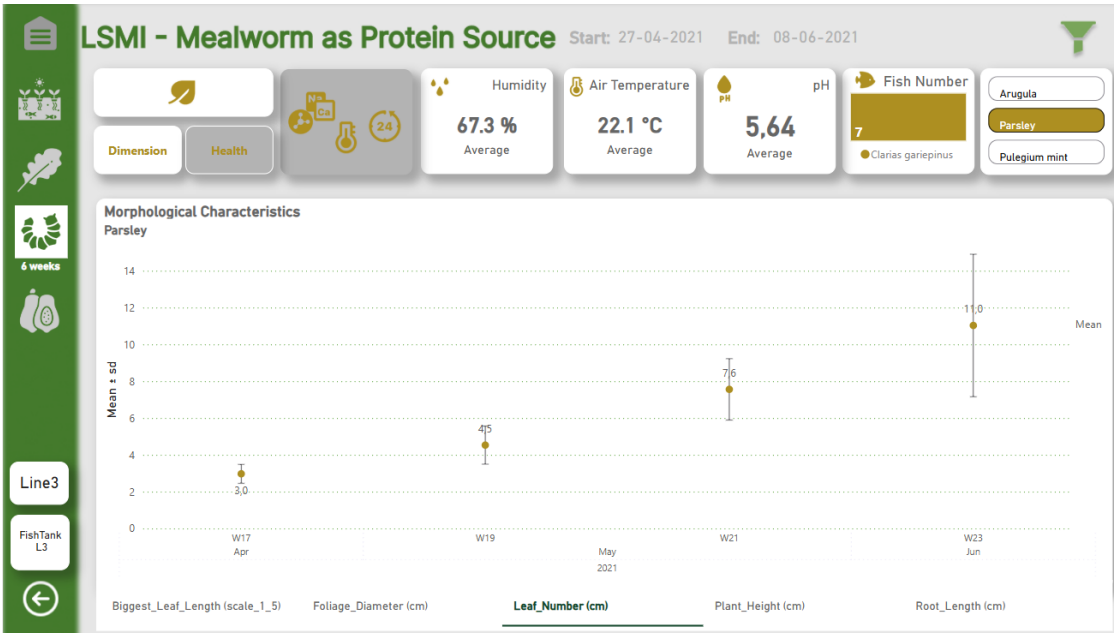


Figure 7.17: *Power BI - Mealworms - Page 4b*

weeks of the experimental period. The chart helps evaluate the growth patterns of plant specimens under the influence of aquaponic conditions specific to **Line1** and the **DWC L1** tank, where 8 specimens of *Barbus barbus* were present.

The page **Figure 7.17: Power BI - Mealworms - Page 4b** also contains a **Health** button, which redirects the user to a subsequent page that displays information related to the health status of parsley plants, Figure 7.18.

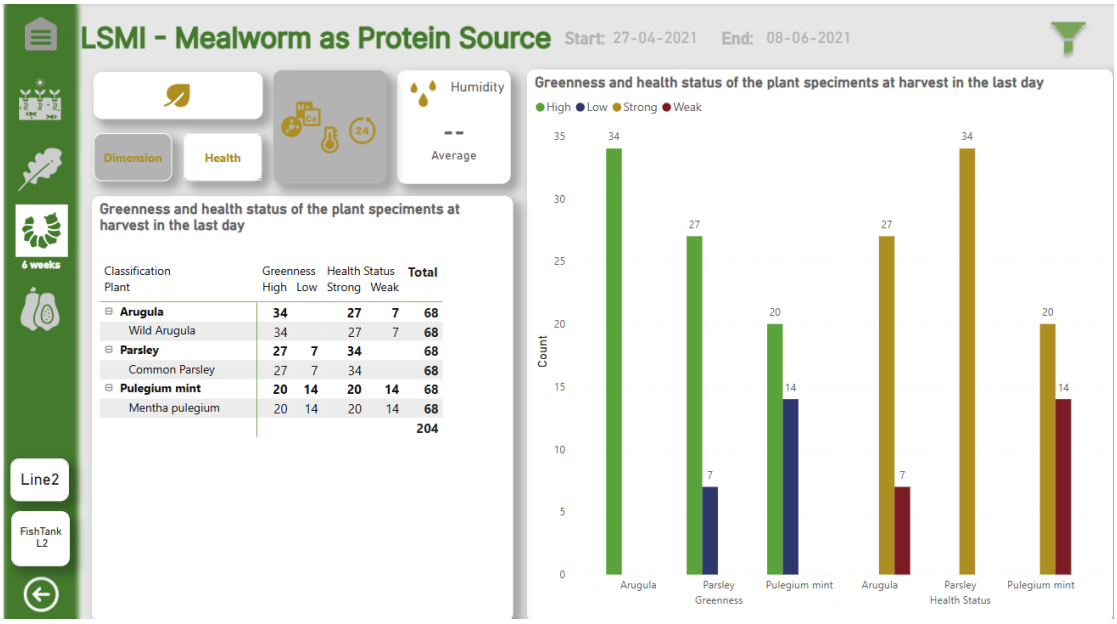


Figure 7.18: *Power BI - Mealworms - Page 4c*

This includes data on greenness levels (high or low) and the strength of the specimens

(strong), recorded under different light intensities. The information is presented in both tabular and graphical formats, allowing for easy comparison of plant performance based on environmental conditions.

Papaya Trees

The pages associated with the Papaya Trees project, cover the experimental period from May 2, 2022, to May 1, 2023 and displays only the system lines relevant to the papaya tree project (Line2 and Line3), providing a focused and context-specific view of the data.

The image below, Figure 7.19, presents the first monitoring dashboard for this project, which follows a structure similar to the first page of the previous projects. It includes key environmental indicators, the accumulated daily water balance over time.

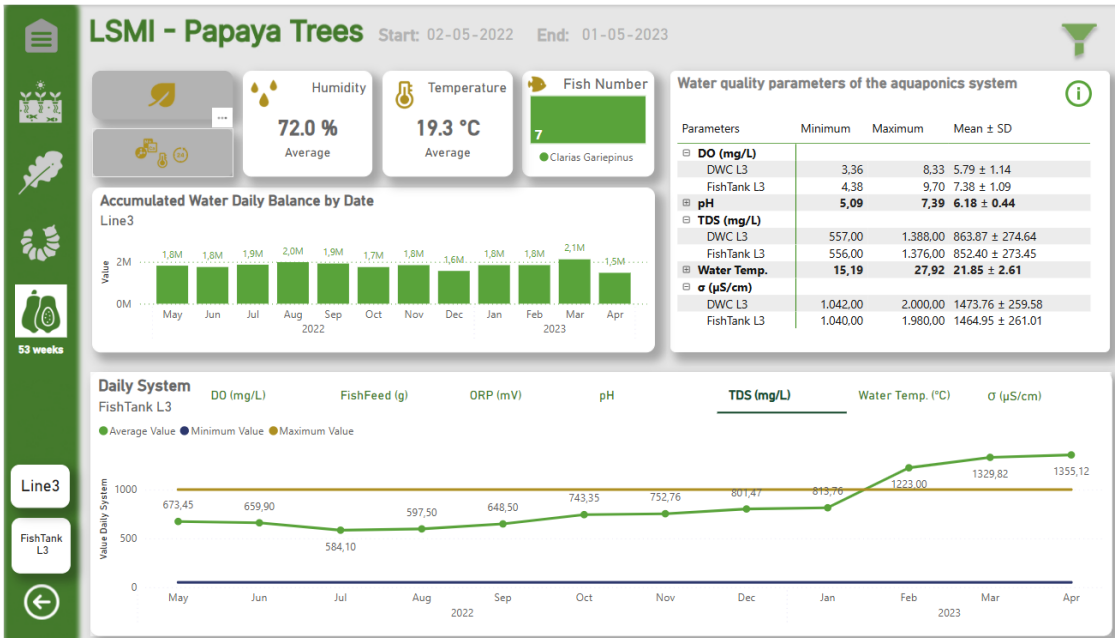


Figure 7.19: *Power BI - Papaya Trees - Page 5*

As previously mentioned, the pages related to this project follow a structure and logic similar to those of the earlier projects. Therefore, to avoid redundancy, these pages will not be described in detail here. The full Power BI report is available for consultation on: Download Power BI Desktop File.

It is worth noting that the chart titled *Water quality parameters of the aquaponics system* includes an information icon (i) that functions as a button. When clicked, it redirects the user to a dedicated information page, Figure 7.20. From that page, users can easily return to their previous location by clicking the back button located in the bottom-left corner.

