INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Migração de uma Infraestrutura Cloud: De Azure para Microsoft Fabric

ANDRÉ FILIPE VALÉRIO CONCEIÇÃO
Setembro de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Cloud Infrastructure Migration: From Azure to Microsoft Fabric

André Conceição

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Information and Knowledge Systems

Advisor: Dr. Telmo Matos
Co-Advisor: Dr. Paulo Oliveira

Porto, September 28, 2025

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.The exceptions are explicitly recognised in the section "Ethical considerations" of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, September 28, 2025

Dedicatory

This document demonstrates the efforts I have made over the past five years, encompassing not only work but also decisions, personal development, and new experiences, during which I was tested numerous times and consistently achieved my goals.
I dedicate this thesis to my family, my girlfriend, and my son, who mean everything to me and serve as the foundation from which I find the strength to accomplish all my goals.

Abstract

This dissertation addresses a project for the company Business to Future (B2F), specialized in Business Intelligence (BI) solutions and also in Custom Development solutions (or software development). One of B2F's clients is EDULOG, a think tank belonging to the Belmiro de Azevedo Foundation, dedicated to Education, which proposed a new challenge that would be the migration of its Microsoft Azure (Azure) database to Microsoft Fabric (Fabric). EDULOG manages large volumes of statistical data on public web platforms related to Education and Employment in Portugal and globally. This study serves to investigate the technical and operational challenges associated with the migration of EDULOG's cloud infrastructure from Azure to Fabric, with the objective of using the innovative functionalities of Fabric, ensuring a low-impact transition.

Fabric is a platform for data analysis and workload management, in which it offers significant advantages, such as resource optimization, scalability, enhanced security, and centralized administration. The transition from an extensive and complex Azure infrastructure to Microsoft Fabric involves several processes, such as the reconfiguration of services, the maintenance of operational continuity, the transfer of large volumes of data, and the incorporation of new functionalities in work processes, without causing interruptions.

The study contains a comprehensive analysis of the existing Azure infrastructure at EDULOG, documenting its architecture, services, data flow logic, and operational processes. It also compares the capabilities and advantages of Microsoft Fabric, especially in workload management and use cases similar to those of EDULOG. Based on this analysis, a detailed migration methodology was developed to deal with fundamental issues, such as minimizing downtime, ensuring compatibility, strengthening security, and improving operational processes.

It also compares the capabilities and advantages of Microsoft Fabric, especially in workload management and use cases similar to those of EDULOG. Based on this analysis, a detailed migration methodology was developed to work with fundamental issues, such as minimizing downtime, ensuring compatibility, strengthening security, and improving operational processes.

The thesis evaluates performance after migration, cost efficiencies, and the overall effectiveness of the system to measure the success of the migration. In addition, it offers practical recommendations for optimizing the new Fabric environment.

This work was developed through a practical case study on cloud migration and a strategic description for organizations facing similar challenges, showing how they can balance technical innovation with operational continuity.

By providing a detailed migration framework and in-depth information on best practices, the study seeks to ensure a smooth transition for EDULOG, improving its data management and analysis capacity in a rapidly evolving technological environment.

Keywords: Cloud Migration, Microsoft Azure, Microsoft Fabric, Data Integration, Infrastructure Optimization

Resumo

Esta dissertação aborda um projeto para a empresa B2F, especializada em soluções de BI e também em soluções de Desenvolvimento à Medida (ou desenvolvimento de software). Um dos clientes da B2F é a EDULOG, um think tank ao qual pertence à Fundação Belmiro de Azevedo, dedicado à Educação, que propôs um novo desafio que seria a migração da sua base de dados de Azure para Fabric. A EDULOG gere grandes volumes de dados estatísticos em plataformas web públicas relacionadas com Educação e Emprego em Portugal e a nível Global. Este estudo serve para investigar os desafios técnicos e operacionais associados à migração da infraestrutura de cloud da EDULOG do Azure para o Fabric, com o objetivo de utilizar as funcionalidades inovadoras do Fabric, garantindo uma transição de baixo impacto. Fabric é uma plataforma para análise de dados e gestão de workloads, no qual oferece vantagens significativas, tal como a otimização de recursos, escalabilidade, segurança melhorada e administração centralizada. A transição de uma infraestrutura Azure extensa e complexa para o Microsoft Fabric envolve vários processos, tais como, a reconfiguração de serviços, a manutenção da continuidade operacional, a transferência de grandes volumes de dados e a incorporação de novas funcionalidades nos processos de trabalhos, sem causar interrupções. O estudo contém uma análise abrangente da infraestrutura Azure existente na EDULOG, documentando a sua arquitetura, serviços, lógica de fluxo de dados e processos operacionais. Também compara as capacidades e vantagens do Microsoft Fabric, especialmente na gestão de workloads e casos de uso semelhantes aos da EDULOG. Com base nesta análise, foi desenvolvida uma metodologia de migração detalhada para lidar com questões fundamentais, como a minimização de tempo de inatividade, garantia de compatibilidade, reforço da segurança e melhoria dos processos operacionais.

Compara também as capacidades e vantagens do Microsoft Fabric, especialmente na gestão de workloads e casos de uso semelhantes aos da EDULOG. Com base nesta análise, foi desenvolvida uma metodologia de migração detalhada para trabalhar com questões fundamentais, tais como a minimização de tempo de inatividade, garantia de compatibilidade, reforço da segurança e melhoria dos processos operacionais.

A tese avalia o desempenho após a migração, as eficiências de custo e a eficácia geral do sistema para medir o sucesso da migração. Além disso, oferece recomendações práticas para a otimização do novo ambiente Fabric.

Este trabalho foi desenvolvido através de um caso de estudo prático sobre migração de cloud e uma descrição estratégica para organizações que enfrentam desafios semelhantes, mostrando como podem equilibrar a inovação técnica com a continuidade operacional.

Ao fornecer uma estrutura detalhada de migração e informações aprofundadas sobre as melhores práticas, o estudo procura garantir uma transição suave para a EDULOG, melhorando a sua capacidade de gestão e análise de dados num ambiente tecnológico em rápida evolução.

Acknowledgement

First, I would like to thank my dissertation supervisor, Professor Telmo Matos, for his availability and the attention he dedicated to me. And at the same time, I would like to thank Professor Paulo Oliveira for all the support he provided me, as well as for accepting my request, which was made at the last minute. I would also like to thank him for all the valuable feedback he provided.

I would also like to thank B2F for the fantastic opportunity, the support, and the help they provided. In particular, I am grateful to João Conde Pereira for all the time, advice, availability, and flexibility that allowed me to achieve this goal.

I am deeply grateful to my family for all the support, affection, and understanding they have provided me throughout these years, and for encouraging me to continually pursue my dreams.

Finally, I want to express my gratitude to my girlfriend, Cláudia Sameiro, for all the support, understanding, patience, and love she gave me so that I could complete this stage of my life, and for ensuring I never gave up on my dream, even in the face of all the difficulties, without her, this would not have been possible.

Contents

List of Figures

List of Tables

Listings

List of Acronyms

Azure	Microsoft Azure.
B2F	Business to Future.
BI	Business Intelligence.
DW	Data Warehouse.
Fabric	Microsoft Fabric.
ISEP	Instituto Superior de Engenharia do Porto.
RQ	Research Questions.

Chapter 1

Introduction

This thesis was drafted under the scope of the Master's program in Informatics Engineering, specializing in Information and Knowledge Systems, within the Informatics Engineering Department of Instituto Superior de Engenharia do Porto (ISEP). It outlines the fundamental processes and core information that led to the success of realizing the project.
As this first chapter, it puts into context the thesis since it states the problem that will be addressed and states the importance of the work. It indicates general aims, the research methodology used, and a timetable for activities that would be fulfilled in completing the project. In addition, it would states the ethical considerations and can state what impact this research would bring to the organization, the cloud computing domain, and the broader technology landscape.
The final detail is the description of the document structure that will give the reader a roadmap in order to understand what contents and key findings are presented through the different chapters.

1.1 Context

This section is divided into five subsections, where the first one "1.1.1 - Presentation of the problem and its relevance" introduces the problem at the core of this dissertation, stating the issues with the existing infra and why those issues matter. Subsection "1.1.2 - Problem statement" provides a crisp and well-structured statement of the research problem, setting the limits of the research. Subsection "1.1.3 - Background of B2F, EDULOG, EDUSTAT, and the role of cloud infrastructures" develops the institutional and technological background in which this project is conceived, emphasizing the importance of the organizations involved and the role of cloud solutions in their operations. Subsection "1.1.4 – Relevance of the subject matter in today's context of technological advancement" situates the work within the broad trends of digital transformation, accentuating the need for cloud modernization for analytics. Finally, Subsection "1.1.5 – Justification of Migration" provides the reason behind performing the migration from Microsoft Azure (Azure) to Microsoft Fabric (Fabric), outlining the drivers and expected benefits of the technology transition.

1.1.1 Presentation of the problem and its relevance

The ability to collect, manage, and share trustworthy information is essential for organizations with a public purpose in the emerging data-driven world. EDULOG, a thinktank incorporated within the Fundação Belmiro de Azevedo, carries great importance for society in Portugal in that it provides large-scale analyses in two areas of deep socio-economic significance: Education and Employment. Public dissemination of these analyses sees their use

by policymakers, researchers, journalists, and citizens alike. In this context, the credibility, timeliness, and availability of EDULOG data are not just technical considerations but instrumental in informing public debate and developing evidence-based policy. To satisfy this goal, EDULOG is fully dependent on solid cloud infrastructure managed through Business to Future (B2F). For several years, the data ingestion, processing, storage, and serving have depended greatly on Azure. Information volumes increased with time and so did complexity and thus, increased strain on existing architecture. The fragmentation between services, low flexibility were all adding up in as operational overheads began constraining scalability in innovation. Adding new functionality or cross-domain datasets often called for cumbersome workarounds or time-consuming refactoring.
For these reasons, migrating to Fabric goes beyond simply upgrading technology. It is a strategic decision that addresses changes in the operational and social environment. Fabric provides a unified data platform that combines data engineering, analytics, governance, and collaboration within one environment. This convergence needs agility, transparency, and future-readiness, very much in line with EDULOG. This makes operations much simpler, allows near-real-time collaboration by teams, and opens up a new world of advanced data scenarios such as AI, machine learning, and predictive analytics.
In addition to the migration, the shift also encompasses a wider scope of change in digital infrastructure management by public-interest organizations. The organizations engaged in the custody of social data face increasing pressure to use data efficiently, maintain its integrity, and generate insight at an accelerated pace. With the adoption of Fabric, EDULOG now finds itself not just as a tech-efficient organization but also as a progressive custodian of data in the public interest.
Ultimately, migration is not just about infrastructure but about deepening EDULOG's capacity to create social value with data, these data will be available later and can be shown at one company site, EDUSTAT. Ensuring that its platforms remain responsive, scalable, and easy to evolve is thus at the very heart of EDULOG's mission in a fast-evolving digital and policy landscape.

1.1.2 Problem Statement

EDULOG's mission pivots on a dependable, efficient infrastructure that scales to the demands of immense public data on education and employment. Unfortunately, the existing Azure hosting model of operation follows an ETL approach that would bring these limits both operationally and strategically.
Such limitations take different shapes. **Scalability issues** and **non-flexible workflow constructs** confine segmentation of EDULOG from the perspective of advancing its analytical and business requirements. Data processing typically experiences **high waits** owing to the serialization of the current ETL pipelines, as it involves high resource consumption. This architecture also does not facilitate **efficient** and **flexible incorporation** of new features and data streams. Moreover, its system is characterized by **high complexity** since essential items like storage, computation, and analytics are addressed by distinct out-of-loop tools. This fragmentation not only **increases maintenance** overhead but also creates friction across workflows. Resources are scattered, and the cost structure is unpredictable; there are separate licenses for various services that complicate financial planning and resource allocation. The **uncaptured pricing model**, where each component bills the infrastructure, is even more a challenge for effective budgeting, monitoring, and governance.
Accordingly, it matters that EDULOG does not develop its platforms with the new data

needs, but at best, established itself to support continuous innovation. To correct the mismatch, a single interoperable modular and scalable architecture will be necessary that will support operational efficiency-organizational, capability-cost predictability, and advanced capability for Data and Analytics.

1.1.3 Background of B2F, EDULOG, EDUSTAT, and the role of cloud infrastructures

The B2F company focuses on Business Intelligence (BI) solutions, data analyzing and cloud computing infrastructres. With an effort in innovating and deploying cutting-edge technology practices, B2F has been a great partner in providing scalable, flexible and robust solutions suited for all their client needs. One of such clients is EDULOG, an initiative by the Foundation Belmiro de Azevedo, dedicated to data analysis within the areas of Education and Employment. Also functioning as a think tank that collects and processes vast amounts of data towards supporting strategic decision-making of public and private entities aimed at improving the conditions of education in Portugal and worldwide.
The very infrastructure that enables EDULOG to store and analyze, in clear, efficient and systematic manner all their data, and simultaneously assures continuity of operations, scalability and reliability for its functions, is powered by Microsoft Azure and managed by B2F and the name of this infrastructure that lets the user observe and analyze the data is the EDUSTAT. Just like with the advanced cloud environment, this will permit faster integration of new data flows without compromising agility and competitive advantage in the field of education according to EDULOG. Apart from the new cloud advancements regarding technology like Fabric, new functionality integration, process improvement and cost reduction can be realized by EDULOG, on the EDUSTAT, ensuring a future-ready infrastructure.

1.1.4 Importance of the topic in the current context of technological evolution

There is constant progression in the area of cloud computing-with examples as Fabric, which creates a demand for more efficacious infrastructures that can scale and secure as required. Adopting new technologies is something needed for organizations like EDULOG that require massive data for giving strategic insights and ensuring continuity and relevance in their services. This sharpening is not only into a technological upgrade-it is also a strategic response to ever-increasing demands for performance, flexibility, and resource optimization. To analyze and integrate data in real time in a very competitive environment gives an organization a significant competitive advantage; therefore, migrating to a more current platform will allow organizations not only to keep pace with their competition but to lead their industries into the future. This research intends to find and document challenges and opportunities surrounding the phenomenon of technological migration, thereby developing a holistic and practical view of managing complex transitions between complementary cloud platforms. The dissertation seeks to tackle all the migration technicalities EDULOG faces while serving as a larger knowledge base on planning and execution of large-scale technological changes whose retrieval will be of value to other establishments experiencing similar problems. Consequently, the topic is representative of current evolution in technology and adds relevance to the contribution this assignment makes to the management of cloud infrastructures in the future.

1.1.5 Rationale for Migration

The migration to Fabric emerged as a necessity in the light of a confluence of factors, namely: contextual, technological, and organizational. Under the old architecture based on Azure, EDULOG's data platforms had painfully become very complex and widely fragmented, making it more difficult to maintain, scale and evolve as the growing analytical demands.
On the technology side, the need for better use of Microsoft Fabric offered the solution, an all-encompassing platform unifying all basic operations: ingestion, transformation, and visualization happens-without unnecessary jigsaw puzzles of services not communicating with one another. The medallion architecture, orchestration, and easy integration with tools like Power BI provide a cleaner, more automated, and modular method of managing data.
On the organizational side, the migration requires more autonomy, maintainability, and productivity. It empowered B2F and EDULOG by essentially minimizing developer involvement in day-to-day operations through indicator management via a back-office interface alongside guaranteeing constant availability and better scalability. The decision to migrate now was a strategic choice to sort out existing limitations and prepare the platform for future growth, advanced analytics, and stronger alignment between the technical capabilities and analytical teams.

1.2 Objectives and Motivation

This section is divided into two subsections. Subsection "1.2.1 – Objectives" sets the primary and ancillary objectives of the dissertation, i.e., the migration of EDUSTAT from Azure to Fabric, the expected improvement in performance and scalability, the analysis of cost implications, and the design of a methodological model for undertaking similar future projects. Subsection "1.2.2 – Motivation" explains the motivations behind the aspiration to undertake this research, linking EDULOG's and EDUSTAT's institutional needs to the opportunities offered by Fabric, and specifying how such migration forms part of broader trends in technological innovation and data-driven decision-making.

1.2.1 Objectives

The main objective of this research is to migrate the technology infrastructure of EDUSTAT from Azure to Fabric and to carry out the entire transition with minimum operational disruptions. This migration includes adapting existing services, transferring information in a secure manner, as well as implementing new functionalities targeted at continuity of operation and resource optimization.
The other objectives (secondary objectives) are:

- Assess the influence of the migration on general infrastructure performance, including analyzing the improvements regarding scalability and processing efficiency.
- Identify changing operational costs due to the migration and use of Fabric as compared with Azure.
- Present a methodological replication model that could be studied as a reference for future infrastructure migrations, within EDULOG or for other organizations with similar needs.

1.2.2 Motivation

The new technology motivates this study to modernize infrastructures and to derive the advantages and improvements that can be gained by businesses from its application. Fabric seems to be highly useful in driving the optimization of processes and the enhancement of operations; thus, it is needed by every organization to continuously keep up with the developments.

Further motivation for the work comes from the social impact of EDULOG. As an entity that informs and supports the Portuguese population, especially its youth, EDULOG shows young persons possible options in education and careers so that informed decisions can be made about their future. Updating its technological infrastructure would ensure that EDULOG continues to supply current and relevant information on the education system while continuing to consolidate its role within society and assuring the future of education within Portugal.

1.3 Research Method

This section describes the methodological approach used for the dissertation and is divided into four subsections. Subsection "1.3.1 – Methodological Approach" details the approach to research undertaken, combining agile project management with a technical approach based on the Medallion architecture. Subsection "1.3.2 – Project Planning Overview" provides the timeline and task decomposition, plus the planning tools and deliverables that formulated the development of the project. Subsection "1.3.3 – Implementation Structure" discloses how the migration and development activities were organized into environments, processes, and responsibilities to deliver an integrated workflow. Finally, subsection "1.3.4 – Risk Considerations" identifies possible risks that could affect the project and analyzes the strategies given in order to mitigate them, ensuring reliability during the implementation of the dissertation

1.3.1 Methodological Approach

This is an exploratory, applied research-type dissertation, based on the case study of the migration of the EDULOG data infrastructure to the Fabric platform. The practical nature of the project warrants the application of a methodology dedicated to addressing an actual and tangible issue by creating, deploying, and testing a real-world technological solution. The company B2F executed the project on behalf of EDULOG, the company that handles public information about Education and Employment in Portugal. B2F spearheaded the architectural design and technical execution of the migration through software engineering methods and best practices in data architecture to ensure a structured, scalable, and sustainable migration to the new platform.
The solution was designed around a new data architecture in the Medallion model (Bronze, Silver, Gold) and utilized unified tools in the Fabric suite such as Lakehouse, PySpark Notebooks, Orchestration Pipelines, and Power BI Embedded. The emphasis was automation, centralization, and scale of operations pertaining to data ingestion, transformation, storage, and visualization. This strategy made it possible not only to deploy a novel infrastructure, but also to test its technical and organizational effects, making the solution consistent with EDULOG's present and future requirements in the field of management and publication of public data. So the Agile methodology was the project management methodology used,

specifically the Azure DevOps tecnology, and for other way the ETL, specifically the Medallion, was the tecnically methodology used.

1.3.2 Project Planning Overview

The EDULOG migration project structure was designed for simple implementation and alignment with the strategic and operational objectives of the project. The end-to-end process, orchestrated by B2F, was organized according to a hierarchical work breakdown structure (WBS), in Figure 1.1, that detailed the primary tasks and deliverables within each phase of the migration.