It is worth highlighting that in **Figure 7.21 - Power BI - Papaya Trees - Page 5c** (Health),

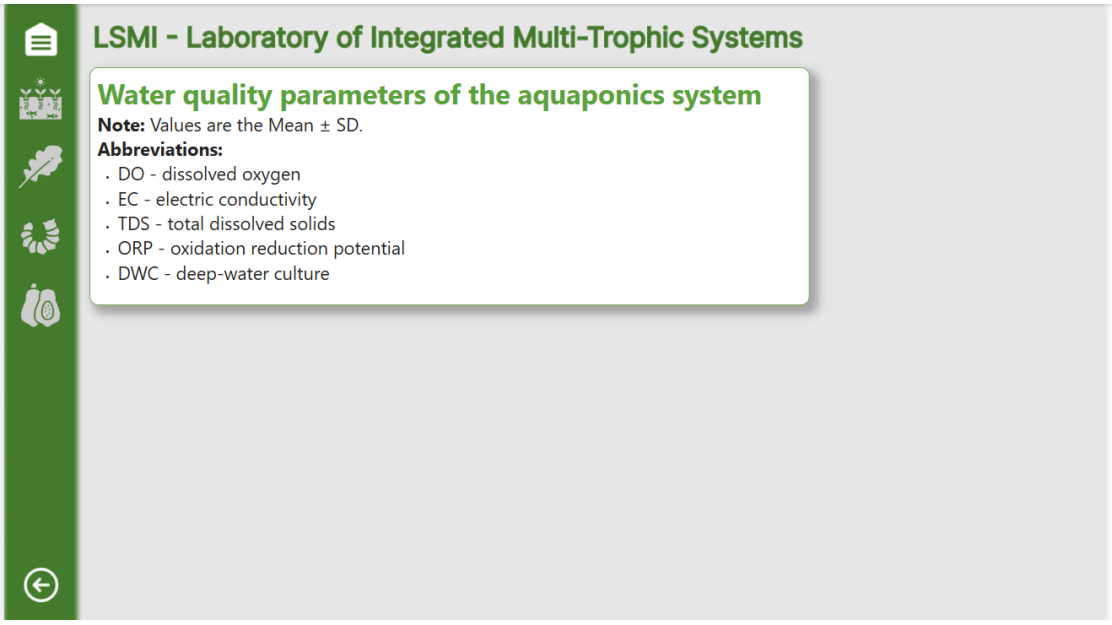
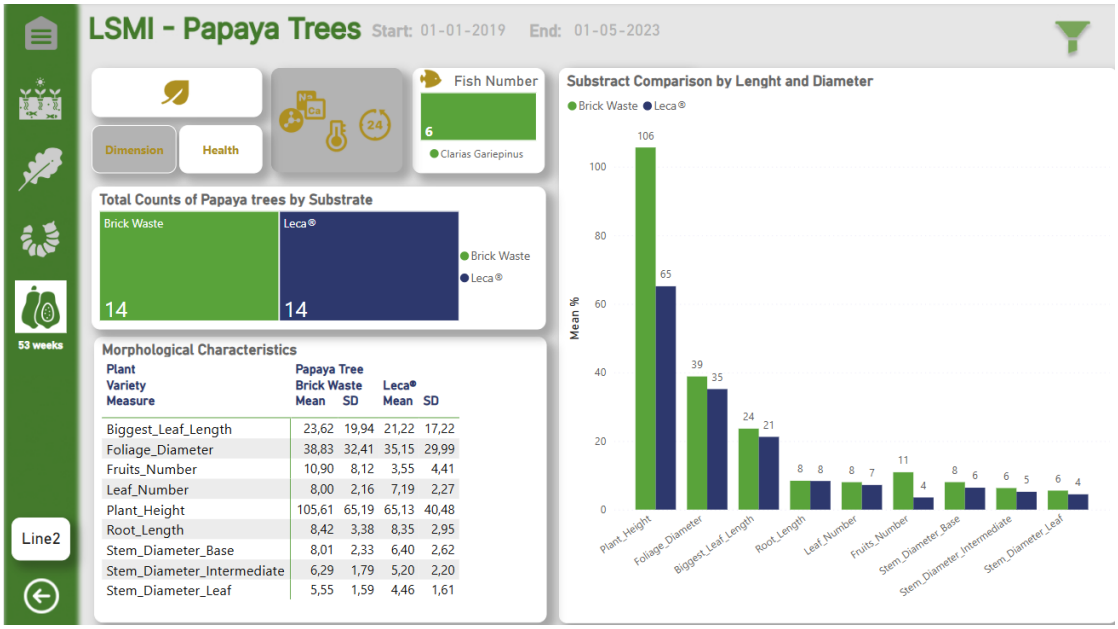


Figure 7.20: *Power BI - Information*

a different approach was used, the page presents morphological data and comparative analysis of papaya trees grown in two different substrates: **Brick Waste** and **Leca®**.



Plant Variety Measure	Papaya Tree Brick Waste Mean	Brick Waste SD	Leca® Mean	Leca® SD
Biggest_Leaf_Length	23,62	19,94	21,22	17,22
Foliage_Diameter	38,83	32,41	35,15	29,99
Fruits_Number	10,90	8,12	3,55	4,41
Leaf_Number	8,00	2,16	7,19	2,27
Plant_Height	105,61	65,19	65,13	40,48
Root_Length	8,42	3,38	8,35	2,95
Stem_Diameter_Base	8,01	2,33	6,40	2,62
Stem_Diameter_Intermediate	6,29	1,79	5,20	2,20
Stem_Diameter_Leaf	5,55	1,59	4,46	1,61

Figure 7.21: *Power BI - Papaya Trees - Page 5c*

It includes a total count of plants per substrate, detailed statistical values (mean and standard deviation) for various growth parameters, and a bar chart illustrating the percentage distribution of each morphological metric by substrate. This visualization provides insight into how each substrate influenced the development of specific plant characteristics such as plant height, leaf number, root length, and stem diameters.

An additional page was created to address specific requirements unique to this project (Health2). To navigate to this page, click the button shown in Figure 7.22 on the page **Figure 7.21 - Power BI - Papaya Trees - Page 5c**.

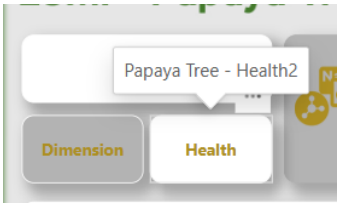


Figure 7.22: *Power BI - Papaya Trees - Button - Health2*

The page Health2 - Figure 7.23 - focuses on the characterization of papaya fruits based on attributes such as surface texture (rough or smooth), presence of seeds, and taste quality. It includes comparative bar charts of fruit diameter and length, as well as average weight, segmented by fruit type and sensory evaluation. Additionally, summary cards display the total number of fruits classified by each trait, and a detailed table presents mean measurements for individual papaya trees. This layout enables a evaluation of fruit quality indicators within the scope of the Papaya Tree project.

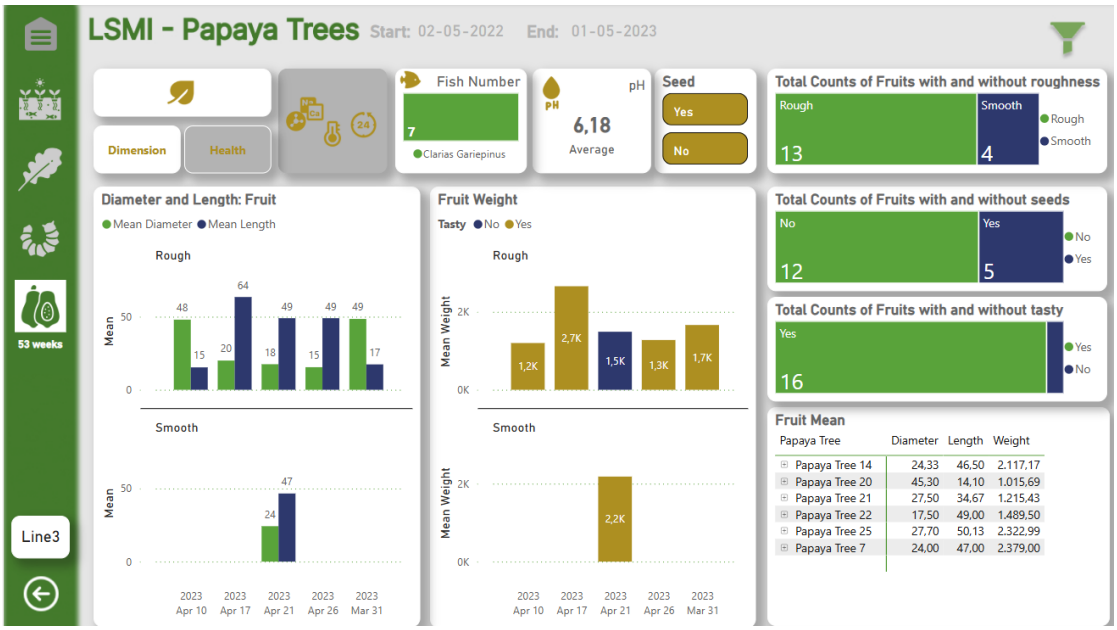


Figure 7.23: *Power BI - Papaya Trees - Page 5d*

8

Usability Evaluation of the Visual Aquaponics Dashboard

This chapter presents the design and results of a usability survey for the aquaponics dashboard. Measuring ease of use, learnability, and satisfaction highlights strengths and reveals areas for improvement, guiding future updates.

8.1 Survey Goals and Target Audience

Accordingly, the primary objective of this survey is to assess how effectively the Power BI dashboard supports monitoring and decision-making within the aquaponic research environment. Specifically, the questionnaire is designed to:

- Evaluate perceived ease of use, learnability, and the need for technical assistance in operating the dashboard (Design & Usability).
- Measure user satisfaction with the clarity and organization of presented information (Content).
- Assess the intuitiveness and efficiency of navigating between different reports, pages, and data views (Navigation).
- Gauge the dashboard's loading times and responsiveness during typical user interactions (Performance).

The survey targets three stakeholder groups directly involved in the aquaponics study students, researchers, and professors to ensure a comprehensive assessment of the dashboard's usability across both operational and supervisory roles. The survey was sent to a total of 5 participants (2 students, 2 researchers and 1 professor).

8.2 Survey Methodology

In the article [89] traces the origins of the System Usability Scale (SUS), created over 25 years ago at Digital Equipment Co. Ltd. SUS was intended as a quick, reliable measure of usability across diverse systems. Its simple yet robust design has enabled widespread adoption in usability testing and has become a standard in subjective usability assessment [89].

The questionnaire combines the 10 core SUS items—alternating positive and negative wording—with four dashboard-specific questions covering Design & Usability, Content, Navigation, and Performance. Table 8.1 is adapted from the original SUS. All items employ a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). The full questionnaire is provided in **Appendix G: Survey**.

Question	Category
1. I find Power BI useful for understanding the information presented.	Content
2. Navigating through Power BI reports feels confusing or overwhelming.	Navigation
3. It is easy for me to find the information I need in Power BI dashboards.	Navigation
4. I would need help from someone with technical skills to use Power BI effectively.	Design & Usability
5. The layout of the dashboard is intuitive.	Content
6. I often feel lost or unsure about how to interact with Power BI reports.	Navigation
7. The dashboard loads and responds quickly to my interactions.	Performance
8. It is not easy to switch between different pages or views in the dashboard.	Navigation
9. The visual design of the dashboard is clear.	Design & Usability
10. The dashboard is slow or unresponsive during my interactions.	Performance

Table 8.1: *Mapping of survey questions to their respective categories*

Table 8.2 shows how often each theme was raised by participants. Navigation topped the list with four mentions, while Content, Performance and Design & Usability were each noted twice.

Category	Number of questions
Content	2
Navigation	4
Design & Usability	2
Performance	2

Table 8.2: *Frequency of mentions by dashboard category*

The survey was deployed online and distributed by email to all prospective respondents. Participants had 5 days to complete the questionnaire. Participation was volun-

tary and anonymous, required approximately 5–10 minutes, and collected no personal identifiers.

Raw responses were screened for completeness; any submission missing more than two SUS items was discarded. Valid responses were exported as a xlsx file and imported into Microsoft Fabric for statistical analysis. SUS scores were calculated following Brooke's guidelines [89]:

Instructions for Calculating:

1. For odd-numbered items (1, 3, 5, 7, 9): subtract 1 from the score.
2. For even-numbered items (2, 4, 6, 8, 10): subtract the score from 5.
3. Add all the adjusted scores together.
4. Multiply the result by 2.5.

The SUS scores were interpreted according to the following usability bands.

Interpretation of Results [89]:

- **90–100:** Excellent usability
- **80–89:** Good usability
- **70–79:** Acceptable
- **60–69:** Marginal
- **Below 60:** Problematic

8.3 Results Analysis: SUS Score – Calculation and Interpretation

In order to characterize our study sample and summarize the usability outcomes, Table 8.3 first presents the distribution of survey invitations and respondent profiles. It shows that invitations were sent to 2 students 2 researchers and 1 professor, yielding five valid replies: three participants with prior Power BI experience and two without.

Question	Answer	Number of Questions
Did you have previous experience as a Power BI user?	Yes	3
Did you have previous experience as a Power BI user?	No	2
What is your occupation?	Student	2
What is your occupation?	Researcher	2
What is your occupation?	Professor	1

Table 8.3: *Distribution of responses for prior Power BI experience and occupation*

Table 8.4 reports the descriptive statistics for the 10-item SUS scores collected on June 29, 2025. Across all five respondents, the SUS total ranged from a minimum of 50 to a maximum of 100, with an average score of 74.5 (SD = 1.581). According to established benchmarks, this places the dashboard's perceived usability in the **Acceptable** category. Together, these tables provide both a clear account of who participated and how they rated the system's usability.

Date	N	Minimum	Maximum	Average	SD	Classification
6/29/2025	5	50	100	74.5	1.581	Acceptable

Table 8.4: *Summary statistics of SUS on 29/06/2025 (N - Sample size)*

Additionally, a complementary analysis was performed beyond the scope of the original SUS methodology in order to identify the best - and worst - rated usability categories. The process first segregates questions by parity (even-numbered *vs*. odd-numbered) and then tabulates how respondents distributed their answers across a five-point Likert scale for each usability theme. In the resulting table, each row represents a specific parity group and theme (Design & Usability, Navigation, Performance, or Content), while the five response columns: *Strongly agree*, *Agree*, *Neutral*, *Disagree* and *Strongly disagree*, display the absolute counts of participants selecting each option. This layout makes it straightforward to compare positively phrased items (odd-numbered) against negatively phrased items (even-numbered) within each theme, thereby highlighting both areas of strength and points of friction in the dashboard's usability, Table 8.5.

Type Questions	Category	S. agree	Agree	Neutral	Disagree	S. disagree
Even	D&U	0	0	2	2	1
Even	Navigation	0	2	3	5	5
Even	Performance	0	0	2	2	1
Odd	Content	4	5	1	0	0
Odd	D&U	4	0	0	1	0
Odd	Navigation	1	3	1	0	0
Odd	Performance	1	2	1	1	0

Table 8.5: *Response distribution by question parity and survey theme; S. - Strongly; D&U - Design & Usability*

8.4 Discussion and Conclusion

Overall, the SUS average score of 74.5 (SD = 1.581) places the dashboard firmly within the "Acceptable" usability category (70–79). While this indicates that the interface meets baseline expectations for functionality and learnability, it also highlights room

for improvement. Specifically, transitioning from *Acceptable* to *Good* usability (80–89).

It is important to pinpoint the dashboard's areas for improvement. Based on Table 8.5, The analysis treats *Agree* and *Strongly agree* on odd-numbered (positively phrased) questions and *Disagree* and *Strongly disagree* on even-numbered (negatively phrased) questions as indicators of **good usability**:

- **Content:** Odd questions received 4 *Strongly agree* and 5 *Agree* responses (9 out of 10), demonstrating very strong satisfaction with the dashboard's informational clarity and usefulness.
- **Navigation:** Odd questions garnered 1 *Strongly agree* and 3 *Agree* (4 positives), while even questions saw 5 *Disagree* and 5 *Strongly disagree* (10 positives). A total of 14 positive signals indicates users generally find navigation straightforward.
- **Design & Usability:** Odd items achieved 4 *Strongly agree* (4 positives), and even items had 2 *Disagree* and 1 *Strongly disagree* (3 positives), for a combined 7. This suggests moderate approval of the visual layout, with room to refine aesthetics and intuitiveness.
- **Performance:** Odd questions yielded 1 *Strongly agree* and 2 *Agree* (3 positives), and even questions produced 2 *Disagree* and 1 *Strongly disagree* (3 positives), totaling 6. While responsiveness is acceptable, it represents the weakest dimension relative to the others.

In summary, Content and Navigation are the strongest usability dimensions, whereas Design & Usability and especially Performance — with fewer positive responses — should be prioritized for improvement. The SUS results (mean = 74.5, SD = 1.581, Table 8.4) place the dashboard in the Acceptable usability range. While users praised its content clarity and basic performance, the detailed parity analysis, reveals that navigation and design aesthetics still can be improved. It is also worth noting that two respondents had no prior Power BI experience, which may have negatively impacted their ratings, and that all evaluations were conducted in Power BI Desktop, an environment less intuitive than the published, browser based workspace. Moreover, this study's small sample (N = 5), composed primarily of technical users, limits broader generalization.

9 Conclusion

This thesis aimed to design and implement a BI solution to support data analysis and monitoring of an integrated aquaponics system at IPLeiria. The project covered the entire data pipeline, from data collection and integration, through dimensional modelling and transformation processes, to the development of a comprehensive Power BI reporting layer.

The implementation of a lakehouse, based architecture, combined with structured ETL processes, enabled the consolidation of heterogeneous data sources into a unified analytical model. The dimensional data model provided a robust semantic layer that supports scalable and user-friendly reporting.

Power BI was used as the front-end platform, where dashboards and interactive reports were developed to provide researchers and stakeholders access to critical operational and biological metrics. The interface supports filtering, interactivity through bookmarks and tooltips, and page-to-page navigation tailored to each experimental project.

The solution addresses the initial challenges identified in the aquaponics system, namely the need for centralized data access, reliability of information, and flexible analytics tailored to multiple research projects. The use of Microsoft Fabric technology, built on a Lakehouse architecture, enabled secure integration and scalability, while ensuring a low learning curve for users.

9.1 Study Limitations

As part of the conclusions, it is essential to acknowledge the key limitations of this project. The full Microsoft Fabric Power BI capabilities could not be leveraged due to trial-version restrictions, which constrained features such as integrated Power BI.

The System Usability Scale evaluation places the dashboard in the Acceptable usabil-

ity range, with clear strengths in Content and Navigation but identifiable points for improvement in Design and Performance. Familiarity issues with Power BI and use of Desktop instead of the browser workspace may have lowered scores in this small technical sample.

Another point to keep in mind is that, as mentioned in the thesis, the Excel files were standardized; however, they could be further optimized to eliminate the need for transformations in Dataflow Gen2. These optimizations were deferred because extensive changes might overwhelm those maintaining the files, so any modifications should be introduced gradually to allow users time to adapt. This change would not only adhere to best practices for separation of concerns but also improve both performance and long-term maintainability.

9.2 Future Work

An immediate enhancement to the current process would involve restructuring the Excel source files so that all necessary data harmonization and cleansing occur downstream in the ETL transformation phase — eliminating the need for ad-hoc adjustments during Dataflow Gen2 ingestion.

Additionally, the Power BI Desktop reports should be further optimized, streamlining visuals and improving load times, to enhance both design and performance. Future studies should engage a broader, more diverse user base and evaluate the dashboard in its published online form to validate and extend these findings.

Beyond this, real-time ingestion could be introduced via streaming, allowing the system to capture live sensor feeds and respond dynamically to changing biological conditions. Building on these real-time capabilities, incorporating predictive analytics and ML models into the Lakehouse would further support proactive decision-making, for example, by forecasting water quality trends or fish growth metrics.