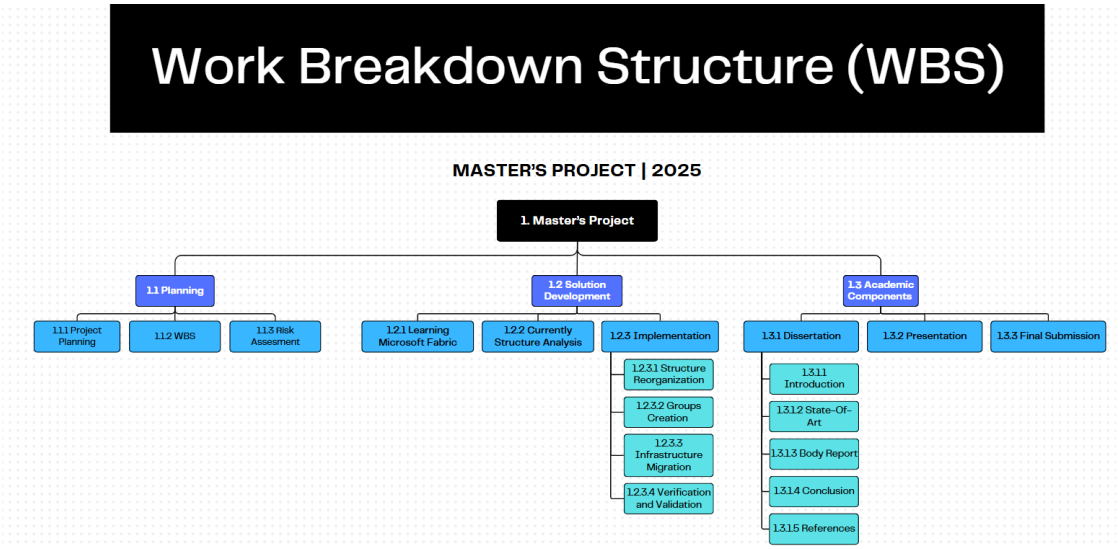
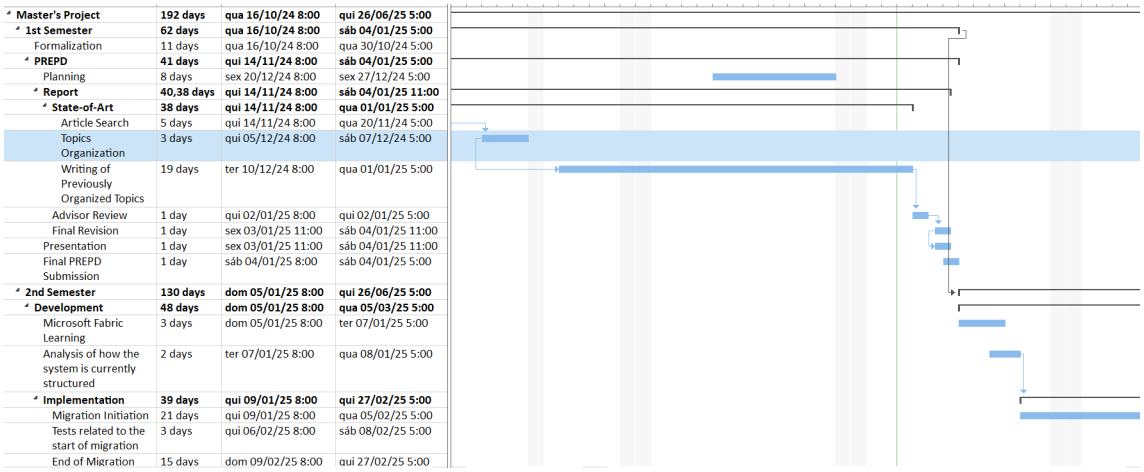


Figure 1.1: Work Breakdown Structure (WBS)

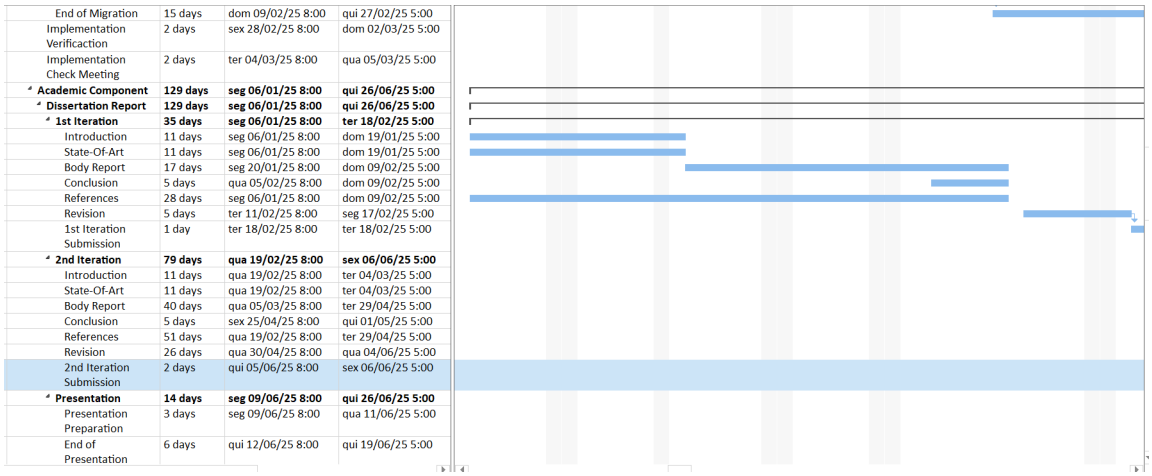
To make the implementation of the Work Breakdown Structure (WBS) easier and allow effective coordination, a detailed project schedule was developed with the help of Gantt diagram illustrations.
The Gantt Chart diagram was divided in 2 figures, the figure 1.2 represents the first part of the project timeline, it includes the planning phase, literature review process, architectural analysis, and the migration performance to Fabric. Each task is scheduled with defined periods of time and interdependencies for transparency over key paths and monitoring the progress across the entire academic year.



Master's Project	**192 days**	**qua 16/10/24 8:00**	**qui 26/06/25 5:00**
1st Semester	**62 days**	**qua 16/10/24 8:00**	**sáb 04/01/25 5:00**
Formalization	11 days	qua 16/10/24 8:00	qua 30/10/24 5:00
PREPD	**41 days**	**qui 14/11/24 8:00**	**sáb 04/01/25 5:00**
Planning	8 days	sex 20/12/24 8:00	sex 27/12/24 5:00
Report	**40,38 days**	**qui 14/11/24 8:00**	**sáb 04/01/25 11:00**
State-of-Art	**38 days**	**qui 14/11/24 8:00**	**qua 01/01/25 5:00**
Article Search	5 days	qui 14/11/24 8:00	qua 20/11/24 5:00
Topics Organization	3 days	qui 05/12/24 8:00	sáb 07/12/24 5:00
Writing of Previously Organized Topics	19 days	ter 10/12/24 8:00	qua 01/01/25 5:00
Advisor Review	1 day	qui 02/01/25 8:00	qui 02/01/25 5:00
Final Revision	1 day	sex 03/01/25 11:00	sáb 04/01/25 11:00
Presentation	1 day	sex 03/01/25 11:00	sáb 04/01/25 11:00
Final PREPD Submission	1 day	sáb 04/01/25 8:00	sáb 04/01/25 5:00
2nd Semester	**130 days**	**dom 05/01/25 8:00**	**qui 26/06/25 5:00**
Development	**48 days**	**dom 05/01/25 8:00**	**qua 05/03/25 5:00**
Microsoft Fabric Learning	3 days	dom 05/01/25 8:00	ter 07/01/25 5:00
Analysis of how the system is currently structured	2 days	ter 07/01/25 8:00	qua 08/01/25 5:00
Implementation	**39 days**	**qui 09/01/25 8:00**	**qui 27/02/25 5:00**
Migration Initiation	21 days	qui 09/01/25 8:00	qua 05/02/25 5:00
Tests related to the start of migration	3 days	qui 06/02/25 8:00	sáb 08/02/25 5:00
End of Migration	15 days	dom 09/02/25 8:00	qui 27/02/25 5:00

Figure 1.2: Gantt Chart Part 1

In the second figure, figure 1.3, we can focus on the academic part of the project, and the process of writing the dissertation is structured in iterations. These include the writing of introduction, state of the art, body report, references, and conclusions. Review cycles and submission milestones are also accommodated in the timeline in order to keep the writing within institutional windows for review.



End of Migration	15 days	dom 09/02/25 8:00	qui 27/02/25 5:00
Implementation Verificaction	2 days	sex 28/02/25 8:00	dom 02/03/25 5:00
Implementation Check Meeting	2 days	ter 04/03/25 8:00	qua 05/03/25 5:00
Academic Component	**129 days**	**seg 06/01/25 8:00**	**qui 26/06/25 5:00**
Dissertation Report	**129 days**	**seg 06/01/25 8:00**	**qui 26/06/25 5:00**
1st Iteration	**35 days**	**seg 06/01/25 8:00**	**ter 18/02/25 5:00**
Introduction	11 days	seg 06/01/25 8:00	dom 19/01/25 5:00
State-Of-Art	11 days	seg 06/01/25 8:00	dom 19/01/25 5:00
Body Report	17 days	seg 20/01/25 8:00	dom 09/02/25 5:00
Conclusion	5 days	qua 05/02/25 8:00	dom 09/02/25 5:00
References	28 days	seg 06/01/25 8:00	dom 09/02/25 5:00
Revision	5 days	ter 11/02/25 8:00	seg 17/02/25 5:00
1st Iteration Submission	1 day	ter 18/02/25 8:00	ter 18/02/25 5:00
2nd Iteration	**79 days**	**qua 19/02/25 8:00**	**sex 06/06/25 5:00**
Introduction	11 days	qua 19/02/25 8:00	ter 04/03/25 5:00
State-Of-Art	11 days	qua 19/02/25 8:00	ter 04/03/25 5:00
Body Report	40 days	qua 05/03/25 8:00	ter 29/04/25 5:00
Conclusion	5 days	sex 25/04/25 8:00	qui 01/05/25 5:00
References	51 days	qua 19/02/25 8:00	ter 29/04/25 5:00
Revision	26 days	qua 30/04/25 8:00	qua 04/06/25 5:00
2nd Iteration Submission	2 days	qui 05/06/25 8:00	sex 06/06/25 5:00
Presentation	**14 days**	**seg 09/06/25 8:00**	**qui 26/06/25 5:00**
Presentation Preparation	3 days	seg 09/06/25 8:00	qua 11/06/25 5:00
End of Presentation	6 days	qui 12/06/25 8:00	qui 19/06/25 5:00

Figure 1.3: Gantt Chart Part 2

Finally, figure 1.4 presents a simplified timetable with the most significant submission dates and iteration windows. It consolidates the most significant scholarly milestones, migration review, submitting the final report, and presentation deadlines, and provides an overview of the overall anticipated route and time constraints of the entire master project.

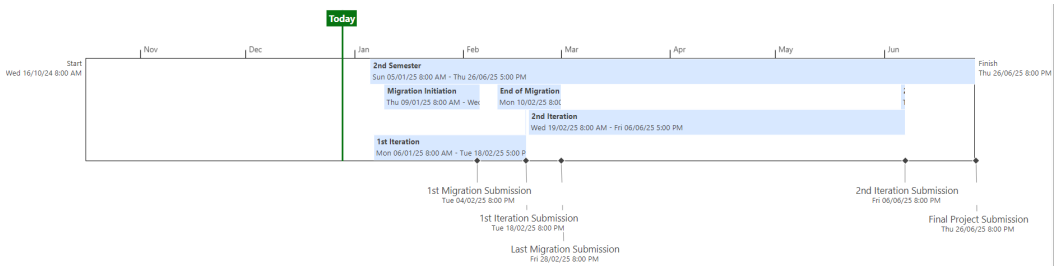


Figure 1.4: Milestone Map

In the overview, the project began with a detailed analysis of the existing Azure-based setup, leading to the definition of the new architecture to be deployed in Fabric. After setting the base architecture, a development environment was built and organized to support the upcoming implementation tasks. This also included the creation of necessary production and development environments (EDU Engineering DEV and EDU Engineering), as well as specifying access and deployment pipelines. The second step was defining ingestion workflows for APIs and file-based data sources. The ingestion workflows were supposed to load the raw data into the Bronze Layer of the architecture. Transformation logic was then utilized to move the data from Bronze to Silver and Gold layers according to the Medallion architecture pattern to have scalable, structured, and reusable data pipelines. After the data layers were implemented, modeling structures were put in place to support the development of dashboards and indicators, the front-end functionalities of EDULOG's data platform. Finally, automation logic and orchestration pipelines were also put in place to manage and orchestrate the execution of the whole ETL process to allow complete automation and integration with EDULOG's backoffice system.
Planning and execution of these phases were done in a modular and logical way, allowing lean and agile development, verification, and deployment of components, and continuous and stable delivery of EDULOG data.

1.3.3 Implementation Structure

EDULOG migration project architecture was planned in phases, where every component of the new architecture could be implemented, validated, and deployed in a modular and scalable manner. Solution was implemented end to end on the Fabric platform leveraging its native data storage, data transformation, orchestration, and visualization abilities.
The design adhered to the Medallion pattern, having three main data layers: Bronze, for raw data storage; Silver, where cleaned and normalized data is stored in delta table format ready for standardization; and Gold, supporting business-level tables and indicators ready for reporting and visualization. The design provided a clean separation of concerns and facilitated reuse of logic across several data sources and indicators. Two isolated work areas were created to support development: EDU Engineering DEV, where all the components were developed and tested; and EDU Engineering, or the production area where the solution implemented was finished. The pieces included ingestion notebooks written in PySpark, orchestration pipelines which process and execute ETL jobs, and a fully structured Lakehouse (EDULAKE) and Warehouse (EDUHOUSE) to store and process data. The deployment followed a pattern of continuous integration and delivery, and CI/CD pipelines to automate the components' transition between environments. The EDULOG's backoffice interface was fully integrated into the solution with indicators' configuration and filtering configured without developers.

Chapter 3, on Solution Analysis, provides a detailed presentation of the architecture, components, and automation logic.

1.3.4 Risks Considerations

In the planning and implementation of the EDULOG migration project, several classes of risk were determined and controlled in an effort to successfully deliver the project. The risks were technical in terms of potential incompatibility between existing Azure components and the new Fabric environment and the challenge in mapping ingestion and transformation operations to the new architecture. These were handled using early-stage prototyping and separated development environments (DEV) before production deployment.
System downtimes or data mismatches due to operational risks during the process of migration were dealt with automation, CI/CD pipelines, and strict orchestration of ETL pipelines. They restricted human error risks to the minimum extent possible and also had traceability in each activity in the lifecycle of the data. Organizationally, the migration created changes in team roles and workflows, namely concerning the backoffice's increased autonomy in processing indicators. To balance these changes, the solution was introduced with configuration tools that are simple to use and complete documentation to reduce developer dependency and encourage easier uptake.
Throughout, the project adopted a risk-aware approach, focusing on modular development, testing in a safe environment, and automation to deliver a smooth and stable transition.

1.4 Research Questions

Research Questions (RQ) specify the investigation and make certain the study meets its objectives. This section presents the primary questions that guide the analysis of the migration from Azure to Fabric regarding technical, operational, and strategic aspects. The inquiry aims to gather best practices during the investigation concerning the benefits of Fabric, thus formulating a better understanding of cloud migration processes.
The **RQ** are:

1. What are the main challenges and best practices in migrating cloud infrastructures between platforms such as Microsoft Azure and Microsoft Fabric?
2. What are the technical and operational implications of migrating data from Microsoft Azure to Microsoft Fabric?
3. What migration frameworks or methodologies are recommended for transitioning from Microsoft Azure to Microsoft Fabric?
4. What are the performance benefits of Microsoft Fabric compared to Microsoft Azure in large-scale data processing and workload management?

1.5 Ethical Concern

There are certain ethical concerns involved in the migration of EDULOG's cloud infrastructure from Microsoft Azure to Microsoft Fabric. This section describes some of the challenges that arise in this context and the various strategies for dealing with those challenges:

- **Data Privacy and Security:** Such an activity or action is of utmost importance for keeping confidential sensitive educational and employment information as it were the

cases. Various measures have been taken and developed to ensure the confidentiality of sensitive educational and employment records such as encryption, anonymization, and access control. The whole process of moving will be in accordance with standards such as the General Data Protection Requirement (GDPR) to ensure that all relevant DATA are safely and responsibly handled.

- **Operational Continuity:** Minimizing interference with EDULOG's platforms during the migration period is an ethical priority. Critical insights from the organization reach students, educators, and policymakers. Therefore, even a few hours of downtime or a service hold can lead to big damage to these stakeholders. Therefore, plans have been made to ensure continuity so as not to lose the trust of the users.
- **Transparency in Decision-Making:** The choice of Fabric and the different methods regarding the migration were done at a relatively upfront level with emphasis on strong arguments that are in evidence and based on organizational needs. This kind of approach ensures accountability and trust with stakeholders.
- **Data Governance and Usage:** The data is used strictly for its defined purposes under stringent governance policies. It is also a key priority to prevent misuse or unauthorized access, for which very strong monitoring systems are in place.
- **Social Responsibility:** This is intended for migration, so that EDULOG can strengthen the level of educational insights it can offer students and the society at large. It is, however, clearly under a moral obligation to use technology not just for its own sake but more so to the benefit of humanity-things like empowering people through better access to information.

The project will be ethically grounded in the above ethical concerns, ensuring responsible migration with security, fairness, and benefit to society, meeting EDULOG's mission to serve the community effectively.
Furthermore, Grammarly was used, but only for grammatical correction and punctuation, when the latter was non-existent.

1.6 Contribution

In the field of cloud computing and the methodologies that accompany migration, this study will make significant contributions to both practice and academic, but will focus on the migration of EDULOG's infrastructure from Azure to Fabric. One of the main contributions to this field would be a supergeneralized framework for cloud migration. The framework was developed to address important facets of the migration process like minimizing operational downtime, ensuring data integrity, and optimizing resource utilization. It is thus a replicable and adaptable model for organizations facing similar challenges, providing the approach to be taken to planning, execution, and validation of cloud migration. Beyond the framework, it evaluates Fabric most thoroughly, without leaving out the important new features benefits over it when compared with Azure. This entails an analysis of the scope of Fabric, among many features, integrated data analytics, and area-capable management. Therein, has the possibility to learn how organizations can harness its architecture to revamp business operations and improve efficiency. At the core of the study is a comprehensive appraisal of the two platforms and provides the decision-makers with the necessary information on the benefits of potential derivatives with the implementation of Fabric.
The entire research provides a practical demonstration in the real-life situation where the

EDULOG infrastructure upgrade is being used as a progress report on the problems faced during the upgrade, the remedies applied and the result achieved. This deployment of the framework also serves as a good example for demonstrating its viability and effectiveness and hence, a very significant reference for such transitions. In addition, the study directly improves the technology capabilities of EDULOG. It has upgraded thus EDULOG today can now also process and analyze bigger amounts of data to enhance flexibility, continuity in operations, and productivity. Such an advanced facility strengthens the fulfilment of the mission where critical insights and accessible educational data are provided to student support, educator and policy developer. Apart from its immediate effect in EDULOG, its outcome has wider importance to the education sector within Portugal. The future-ready infrastructure that EDULOG has makes this study indirect in contribution to society's development, to make such an organization still the leading resource on educational guidance and policymaking. This research further closes the gap between theory and practice in a manner that combines academic robustness with actual realization, ensuring that the findings are rigorous and methodologically sound while also being applicable in the real world.
In sum, this study advances cloud migration with well-articulated guidance on how to negotiate the nuanced realms of migrating platforms. More important, this insight and methodology will excite practitioners and scholars, thereby making this work yet another major step in making contributions towards the continually transformative cloud computing technologies.

1.7 Document Overwiew

This dissertation is structured into seven major chapters, with each chapter addressing a specific stage of the research, design, and development of the Microsoft Fabric–based analytical architecture for EDULOG, with particular emphasis on the EDUSTAT platform. Chapter 1 introduces the background, motivation, objectives, scope, and methodology of the project and concludes with the structure of the organization of the document. Chapter 2 presents the state of the art, compiling the main concepts, technologies, and methodologies relevant to the task, such as cloud platforms, Medallion architecture, and comparative analysis of Microsoft Azure and Microsoft Fabric. Chapter 3 focuses on the analysis of the present infrastructure and functional and non-functional requirements. Chapter 4 outlines the solution design, in which the architectural planning and the design decisions are described, such as technical and storage architecture, model logic, definition and indicator tables, and methodological approaches used. It also includes C4 diagrams and logical views to describe the solution proposed. Chapter 5 describes the execution of the solution in the EDU Engineering DEV environment, such as storage and warehousing, data transformation and extraction, orchestration, and process automation and is accompanied by screenshots and step-by-step walkthroughs of the processes and notebooks executed. Chapter 6 discusses the testing and cost analysis, comparing the functionality and prices of Azure and Fabric with empirical testing and official prices, against the significant advances achieved using the new platform. The dissertation ends by summarizing the key findings and contributions, describing the advantages of Microsoft Fabric for EDULOG and EDUSTAT, and delineating the preparedness of the platform for scalability and innovation in the future.

Chapter 2

State-of-the-Art

In recent years, sheer growth in the volume and complexity of data has encouraged organizations to look for modern, scalable, and agile cloud-based data architectures. These architectures must enable not only robust data storage and processing, but also efficient modeling, visualization, and governance, especially on public data platforms where transparency, autonomy, and reliability are essential.
This chapter provides information about the state-of-the-art technologies, methodologies, and architectural patterns on which the solution presented in this dissertation is based. It presents relevant literature on cloud platforms like Azure and Fabric and elaborates on data architecture patterns such as Data Lakes, Data Warehouse (DW), and the emerging Lakehouse pattern. Special attention is given to the Medallion architecture, a tiered data processing model (Bronze, Silver, Gold) taking a central role in the system under consideration, and emerging trends in cloud migration and indicator modeling. Furthermore, this chapter emphasizes the critical drawbacks and issues identified in modern solutions, placing the proposed solution in broader technological and research contexts.
By critical evaluation of what has been conceptually dealt with and proposed, the chapter constructs the theoretical foundation that justifies decisions in architecture made in ensuing chapters, namely the design and implementation of the EDU Fabric platform.

2.1 Cloud Platforms for Modern Data Architectures

Cloud platforms are now central to data-driven organisations within this new age, with the capabilities, scalability, and elasticity required to consume, process, and analyse vast amounts of structured and unstructured data. Of the available options, Azure and the newly announced Fabric are two market-leading platforms varying in paradigm architecture and features.
Azure has come into wide application because of its componentized nature, such as Azure Data Lake, Synapse Analytics, and Azure SQL Database. It supports organizations to build scalable and secure data solutions through the combination of independent pieces for ingestion (e.g., Data Factory), transformation (e.g., Synapse pipelines), and visualization (e.g., Power BI) [Fehling et al. 2013] [Alouffi et al. 2021]. However, this componentized approach usually introduces architectural complexity, requires a lot of orchestration effort, and might introduce difficulty in data consistency maintenance among distributed services.
On the other hand, Fabric emerges as a unified platform that integrates each layer of the data stack - from ingestion and storage to modeling and visualization — into a single environment. With its first-class support for Lakehouse pattern, Delta tables, Medallion architecture, and Power BI Embedded, it allows seamless movement between data engineering and business intelligence workflows. The literature suggests that Fabric addresses some of the most

significant limitations observed in pipelines based on Azure, such as the need to manually manage multiple environments or replicate logic between disparate components [Lopes 2023]. Further, Fabric leverages consumption-based architecture with centralized compute, which provides better cost control and performance monitoring compared to Azure's traditional architectures. Its simplicity, automation, and data product governance are features that are easy to resist for organizations undergoing digital modernization or cloud migration. In the context of this dissertation, the migration decision from a conventional setup based on Azure to Fabric is motivated by the need to reduce operational complexity, improve governance and facilitate a more flexible and self-managed data modeling landscape.

2.2 Data Storage and Processing Models

Modern data structures can be based on multiple model of system storage and computation, each designed to satisfy specific organizational needs and technical constraints. The system's top three paradigms are the DW, the Data Lake, and the Lakehouse, each of them with different trade-offs between system structure, scale, and flexibility [Fehling et al. 2013]. The DW design is used for structured data and analytical SQL queries. Schema-on-write regulations are enforced in it, i.e., the data must obey pre-defined schema before storage. The design includes strong consistency, indexing, as well as report and business intelligence operations performance. However, its rigidity can limit its agility and scalability in dealing with semi-structured or unstructured data [Jamshidi, Ahmad, and Pahl 2013]. In contrast, Data Lake seeks to ingest and store very large volumes of raw data in a wide variety of formats — structured, semi-structured, and unstructured — with schema-on-read logic. Although this pattern is cost-effective and flexible in storage, it presents data governance, quality, and performance issues for analytical workloads [Sabiri et al. 2015]. The Lakehouse architecture brings the good from both worlds and combines the flexibility of Data Lakes with the reliability and performance of DW. It enables schema enforcement, ACID transactions, and time travel by utilizing technologies such as Delta Lake [Lopes 2023]. It's a one stop source of truth for production-quality analytics and exploratory processing by data engineers and analysts.
In Fabric, this is done by having the pattern applied to the OneLake storage layer that stores data in a single system of storage with direct querying enabled through SQL or Spark engines. Fabric ability to support Delta tables guarantees transactional consistency, scalability, and compatibility with medallion-based data pipelines (Bronze, Silver, Gold) [Lopes 2023]. Within the context of the solution outlined, the blending of the Lakehouse paradigm for data processing (EDULAKE) and an organized DW layer (EDUHOUSE) is also supportive of flexibility as well as governance. While EDULAKE performs ingestion and transformation in open formats, EDUHOUSE is a curated, business-focused repository that supports stable and high-performant access to indicators and reports.

2.3 Medallion Architecture (Bronze, Silver, Gold)

The Medallion architecture is a layered model of data organization and processing in a scalable and sustainable manner. It has broad applicability in current data platforms to organize data transformations into incremental steps, each delivering increasing amounts of quality, refinement, and business value. The model is especially suited to Lakehouse architectures and is fully supported in Fabric environments [Lopes 2023].
The architecture is generally outlined into three layers. The Bronze Layer is designed for

capturing raw ingested data from various kinds of sources, which may include files, APIs, or databases. It basically stores a given original record in its unaltered form and serves as an immutable primary-source record for downstream processing, ensuring traceability and reprocessing if needed. The Silver Layer is responsible for transforming and cleaning the data. At this point, data is normalized and standardized onto a common tabular form-finally into either business or platform standards. For EDU Fabric, this means standardizing all inputs (e.g., Excel files, public APIs) into a four-column format: "Dataset", "Valor", "Filtro", and "Significado". This uniformity increases consistency, which in turn allows automation and separates application logic from source-specific data formats. And lastly, the Gold Layer represents the business-side modeling of outputs, which involves indicators, dashboards, and other materialized tables optimized for analytics and reporting. In EDU Fabric, this features the automated generation of Indicator Tables based on user-configurable logic-driven Definition Tables held in the platform's metadata layer.
By structuring the architecture in this way, organizations gain several advantages. Data is reusable and modular, with each level having a specific purpose and ownership. It also enhances governance and debugging since issues can be traced at the correct place in the flow. In addition, segregation of the ingestion, transformation and modeling steps enables parallel development, enhanced performance management, and greater flexibility in adding new datasets or indicators. These multilayered strategies will be tightly suited to the needs of public platforms like EDULOG, where data needs to be traceable, rule-governed, and dynamically adjustable to the needs of changing policy and analyses.

2.4 Cloud Migration Strategies

The evolution of data platforms has prompted organizations to rethink their foundations of architecture, particularly as the limitations of old systems become progressively incompatible with demands for real-time processing, governance, and scalability. Cloud migration, when done strategically, presents a path toward more resilient, modular, and integrated solutions. But the success of such migrations depends not just on technical readiness, but also on the selection of an appropriate migration strategy that balances agility, risk, and long-term architectural goals.

2.4.1 Migration Approaches: From Lift-and-Shift to Refactoring

Migration strategies tend to be positioned on a continuum, from low-effort rehosting to full architectural redesign. It is at one end that the "lift-and-shift" option lies, with the activity of replicating systems in place to the cloud without changing their internal design or logic. While simple and convenient, this strategy has the disadvantage of leaving behind legacy inefficiencies and failing to exploit the optimacies of cloud-native services. Replatforming addresses this by offering a halfway house solution that optimizes to a limited extent while in transit — e.g., replacing a virtual machine with a managed service — but leaves the underlying architecture of the original solution in place.
More extreme is the refactoring approach, which restructures systems to take full advantage of native features of the cloud. It involves re-architecting services for increased modularity, automating processes, decoupling tightly coupled components, and moving to scalable storage and processing models. Although refactoring involves more effort and resources, it is usually the optimum for achieving long-term efficiency, maintainability, and flexibility. In practice, mass migrations are hybrid in nature, selectively applying these approaches based

on the criticality and technical maturity of specific components of the system [Fehling et al. 2013] [Jamshidi, Ahmad, and Pahl 2013].

2.4.2 Architecture-Driven Modernization (ADM)

One of the most widely accepted approaches to cloud migration - especially in public sector and regulated environments - is Architecture-Driven Modernization (ADM). In contrast to recommending broad system replacement, ADM is about gradual change under architectural analysis guidance. ADM entails the determination of reusable legacy assets, compartmentalization of system boundaries, and phased introduction of new technology without compromising core functions. It enables organizations to modernize at a sustainable rate, reducing risk, minimally disturbing users, and maintaining compatibility along the way.
ADM is perfectly placed in the conditions of cost constraint, regulation, or dependencies that make the full reimplementation not feasible. With systems being segmented into modular pieces and interfaces specified for old and new pieces, organizations are able to innovate without jeopardizing mission-critical processes. ADM also finds its place in agile governance and continuous improvement to build conditions of scalable modernization that respects institutional continuity [Sabiri et al. 2015].

2.4.3 Best Practices in Cloud Migration

Successful migration programs also depend on the deployment of best practices offering quality, continuity, and robustness during the transition. Phased implementation is one of the most important principles where migration is performed in clearly defined phases. This allows teams to constantly check results and react to issues without undermining the overall system. Hybrid coexistence of new and old platforms is also common, particularly when legacy components must remain active while adding new infrastructure.
A stable test environment — usually a sandbox or staging level — is required to verify configurations, compatibility, and data quality before production deployment. Also important is the use of CI/CD pipelines that automate integration, testing, deployment, and rollback. These enable faster iteration speed and reduce the risk of human error, especially in complex analytical systems where data lineage and transformation accuracy are key.
By applying these principles, organizations build migration flows that are not only technically robust but also organizationally sustainable, and innovation is attained without instability.

2.4.4 Migration Strategy in EDU Fabric

The migration strategy adopted for EDU Fabric is one of deliberate departure from the limitations of an Azure-based architecture to a single and manageable data platform on Microsoft Fabric. Rather than adopting a simple lift-and-shift replication of existing components, the approach was founded on the principles of Architecture-Driven Modernization. Some of the established components of the legacy system - such as proven data sets and existing metadata — were retained and integrated into the new structure, while others, particularly related to modeling logic and data orchestration, were fully re-factored.
The resulting platform leverages Fabric support for Lakehouse storage (via OneLake and Delta tables), uses the Medallion architecture to separate ingestion, transformation, and modeling layers, and introduces SQL-based automation to enable automated indicator creation. Continuous integration and delivery pipelines control content movement between development and production environments, enabling repeatable and traceable deployments.

The migration was done incrementally, so platform capabilities were always accessible to end users during the migration, while at the same time bringing new, contemporary capabilities to supplant formerly manual or disconnected activity.
This iterative and hybrid approach enabled the EDU Fabric platform to evolve from a disjointed, manually orchestrated setup to an organized, scalable, and automated data solution, according to best practices in public data management and cloud modernization.

2.5 Indicator Modeling and Automation

In traditional BI systems, performance measures and metrics are strongly and statically coupled as indicators. The calculation logic for each indicator - time intervals, filters, aggregations, and dimensional drill-downs - is typically hardcoded into SQL queries, dashboards, or ETL programs. Although such an approach may suffice in simple or stable situations, it is progressively unviable in dynamic public sector settings, where indicators must be constantly re-calibrated to keep up with shifting policies, data sources, and stakeholder needs. The hardcoded indicator logic has several drawbacks. Each change usually involves the technical teams and, therefore, update cycles are slow, and minor business changes depend on developers. In addition, the repetition of logic between layers-for instance, transformation scripts and visualization dashboards- leads to inconsistency, lack of transparency, and auditing or validation of outputs, which is especially problematic in public-facing data platforms where traceability and trust are crucial.
To overcome these shortcomings, a new paradigm has emerged that works on parametrized and automated indicator modeling. By decoupling indicator setup from deployment by shifting business logic into definition layers that are metadata-stored and managed through user-exposed backoffice interfaces, the platform reads parameters set in definition tables instead of hardcoding filters and calculations and dynamically builds indicators. These can specify dataset to be drawn upon, aggregation type to be employed, filters available (e.g., region, gender, age band), and inclusion of derived fields. In systems like EDU Fabric, this strategy enables business users to define or revise indicators without requiring ETL pipeline or SQL code modifications. Once configured, such definitions are read automatically by transformation engines that generate equivalent indicator tables and views, which reporting layers consume. Such modular design enables reusability, consistency, and fast iteration, in addition to reducing developer effort and platform maintenance. The employment of dynamic indicator modeling also enables the platform to manage schema change — say, the addition of new columns — with no system downtime. When a new attribute is discovered in the ingestion layer, it gets automatically added to the metadata constructs (e.g., filter dimensions) and is available for use in indicator configuration. This facilitates long-term evolution of the platform so that it can expand without much friction.
These kinds of automated model approaches are particularly well-suited to public administration contexts, where policy responsiveness, transparency of data, and non-technical autonomy are esteemed. By enabling analysts and subject-matter experts to define and maintain indicators directly, the platform maximizes responsiveness without compromising governance, repeatability, and auditability.

2.6 Gaps Identified and Contribution Opportunity

Despite the evolution of today's cloud platforms and data modeling best practices, there are a few areas of incompatibility in the actual implementation of such technologies, particularly

for public-sector analytical systems. Most of the existing solutions excessively employ fragmented architecture, custom ETL scripts, and hard-coded business rules. Such constraints limit flexibility, increase maintenance costs, and diminish the autonomy of non-technical users who are dependent on rapid and assured access to updated indicators.
One of the major limitations that are commonly encountered with Azure-based architectures is that there is no integration of data services. In the previous configurations, ingestion, transformation, modeling, and visualization are separated across separate tools — such as Data Factory, Synapse, SQL Server, and Power BI — with a degree of integration. It has a tendency to build complex deployments, duplicated logic, and governance problems whenever systems must scale or modify at rapid rates. Despite the evolution of contemporary cloud platforms and data modeling best practices, there exist certain incompatibilities in the real-world application of such technologies, particularly for public sector analytical systems. Most existing solutions excessively rely on fragmented architecture, custom ETL scripts, and hard-coded business rules. These aspects limit flexibility, increase maintenance costs, and diminish the autonomy of non-technical users who depend on rapid and assured access to updated indicators. Furthermore, very few deployments offer schema evolution transparently and automatically. In extremely dynamic data contexts — i.e., where education and government statistics are embodied — new columns or structural changes must be added with the minimum level of friction. However, systems are generally incapable of ingesting these changes without perturbation or reconstruction.
Literature addresses a lot of such concerns at a theoretical level but lacks sufficient example cases of platforms on which Lakehouse storage, Medallion architecture, SQL-based modeling, and dynamic configuration of indicators get bundled under one model of governance. Solutions usually take the form of cloud migration or modeling logic, but very rarely both simultaneously.
The present dissertation proposes EDU Fabric as an architectural solution to these shortcomings. By combining Microsoft Fabric's unified platform with modular and metadata-driven modeling, the solution addresses both operational and analytical deficiencies. It offers an automated, scalable, and sustainable indicator system for the public sector — enabling flexibility without compromise on control, and empowering functional users without sacrifice of governance. In so doing, this project delivers not only a tangible implementation, but also a reproducible pattern that can serve as a foundation for similar modernizations in other areas of public data publication.
The literature review gives an overview of currently existing researches and methodology in cloud migration, such as transition from traditional platforms like Azure to state-of-the-art solutions, such as Fabric. It provides an examination of vital challenges like scalability, security, operational continuity, and data integrity and offers examples of best practices and frameworks for successful migration. This section collects knowledge from systematic reviews and case studies and identifies gaps in current research knowledge while laying the foundation for advancing cloud migration strategies through this research study.

2.7 Research Questions

The purpose of the Research Question (RQ) in this study is to reach to primary and secondary objectives as outlined earlier. These objectives include a successfully migrated EDULOG technology infrastructure from Azure to Fabric with minimum disruption, with continuity in operations, secure data transfer, and resource optimization. This research will also assess what the effects of migration are on performance and operational costs and scalability,

as well as propose a methodological framework for replication by other organizations in situations such as these. Each of the RQ is tailored to specific aspects of the migration process, from understanding the adversities to examining the benefits of Fabric. Collectively, they drive the study's attempts to generate practical insights into cloud migration and contribute to the larger corpus. Below is a summary of the RQ, in the table 2.1:

Table 2.1: Research Questions Table

ID	Research Question
RQ1	What are the main challenges and best practices in migrating cloud infrastructures between platforms such as Microsoft Azure and Microsoft Fabric?
RQ2	What are the technical and operational implications of migrating data from Microsoft Azure to Microsoft Fabric?
RQ3	What migration frameworks or methodologies are recommended for transitioning from Microsoft Azure to Microsoft Fabric?
RQ4	What are the performance benefits of Microsoft Fabric compared to Microsoft Azure in large-scale data processing and workload management?

2.7.1 RQ1: What are the main challenges and best practices in migrating cloud infrastructures between platforms such as Microsoft Azure and Microsoft Fabric?

The **RQ1** has been set to identify the critical barriers the organizations face in moving to the cloud, the service compatibility, data security, operational downtimes, and other hurdles. It will also address best practices such as phased migration, testing strategies, and stakeholder communication that critical organization stakeholders may use to ensure a smooth transition. This question is important in terms of providing relevant practical insights into risk minimization and challenge mitigation during the process.

2.7.2 RQ2: What are the technical and operational implications of migrating data from Microsoft Azure to Microsoft Fabric?

The **RQ2** is concerned with the technical complexity of migrating data-from maintaining its integrity, reconfiguration, and appropriate secure transfer mechanism, to examining operational considerations such as changes to the workflow or service disruptions that affect the migration in line with EDULOG operational objectives. It understands these implications, making planning efficient migration with limited negative effects possible.

2.7.3 RQ3: What migration frameworks or methodologies are recommended for transitioning from Microsoft Azure to Microsoft Fabric?

The **RQ3** seeks to discover well-established frameworks and methodologies that can be adapted to suit the needs of this migration, such as iterative migration models or hybrid forms. Through identifying effective strategies, it seeks to devise a repeatable framework

supporting systematic migrations with some of the most important challenges being addressed, not just for EDULOG but for other organizations requiring similar assistance.

2.7.4 RQ4: What are the performance benefits of Microsoft Fabric compared to Microsoft Azure in large-scale data processing and workload management?

The **RQ4** concerns how Fabric provides improved operational efficiencies in data processing, workload distribution, and scalability as opposed to Azure. By analyzing those performance metrics, this study avails a detailed appraisal of the advantages of adopting Fabric thereby giving a solid case for migration and its consequent long-term benefits to EDULOG.