Another avenue for future work is a detailed cost and performance comparison between Microsoft Fabric and Databricks in production scenarios. Such an evaluation should measure not only licensing and compute expenses but also factors like development velocity, operational overhead, and scalability under realistic workloads. By benchmarking both platforms on identical aquaponics workloads, ranging from batch ETL to streaming ingestion and advanced analytics, this study would guide long-term platform strategy.

Additionally, a dedicated data analyst with in-depth knowledge of the aquaponics system could further enhance the solution by crafting narrative-driven Power BI reports. By combining domain expertise with advanced data storytelling techniques, such as guided report tours, contextual annotations, and dynamic bookmarks, this specialist

would transform raw metrics into cohesive insights, improving stakeholder engagement and supporting more informed decision-making.

In summary, this thesis has demonstrated the value of combining modern Lakehouse architecture with self-service BI tools to drive sustainable innovation in biological research. The proposed future work will deepen that integration, enhance real-time responsiveness, and strengthen the platform's analytical capabilities.

References

[1] F. Sebastião, D. C. Vaz, C. L. Pires, *et al.*, "Nutrient-efficient catfish-based aquaponics for producing lamb's lettuce at two light intensities," *Journal of the Science of Food and Agriculture*, vol. 104, no. 11, pp. 6541–6552, 2024. DOI: 10.1002/jsfa.13478. [Online]. Available: https://doi.org/10.1002/jsfa.13478.

[2] S. Goddek, A. Joyce, B. Kotzen, and G. Burnell, *Aquaponics Food Production Systems: Combined Aquaculture and Hydroponic Production Technologies for the Future*. Cham: Springer International Publishing, 2019, Open Access under CC BY 4.0, ISBN: 978-3-030-15943-6. DOI: 10.1007/978-3-030-15943-6. [Online]. Available: https://doi.org/10.1007/978-3-030-15943-6.

[3] P. Chapman, J. Clinton, R. Kerber, *et al.*, *Crisp-dm 1.0: Step-by-step data mining guide*, Retrieved from DaimlerChrysler website, 2000. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://mineracaodedados.wordpress.com/wp-content/uploads/2012/12/crisp-dm-1-0.pdf.

[4] Fraunhofer Institute for Surface Engineering and Thin Films IST, *Implementation strategy for a data-mining project with crisp-dm in surface technology*, Accessed: 2025-04-19, 2024. [Online]. Available: https://www.ist.fraunhofer.de/en/expertise/simulation-digital-services/data-acquisition-model-based-process-optimization/crisp-dm-surface-technology.html.

[5] A. M. Shimaoka, R. C. Ferreira, and A. Goldman, *The evolution of crisp-dm for data science: Methods, processes and frameworks*, Preprint, Open Access under CC BY-SA 4.0, Oct. 2024. DOI: 10.13140/RG.2.2.22493.42721. [Online]. Available: https://doi.org/10.13140/RG.2.2.22493.42721.

[6] A. R. da Silva Mendes, "Quality data mart: Desenvolvimento de dashboards relativos à qualidade de produtos e processos," Supervised by Prof. Helena Maria Pereira Pinto Dourado e Alvelos, Relatório de Projeto de Mestrado, Universidade de Aveiro, Departamento de Economia, Gestão, Engenharia Industrial e Turismo, 2020.

[7] E. R. de Oliveira and E. C. L. Pereira, *Desenvolvimento de um roteiro para a análise de projetos e carga de trabalho utilizando ferramentas de business intelligence com base em dados disponíveis no setor de bens de consumo*, Trabalho de Conclusão de Curso

(Bacharelado em Engenharia Mecânica) — Universidade Tecnológica Federal do Paraná (UTFPR), Advisor: Profª. Drª. Cleina Yayoe Okoshi, Curitiba, Brasil, 2022.

[8] L. A. d. Andrade and M. A. Bovério and F. Camilotti and F. d. F. Borges, "Aquaponia e sua relação com a sustentabilidade," vol. 13, no. 1, 2021.

[9] A. M. P. and A. R. R. Yanes, "Towards automated aquaponics: A review on monitoring, iot, and smart systems.," vol. 263, 2020.

[10] B. König, R. Junge, A. Bittsánszky, M. Villarroel, and T. Komives, "On the sustainability of aquaponics," vol. 2, no. 1, 2016.

[11] M. Krastanova, I. Sirakov, S. Ivanova-Kirilova, D. Yarkov, and P. Orozoza, "Aquaponic systems: Biological and technological parameters," *Biotechnology & Biotechnological Equipment* (*Taylor & Francis Group*), vol. 36, no. 1, 2022. DOI: 10.1080/13102818.2022.2074892.

[12] R. Chang, *Chemistry*. New York: McGraw-Hill, 1998.

[13] S. Singh, R. Yadav, R. K. Srivastava, and R. Sharma, "Optimization of potassium (k□) supplementation for growth enhancement of *spinacia oleracea* l. and *pangasianodon hypophthalmus* in an aquaponic system," *Aquaculture Reports*, vol. 26, p. 101 276, 2022. DOI: 10.1016/j.aqrep.2022.101276. [Online]. Available: https://doi.org/10.1016/j.aqrep.2022.101276.

[14] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Indianapolis: John Wiley & Sons, 2013, ISBN: 978-1118530801.

[15] L. T. Moss and S. Atre, *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*. Addison-Wesley Professional, 2003.

[16] V. Shah, *What is modern bi? how is it different from traditional bi?* Accessed 4 12, 2025, Oct. 2024. [Online]. Available: https://www.thoughtspot.com/data-trends/business-intelligence/modern-bi.

[17] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[18] H. J. Hadi, A. H. Shnain, S. H. Shaheed, and A. b. H. Ahmad, "Big data and five v's characteristics," *International Journal of Advances in Electronics and Computer Science*, vol. 2, no. 4, pp. 2393–2398, 2015, Accessed: 2025-07-05. [Online]. Available: https://www.researchgate.net/publication/332230305_BIG_DATA_AND_FIVE_V%27S_CHARACTERISTICS.

[19] R. Jain, "Big data and competition law: Navigating trade practices in the digital age," *Journal of Law, Market and Innovation*, vol. 4, no. 1, 2025.

[20] H. Al-Aqrabi, L. Liu, R. Hill, and N. Antonopoulos, "Cloud bi: Future of business intelligence in the cloud," *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 85–96, 2015. DOI: 10.1016/j.jcss.2014.06.008.

[21] M. Armbrust, A. Fox, R. Griffith, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[22] P. Mell and T. Grance, "The nist definition of cloud computing," National Institute of Standards and Technology, Tech. Rep. Special Publication 800-145, 2011.

[23] D. Kumar, *Data lake*, Accessed 4 12, 2025, Sep. 2024. [Online]. Available: https://medium.com/@danushidk507/data-lake-0a93f3b546fa.

[24] M. Armbrust, T. Das, S. Zhu, *et al.*, "Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics," *Communications of the ACM*, vol. 64, no. 12, pp. 54–63, 2021.

[25] Datafortune, *The future of self-service bi: Trends and innovations to watch out for*, Accessed 4 12, 2025, Jul. 2024. [Online]. Available: https://datafortune.com/the-future-of-self-service-bi-trends-and-innovations-to-watch-out-for/.

[26] B. Balusamy, N. R. Abirami, S. Kadry, and A. H. Gandomi, *Big Data: Concepts, Technology, and Architecture*. Hoboken, NJ: Wiley, 2021, ISBN: 978-1-119-70182-8.

[27] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical – don't focus on big data; focus on the data that's big," IDC, Tech. Rep., 2017, Sponsored by Seagate. [Online]. Available: https://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf.

[28] M. Shahnawaz and M. Kumar, "A comprehensive survey on big data analytics: Characteristics, tools and techniques," *ACM Computing Surveys*, vol. 57, no. 16, pp. 1–33, 2025. DOI: 10.1145/3718364.

[29] S. Ponnusamy and P. Gupta, "Scalable data partitioning techniques for distributed data processing in cloud environments: A review," *IEEE Access*, vol. 12, 2024. DOI: 10.1109/ACCESS.2024.10507385. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10436080.

[30] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2015.

[31] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2012.

[32] M. Zaharia, T. Das, H. Li, *et al.*, "Apache spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[33] A. A. Harby and F. Zulkernine, "From data warehouse to lakehouse: A comparative review," in *2022 IEEE International Conference on Big Data (Big Data)*, Presented at IEEE Big Data 2022, 17–20 December 2022. Added to IEEE Xplore on 26 January 2023, Osaka, Japan: IEEE, 2022, ISBN: 978-1-6654-8045-1. DOI: 10.1109/BigData55660.2022.10020719. [Online]. Available: https://doi.org/10.1109/BigData55660.2022.10020719.

[34] E. Mehmood and A. Afzal, *Connected lakehouse: The future of modern data warehousing & analytics*, Accessed: 2025-04-13, Systems Ltd, Nov. 2021. [Online]. Available: https://www.systemsltd.com/blogs/connected-lakehouse-future-modern-data-warehousing-analytics.

[35] P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, and M. Zaharia, "Analyzing and comparing lakehouse storage systems," in *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2023. [Online]. Available: %5Curl%7Bhttps://www.cidrdb.org/cidr2023/papers/p92-jain.pdf%7D.

[36] S. Yu, "Acid properties in distributed databases," University of Helsinki, Tech. Rep., 2009, Advanced eBusiness Transactions Seminar paper. [Online]. Available: https://www.cs.helsinki.fi/group/cinco/teaching/2009/advanced-businesstransactions-seminar/papers/ACID_in_Distributed_Database_Shiwei_Yu.pdf.