2.8 Search Metodology

The procedure used to identify, collect and analyze relevant literature and data sources for this study is explained in this section. Methodologies were robust and systematic in searching for relevant high-quality research material covering relevant topics in line with RQ and objectives of this project. The methodology based on research uses academic databases such as IEEE Xplore and Google Scholar, as well as b-on, for the analyze of peer-reviewed articles, case studies, and systematic reviews. Specific keywords were designed to correspond to the main subjects under research by this research, such as cloud migration, Azure, Fabric, and best practices in infrastructure transitions. Following a well-structured search strategy, this section serves to safeguard the credibility and pertinency of the materials under analysis, thereby forming the bedrock for what are essentially the theoretical and practical contributions of the study to knowledge.
Following the research questions outlined above and Data sources used, specific queries were framed that would facilitate retrieval of relevant and high-quality literature. These queries were framed within the key themes of the study, cloud migration, Microsoft Azure, Microsoft Fabric, and methodology. In the table 2.2 are the queries for all data sources:

Table 2.2: Research Queries Table

Data Source	Research Query
IEEE	("cloud migration" OR "migration framework") AND ("cloud computing security" OR "live migration" OR "architectural options") AND ("systematic review" OR "case studies") AND ("Microsoft Azure" OR "Microsoft Fabric")
B-on	("cloud infrastructure migration" OR "cloud migration framework") AND ("Microsoft Azure" OR "Microsoft Fabric") AND "case studies"
Google Scholar	"cloud migration" AND ("Microsoft Azure" OR "Microsoft Fabric") AND "technical challenges" AND "operational implications" AND "best practices"

2.8.1 Inclusion Criteria

In this regard, each of the databases related to this study had specific inclusion criteria applied to select the literature that was considered relevant and of high quality. Selection was restricted to only academic journals and peer-reviewed articles so that the sources could be substantiated for their credibility and scholarly rigor. In addition, the search period was limited to publications between 2020 and the present to align with the latest advancements and trends in cloud migration technologies and methodologies. This condition ensures that the study covers information very much to date, especially on novelties such as Fabric and new cloud migration frameworks. Thus, through a consistent application of these inclusion criteria in all sources of data, the study ensures their relevance, quality, and timeliness for the analysis of materials. For this reason, the inclusion criteria used are:

- **Academic Quality:** Only peer-reviewed articles, academic journals, and scholarly publications were selected to ensure credibility and rigor.
- **Time Frame:** The use of cloud migration technologies and strategies from 2020 up to the present to find recent innovations and recent advances in cloud migration techniques.

2.8.2 Exclusion Criteria

Once a study has established its inclusion criteria, it must also define exclusion criteria to help narrow the focus of the literature review and streamline the research process. Exclusion criteria include the sources that lack correspondence with the objectives of a study so that the selected materials remain relevant and highly qualitative. The systematic application of exclusion criteria ensures that the study stays focused on the research questions while maintaining academic integrity.
The exclusion criteria applied in this study include:

- **Non-English Publications:** In order to make sure that the articles can be understood and accessible, non-English articles were excluded since the research in itself is in English and the analysis is done in English.
- **Lack of Relevance to the Topic:** Those articles that didn't relate directly to cloud infrastructure migration on Azure, Fabric, or any such frameworks were excluded. This way, literature can be aligned with the appropriate research objectives and questions.
- **Non-Cloud-Specific Content:** There were also other articles that majorly revolved around technology but not to the extent of cloud computing and infrastructure migration or methodologies since relevance was part of the different research questions.

2.8.3 Data Research

After applying the queries to each Data source and implementing the inclusion criteria, the resulting data is summarized in the table 2.3.

After the preliminary research process, the exclusion criteria were implemented and applied to the further searches. This led to a specific number of documents excluded based upon each criterion. This further ensured that the final literature that was selected was explicitly relevant to the objective of this study. In the table 2.4 follows the number of documents excluded per criterion.

Table 2.3: Research Result Table

Data Source	Number of Documents Results
IEEE Xplore	22 documents
B-on	1 document
Google Scholar	2 documents

Table 2.4: Research Exclusion Table

Exclusion Criteria	Number of Documents Excluded
Non-English Publications	1 documents affected
Lack of Relevance to the Topic	7 document affected
Non-Cloud-Specific Content	6 documents affected

PRISMA Flow Diagram

The searching process, as the figure 2.1 involved conducting a thorough search through the main three databases: IEEE Xplore, B-on and Google Scholar. This initial search resulted in the generation of 25 records. After the duplicates were excluded, the remaining articles were assessed for relevance or specificity according to the inclusion and exclusion criteria that had been previously defined. As presented in the PRISMA diagram, other exclusion criteria included among others,"Non-English Publications" (1 document), "Lack of Relevance to the Topic" (12 documents), "Non-Cloud Specific Content" (6 documents). After such a rigorous filtering process, the final outcome was 6 eligible records which were found appropriate for being included in the study.

2.9 Results of Research Questions

This part discusses the outcomes of the research questions stated earlier, along with the contribution of the reviewed literature to these results. Each of these questions will be systematically addressed below:

2.9.1 RQ1: What are the main challenges and best practices in migrating cloud infrastructures between platforms such as Microsoft Azure and Microsoft Fabric?

The movement of cloud infrastructures from one platform to another, typically from Azure to Fabric, entails many hurdles. This is important to note as one of these challenges is dependency on older systems that have been core business integrated processes. Legacy systems tend to block quick migrations and require architectural modifications to adopt the

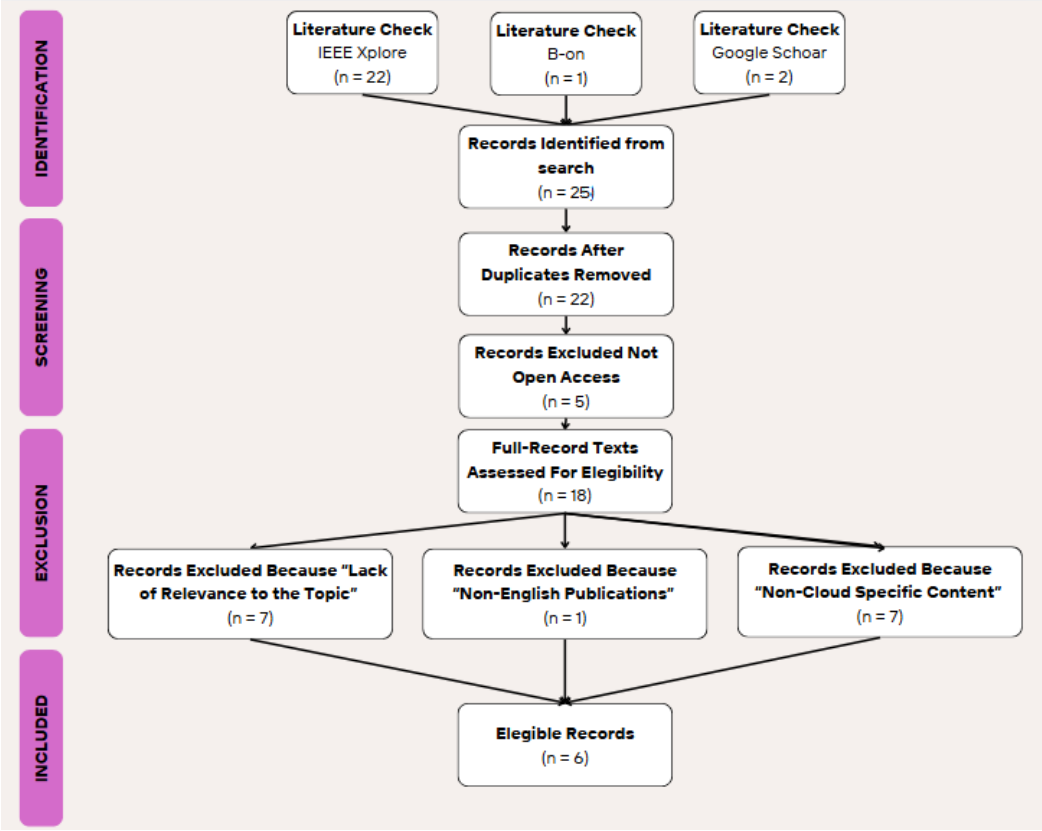


Figure 2.1: PRISMA Flow Diagram

new platform fully. Jamshidi et al. (2013) say such dependencies cause delays and cost overruns for compatibility modifications [Jamshidi, Ahmad, and Pahl 2013]. The other key challenge is the management of multi-clouds such that they become interoperable. Typically, switching from one platform to another requires resolving any data compatibility issues while maintaining a uniform level of services Fehling et al., (2013). These authors recognize this concern as being among the most crucial barriers in integrating cloud applications with existing systems during migration [Fehling et al. 2013]. Migration, especially that of sensitive data, creates security and compliance risks. In this regard, Alouffi et al. (2021) reported that secure data transfer is ensured via encryption and strong authentication measures that would minimize such risks and comply with laws like GDPR [Alouffi et al. 2021]. Cost overruns and resource management are yet other common challenges to migration activities. Migrations are often found to have precise funding allocation due to the unforeseen complexities usually associated with legacy applications refactoring. Effective resource allocation and planning are thus essential in preventing cost overrun and time delays (Sabiri et al., 2015) [Sabiri et al. 2015].

Some best practices aimed at addressing these challenges include carrying out a very thorough pre-migration assessment for evaluating the current infrastructure and identifying any potential dependencies and bottlenecks. According to Fehling et al. (2013), this property is quite important when planning for an effective migration strategy [Fehling et al. 2013]. Gradual migration techniques such as migrating the non-critical systems first help to reduce risk and to allow for iterative test and fine tuning. According to Jamshidi et al. (2013), hybrid approaches are proposed to lower the impact of large-scale migration [Jamshidi, Ahmad, and Pahl 2013].

Those systematic and reproducible migration processes can be achieved through structured frameworks such as Architecture Driven Modernization (ADM). According to Sabiri et al. (2015), this approach is efficient in adapting platform-independent models to platform-specific requirements [Sabiri et al. 2015]. Also necessary has testing into more extreme sandbox settings. According to Thoutam(2024) demonstrating how testing is done before launching it entirely can ensure its performance to resolve all remaining issues [Thoutam 2024]. In the end, therefore, comprehensive data governance systems; access controls; audit trails will serve as bedrock pillars for continuously enforcing synergetic integrity and security of data throughout the migration exercises. According to Alouffi et al. (2021), strict protocols need to be applied to assure quality in the data during migration [Alouffi et al. 2021].
Among the above challenges with the best practices mentioned in this paper, it would seamlessly take organizations like EDULOG into the Fabric environment without causing many operational interruptions and maximizing long-term benefits.

2.9.2 RQ2: What are the technical and operational implications of migrating data from Microsoft Azure to Microsoft Fabric?

There is a significant migration of information from Azure to Fabric that implies several changes in technical and operational dimensions, and the changes that is needed to take into account before undertaking the transition to ensure that it is seamless and successful. Among the major dynamic change challenges on the technical side is the architectural adaptation, which requires total architectural modification to fit into the integrated analytics ecosystem designed to leverage Fabric's high-end features. According to Fehling et al. (2013), the reconstruction of services and workflows necessary for compatibility both at the source and target platforms, which typically transforms legacy architectures into microservices, is also essential for scalable and resource-efficient architecture [Fehling et al. 2013].
Another important technical aspect of migration is data consistency. As pointed out by Jamshidi et al. (2013), the integrity issues can arise in data transfer and can disrupt operations, furthermore, the quality of data may be compromised. To preserve data integrity during processing and use, data replication and real-time synchronization techniques are proposed to manage these two risks in particular, handling transactional data [Jamshidi, Ahmad, and Pahl 2013]. Moreover, the inclusive capabilities of Fabric, for example, using data pipelines and real-time analytics, may require redesigning existing data models, as indicated by Thoutam (2024), who suggests adjusting traditional ETL workflows to match the Medallion architecture of Fabric that provides a better structure of data and processing [Thoutam 2024]. Most importantly, the data should be transferred under the very secure protocols. Alouffi et al. (2021) emphasize that comprehensive encryption schemes and authentication mechanisms should be implemented to protect sensitive information and to guarantee compliance with the security standard [Alouffi et al. 2021].
Operation consequences also assume a significant part in migration. A minimum downtime is necessary for business continuity. Sabiri et al. (2015) indicated a phased migration that could enable critical systems to remain operational and permit other components to change gradually, thus reducing interruptions in service delivery [Sabiri et al. 2015]. Similarly, personnel training would lead to adaptation where it was necessary for teams to be acquainted with the integrated analytics environment of Fabric with real-time data insight into centralized workload management capabilities. There must also be effective training programs to ensure a smooth transition and employee readiness for the new platform [Thoutam 2024] [Sabiri et al. 2015]. Resource allocation and cost considerations for additional operations.

Conversely, migration projects are likely to incur uncovered charges for infrastructure upgrades, software licensing, and human resource training. As highlighted by Sabiri et al. (2015), strategic resource allocation becomes necessary in tempering expenditures because it optimizes everything to the maximum cost-effectiveness during and after the migration process [Sabiri et al. 2015]. Last but not least, it is extremely important to do very extensive testing and quality assurance, as it is with any complex initiative that runs the risk of issues. Fehling et al. (2013) suggest sandbox environments that test real-world cases to check the migrated service before massive deployment, and that keeps operational integrity throughout [Fehling et al. 2013].
After dealing with such technical and operational implications with meticulous planning and execution, now organizations can effectively pass through the transition complexities into Fabric. Such an approach not only minimizes risks associated with much benefit on any modern cloud analytics adoption but also provides operational continuity in line with long-term strategic success.

2.9.3 RQ3: What migration frameworks or methodologies are recommended for transitioning from Microsoft Azure to Microsoft Fabric?

Transitioning from Azure to Fabric requires raw frameworks and methodologies to assure a systematic and efficient transfer. The literature report a variety of strategies dealing with diverse complexities of migration into the cloud, underlining their structural, scalable, and repeatable nature. One of the most popular methods is Architecture Driven Modernization (ADM), which provides a clear path forward for understanding and transforming legacy systems into cloud-compatible architectures. According to Sabiri et al. (2015), this process starts from analyzing the existing infrastructure in order to create a Platform Independent Model (PIM) blueprint, and is then adapted into a Platform Specific Model (PSM) for the particular target cloud environment, such as Fabric. This method assures critical components of a legacy system are preserved while adapting them to the capabilities of the new platform [Sabiri et al. 2015]. Another framework which is recommended for adoption is the service migration pattern, emerging from the works of Fehling et al. (2013). These patterns provide decision-making support with respect to the appropriate migration strategy according to the type of applications architecture and operational requirements. Most important patterns are incremental migration, where less critical services are moved to develop the safety margin, to hybrid migration, which is a transition process having features of both on-premises execution and cloud deployment. These are patterns of great emphasis on being scalable and flexible, adjusting to the dynamics of changing requirements for migration [Fehling et al. 2013]. Iterative and modular methods are considered by the literature in facilitating the migration process. For example, Jamshidi et al. (2013) recommend splitting the migration task into smaller, manageable phases. Such an approach allows one to manage complexity by adopting an iterative approach, where one issue is treated at a given time, thus determining early and resolving complications before proceeding to the following phases. This approach also maintains continuous monitoring and optimization throughout the process of migration [Jamshidi, Ahmad, and Pahl 2013].
Thoutam (2020) emphasizes on the pre-migratory evaluations and assessment frameworks as well. It should be carried out by thoroughly analyzing the present architecture and testing these in sandbox environments to discover any potential problem areas and to verify interoperability with Fabric including data pipelines, scale-out features, and workloads-to optimize the system post-migration [Thoutam 2024]. Governance framework has substantial role in establishing a comprehensive migration model to keep high consistency and security. This is

the aspect in which integrity of data is preserved, access control specifies restricted usage, and compliance with regulations is achieved. Solomon et al. (2021) continue to stress that sound governance practices form the foundation of risk mitigation leading smooth transitions to new platforms [Alouffi et al. 2021]. Through such frameworks and methodologies, organizations like EDULOG are helped by ensuring that their migration to Fabric is thorough and as risk-free, while still maximizing the platform's potential to support advanced analytics and scalable operations.

2.9.4 RQ4: What are the performance benefits of Microsoft Fabric compared to Microsoft Azure in large-scale data processing and workload management?

Fabric has several performance advantages over Azure, especially concerning large-scale data processing and workload management. These benefits stem from advanced architectural optimizations; unifying capabilities within data analytics; and driving efficiencies toward analytics-based workloads. The scalability level greatly emphasized by Fabric is one of its premier features. It enables seamless scaling of compute and storage layers, dynamically providing resources with the workload requirements. This allows for high performance under heavy loads while avoiding bottlenecks. Lopes (2023) demonstrates how technology like Apache Spark and Delta Lake enhances the capacity of the platform's processing of diverse kinds of data through Pull Within Fabric [Lopes 2023]. Another advantage is that Fabric features a unified environment for data analytics, which combines batch as well as streaming data processes into one platform. By combining both a data lake and a DW under a single platform, it removes the need for complex ETL procedures, thus improving performance and expediting data processing. Lakehouse architecture facilitates better workflows that allow faster insight into how operations are performed [Lopes 2023].
Fabric also incorporates state-of-the-art optimization techniques such as Z-ordering and compaction, which greatly improve both the performance and query times at a lower amount of scanned data. Lopes (2023) has demonstrated the importance of such methods for high-frequency analytics operations, noting the great improvement offered by these features over previously used techniques. Furthermore, Fabric increases cost efficiency owing to its pay-as-you-go model and elastic resources that allow precise resource allocation depending on workload needs, thus eliminating over-provisioning and improving operational costs, especially in large-scale data processing as observed by Fehling et al. (2013) [Fehling et al. 2013] [Lopes 2023].
Fabric has a comprehensive set of features that allows it to grant both security and compliance, for example, it has encryption at rest and encryption in transit, strict access policies, and compliance with standards such as GDPR. These features are secure data processing mechanisms that are put into place for very high-performance environments, which is a necessary requirement for organizations dealing with any sensitive data. As exemplified in Alouffi et al. (2021), these security protocols play a vital role in upholding the integrity of operations-an area where Fabric users testify to a strong capability [Alouffi et al. 2021] [Lopes 2023]. Real-life examples further illustrate the capabilities of Fabric, with a case study by Thoutam (2024) showing the advancement of operational efficiencies, scalability, and innovation achieved by organizations transitioning to Fabric. The capacity of the system to achieve high-throughput workloads with low latency and consistent performance has made it a solution of choice for highly analytical companies.
So, in the end, Fabric guarantees vast performance benefits over Azure, namely, better scalability, faster insights, reduced operational expenses, and enhanced security. All of these

factors make Fabric the thing to go for those that wanted to make their organizations capital-efficient for data processing and workload management, where excellence in operations is required.

Chapter 3

Solution Analysis

This chapter aims to describe the context and needs that originated the proposed solution's definition. Section 3.1 starts with an analysis of the existing system and the main problems that have been observed in the Azure-based architecture, including cases of complexity, scalability, and governance, thus impeding its ability to support the analytical needs of EDULOG. Section 3.2 identifies the derived requirements from the analysis, classifying them into functional and non-functional ones. These requirements serve as the basis for the alternatives evaluation and for the justification of the migration to Fabric, to assure that the proposed solution would specially address both institutional and technological needs of EDUSTAT.

3.1 Overview of the Current Solution

The EDULOG solution involves the complete migration of its data architecture, represented in Figure 3.1 to Fabric with the objective of having a more automated, modular and scalable data platform. The architecture is on the Medallion architecture model, whereby data is organized in three layers — **Bronze**, **Silver**, and **Gold** — for standardization, consistency, and reusability throughout the data lifecycle.

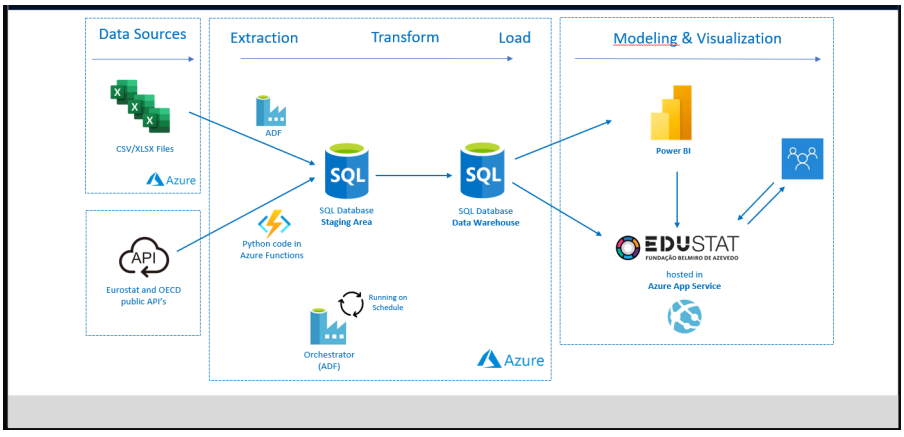


Figure 3.1: Oldest Architecture EDUSTAT

In the architecture of the figure 3.1 we can observe that first we have two types of data sources (Excel files or API's (for the Eurostat and OECD public API's)), in the first one the extraction is realized by Azure Data Factory, and the second is by Python code in Azure Fuctions, and both are saved in SQL Databases, named Staging Area, after the transformation the data was loaded to the DW, and the orchestration runes in every schedule, in Azure Data Factory. Finally, in the final part, the data can be shown in Power BIs, then later the EDUSTAT (Front-Office) take this Power BIs and show them to the users or in other way the EDUSTAT get the data directly from the DW (all these processes was made by B2F.
By organizing the platform in layers, in the Fabric, decoupling ingestion from modeling and transformation, and unifying data access in a centralized warehouse, the solution enhances both technical governance and operational agility. This architecture provides EDULOG with a firm foundation for growth, with the capacity to onboard new data sources faster and greater flexibility for analytical initiatives further down the line.

3.2 Requirements Definition

In order to guide the development and design of the proposed solution, it was important to define a clear set of functional and non-functional requirements. These requirements reflect both the technical goals of the platform and the operational requirements for processing public data in governed, reliable, and scalable form.
Functional requirements ensure that the solution provides the expected functionalities in terms of a system and user perspective, while the non-functional requirements specify quality properties such as performance, maintainability, and traceability — all primary drivers of long-term sustainability and goals alignment with institutions.

3.2.1 Functional Requirements

Functional requirements of the solution were defined to get the solution working efficiently through the entire data life cycle — from consumption through visualization — and to give freedom and autonomy. Among the main needs is the ability to consume data automatically from diverse, disparate sources, such as Excel files, and APIs, from the Eurostat and/or OECD sources. Once consumed, all information needs to be converted and normalized to a single tabular format, allowing for uniform processing irrespective of source origin or format.
Another main requirement is the utilization of a parametrized indicator configuration model, in which users can define metrics and calculations through back office-controlled definition tables. These configurations drive filters, grouping dimensions, aggregation rules, and business rules that are dynamically interpreted to generate output views and materialized tables — so-called Indicator Tables. This approach obviates the need for technical personnel for each new metric or modification.
In addition, the solution must also support real-time integration with visualization platforms such as Power BI Embedded to allow dashboards to be rendered in a smooth manner and users to access analytical outputs. Environment separation (i.e., development (DEV) and production (PROD)) must be supported to serve a secure CI/CD process so that changes and new work may be tested prior to being made available for publication. Lastly, the solution must allow for users to set, track, and verify indicators themselves but remain aligned with the institution's overall data governance model.

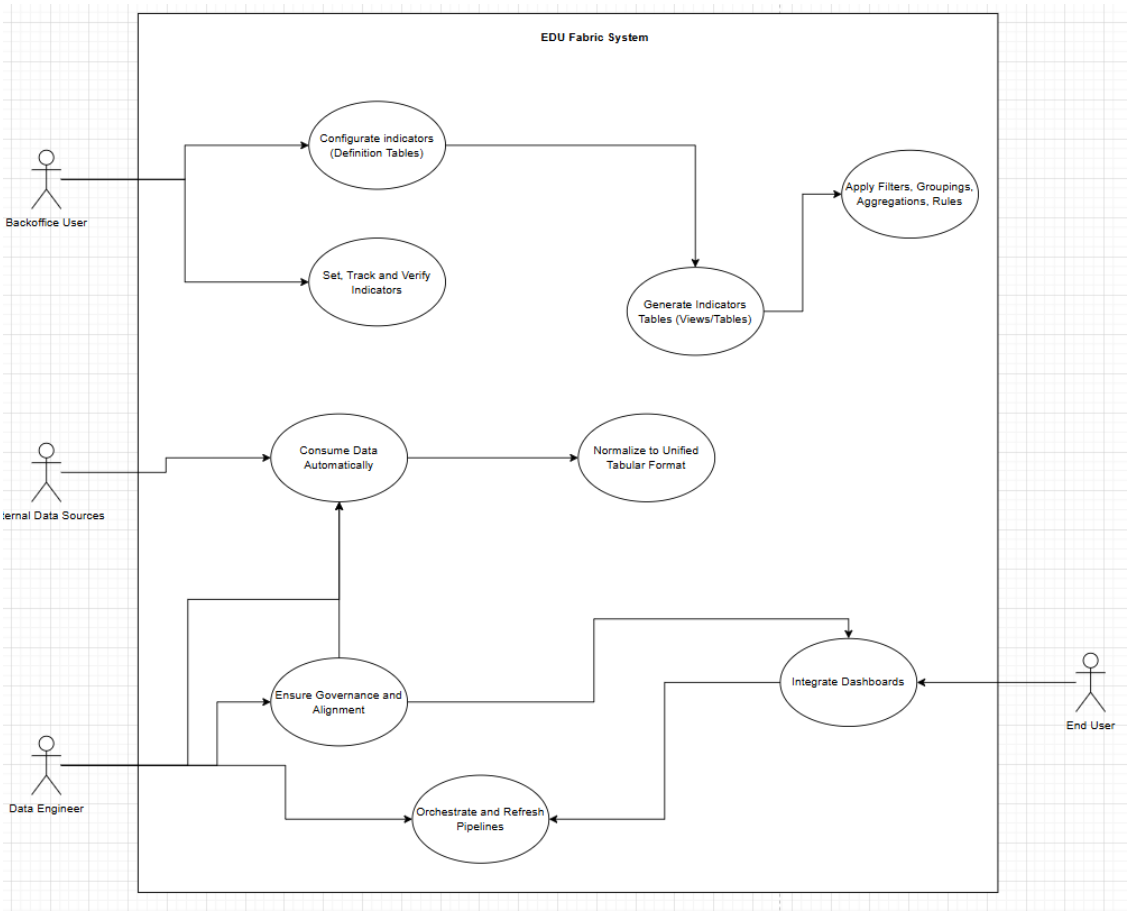


Figure 3.2: Use Cases Diagram

For better visual representation of functional requirements, a use case diagram was suggested in Figure 3.2. The diagram represents the interaction between diverse user roles, such as the data engineers, backoffice users, DevOps, and end users, with the core functions of the EDU Fabric system. It shows the processes of automated ingestion of data, its normalization into a common tabular format, the parameterized definition of indicators, orchestration pipelines, governance alignment, CI/CD support, and Power BI Embedded integrations. These use cases constitute the analytical work process in totality, from consuming raw data to displaying dashboards and indicators.

3.2.2 Non-Functional Requirements (FURPS+)

Functionality (F)

The solution must ensure data governance and security, aligning with institutional standards and guaranteeing that users only access the information they are authorized to view. It must also provide interoperability between data sources and consistency in applying business rules across all generated indicators.

Usability (U)

The system should enable ease of use for backoffice users when configuring indicators through definition tables, reducing dependence on technical personnel. Documentation and a

user-friendly interface are essential so that users can configure, track, and validate indicators without requiring advanced technical knowledge.

Reliability (R)

The platform must ensure stable, accurate, and available analytics outputs. ETL operations should perform automatically and reliably and must integrate multiple heterogeneous sources of data. The system should be resilient against failures and be able to maintain data integrity within ingestion, transformation, and publishing.

Performance (P)

The solution must be extremely scalable and performant, able to process large volumes of data from many different sources in near real time. Performance improvement over the previous Azure-based solution must happen through faster processing of ETL and lower latency in rendering dashboards via Power BI Embedded.

Supportability (S)

The solution must be maintainable, monitorable, and evolvable with ease continuously. CI/CD pipelines must allow secure promotion of the changes from DEV to the PROD environment so that new functionality or fixes can be added with minimal risk. New data sources or analytics requirements must be simple to accommodate without reengineering the entire architecture.

+ (Other Constraints)

The platform must be capable of satisfying institutional and legal requirements for processing public statistical data, e.g., traceability and audibility of processes. It should be in compliance with packaging and operations standards as defined under EDULOG's IT governance model.

Chapter 4

Solution Design

In this chapter, the architectural design of the proposed solution is introduced, which describes how the requirements set down in the previous chapter are translated into a technically viable and structured system. It begins by describing the methodological approach, combining agile project management practices with Medallion architecture as the technical guideline for design. The chapter then introduces the technical and storage architecture, comparing the previous constraints of the Azure-based environment with the better Fabric design, and explaining how its building blocks intercompose in order to support the whole analytical life cycle. Special attention is given to the EDUHOUSE data warehouse, modelling logic, and definition and indicator tables employed, which enable the generation of dynamic and configurable indicators in the absence of continuous developer interaction. The chapter also contains C4 diagrams and logical views illustrating the structural and behavioral components of the solution. Finally, the justification of the most critical design decisions is presented, wherefore Fabric, SQL-based modeling, CI/CD, and the Medallion architecture were chosen as the foundation of this migration.

4.1 Methodological Approach

This section describes the methodological assumptions that guided the solution suggested in both the project management and technical architecture perspectives, as we already talked about in Chapter 1.
Firstly, it describes the Agile methods utilized in tackling the execution and planning process throughout the period, where the project management technology used was DevOps. Secondly, it describes the technical method utilized, the ETL, specifically the Medallion structure, in structuring the layers of the system and enabling a framework that is both modular and scalable.

4.1.1 Project Management Approach

To ensure proper coordination of tasks and timely completion of project components, an Agile-based approach was adopted for the management of this project and dissertation. The development process was carried out in iterative cycles, facilitating continuous improvement of the solution and the conformity of both technical goals and academic milestones.
The project was managed on Azure DevOps, allowing for the planning of tasks, sprints, for the project delivery, and deliverables throughout the academic year. Task planning, sprint planning, and team management were conducted following Agile principles, allowing for flexible and incremental development flow. The project was divided into phases - research, architectural design, implementation, and documentation - each divided into user stories and

work items, allowing simple visibility over priorities and workload.
In addition, this Agile approach supported an ongoing process of validation in which working segments of the solution — such as ingestion pipelines, data models, and dashboards — were tested, reviewed, and refined iteratively. This created a balance of effort, reduced implementation risk, and fostered a disciplined rhythm in both the development and reporting tracks of the project.

4.1.2 Technical Methodology

Technically, the solution was designed on the lines of the ETL, specifically the Medallion Architecture, a layering data modeling approach commonly used in current analytical platforms. This paradigm structures data transformation and storage in three incremental levels — Bronze Layer, Silver Layer, and Gold Layer — each with a specific function in the data life cycle and enabling clear separation of concerns, reuse, and traceability, already documented in the previous chapter.
The layered approach made the platform support scalable and automated data flows with minimum hard-coding or manual changes. The usage of SQL Views and stored procedures inside Fabric rather than using notebooks promotes maintainability, transparency, and direct integration into the semantic models consumed by the Power BI dashboards.
With the solution design conforming to the Medallion Architecture, the system acquires modularity, auditability, and flexibility-the key features for public sector data platforms where transparency and evolution are fundamental requirements.

4.2 Technical Architecture Overview

This section tells us about the technical architecture of the solution envisioned, with a focus on the architectural decisions facilitating performance, scalability, and maintainability. It begins by looking at the transition from the present Azure-based setup to a conformant Microsoft Fabric environment before detailing the main architectural components — including the adopted platform, data storage patterns, and semantic framework — which together make possible the system's entire data life cycle, from ingestion to visualization, in this same section we will talk about the class diagrams.

4.2.1 Platform Architecture and Design Rationale

Fabric was chosen as the focal platform for the solution because of its ability to consolidate all data lifecycle phases into a single location. Unlike the current architecture — that is, a disjointed collection of Azure services such as Data Factory, Synapse Analytics, and standalone storage layers — Fabric offers an integrated stack that can serve ingestion, transformation, modeling, and visualization from a shared infrastructure and governance model.
Since Fabric offers native support for Lakehouse architecture based on OneLake and Delta Tables, there is no need to manage many different disconnected services and formats. Hence, this decision has lessened the complexity of orchestration and automation and also has upbeat performance, consistency, and auditability on different layers. With the centralizing of data storage and compute under a single engine, Fabric makes access control easier, increases scalable capabilities, and smoothens the monitoring of usage and costs.
Another significant factor in choosing Fabric was its native connectivity with Power BI Embedded, facilitating direct interactivity of data models with front-end visualizations without adding additional transformation layers. In this setup, the modeling logic developed in the

platform's Warehouse — SQL views and stored procedures can be made available to the dashboards directly, so transparency and consistency exist. The platform must support established methods such as CI/CD pipelines, environment promotions (DEV to PROD), and version control to ensure quality and stability in an iterative and highly collaborative development environment. Multi-programming interfaces available in the system (e.g., PySpark, T-SQL) are granted where flexibility is necessary, but the solution deliberately defaults to SQL for maintainability and performance.
Overall, Fabric was chosen not only as a cloud service but as a strategic enabler for automation, modularity, and governance — all of which are necessary for the long-term sustainability and development of analytical platforms such as EDU Fabric.

4.2.2 Comparison between Legacy Azure and Microsoft Fabric

Here we can observe the oldest architecture of the systems, in Azure, in comparison with the newest architecture, in Fabric.

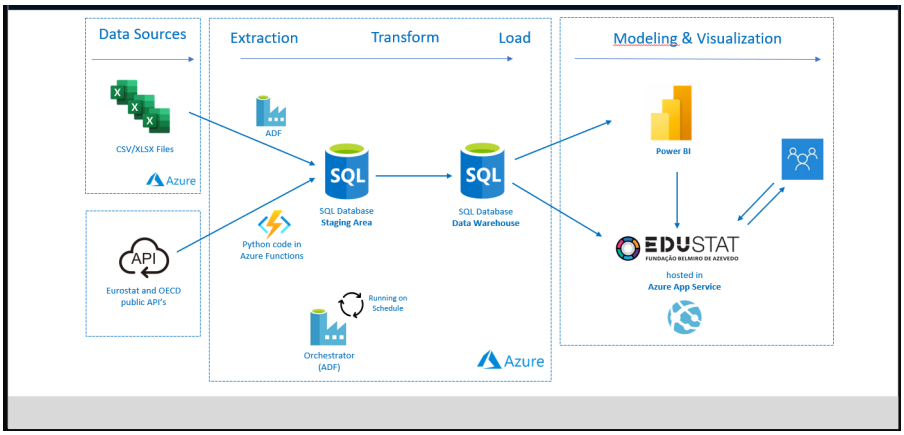


Figure 4.1: Oldest Architecture EDUSTAT

As mentioned in the previous chapter, the explanation of the actual architecture of EDUSTAT states that the entire process is hosted in Azure, as is observed in Figure 4.1.

Similarly to this architecture, we have the architecture of the infrastructure of the Brighter Future, This project was donated to EDULOG, and the architecture was very similar to the architecture of EDUSTAT, but this project wasn't made by B2F, but when it was donated to EDULOG, they migrated the infrastructure too.

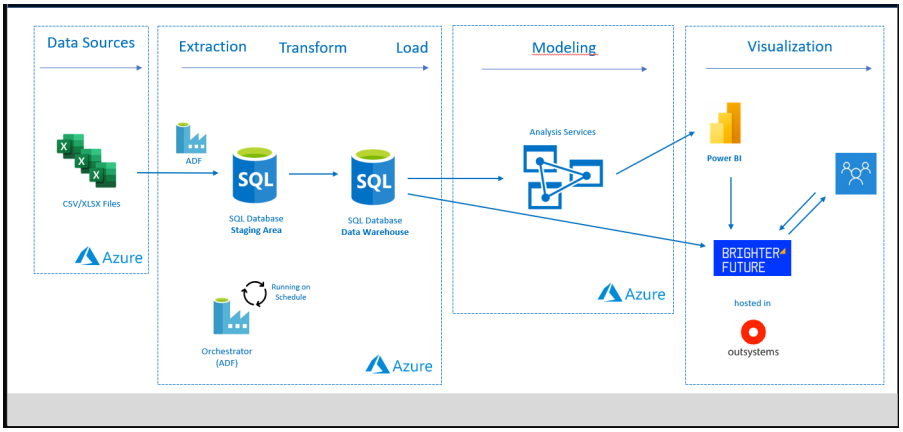


Figure 4.2: Oldest Architecture Brighter Future

In the figure 4.2, like we had in the last figure, the data source was in Excel files, they are extracted by Azure Data Factory to the Staging Area, some transformations happened, and the data was loaded into the DW. By here, two thing happen, or some tables was saved in analysis services (tabular), here we create like a copy of each tables just to load this tables into Power BI, and the Brighter Future site consume this dashboards to show them to the users, or in other way the site can take the data directly from the DW, all the site was hosted in Outsystems.

Finally, as we can observe in the figure 4.3, in the newest architecture, we've the Medallion architecture, where the bronze layer has the raw data. This data could be displayed in two ways, from Excel Files or by API's. This data will be stored in the Lakehouse, after by Pyspark in notebooks, the data will be stored in the silver layer, in the Lakehouse too, but now in cleaned tables, with every modification that we have already added, when everything is correct, by SQL we load the data into gold layer, in DW.
The visualization of the data can appear in two ways, one, as happens in the oldest architecture, the data was loaded into Power BIs and the EDUSTAT will consume these dashboards, or directly from the DW, and this data can be accessed by users.

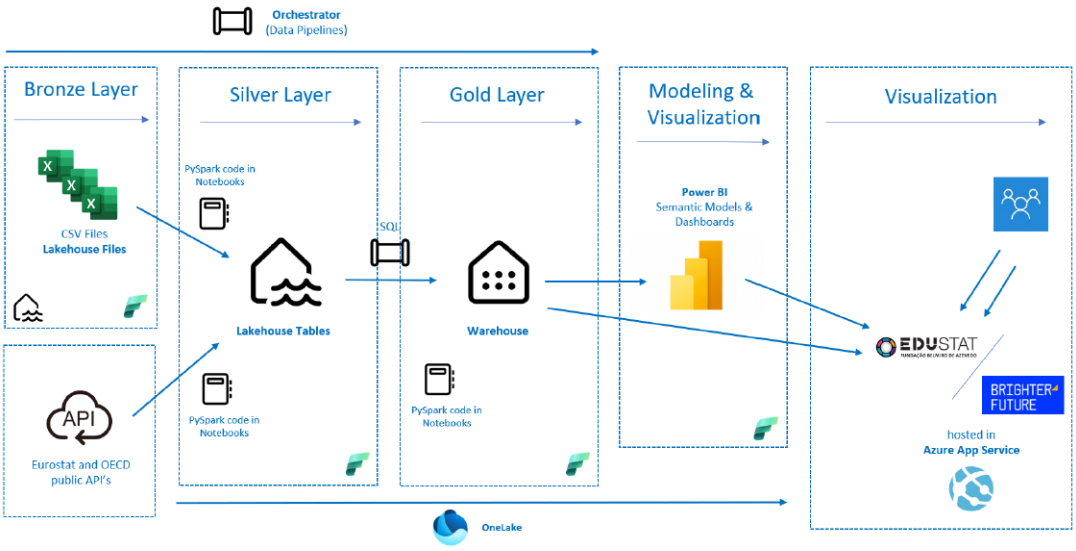


Figure 4.3: Newest Fabric Architecture

4.2.3 Storage Architecture

In the actual data systems, there exist three paradigms for data storage, as we can see in the table 4.1, the Data Lake, the DW, and recently the Lakehouse architecture. Each one has its limitations and strengths, depending on the application, variety of data, and governance requirements.