[37] J. Giceva and M. Sadoghi, "Hybrid oltp and olap," in Jan. 2018, pp. 1–8, ISBN: 9783319639628. DOI: 10.1007/978-3-319-63962-8_179-1. [Online]. Available: https://www.researchgate.net/publication/4140602_OLTP_and_OLAP_data_integration_A_review_of_feasible_implementation_methods_and_architectures_for_real_time_data_analysis.

[38] A. A. Harby and F. Zulkernine, "Data lakehouse: A survey and experimental study," *Information Systems*, vol. 127, p. 102 460, 2025, ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2024.102460. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437924001182.

[39] B. John, "Comparative analysis of data lakes and data warehouses for machine learning workflows: Architecture, performance, and scalability considerations," Apr. 2025. [Online]. Available: https://www.researchgate.net/publication/390532876_Comparative_Analysis_of_Data_Lakes_and_Data_Warehouses_for_Machine_Learning_Workflows_Architecture_Performance_and_Scalability_Considerations.

[40] A. Chikhalkar, M. Brünninghaus, S. Deppe, E. Bicker, and C. Röcker, "A data pipeline concept for digitizing services in small and medium-sized companies," *International Journal on Informatics Visualization*, vol. 9, no. 1, pp. 333–341, Jan. 2025. [Online]. Available: %5Curl%7Bhttps://joiv.org/index.php/joiv/article/view/3796%7D.

[41] M. Databricks, *What is a medallion architecture?* Accessed 4 12, 2025, 2020. [Online]. Available: %5Curl%7Bhttps://www.databricks.com/glossary/medallion-architecture%7D.

[42] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite, *The Data Warehouse Lifecycle Toolkit*. Indianapolis: John Wiley & Sons, 1998, ISBN: 978-0471255475.

[43] W. H. Inmon, *Building the Data Warehouse*, 4th. Wiley, 2005.

[44] S. Pandey. "Data lake table formats: Apache iceberg vs apache hudi vs delta lake." Medium article, accessed 4 17, 2025. (Aug. 2023), [Online]. Available: https://medium.com/@shashwat.pandey/data-lake-table-formats-apache-iceberg-vs-apache-hudi-vs-delta-lake-10b67a1d587.

[45] H. P. Salim, "A comparative study of delta lake as a preferred etl and analytics database," *International Journal of Computer Trends and Technology* (*IJCTT*), vol. 73, no. 1, pp. 65–71, Jan. 2025, ISSN: 2231-2803. DOI: 10.14445/22312803/IJCTT-V73I1P108. [Online]. Available: https://doi.org/10.14445/22312803/IJCTT-V73I1P108.

[46] P. Bhosale, "Scalable metadata management in data lakes: The role of apache iceberg," *International Journal on Science and Technology* (*IJSAT*), vol. 3, 2024, E-ISSN: 2229-7677. [Online]. Available: https://www.ijsat.org/papers/2024/3/1409.pdf.

[47] V. Belov and E. Nikulchev, "Analysis of big data storage tools for data lakes based on apache hadoop platform," *International Journal of Advanced Computer Science and Applications* (*IJACSA*), vol. 12, no. 8, 2021. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://thesai.org/Downloads/Volume12No8/Paper_64-Analysis_of_Big_Data_Storage_Tools.pdf.

[48] S. Pandey. "Data lake table formats: Apache iceberg vs apache hudi vs delta lake." Accessed June 18, 2025, Medium. (2023), [Online]. Available: https://shashwat-pandey.medium.com/data-lake-table-formats-apache-iceberg-vs-apache-hudi-vs-delta-lake-10b67a1d587.

[49] F. Databricks. "Databricks named a leader in the 2024 forrester wave for data lakehouses." Accessed on 4 17, 2025. (Apr. 2024), [Online]. Available: https://www.databricks.com/blog/databricks-named-leader-2024-forrester-wave-data-lakehouses.

[50] D. Pálma. "Databricks vs snowflake: The ultimate data warehouse showdown for 2025." [Online] Accessed: 4 18, 2025. (2024), [Online]. Available: https://estuary.dev/databricks-vs-snowflake/.

[51] Gerd Saure, Sneha Gunda and Vahid Doustimajd, *What is microsoft fabric?* Accessed on 4 18, 2025, 2025. [Online]. Available: https://learn.microsoft.com/en-us/fabric/fundamentals/microsoft-fabric-overview.

[52] Pricing Google Cloud. "Pricing overview." Accessed on 4 18, 2025. (2024), [Online]. Available: https://cloud.google.com/pricing.

[53] Polestar. "Lakehouse big four: Aws, snowflake, azure & google cloud." Accessed on 4 18, 2025. (2023), [Online]. Available: https://polestarllp.com/blog/lakehouse-big-four-aws-snowflake-azure-google-cloud.

[54] BigQuery Google Cloud. "Bigquery: From data warehouse to autonomous data and ai platform." Accessed on 4 18, 2025. (2024), [Online]. Available: https://cloud.google.com/bigquery?hl=Eng.

[55] Google Cloud. "Use the spark bigquery connector." Accessed on 4 18, 2025. (2024), [Online]. Available: https://cloud.google.com/dataproc/docs/tutorials/bigquery-connector-spark-example.

[56] Kanerika. "Databricks vs snowflake vs fabric: A complete comparison guide." Accessed on 4 18, 2025. (2025), [Online]. Available: https://kanerika.com/blogs/databricks-vs-snowflake-vs-fabric/.

[57] Atlan. "Working with apache iceberg on databricks: A complete guide [2025]." Accessed on 4 18, 2025. (2025), [Online]. Available: https://atlan.com/know/iceberg/databricks-apache-iceberg/.

[58] D. L. Databricks. "What is delta lake?" Accessed on 4 18, 2025. (2025), [Online]. Available: https://docs.databricks.com/aws/en/delta/.

[59] Snowflake. "External tables: Apache iceberg support." Accessed on 4 18, 2025. (2024), [Online]. Available: https://docs.snowflake.com/en/user-guide/tables-external-intro.

[60] C. Vukos-Walker. "Data quality native support for iceberg format (preview)." Accessed on 4 18, 2025. (2025), [Online]. Available: https://learn.microsoft.com/en-us/purview/unified-catalog-data-quality-iceberg.

[61] D. Coelho. "Lakehouse and delta lake tables." Accessed on 4 18, 2025. (2023), [Online]. Available: https://learn.microsoft.com/en-us/fabric/data-engineering/lakehouse-and-delta-tables.

[62] Microsoft Fabric Interface, *Microsoft fabric interface overview*, Accessed on 4 18, 2025, 2025. [Online]. Available: https://learn.microsoft.com/en-us/fabric/.

[63] Data Flows Gen2 Microsoft Learn, *Understand dataflows gen2 in microsoft fabric*, Accessed on 4 18, 2025, 2025. [Online]. Available: https://learn.microsoft.

com/en-us/training/modules/use-dataflow-gen-2-fabric/2-dataflows-gen-2.

[64] W. A. Mark Pryce-Maher, *What is data warehousing in microsoft fabric?* https://learn.microsoft.com/en-us/fabric/data-warehouse/data-warehousing, Accessed: 04 03 2025, Aug. 2024.

[65] W. A. Mark Pryce-Maher, *What is a lakehouse in microsoft fabric?* https://learn.microsoft.com/en-us/fabric/data-engineering/lakehouse-overview, Accessed: 2025-03-04, Feb. 2025.

[66] William Assaf. "Microsoft fabric decision guide: Choose between warehouse and lakehouse." Accessed on 4 18, 2025. (2025), [Online]. Available: https://learn.microsoft.com/en-us/fabric/fundamentals/decision-guide-lakehouse-warehouse.

[67] Paul Inbar, Jene Zhang. "How to use microsoft fabric notebooks." Accessed on 4 18, 2025. (2025), [Online]. Available: https://learn.microsoft.com/en-us/fabric/data-engineering/how-to-use-notebook.

[68] Whitney Henderson, *What is data factory in microsoft fabric?* Accessed on 4 18, 2025, 2025. [Online]. Available: https://learn.microsoft.com/en-us/fabric/data-factory/data-factory-overview.

[69] Michael Sanky and Michael Ortega. "Introducing lakehouse for healthcare and life sciences: Delivering better patient outcomes with data and ai." Accessed: 2025-04-18. (Mar. 2022), [Online]. Available: https://www.databricks.com/blog/2022/03/09/introducing-lakehouse-for-healthcare-and-life-sciences.html.

[70] Finance Databricks. "Databricks launches lakehouse for financial services to accelerate data-driven innovation across the industry." Accessed: 2025-04-18. (Feb. 2022), [Online]. Available: https://www.databricks.com/company/newsroom/press-releases/databricks-launches-lakehouse-for-financial-services-to-accelerate-data-driven-innovation-across-the-industry.

[71] Opendatasoft, *Harnessing open data to create smart communities: Why you don't need to be a major city to benefit from becoming smart*, https://www.opendatasoft.com/wp-content/uploads/2023/01/202212_Smart-cities_V3.pdf, Accessed: 2025-04-18, Dec. 2022.

[72] Zihang Lin and Yian Yin and Lu Liu and Dashun Wang, "Sciscinet: A large-scale open data lake for the science of science research," *Scientific Data*, vol. 10, 2023. DOI: 10.1038/s41597-023-02198-9. [Online]. Available: https://www.nature.com/articles/s41597-023-02198-9.

[73] Redress Compliance, *Case study: H&m's use of ai to optimize fashion operations and customer experience*, Accessed: 2025-04-18, Feb. 2025. [Online]. Available: https:

//aiexpert.network/case-study-how-hm-leverages-ai-for-supply-chain-efficiency-and-customer-experience/.