	Data Lake	**Data Warehouse**	**Lakehouse**
Type	Raw (unstructured), semi-structured	Structured	Structured, semi-structured, unstructured
	Relational, non-relational	Relational	Relational, non-relational
Schema	Schema on read	Schema on write	Schema on read, schema on write
Format	Raw, unfiltered processed, curated	Processed, vetted	Raw, unfiltered processed, curated, delta format files
Sources	Big Data, IoT, social media, streaming data	Application, business, transactional data, batch reporting	Bid data, IoT, social media, streaming data application, business, transactional data, batch reporting
Scalability	Easy to scale a low cost	Difficult and expensive to scale	Easy to scale a low cost
Users	Data scientists, data engineers	Data warehouse professionals, business analysts	Business analysts, data engineers, data scientists
Use Cases	Machine learning, predictive analytics, real-time analytics	Core reporting, BI	Core reporting, BI, machine learning, predictive analytics

Table 4.1: Storage Architecture Differences

Talking firstly about the Data Lake, it can hold semi-structured, and raw (unstructured) data in their native formats, based on schema-on-read. Even if this approach is very scalable, low-cost, and cheap as a storage option, it lacks ontologies and performance guarantees needed for business analytics, making it less fit for use cases demanding strong data consistency, data lineage, or some form of interactive querying. This type of data storage doesn't permit any ACID transaction.
On the other hand, we have a DW that can hold a structured, schema-on-write environment optimized for analytical queries and business reporting. It provides greater performance and data integrity considerations through indexing, relational modeling, and optimized query execution. However, DWs are commonly less flexible with heterogeneous or fast-evolving data sources and require predefined schemas and ETL pipelines that may sometimes add a layer of rigidity and overhead in dynamic environments. The DW permits ACID transactions.
To combat all of these limitations, with the same strengths, we have the junction of Data Lake and DW, we have the Lakehouse, and it was conceived as a hybrid that makes Data Lakes more flexible and DWs more reliable and faster. By leveraging Lakehouse technology, Delta Lake provides ACID transactions - the properties that enforce the atomicity, consistency, isolation, and durability of operations, where atomicity memories operations are either carried out entirely or not at all, the consistency maintains data integrity, the isolation does not allow operations to occur concurrent with one another, and finnaly the durability ensure that once an operation has been committed, even a crash cannot undo it. Having these guarantees, data can be handled safely and reliably in complicated and distributed environments. Moreover, Delta Lake supports time travel and schema evolution so that users can access an older version of the data or make structural modifications without risking consistency. This enables organizations to persist raw and processed data in a single unified platform, without compromising scalability or governance.
For this project, the Lakehouse architecture - via Onelake in Fabric - was chosen as the foundation for the EDULAKE layer (the bronze and silver layers). The EDULAKE layer maintains data in open formats and supports ingesting from different sources (files and APIs) and version control, schema flexibility, and SQL and Spark engine compatibility. For ready and business purposes, information, the EDUHOUSE module (gold layer) is the structured DW component of the solution in which finalized indicator outputs are created and optimized for reporting. This mixed architecture enables the platform to leverage the best of two worlds: Lakehouse storage's flexibility in ingestion and transformation, and Warehouse storage's reliability in reporting and dashboarding integration. It also accommodates modular pipeline design, incremental processing, and long-term data asset maintainability.

4.2.4 C4 Diagrams

To complement the technical architecture and provide a concise graphical overview of the system structure, this section presents a set of C4 diagrams illustrating the logical and development views of the solution. The diagrams were prepared to demonstrate how the different components communicate and data flows from ingestion to visualization, both at a high-level overview and in detail for significant processes. Starting with the Logical View Level 1, as shown in Figure 4.4, we have our system, EDUSTAT, and as we already talked in other chapters, we have two ways of data insertion, or by Files (Excel or CSV files) or by API's (OECD or Eurostat), and who will consum from our system was the user.

Deeply, we have the Logical View Level 2, as we can observe in Figure 4.5, we can start with the insertion of Files into the Files folder, inside the system, the users in the backoffice realized this, this Files folder was the Bronze Layer in the Medallion architecture, in other side,

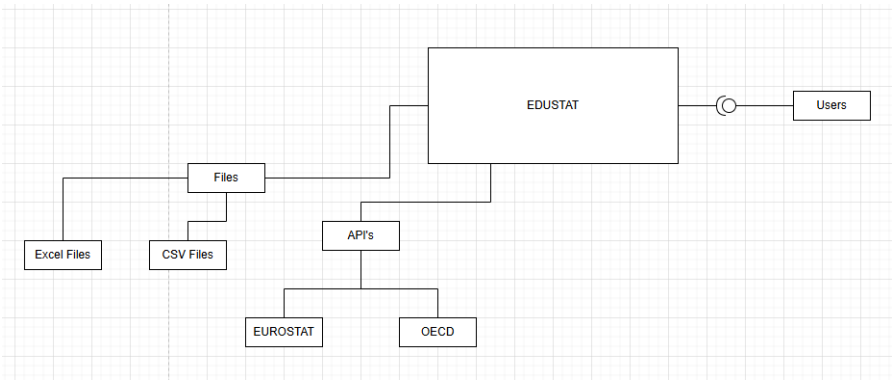


Figure 4.4: Logical View Lv1

we have the API's, from the notebook in PySpark the data was extracted and transformed into the Silver Layer, in other words into the Delta Tables, other different way to insert data into the Delta Tables is from the notebooks applied into the Files folder, that will transform the data into Delta Tables structure, so the Delta Tables consume the information from the Files. Both are in the Lakehouse, and the DW consumes the info from the Lakehouse.

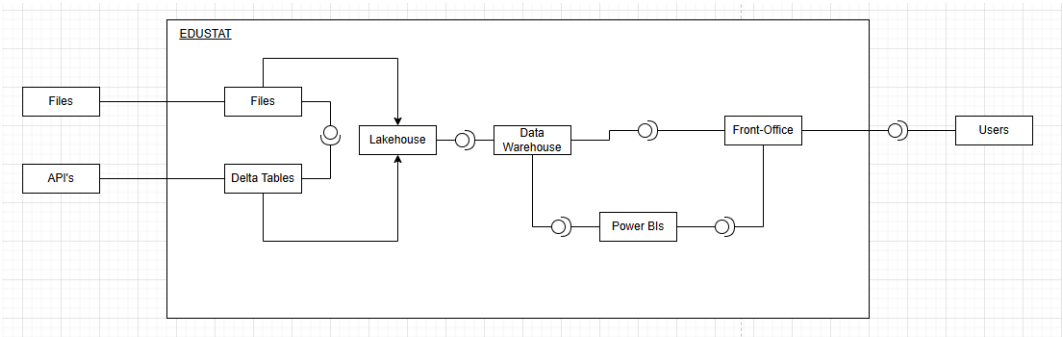


Figure 4.5: Logical View Lv2

The front office has two ways to take the information, either via the consumption of the information in the DW, directly, or via the consumption of data in the Power BI's (dashboards), and the final user will use the front office to access the system.

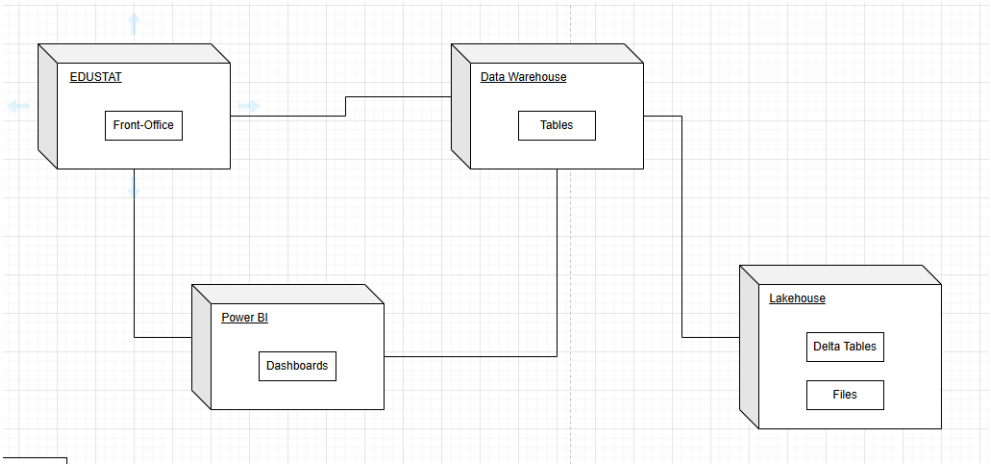


Figure 4.6: Development View Lv2

Finally, in the last diagram, we have the Development View Lv2, as shown in Figure 4.6, where we have two connections: one between the EDUSTAT system, which includes the Front Office, and Power BI for the dashboards. The other connection is between the EDUSTAT and the DW, where we have tables. We also have one connection between the DW and the Power BI. At the end, we have a connection between the DW and the Lakehouse, where we have the Delta Tables and the Files. The proposed solution will then be presented.

4.3 Solution Structure

The solution is architectured to support the entire data life cycle — ingestion through visualization — modularly, layer by layer, and automated. Architecture relies on the Medallion model which organizes data into three sequential layers — Bronze, Silver, and Gold — in which every stage is dedicated to different phases of data processing. Such layered composition supports consistency, traceability, and scalability throughout the platform.
Essentially, the approach adopts the Lakehouse model, combining the scale and flexibility of traditional data lakes with the modeling, performance, and governance of DW. In this approach:

- The **Bronze Layer** ingests raw data from CSV files and public APIs (e.g., Eurostat and OECD);
- The **Silver Layer** normalizes and structures that data into reusable delta tables;
- And the **Gold Layer** offers business-grade output in the form of pre-designed fact, dimension, and indicator tables.

Two core storage components maintain this model:

- **EDULAKE**, where raw and normalised data is stored in delta form;
- **EDUHOUSE**, where ultimate indicators, definitions, and dimensions are deposited in a form where they can be analyzed and visualized.

The overall architecture of the solution is depicted in Figure 4.7, which illustrates the end-to-end data pipeline, from ingest in the Bronze Layer to final visualization delivered to such

outlets as EDUSTAT and Brighter Future, encompassing semantic modeling using Power BI Embedded as well as orchestration via native Fabric pipelines.

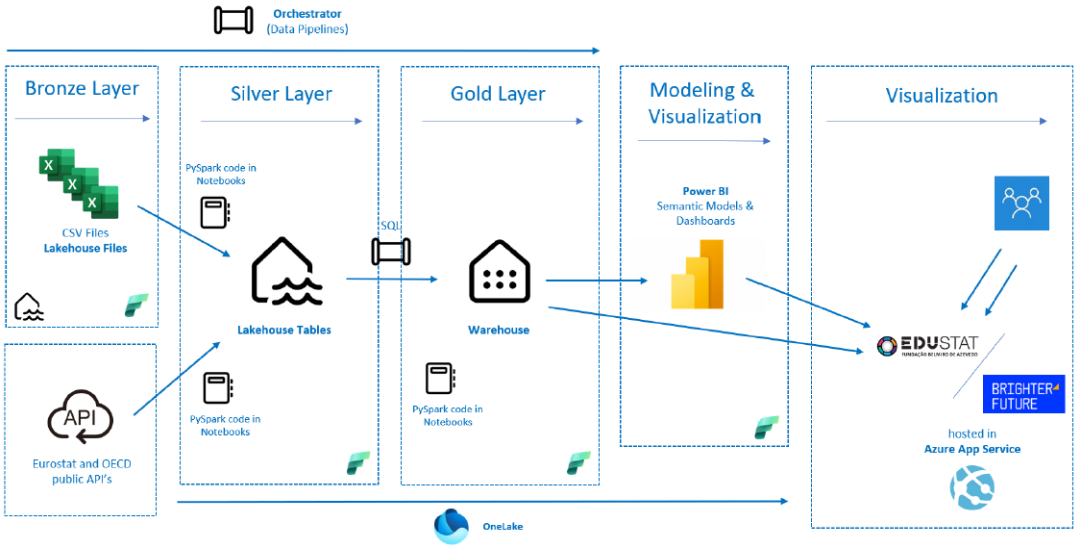


Figure 4.7: Architecture EDU Fabric

This unified architecture affords a highly flexible and manageable platform model for EDULOG. It allows new sources of data to be integrated quickly, it promotes modularity and reuse, and it permits non-technical users to define indicators via the backoffice of the platform — enabling an agile and scalable approach to providing public data.

4.4 Layers of Structure

EDU Fabric solution conforms to the Medallion architecture model that decomposes the data lifecycle into three logical layers, namely Bronze, Silver, and Gold. All of them cover a specific spot in the pipeline of processing for more modularity, better separation of concerns, and easier platform maintenance. The stratified approach facilitates a process where raw data are safely ingested and stored and later transformed to reusable forms, and finally modeled to harvested outcomes suitable for analyzing and publishing. Each layer, in the coming sections, will be described individually with its functionality, architecture, and worth to the entire ecosystem.

4.4.1 Bronze Layer – Data Ingestion

The Bronze Layer is the first phase in the EDU Fabric data pipeline. Its primary function is to store raw, unprocessed data in its original form, with complete traceability and no information loss during ingestion. This layer forms the foundation for all the transformations and standardizations that occur in the Silver and Gold layers. Inside EDU Fabric, the Bronze Layer receives data from multiple sources, all of which have the same structure, and these sources require manual entry for file imports (CSV files). The data is verified manually and uploaded by analysts into the Microsoft Fabric Lakehouse Files directory. Figure 4.8 demonstrates the EDU Fabric architecture and how CSV files and API responses, from Eurostat and OECD, are piped into the Bronze Layer.

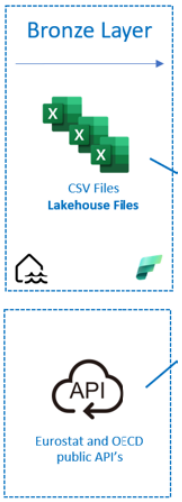


Figure 4.8: Bronze Layer

All data consumed is stored within a single Lakehouse instance named EDULAKE. Storage is also stored in a folder structure organized by source, with each folder containing either a single or multiple files. This keeps the data in its original format and also facilitates quick access and processing in subsequent stages. Every ingested dataset has a specific indicator attached to it that becomes the basis for its following transformations.
The data (files) was inserted by users into the Bronze Layer, not directly, but they insert the files in the backoffice and the data was ingested into the Bronze Layer.
A standard record ingested into the Bronze Layer includes common metadata fields of dataset name, value, and dimensions of gender, age, and region. Table 4.2 below displays a standard sample from the "dgeec_bols1" dataset.

Linha	Dataset	Valor	Género	Idade	NUTS I	Bolseiro
254	dgeec_bols1	10	Feminino	20	Continente	Sim
...						

Table 4.2: Bronze Layer Table

The Bronze Layer architecture offers support for data lineage, versioning, and reprocessing. Through the separation of raw ingestion and transformation logic, the platform is more flexible, better governed, and able to evolve consistently throughout the entire data lifecycle.

4.4.2 Silver Layer – Transformation and Normalization

The Silver Layer is a critical part of the EDU Fabric pipeline that exists between raw data intake (Bronze) and business-conformant data modeling (Gold). It is primarily to transform and normalize data from various sources into a standard format. Through this unification, all datasets are presented according to a single schema so that automation is enabled in the

downstream modeling and indicator creation process.
In our system, we will have two different ways that happen in the Silver Layer, one for the data that we already have in the Bronze Layer, and the other for the data that is not already in the Lakehouse.
Relatively to the data not being in the Lakehouse, we have one way that the data was extracted and transformed in the same manner, adding to the Silver Layer directly. This happens with the data that comes from the Endpoints (APIs), the Eurostat and OECD data, this happens for a single reason, to avoid redundancy in the process and because Eurostat and the OECD consider datasets with millions of lines, a Bronze Layer was not considered for these datasets, only silver layer. This procedure occurs in a PySpark notebook, which simultaneously performs the extraction and transformation.
We used PySpark because it offers significant performance advantages compared to traditional tools, especially in handling large volumes of data or ingesting files concurrently. With its distributed process model, which enables concurrent running of ingestion processes on separate nodes, processing is fast and efficient with minimal processing time and system usage. Each notebook integrates with foreign endpoints, interprets returned payloads, and stores the output in a clean, uniform format. With the modular nature of the ingestion layer, new data sources can be added or exchanged without requiring changes to the rest of the pipeline, allowing the system to scale and be flexible.
In the data that was ingested into the Bronze Layer, the files case, PySpark notebooks transform the data by applying transformation logic to convert the source-specific formats received into a standard table, or, as it is named in the Silver Layer, a delta table, that supports ACID transactions. All notebooks have been designed to reshape these diverse structures into one model using the fields: "Dataset", "Valor", "Filtro", and "Significado". Figure 4.9 illustrates this data transformation architecture within the Silver Layer, capturing the way PySpark notebooks behave in the Lakehouse to produce structured Delta Tables.

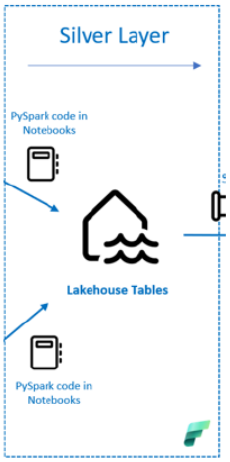


Figure 4.9: Silver Layer

This universal table format enables all sets of data — wherever they originate — to be brought to a common schema. This consistency is essential for automating indicator modeling, limiting human intervention to an absolute minimum, and facilitating the easy integration with the backoffice. These tables are kept in Delta format in EDULAKE, benefiting from ACID transactions, versioning, schema enforcement, and incrementally efficient update.
One original record from the Bronze Layer can produce more than one row in the Silver Layer during this process, creating the same columns structure for all tables. For example,

if there is a single record with more than one attribute (e.g., gender, age, region), every attribute is unwound into a separate row with the same Dataset and Valor values and with differing "Filtro" and "Significado" pairs, the name of the process was unpivoting, when we transform different columns into different rows. This flattening semantics is illustrated in Table 4.3, where one record from dataset "dgeec_bols1" is unfolded into four structured rows in the Silver Layer.

Linha	Dataset	Valor	Filtro	Significado
254	dgeec_bols1	10	Género	Feminino
254	dgeec_bols1	10	Idade	20
254	dgeec_bols1	10	NUTS I	Continente
254	dgeec_bols1	10	Bolseiro	Sim

Table 4.3: Silver Layer Table

This structure is the foundation for facilitating flexible filtering and dynamic indicator modeling in Gold Layer. By elevating the native source structures into a generic and consistent form, the Silver Layer maintains the pipeline of data modular, scalable, and manageable while still preserving the precision and conformance required by high-quality visualizations and analytics.

4.4.3 Gold Layer – Storage and Modeling

The Gold Layer is the final stage in the EDU Fabric data architecture, where transformed, business-ready data is exposed for visualization, analysis, and publication. It is the layer where all of the structured data from the previous phases is brought together and stored in a relational, high-performance warehouse environment named EDUHOUSE. Data is mapped into fact and dimension tables here and mixed with configurable logic in order to generate final indicators.
In the Gold Layer, unlike what happens in the Silver Layer, the different rows are transformed into different columns, the opposite process of unpivoting, here, the name of the process was pivoting.

The layer structure, and how it is being integrated with the Warehouse component, can be observed from Figure 4.10, which shows the loading process from Lakehouse and preparation of final output in the Gold Layer.

The next sub-subsections explain the design of EDUHOUSE, dynamic logic for indicator modeling, role of definition tables, and benefits of this module and scalable design.

EDUHOUSE Overview

The EDU Fabric architecture's Gold Layer is implemented in a centralized DW named EDUHOUSE. The warehouse is designed to store fully curated and structured data in support of business-level indicators and analytical results, such as those released in platforms like EDUSTAT and Brighter Future. While the Bronze and Silver layers operate within the Lakehouse platform (EDULAKE), the Gold Layer takes the architecture to a relational and

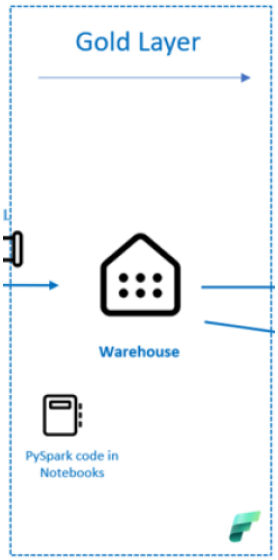


Figure 4.10: Gold Layer

highly structured system, where data is stored in SQL-based tables. The tables are performance, consistency, and semantic modeling optimized.