[74] T. Morgan, *Business Rules and Information Systems: Aligning IT with Business Goals*. Boston, MA, USA: Addison-Wesley Professional, 2008, ISBN: 0-201-74391-4.

[75] R. R. Pansara, "Master data quality and business rules: A comprehensive analysis," *Saudi Journal of Engineering and Technology*, vol. 9, no. 2, pp. 34–43, Feb. 2024. DOI: 10.36348/sjet.2024.v09i02.001. [Online]. Available: https://saudijournals.com/media/articles/SJEAT_92_34-43.pdf.

[76] J. Wang and J. L. Kourik, "Data warehouse snowflake design and performance considerations in business analytics," *Journal of Applied Information Technology*, vol. 3, no. 2, pp. 45–53, 2015. [Online]. Available: https://www.jait.us/uploadfile/2015/1027/20151027105124540.pdf.

[77] Databricks Documentation, *Databricks free edition*, https://docs.databricks.com/aws/en/getting-started/free-edition, Last updated June 11, 2025; accessed June 29, 2025, Jun. 2025.

[78] Databricks Documentation, *Databricks free edition*, https://docs.databricks.com/aws/en/getting-started/free-edition-limitations?, Last updated June 11, 2025; accessed June 29, 2025, Jun. 2025.

[79] Microsoft Corporation, *Microsoft fabric trial capacity*, https://learn.microsoft.com/en-us/fabric/fundamentals/fabric-trial, Accessed June 29, 2025, May 2025.

[80] D. Klopfenstein, *Sharepoint folder connector overview*, https://learn.microsoft.com/en-us/fabric/data-factory/connector-sharepoint-folder-overview, Accessed: 2025-03-04, Dec. 2024.

[81] William Assaf, *Sql analytics endpoint performance considerations*, https://learn.microsoft.com/en-us/fabric/data-warehouse/sql-analytics-endpoint-performance, Accessed: 2025-03-04, Apr. 2024.

[82] S. G. Ted Vilutis. "Create shortcuts in lakehouse." Accessed: 2025-05-01. (2024), [Online]. Available: https://learn.microsoft.com/en-us/fabric/data-engineering/lakehouse-shortcuts.

[83] JeneZhang, *Develop, execute, and manage microsoft fabric notebooks*, Microsoft, https://learn.microsoft.com/en-us/fabric/data-engineering/author-execute-notebook, Accessed: 2025-03-04, Apr. 2024.

[84] W. Assaf, *Default power bi semantic models in microsoft fabric*, Microsoft, https://learn.microsoft.com/en-us/fabric/data-warehouse/semantic-models, Accessed: 2025-03-04, Jun. 2024.

[85] W. Assaf, *Quickstart: Create your first dataflow to get and transform data*, Microsoft, https://learn.microsoft.com/en-us/fabric/data-factory/create-first-dataflow-gen2, Accessed: 2025-02-27, Dec. 2024.

[86] M. Wagle, *Dataflow gen2 pricing for data factory in microsoft fabric*, Microsoft, https://learn.microsoft.com/en-us/fabric/data-factory/pricing-dataflows-gen2, Accessed: 2025-02-27, Dec. 2024.

[87] DataCamp. "What is a power bi hierarchy?" Accessed: 27 April 2025. (2024), [Online]. Available: https://www.datacamp.com/tutorial/power-bi-hierarchies.

[88] D. Iseminger. "Dax basics in power bi desktop." Accessed: 2025-04-27. (2024), [Online]. Available: https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-quickstart-learn-dax-basics.

[89] J. Brooke, "Sus: A retrospective," *Journal of Usability Studies*, vol. 8, no. 2, pp. 29–40, Feb. 2013, Independent consultant, Sonning, Reading, UK.

Appendices

A

Notion: Board and Calendar

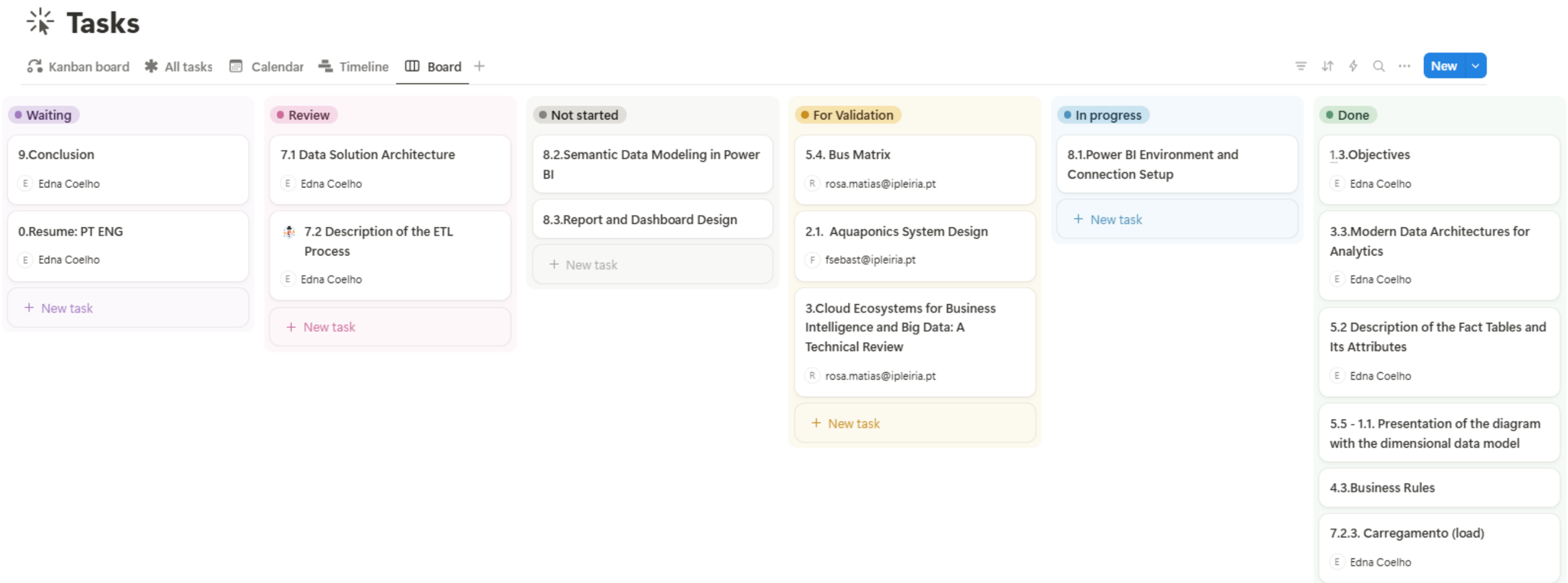


Figure A.1: *Board in Notion visualizing the status of master's thesis tasks across workflow stages: Waiting, Review, Not Started, For Validation, In Progress, and Done.*

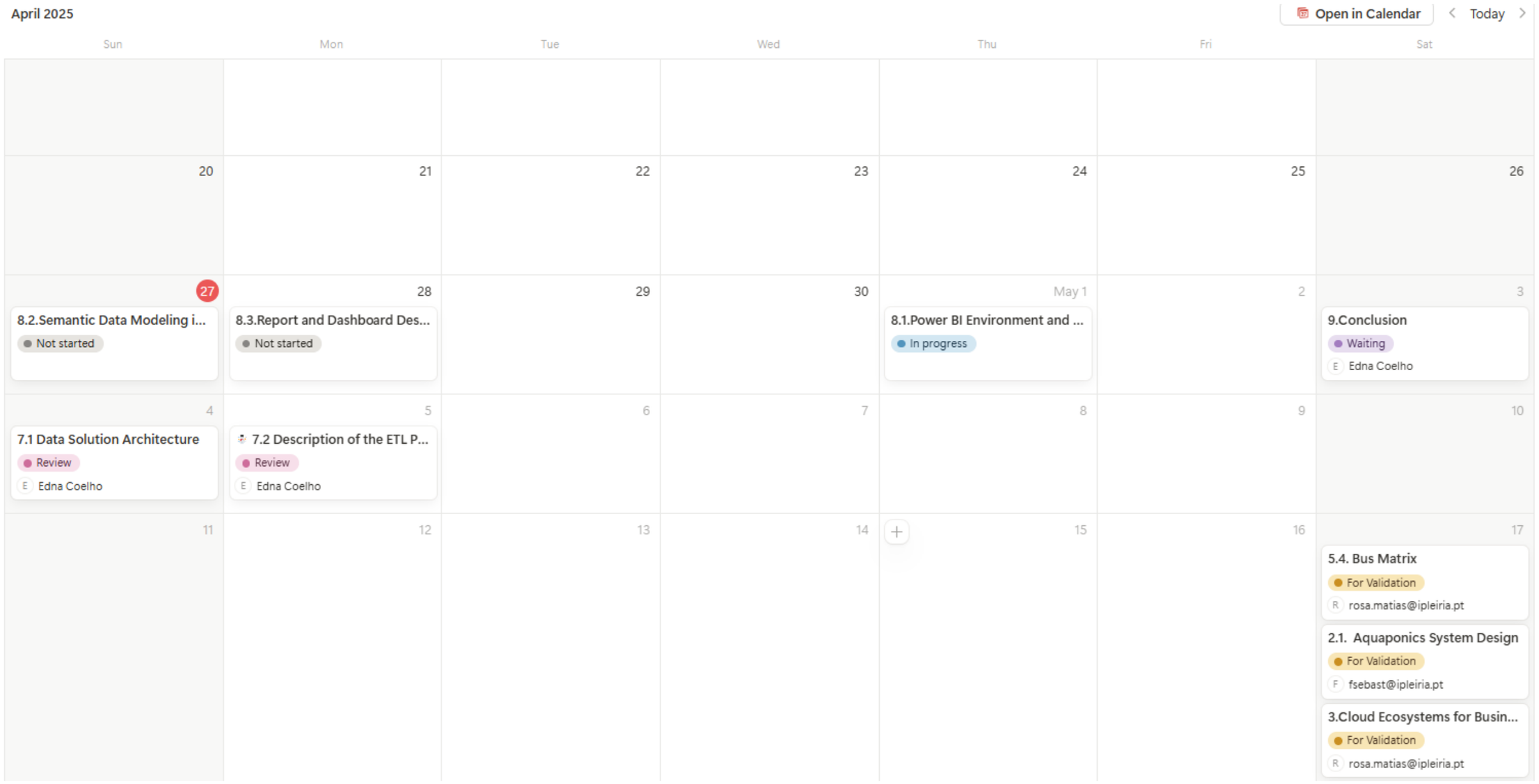


Figure A.2: *Notion calendar view (April–May 2025) displaying scheduled master's thesis tasks, their workflow status, and assigned stakeholders.*

B

Bus Matrix

Bussiness Process	Grain	Measures	Dimension Date	Dim Time	Dim AM PM	Dim Measures	Dim Fish Attributes	Dim Fishtank DWC	Dim Lines	Dim Plants	Dim Projects	Dimensions Counts
Fact Fish Events	One row for each fish in a tank on a given day that either moved to another tank or died	Number Fish	✓				✓	✓				3
Fact Daily System	One row represents a record of a parameter value at a specific AM or PM in a tank	Value	✓		✓	✓		✓				4
Fact Water Consumption	One row represents a record of a parameter value at a specific day in a line	Value	✓			✓			✓			3
Fact Atmospheric Condition	One row represents a record of a parameter value at a specific time in a line	Value	✓	✓		✓			✓			4
Fact Measurements Nutrients Metals	One row represents a record of a parameter value at a specific day in a tank	Value	✓			✓		✓				3
Fact Plants Development	One row represents a record of a parameter value for a specific day in a line that contains a plant or fruit.	Value	✓			✓			✓	✓ Fruit	✓	5

Figure B.1: *Bus Matrix*

C

Diagram with the Dimensional Data Model of LSMI

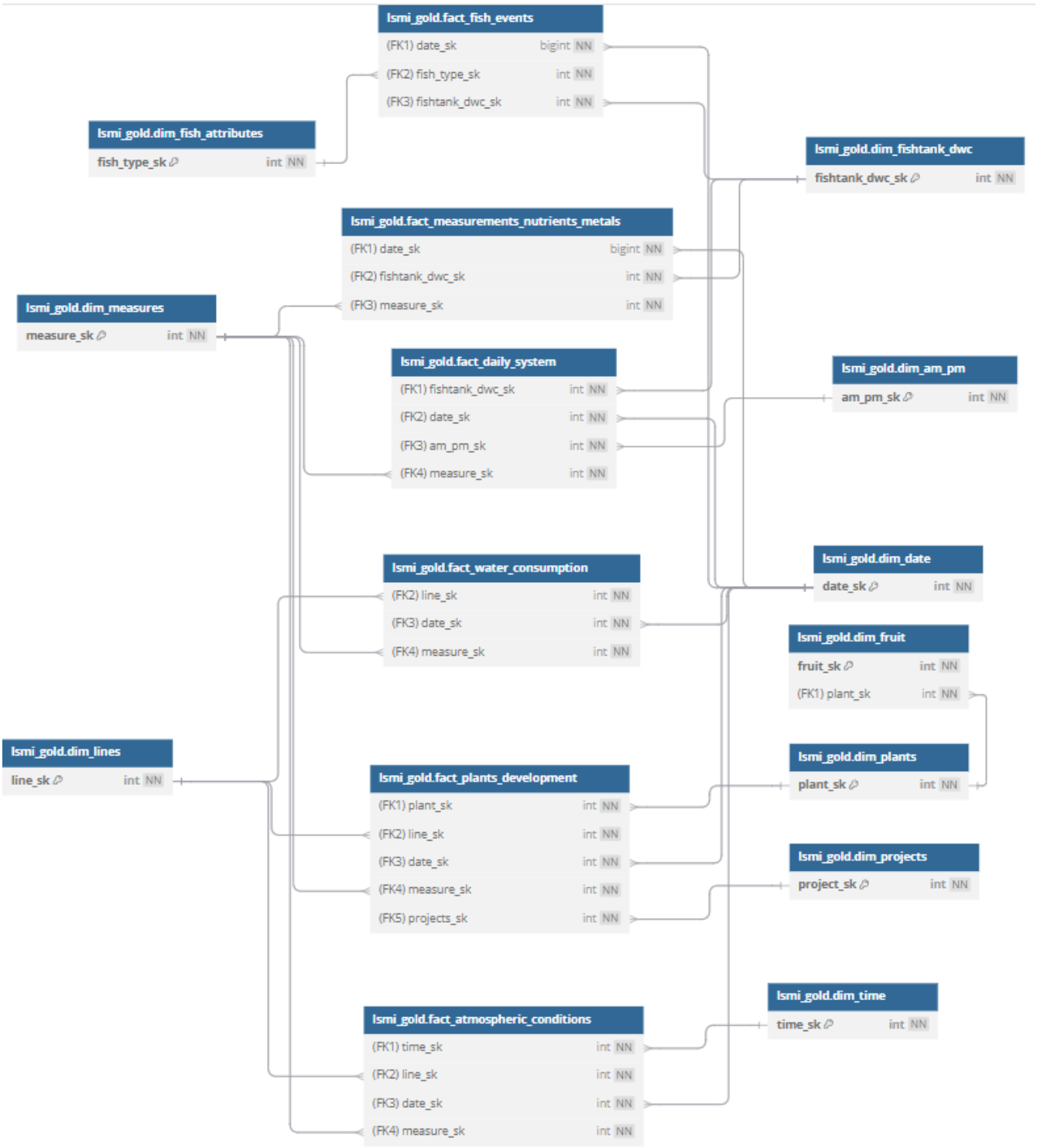


Figure C.1: *Diagram with the dimensional data model of LSMI*

D
Data Mapping

gold_table	column_name_gold	description	data_type	calculated
fact_atmospheric_conditions	value	Values of metrics or parameters	Decimal	No
fact_daily_system	value	Values of metrics or parameters	Decimal	Yes
fact_fish_events	fish_out_in	This field indicates whether a fish has entered or left the tank	String	No
fact_fish_events	number_fish	Number of fish	Int	No
fact_measurements_nutrients_metals	value	Values of metrics or parameters	Decimal	Yes
fact_plants_development	value	Values of metrics or parameters	Decimal	No
fact_water_consumption	value	Values of metrics or parameters	Decimal	Yes

Table D.1: *Table mapping the measures columns of the fact tables between the source and the target*

gold_table	silver_table	column_name_silver	bronze_table	column_name_bronze	relationship	domain
fact_atmospheric_conditions	atmospheric_conditions	value	atmospheric_conditions_shpt	value	No	Decimal numbers
fact_daily_system	daily_system	value	daily_system_shpt	value	No	Decimal numbers
fact_fish_events	fish_events	fish_out_in	fish_events_shpt	fish_out_in	No	String Out or In
fact_fish_events	fish_events	number_fish	fish_events_shpt	number_fish	No	Number integer
fact_measurements_nutrients_metals	measurements_nutrients_metals	value	measurements_nutrients_metals_shpt	value	No	Decimal numbers
fact_plants_development	plants_development	value	plants_development_shpt	value	No	Decimal numbers
fact_water_consumption	water_consumption	value	water_consumption_shpt	value	No	Decimal numbers

Table D.2: *Table mapping the measures columns of the fact tables between the source and the target (continuation)*