The EDUHOUSE warehouse contains three broad categories of tables:

- **Fact Tables** ("F_" or "FACT_"): These contain the numerical values from every data set, processed and aligned to analyze. Every data set consumed through the pipeline contains a corresponding dedicated fact table that maintains the original numeric values and contains references to numerous dimensions. Figure 4.11 shows examples of the fact tables generated for different datasets in the EDUHOUSE warehouse.

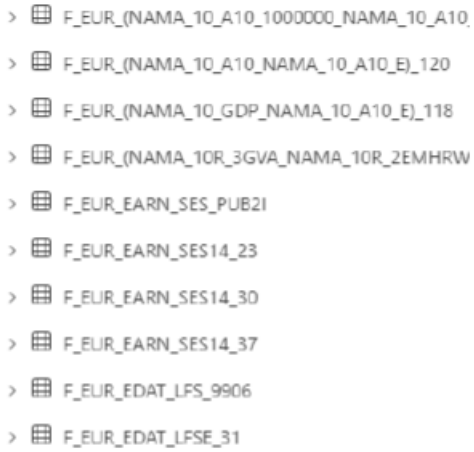


Figure 4.11: Fact Tables

- **Dimension Tables** ("DIM_"): These store categorical labels such as regions (NUTS), institutions, genders, or indicator metadata. These tables enable flexible segmentation and are a very critical component of the model-based process that powers indicators

and dashboards. Figure 4.12 presents a view of the available dimension tables used to enrich and contextualize fact data.

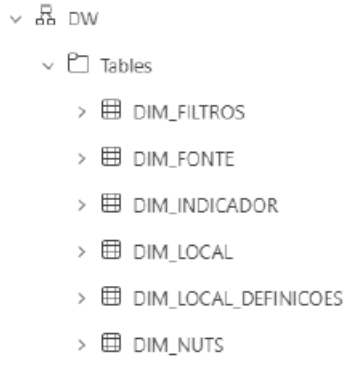


Figure 4.12: Dimension Tables

Configuration and Output Tables (explained in the following sections): These are the definition tables (which store user-configured logic) and final indicator tables (automatically generated output based on those definitions). This isolation of fact and dimension tables is at the center to allow semantic modeling, best data joins, and user-driven indicator creation. The Gold Layer is the most user-centric and highest-level component of the architecture, where data preparation converges into consumable, high-quality analytical products.

Modeling Logic

The rationale for modeling in the EDU Fabric architecture is to achieve an extremely flexible, dynamic, and autonomous process for the generation of business indicators. The core of this rationale is data structure standardization at the Silver Layer, where all datasets — no matter from where they come — are transformed into a homogeneous tabular structure with fields "Dataset", "Valor", "Filtro", and "Significado".
This uniformity allows subsequent steps, such as modeling and indicator generation, to be independent of where or in what form the data originated. In the Gold Layer, the architecture creates a distinct segregation of:

- **The Data Warehouse (DW)**, where fact and dimension tables, normalized and stored;
- **The Definition Tables**, where every dynamic rule for building business indicators from user-definable settings managed via the backoffice.

This approach enables the platform to create indicators on the fly without developer intervention, to scale horizontally when new data sources or business requirements appear, and to maintain a good separation between raw data storage and business logic.
One of the main tenets of this design is source agnosticism: the structure of the DW is decoupled from the data origin (file or API), as opposed to legacy models that needed to have an individual ETL per source. Additionally, the system is schema-evolution enabled, wherein new fields get automatically integrated into the modeling logic without any structural modifications. The dynamic modeling process eventually produces indicator-specific results, which are materialized into specialized tables and views to be used in the visualization and report layers. The specific organization and functions of the definition and indicator tables — and illustrations of their application — are explained in the next sub-subsection.

Definition and Indicator Tables

One of the most significant characteristics of the EDU Fabric architecture is the distinction between the business logic definition and storing final results of indicators. It is applied through two table sets in the Gold Layer: the Definition Tables and the Indicator Tables.
The Definition Tables are the parameterization layer governing the way in which each indicator is constructed. The tables are governed by a given backoffice interface, where non-technical users can establish various aspects of indicator creation such as:

1. Which columns to exclude or include;
2. Which filters apply to a given sets of datasets;
3. How calculated fields or column matches are defined;
4. Which dimensions should be treated as categorical or not included within dimensionalization.

The architecture involves a series of specialized tables — such as "DEFINICOES_COLUNAS_EXISTENTES", "DEFINICOES_FILTROS_EXISTENTES", "COLUNAS_MANUAIS", and so on — that capture different dimensions of indicator configuration. These tables are part of the schema PARAMS, the metadata layer for the system's modeling logic.

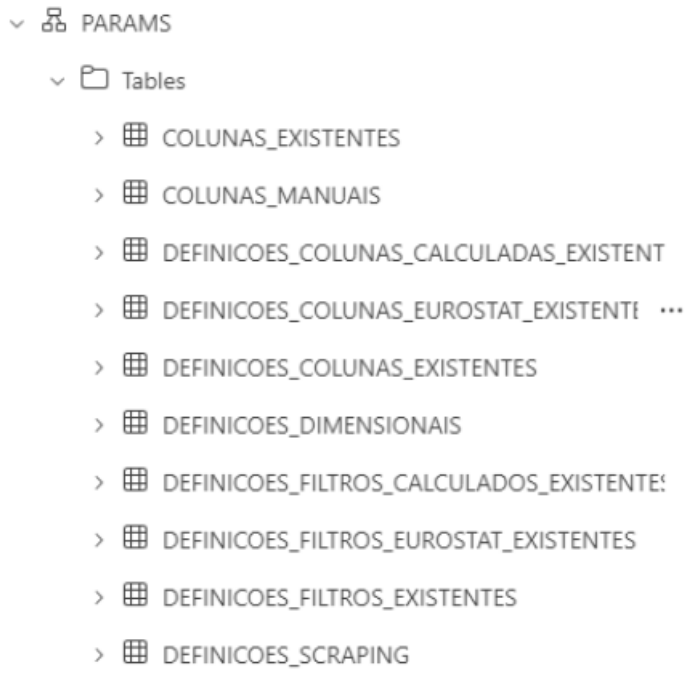


Figure 4.13: Definition Tables

Figure 4.13 displays this metadata layer graphically, emphasizing the main definition tables behind the modeling process.
Once definitions are registered, the ETL process translates them and creates the final outputs: the Indicator Tables. One indicator table per indicator ("INDICADOR_###") that contains only the data relevant to each indicator. These tables are created automatically and are ingested directly into the reporting and visualization interfaces (e.g., the EDUSTAT

and Brighter Future platforms).
This process is shown in Figure 4.14, in which a sample list of indicators automatically generated by the EDU Fabric system materialized is shown.

Figure 4.14: Indicator Tables

For further traceability and debugging, every indicator also has an additional associated SQL view that stores the query that was executed to create the materialized table. This is provided for purposes of opening up transparency into the logic used and for validation or duplication of results as required. The use of this dual-table schema, definitions for configuration, and indicators for output, renders the platform adaptable, through the easy development or reconfiguration of indicators; regular, through the central enforcement of modeling rules; and sustainable, through any changes to definitions being propagated automatically, without the requirement of manual updates to the ETL process.
Together, the Definition and Indicator Tables apply a business-friendly and scalable modeling paradigm that separates configuration from execution, enabling EDULOG to produce high-quality indicators cost-efficiently and autonomously.

Benefits of This Structure

The architectural design of the EDU Fabric boasts an immense performance, scalability, maintainability, and user self-containment benefit. With structured data on standard layers (Bronze, Silver, Gold) and isolating business logic from data storage by using definition and indicator tables, the platform has a firm analytical base for sustainable development and versatility. Perhaps the single most valuable aspect of this architecture is its scalability. Because the Silver Layer ("Dataset", "Valor", "Filtro", "Significado)" uses a standardized

tabular model, the system can easily add new sources of data—either file-based or public API-based—without modifying the downstream architecture. That allows new datasets to be quickly and uniformly added regardless of source.
Furthermore, this solution enhances the operation and automation. Since the architecture is based on Delta Lake tables and partitioned storage, it inherits compression, transaction integrity (ACID), and incremental updates for processing data. This is further reinforced by automated pipelines for orchestrating and loading data with minimum human input, leading to lesser latency and fewer operational efforts. Another major benefit is modularity. Each layer in the architecture is responsible for its own thing, enabling a clean separation of concerns between ingestion (Bronze), transformation (Silver), and modeling (Gold). This not only improves system organization, but also makes debugging, versioning, as well as reusability of logic across indicators and sources more straightforward.
Also, the use of definition tables and dynamic generation of indicator outputs gives a very good level of independence for users. New indicators can be directly defined by analysts and business users through the backoffice, in a non-technical way without developer involvement. This dramatically accelerates the development of insights and reduces the bottlenecks that typically accompany centralized ETL development.
Finally, the overall architecture ensures maintainability and consistency. As modeling rules are kept in centrally controlled definition tables, any change to filters, column mappings, or business logic will automatically be propagated to the corresponding indicators. Central control avoids redundancy and inconsistencies and makes the long-term maintenance of the analytical system easier.
In summary, the EDU architecture Fabric successfully combines flexibility, performance, and governance to create a robust and resilient platform for the management and dissemination of public data indicators.

Why was the Medallion architecture (Bronze, Silver, Gold) adopted?

Medallion architecture (Bronze, Silver, Gold) was adopted as the foundation of EDU Fabric due to its strong support for layered modularity, reusability of data, and progressive refinement, one of the documents that reinforce the idea of the implementation of the Medallion Architecture was "Understand medallion lakehouse architecture for Microsoft Fabric with OneLake", [Microsoft n.d.(d)]. There is a defined role for each layer in the system: the Bronze Layer consumes raw data from files and APIs; the Silver Layer unifies all data into a common structure, ready for reuse; and the Gold Layer focuses on business modeling and the generation of curated output for end-users. This design allows for the addition of new data sources with no disruption to downstream logic and maintains transformations as discrete, auditable, and reusable across a range of indicators and contexts. By embracing the Medallion model, EDU Fabric is able to attain a highly maintainable and scalable architecture that can evolve very quickly with emerging analytical needs.

Chapter 5

Solution Implementation

This chapter offers the real deployment of the designed architecture, demonstrating how the migration from Azure to Fabric was deployed in the EDU Engineering DEV environment. It begins with the description of the organization of the environment into thematic folders, such as data extraction and transformation, orchestration, machine learning experiments, storage, and integration into the EDUSTAT platform. The chapter then emphasizes the storage and warehousing features, i.e., EDU Lakehouse and EDUHOUSE, that manage data consolidation, standardization, and modeling. The subsequent sections detail the extraction and transformation activity, where notebooks automatically trigger ingestion from heterogeneous sources such as Eurostat APIs and Excel sheets and restructure them into the homogenized Silver layer form. Patterns of orchestration are also being addressed, showing how pipelines and CI/CD practices provide automation, consistency, and orchestrated promotion across environments. Finally, the chapter includes real implementation proof in the form of screenshots, process examples, and tables that prove the solution efficiency and validate the value of Microsoft Fabric for EDULOG's analytics platform.

5.1 Technical Justifications

In order to ensure that architectural decisions in EDU Fabric were solely driven by its strategic objectives, some fundamental architectural questions were brought under the limelight while designing and implementing the solution:

- Why was Microsoft Fabric chosen as the central platform?
- Why were SQL views and stored procedures preferred over notebooks for modeling?
- Why was Continuous Integration/Continuous Delivery (CI/CD) implemented?

These are elaborated in the subsequent subsections with explanations for the most ideal technical decisions in the solution.

5.1.1 Why was Microsoft Fabric chosen as the central platform?

Fabric was chosen as the hub platform since it provided a unified data engineering, data warehousing, orchestration, and business intelligence strategy. Unlike traditional solutions using separate tools for each layer of the data stack, Fabric unifies ingestion, transformation, storage, and visualization in a single environment. Its native support for Lakehouse architecture (combining the data lake's flexibility and DW's structure), Delta tables, Power BI Embedded, and notebook execution in PySpark or SQL provides an end-to-end solution that fits the needs of firms such as EDULOG, which require agility along with governance

while handling high-volume public datasets.
Fabric also simplifies resource management, offering consumption-based delivery with centralized compute, reducing complexity and simplifying the process of getting all the layers of the platform to operate together under a single governance model.

5.1.2 Why were SQL views and stored procedures preferred over notebooks for modeling?

Although Fabric is designed to deal with diverse languages and interfaces (PySpark notebooks being a case in point), the Gold Layer transformation and modeling logic were specifically coded using SQL views and stored procedures. This is an acceptable choice for several reasons:

- **Maintainability**: SQL offers better readability and normality in the modeling of business logic compared to PySpark notebooks that are more geared towards ingest or computation-intensive workloads.
- **Performance**: SQL actions in Fabric's Warehouse are tuned and include indexing, caching, and query plans.
- **Integration**: Use of SQL views allows for the modeling logic to be simply inserted into semantic models that are consumed by Power BI, with no additional levels of transformation and conversion.

The choice allows business logic transparency, auditing ease, and full correlation with the layer of data visualization.
Also, the Gold Layer will be in the DW, so the used language will be SQL, while PySpark notebooks are better in terms of performance and speed to realize extractions and transformations, much faster compared to SQL queries, as analyzed in the "Decision Guide for Selecting an Analytical Data Store in Microsoft Fabric", [Trofimov n.d.].

5.1.3 Why was Continuous Integration/Continuous Deployment (CI/CD) implemented?

The utilization of CI/CD pipelines (Continuous Integration and Continuous Deployment) within EDU Fabric ensures that the transfer between development (EDU Engineering DEV) and production (EDU Engineering) environments is traceable, controlled, and automated, as analyzed in the "CI/CD for pipelines in Data Factory in Microsoft Fabric", [Microsoft n.d.(b)]. This is particularly important in publicly facing platforms like EDUSTAT and Brighter Future, where data reliability, availability, and consistency are paramount.
With CI/CD, new features — such as tables, views, and pipelines — can be promoted across environments with minimal risk, causing deployments to be repeatable and auditable, and thus reducing human error. Furthermore, teams can develop in parallel, safely experimenting with features prior to promoting to production. This approach enforces platform governance, reduces deployment time, and enables continuous evolution of the analytical system without compromising stability.

5.2 Solution Implementation

The implementation of the recommended solution occurs in the EDU Engineering DEV environment, a dedicated working space in Fabric used for structuring, planning, and performing all development elements. The working space was structured into thematic folders, as shown in Figure 5.1, which categorized activities under data extraction and transformation, orchestration, machine learning experimentations, storage and warehousing, and website integration. Notably, data extraction, transformation, orchestration, storage, and warehousing are crucial for this thesis.

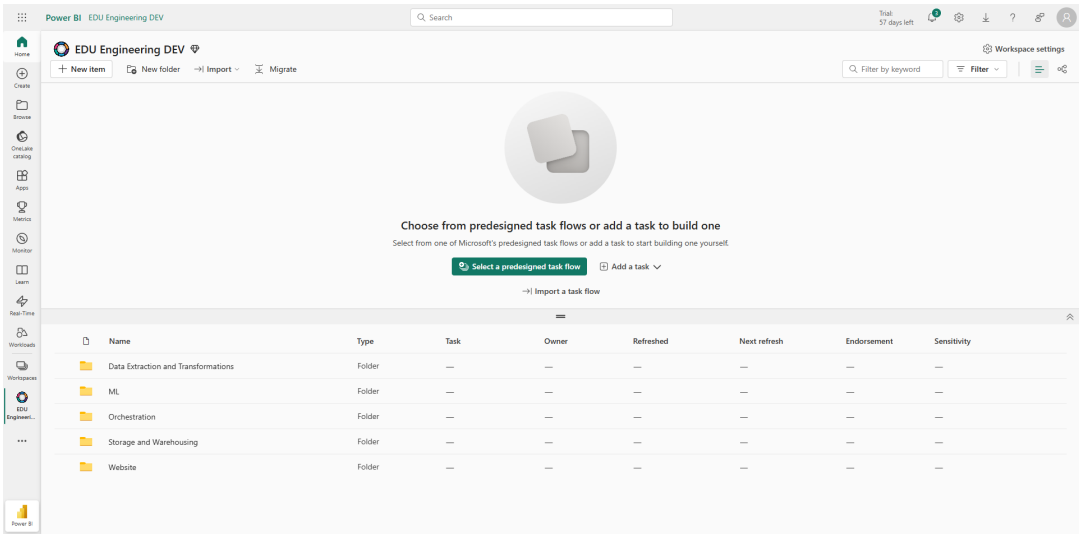


Figure 5.1: EDU Engineering DEV

The environment organizes its content into folders that represent the most significant aspects of the solution: integration, warehousing, orchestration, and data processing. The organization offers modularity and simplicity in managing tasks during development.
Furthermore, this chapter will present the solution's concrete deployment in three major blocks, as previously discussed. Firstly, it accounts for the Storage and Warehousing processes adopted to organize and display information through the EDULAKE and the EDUHOUSE, in other words, the Lakehouse and Data Warehouse (DW), respectively. Secondly, it addresses the Data Extraction and Transformation, where we have all the scripts in PySpark that will perform the transformation and/or extraction. Finally, it accounts for the Orchestration, where we have all the pipelines.
In combination, these components generate the operational nucleus of the system that transforms raw inputs into processed outputs, which can be visualized and analyzed.

5.2.1 Storage and Warehousing

The architecture for the storage and warehousing solution was deployed into the "Storage and Warehousing" dedicated folder in the EDU Engineering DEV environment of Fabric. As shown in Figure 5.2, this folder contains two significant components of the platform: the EDULAKE, which is responsible for the Lakehouse layer, and the EDUHOUSE, which is used for the DW and rendering final business models and reports.

Both parts have different yet complementary roles; the EDULAKE handles the unstructured Bronze Layer with the files that users insert, and the structured Silver Layer operates over

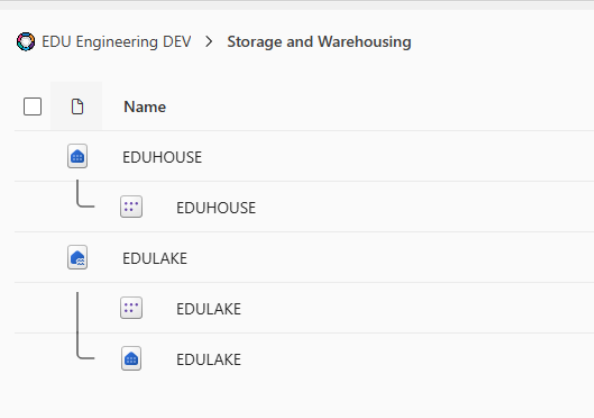


Figure 5.2: Storage and Warehousing

Delta Tables. On the other hand, we have EDUHOUSE, which contains the Gold Layer, expressed in the form of structured tables, following all the transformations. Calculated tables, followed by SQL views, are to be discarded in the Power BI Embedded Integration. The following subsections discuss these components, explaining their functionality, structure, and implementation logic.

EDULAKE

The EDULAKE, as shown in the Figure 5.3, was divided into two different folders: the Tables (for Silver Layer) and the Files (for Bronze Layer).