E

Semantic Model

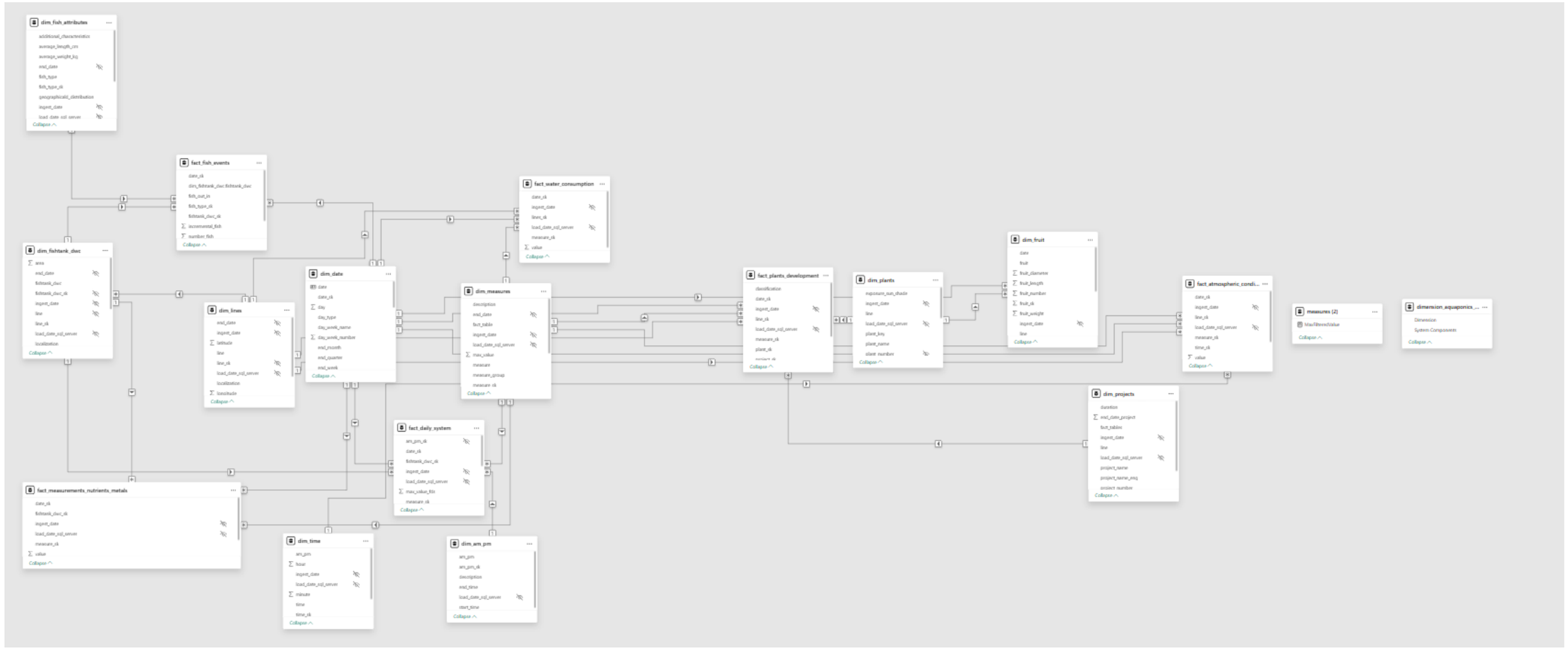


Figure E.1: *Model View in Power BI*

F

DAX Measures

measuregroup_name	measure_name	expressions
fact_daily_system	[Measures].[daily_value_dynamic]	SWITCH(TRUE(), -- Se a medida for CaOH2, CaCO3 ou Tenebrios SELECTEDVALUE('dim_measures'[measure]) IN { "CaOH2", "CaCO3", "Tenebrios"}, SUM('fact_daily_system'[value]), -- Caso contrário (medidas restantes) AVERAGE('fact_daily_system'[value]))
fact_daily_system	[Measures].[axis_title]	SELECTEDVALUE('dim_measures'[measure], "Parâmetro")

fact_daily_system	[Measures].[daily_max_value_dynamic]	SWITCH(
		TRUE(),
		SELECTEDVALUE('dim_measures'[measure]) IN { "CaOH2", "CaCO3", "Tenebrios" }, 5,
		SELECTEDVALUE('dim_measures'[measure]) = "FishFeed", 200,
		AVERAGE('fact_daily_system'[max_value_fds]) // Valor padrão
		)

fact_daily_system	[Measures].[daily_min_value_dynamic]	SWITCH(
		TRUE(),
		SELECTEDVALUE('dim_measures'[measure]) IN { "CaOH2", "CaCO3", "Tenebrios" }, 0,
		SELECTEDVALUE('dim_measures'[measure]) = "FishFeed", 0,
		AVERAGE('fact_daily_system'[min_value_fds]) // Valor padrão
		)

fact_daily_system	[Measures].[daily_value_avg_sd]	VAR MediaValue = AVERAGE(fact_daily_system[value]) VAR DesvioValue = STDEV.P(fact_daily_system[value]) RETURN FORMAT(MediaValue, "0.00") & " ± " & FORMAT(DesvioValue, "0.00")

Figure F.1: *Dax Measures - Part.I*

measuregroup_name	measure_name	expressions
		VAR currentMeasure = SELECTEDVALUE(dim_measures[measure_group]) VAR val = AVERAGE(fact_daily_system[Value]) VAR unit = SELECTEDVALUE(dim_measures[units]) RETURN
fact_daily_system	[Measures].[avg_value_unit_daily]	CONCATENATE(FORMAT(val, "0.0"), CONCATENATE(" ", unit))
		VAR project = SELECTEDVALUE(dim_projects[project_number]) VAR StartProjectDate = CALCULATE(MIN(dim_projects[start_date_project]), dim_projects[project_number] = project) VAR EndProjectDate = CALCULATE(MAX(dim_projects[end_date_project]), dim_projects[project_number] = project) RETURN CALCULATE(SUM(fact_fish_events[number_fish]), ALL(dim_date[date]), dim_date[date_sk] <= EndProjectDate
fact_fish_events	[Measures].[number_fish_end_project]	)

Figure F.2: *Dax Measures - Part.II*

measuregroup_name	measure_name	expressions
		VAR LastRow = TOPN(1, FILTER(fact_fish_events, fact_fish_events[incremental_fish] <> 0 && fact_fish_events[date_sk] <= MAX(dim_date[date_sk])), fact_fish_events[date_sk], DESC) RETURN IF(MINX(LastRow, fact_fish_events[number_fish]) < 0, MINX(LastRow, fact_fish_events[incremental_fish]), MAXX(LastRow, fact_fish_events[incremental_fish])
fact_fish_events	[Measures].[last_incremental_fish]	)

		VAR value_avg = AVERAGE(fact_measurements_nutrients_metals[value])
		VAR min_value_dim = CALCULATE(MIN(dim_measures[min_value]))
		VAR max_value_dim = CALCULATE(MAX(dim_measures[max_value]))
		RETURN
		IF(
		ISBLANK(value_avg),
		BLANK(),
		IF(value_avg > min_value_dim && value_avg < max_value_dim, 1,0)

fact_measurements_nutrients_metals	[Measures].[flag_nm]	)

Figure F.3: *Dax Measures - Part.III*

measuregroup_name	measure_name	expressions
		VAR currentMeasure = SELECTEDVALUE(dim_measures[measure_group]) VAR val = AVERAGE(fact_measurements_nutrients_metals[Value]) VAR unit = SELECTEDVALUE(dim_measures[units]) RETURN
fact_measurements_nutrients_metals	[Measures].[avg_value_unit_metal_nutrient]	CONCATENATE(FORMAT(val, "0.0"), CONCATENATE(" ", unit))

		VAR value_avg = AVERAGE(fact_atmospheric_conditions[value])
		VAR min_value_dim = CALCULATE(MIN(dim_measures[min_value]))
		VAR max_value_dim = CALCULATE(MAX(dim_measures[max_value]))
		RETURN
		IF(
		ISBLANK(value_avg),
		BLANK(),
		IF(value_avg > min_value_dim && value_avg < max_value_dim, 1,0)

fact_atmospheric_conditions	[Measures].[flag_ac]	)
		VAR currentMeasure = SELECTEDVALUE(dim_measures[measure]) VAR val = AVERAGE(fact_atmospheric_conditions[Value]) RETURN SWITCH(TRUE(), ISBLANK(val), "--", currentMeasure = "Readings_Temperature", FORMAT(val, "0.0") & " °C", currentMeasure = "Readings_Relative_Humidity", FORMAT(val, "0.0") & " %", FORMAT(val, "0.0")
fact_atmospheric_conditions	[Measures].[avg_value_unit_atmosferic]	)
fact_plants_development	[Measures].[value_plants_dev_upper_sd]	STDEV.P(fact_plants_development[value])

Figure F.4: *Dax Measures - Part.IV*

measuregroup_name	measure_name	expressions
fact_plants_development	[Measures].[value_plants_dev_upper_sd]	STDEV.P(fact_plants_development[value])
fact_plants_development	[Measures].[value_plants_dev_lower_sd]	- STDEV.P(fact_plants_development[value])
measure	[Measures].[MaxFilteredValue]	VAR DaysWithValue = FILTER(ALL('fact_daily_system'), -- Substitua pelo nome da tabela correta NOT(ISBLANK('fact_daily_system'[daily_value_dynamic])) -- Filtra apenas os dias com valor) RETURN MAXX(DaysWithValue, RELATED(dim_measures[max_value]))
measure	[Measures].[status_filter_measure_fishtank]	VAR SelectedFishtank = SELECTEDVALUE('dim_fishtank_dwc'[fishtank_dwc]) VAR SelectedMeasureType = SELECTEDVALUE('dim_measures'[measure_type]) RETURN IF (SelectedMeasureType = "metals" && SelectedFishtank IN { "FishTank L2", "FishTank L3" }, "out", "in")

Figure F.5: *Dax Measures - Part.V*

G
Survey

Power BI Usability Survey

Based on the SUS by John Brooke

Instructions
Please rate each statement on a scale from 1 to 5, where:
1 = Strongly Disagree
2 = Disagree
3 = Neutral
4 = Agree
5 = Strongly Agree

Respond instinctively – don't overthink each statement.

2. What is your occupation?

- ○ Student
- ○ Researcher
- ○ Professor

3. Did you have previous experience as a Power BI user?

- ○ Yes
- ○ No

3. Power BI Interaction Feedback.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1. I find Power BI useful for understanding the information presented.	○	○	○	○	○
2. Navigating through Power BI reports feels confusing or overwhelming.	○	○	○	○	○
3. It is easy for me to find the information I need in Power BI dashboards.	○	○	○	○	○
4. I would need help from someone with technical skills to use Power BI effectively.	○	○	○	○	○
5. The layout of the dashboard is intuitive.	○	○	○	○	○
6. I often feel lost or unsure about how to interact with Power BI reports.	○	○	○	○	○
7. The dashboard loads and responds quickly to my interactions.	○	○	○	○	○
8. It is not easy to switch between different pages or views in the dashboard.	○	○	○	○	○
9. The visual design of the dashboard is clear.	○	○	○	○	○
10. The dashboard is slow or unresponsive during my interactions.	○	○	○	○	○

Table G.1: *Power BI Dashboard Usability Evaluation*