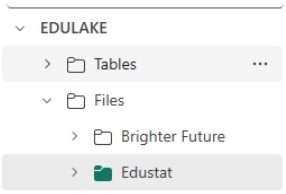


Figure 5.3: EDULAKE Files Components

Moreover, the Files folder is divided into two systems: the Brighter Future and the EDUSTAT folders.
The EDUSTAT folder is the most relevant to this work. The folder is divided into source. Each source contains columns, where each one has its own data type, and includes raw data related to the files. The Lakehouse is unaware of the file content; it only knows that files were stored in the Lakehouse. The only way to determine what is inside the file is through the use of procedures.
One example of data in one file in the Bronze Layer in the Figure 5.4.

```
ano_letivo;nuts1;nuts2;atributo;unidade;valor;dataset
2015/2016;1;;Com AEC;Número;291726;411_XLS
2015/2016;1;;Sem AEC;Número;40038;411_XLS
2015/2016;1;;Alunos por turma com AEC;Número;18.4;414_XLS
2015/2016;1;;Aprendizagem da língua inglesa;Número;132294;412_XLS
2015/2016;1;;Dimensão Europeia na Educação;Número;859;412_XLS
2015/2016;1;;Domínio artístico;Número;175197;412_XLS
2015/2016;1;;Domínio científico;Número;38853;412_XLS
2015/2016;1;;Domínio desportivo;Número;211792;412_XLS
2015/2016;1;;Domínio tecnológico;Número;18768;412_XLS
2015/2016;1;;Ligação da escola com o meio;Número;11432;412_XLS
2015/2016;1;;Solidariedade e Voluntariado;Número;1998;412_XLS
2015/2016;1;;Com AEC;Número;3540;410_XLS
2015/2016;1;;Sem AEC;Número;9;410_XLS
2015/2016;1;;Aprendizagem da língua inglesa;Número;3043;413_XLS
2015/2016;1;;Dimensão Europeia na Educação;Número;29;413_XLS
2015/2016;1;;Domínio artístico;Número;3340;413_XLS
2015/2016;1;;Domínio científico;Número;828;413_XLS
2015/2016;1;;Domínio desportivo;Número;3408;413_XLS
2015/2016;1;;Domínio tecnológico;Número;422;413_XLS
2015/2016;1;;Ligação da escola com o meio;Número;303;413_XLS
2015/2016;1;;Solidariedade e Voluntariado;Número;54;413_XLS
2015/2016;1;4;Com AEC;Número;109790;411_XLS
2015/2016;1;4;Sem AEC;Número;12036;411_XLS
2015/2016;1;4;Alunos por turma com AEC;Número;18.6;414_XLS
2015/2016;1;4;Aprendizagem da língua inglesa;Número;53365;412_XLS
2015/2016;1;4;Dimensão Europeia na Educação;Número;267;412_XLS
2015/2016;1;4;Domínio artístico;Número;68359;412_XLS
2015/2016;1;4;Domínio científico;Número;15614;412_XLS
2015/2016;1;4;Domínio desportivo;Número;81189;412_XLS
2015/2016;1;4;Domínio tecnológico;Número;6269;412_XLS
2015/2016;1;4;Ligação da escola com o meio;Número;3292;412_XLS
2015/2016;1;4;Solidariedade e Voluntariado;Número;465;412_XLS
2015/2016;1;4;Com AEC;Número;1315;410_XLS
2015/2016;1;4;Sem AEC;Número;0;410_XLS
2015/2016;1;4;Aprendizagem da língua inglesa;Número;1187;413_XLS
2015/2016;1;4;Dimensão Europeia na Educação;Número;12;413_XLS
```

Figure 5.4: Bronze Layer Files Example

In the Figure, it is possible to get the information of the "school year", after which nuts1 correspond to the value. Successively, which nuts2, it is possible to see which "attribute", and the corresponding "unit", and the "value", and finally the "dataset" where this entire value was obtained.
In the other folder, we own the Tables folder, where all the content of the Bronze Layer, plus the information that exists in the other two Endpoints (OECD and Eurostat), is stored. In this case, the Lakehouse knows what is in each table.
As can be seen in the Figure 5.5, the Tables folder is divided by schemas, and inside each schema exists one or more Delta Tables.

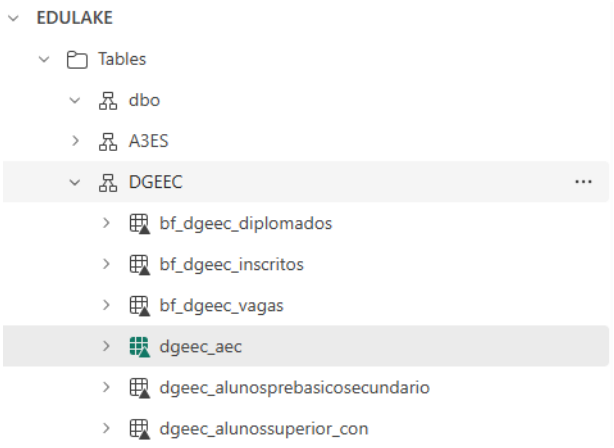


Figure 5.5: EDULAKE Tables Components

To differentiate from the previous process, we are now organizing the content in an agnostic structure due to the diversity of datasets involved, many of which are unrelated to one another. In this layer, all data is standardized under a single structure, regardless of the data source, as already referenced in Chapter 3.
The structure consists of the following fields: RowId, which identifies the original row; Dataset, which allows for grouping data that conceptually belongs together — since a single file may contain multiple datasets; Valor, a flexible field that can represent different types of values depending on the context (e.g., integer, percentage); Filtro, which stores the categorical column name; and finally, Significado, which contains the actual value from that categorical field — not necessarily numeric.
This standardized structure enables uniform processing and transformation of all datasets, regardless of their origin or complexity. An example of this model can be seen in Table 5.1, which is the same dataset shown in the Bronze Layer, but after the application of the transformation notebooks.

	rowid	dataset	valor	filtro	significado
1	2	414_XLS	18.4	ano_letivo	2015/2016
2	23	414_XLS	18.6	ano_letivo	2015/2016
3	65	414_XLS	19.6	ano_letivo	2015/2016
4	86	414_XLS	17.5	ano_letivo	2015/2016
5	105	414_XLS	16.1	ano_letivo	2015/2016
6	124	414_XLS	17.6	ano_letivo	2016/2017
7	147	414_XLS	18.5	ano_letivo	2016/2017
8	170	414_XLS	15.1	ano_letivo	2016/2017
9	191	414_XLS	18.5	ano_letivo	2016/2017
10	235	414_XLS	15.6	ano_letivo	2016/2017
11	258	414_XLS	17	ano_letivo	2017/2018
12	281	414_XLS	16.799999	ano_letivo	2017/2018
13	302	414_XLS	14.5	ano_letivo	2017/2018
14	325	414_XLS	18.9	ano_letivo	2017/2018
15	365	414_XLS	15.8	ano_letivo	2017/2018
16	387	414_XLS	13.9	ano_letivo	2018/2019
17	408	414_XLS	12.7	ano_letivo	2018/2019
18	429	414_XLS	13.8	ano_letivo	2018/2019
19	469	414_XLS	13.2	ano_letivo	2018/2019
20	490	414_XLS	9.6000004	ano_letivo	2018/2019
21	509	414_XLS	15.7	ano_letivo	2019/2020
22	532	414_XLS	15.2	ano_letivo	2019/2020
23	555	414_XLS	14.8	ano_letivo	2019/2020

Table 5.1: Silver Layer Table Example

EDUHOUSE

In Figure 5.6 we can observe all the folders that we have in the EDUHOUSE.

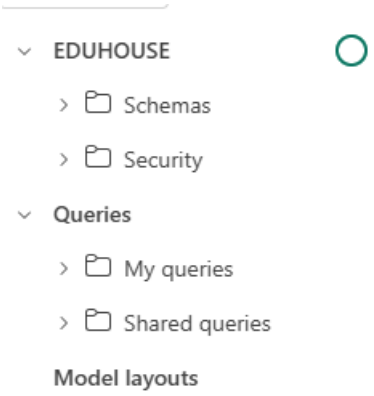


Figure 5.6: EDUHOUSE Components

The most important folder for us is the "Schemas" folder, where we have all the data, as we can see in the Figure 5.7. This data was loaded into the DW.

Figure 5.7: EDUHOUSE Schemas

Everything that has the schema DW is included in the final DW. In this subsection, some of the most essential tables for this project are provided.
In this thesis, we will talk about some of the most essential tables for this process. These tables were: "DW.DIM_FILTROS", and "DW.DIM_INDICADOR", for the realization of calculations, the essential tables were "PARAMS.DEFINICOES_COLUNAS_CALCULADAS_EXISTENTES_MAPPING", "PARAMS.DEFINICOES_COLUNAS_CALCULADAS_EXISTENTES", and "PARAMS.DEFINICOES_FILTROS_EXISTENTES".

Table DW.DIM_FILTROS

This is like a heart table that we have in our DW. In this table, we have listed all the filters that have been created, as shown in the Table 5.2.

	id_sistema	ID_FN	ID_FD	fn	fd	nfn	nfd	ordem_fn	ordem_fd	colapsar	tipofiltro	tipolocal	latitude	longitude	creationdate	alterdate	apagado	bookmark_fd	bookmark_fn
1	1	9	6084	geo	PT1 - Continente	Geografia	Continente	57	2115	0	NULL	1	39.3998720000	-8.2244540000	2025-01-03 15:27:18.84	2025-05-15 14:55:04.40	NULL	NULL	NULL
2	1	6	369	pais	369	País	Portugal	140	1	0	NULL	0	39.3998720000	-8.2244540000	2024-11-22 16:46:10.31	2025-05-15 14:21:57.34	NULL	NULL	NULL
3	1	4	365	nuts2	365	NUTS II	Total	9	8	0	NULL	2	0.0000000000	0.0000000000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
4	1	4	10	nuts2	10	NUTS II	R. A. da Madeira	9	6	0	NULL	2	32.7207500000	-16.9241620000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
5	1	4	9	nuts2	9	NUTS II	R. A. dos Açores	9	7	0	NULL	2	37.7741880000	-25.5573990000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
6	1	4	4	nuts2	4	NUTS II	Norte	9	1	0	NULL	2	41.3735110000	-7.5510930000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
7	1	4	7	nuts2	7	NUTS II	A. M. de Lisboa	9	3	0	NULL	2	38.7729290000	-9.2581280000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
8	1	4	8	nuts2	8	NUTS II	Algarve	9	5	0	NULL	2	37.2425740000	-8.1562140000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
9	1	4	6	nuts2	6	NUTS II	Alentejo	9	4	0	NULL	2	38.5709800000	-7.9096000000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
10	1	4	5	nuts2	5	NUTS II	Centro	9	2	0	NULL	2	39.9569120000	-7.8684880000	2024-11-22 16:46:10.31	2025-05-15 14:53:24.26	NULL	NULL	NULL
11	1	3	364	nuts1	364	NUTS I	Total	8	4	0	NULL	1	0.0000000000	0.0000000000	2024-11-22 16:46:10.31	2025-05-15 14:52:43.84	NULL	NULL	NULL
12	1	3	1	nuts1	1	NUTS I	Continente	8	1	0	NULL	1	39.3998720000	-8.2244540000	2024-11-22 16:46:10.31	2025-05-15 14:52:43.84	NULL	NULL	NULL
13	1	3	3	nuts1	3	NUTS I	R. A. da Madeira	8	2	0	NULL	1	32.7207500000	-16.9241620000	2024-11-22 16:46:10.31	2025-05-15 14:52:43.84	NULL	NULL	NULL
14	1	3	2	nuts1	2	NUTS I	R. A. dos Açores	8	3	0	NULL	1	37.7741880000	-25.5573990000	2024-11-22 16:46:10.31	2025-05-15 14:52:43.84	NULL	NULL	NULL
15	1	5	366	nuts3	366	NUTS III	Total	10	27	0	NULL	3	0.0000000000	0.0000000000	2024-11-22 16:46:10.31	2025-05-15 14:54:39.62	NULL	NULL	NULL
16	1	5	13	nuts3	13	NUTS III	R. A. dos Açores	10	26	0	NULL	3	37.7741880000	-25.5573990000	2024-11-22 16:46:10.31	2025-05-15 14:54:39.62	NULL	NULL	NULL

Table 5.2: DW.DIM_FILTROS

In this figure, we can observe the system's identifier. After that, we have the "ID_FN" and "ID_FD", that is the ID of the filter name and the ID of the filter description, respectively. We have the "fn", the "fd", the "nfn", and the "nfd", in order. They are the filter name and the filter description. The "fn" and "fd" never change, because these columns were the original name and description, and it is to maintain the same information as it was created. After that, we have the "nfn" and the "nfd", which are the new filter name and the new filter description, respectively. This data is different from the "fn" and "fd" because sometimes the user in the backoffice changes the name and the description, so the "nfn" and the "nfd" are different from the original data, these two are shown to the user in the front office.

Table DW.DIM_INDICADOR

There is a row by indicator, and that is where every knowledge starts. When the user chooses which indicator he wants to change, the value he needs to insert to decide which indicator he wants to change is in the column "Indicador".

Table 5.3: DW.DIM_INDICADOR

In this Table, we have many columns, as we can observe in the Table 5.3. Some of them are: the "automatico", where a value of 0 indicates that the data is from a file, and a value of 1 indicates that the data is from an endpoint. The value of the file and the Table is seen in the Lakehouse; other important column is the "principal", this column says if the value is a calculated value or not, if is a calculated value, the value in the "principal" column is 0, if is not the value is 1; other important column is the "ordem_agregado", to create a calculated value we need a value "a" and a value "b", and the calcule is the "a/b", this column tells us which value is the "a", and which is the "b", if the value of "ordem_agregado" is 1, is the value "a", if it is 2, the value is the "b".
As we mentioned, there are calculated values. This process can create calculated indicators, which may generate fictitious values that are not present in the source. When searched for in the database, it should appear in three rows, as shown in Table 5.4, one of which is the principal. The other two are the ones that compose it. From a query, it automatically generates the calculated value. In this case, we used a filter when the indicator is 68.

Table 5.4: Calculated Table

The calculation knowledge was realized from three tables, which we will discuss in the following three subsections.

Table PARAMS.DEFINICOES_COLUNAS_CALCULADAS_ _EXISTENTES_MAPPING

This table is populated automatically in the back office.

	ID	indicador	tabelaa	tabelab	dataseta	datasetb	colunaa	colunab	colunafinal	ativo	creationdate	alterdate
1	505	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	ano_letivo	ano_letivo	ano_letivo	1	2025-06-17 16:05:25.96	NULL
2	511	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	unidade	unidade	unidade	0	2025-06-17 16:05:25.96	2025-06-17 16:13:58.65
3	510	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	recurso	recurso	recurso	0	2025-06-17 16:05:25.96	2025-06-17 16:13:58.19
4	509	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	nuts3	nuts3	nuts3	0	2025-06-17 16:05:25.96	2025-06-17 16:13:57.66
5	508	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	nuts2	nuts2	nuts2	0	2025-06-17 16:05:25.96	2025-06-17 16:13:57.15
6	507	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	nuts1	nuts1	nuts1	0	2025-06-17 16:05:25.96	2025-06-17 16:13:56.52
7	506	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	quadros	natureza	natureza	natureza	0	2025-06-17 16:05:25.96	2025-06-17 16:13:56.03

Table 5.5: Table PARAMS.DEFINICOES_COLUNAS_CALCULADAS_ _EXISTENTES_MAPPING

As we saw in the last subsection, the indicator 68 was a calculated indicator so that we can use that table, Table 5.5, to see which columns of the numerator dataset "a", column "colunaa", will be divided by the columns of the denominator dataset "b", column "colunab", and which columns should connect. It is defined what we will use to connect both datasets. It needs to be the same value, in this case it is the "ano_letivo", because it is the only row that has the column value "ativo" 1, so the calculation is done by the "ano_letivo".

Table PARAMS.DEFINICOES_COLUNAS_CALCULADAS_EXISTENTES

In the Table 5.6 is selected what do you want to include in the division. The column "ordem" can have two different values, "a"/"b", depending on whether it refers to the numerator, "a", or to the denominator, "b".

	ID	indicador	tabela	dataset	ordem	coluna	ativo	creationdate	alterdate
1	1307	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	recurso	1	2025-06-17 16:06:06.74	NULL
2	1306	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	1	2025-06-17 16:06:06.74	NULL
3	1302	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_letivo	1	2025-06-17 16:06:06.74	NULL
4	1308	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	unidade	1	2025-06-17 16:06:06.74	NULL
5	1305	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	1	2025-06-17 16:06:06.74	NULL
6	1304	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts1	1	2025-06-17 16:06:06.74	NULL
7	1303	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	natureza	0	2025-06-17 16:06:06.74	2025-06-17 16:13:59.15
8	1309	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	ano_letivo	1	2025-06-17 16:06:06.74	NULL
9	1315	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	unidade	0	2025-06-17 16:06:06.74	2025-06-17 16:14:02.28
10	1314	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	recurso	0	2025-06-17 16:06:06.74	2025-06-17 16:14:01.73
11	1313	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	nuts3	0	2025-06-17 16:06:06.74	2025-06-17 16:14:01.24
12	1312	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	nuts2	0	2025-06-17 16:06:06.74	2025-06-17 16:14:00.72
13	1311	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	nuts1	0	2025-06-17 16:06:06.74	2025-06-17 16:14:00.24
14	1310	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	natureza	0	2025-06-17 16:06:06.74	2025-06-17 16:13:59.67

Table 5.6: Table PARAMS.DEFINICOES_COLUNAS_CALCULADAS_ _EXISTENTES

In the column "coluna" appears the name of the column that will be used, and the value of the column "ativo" will determine (establish which columns you want to disaggregate information), in other words, you want to see the data by these selected columns, everything that value of "ativo" it is 0 it means you do not wish to see in the final dataset, only the values 1 will appears in the final dataset.
On the other hand, when the value of the column "ativo" is 0 for the denominator, "b", it is because you do not want to use it for the final sum, in this case, the only value where the column "ativo" is 1 is for the column "ano_letivo".

Table PARAMS.DEFINICOES_FILTROS_Calculados_EXISTENTES

The Table 5.7 is used to know what is inside each column, for example, for the column "nuts3", we can see which filters it has, from the column "filtro" or from the column "filtro_descritivo".

	ID	indicador	tabela	dataset	ordem	coluna	filtro	filtro_descritivo	ativo	creationdate	alterdate
1	329604	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	507	Total	0	2025-06-17 16:06:07.81	2025-06-17 16:14:04.88
2	329599	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	31	Tâmega e Sousa	1	2025-06-17 16:06:07.81	NULL
3	329587	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	19	Algarve	1	2025-06-17 16:06:07.81	NULL
4	329601	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	33	Médio Tejo	1	2025-06-17 16:06:07.81	NULL
5	329583	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	14	A. M. do Porto	1	2025-06-17 16:06:07.81	NULL
6	329581	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	11	Alto Alentejo	1	2025-06-17 16:06:07.81	NULL
7	329582	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	12	Alto Tâmega	1	2025-06-17 16:06:07.81	NULL
8	329598	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	30	Beiras e Serra da Estrela	1	2025-06-17 16:06:07.81	NULL
9	329596	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	28	Beira Baixa	1	2025-06-17 16:06:07.81	NULL
10	329588	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	20	Alentejo Litoral	1	2025-06-17 16:06:07.81	NULL
11	329593	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	25	Oeste	1	2025-06-17 16:06:07.81	NULL
12	329585	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	16	Leziria do Tejo	1	2025-06-17 16:06:07.81	NULL
13	329595	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	27	Douro	1	2025-06-17 16:06:07.81	NULL
14	329594	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	26	Região de Aveiro	1	2025-06-17 16:06:07.81	NULL
15	329590	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	22	Alto Minho	1	2025-06-17 16:06:07.81	NULL
16	329602	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	34	Viseu Dão Lafões	1	2025-06-17 16:06:07.81	NULL
17	329600	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	32	Baixo Alentejo	1	2025-06-17 16:06:07.81	NULL
18	329597	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	29	Alentejo Central	1	2025-06-17 16:06:07.81	NULL
19	329584	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	15	Região de Coimbra	1	2025-06-17 16:06:07.81	NULL
20	329603	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	35	Ave	1	2025-06-17 16:06:07.81	NULL
21	329592	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	24	Terras de Trás-os-Montes	1	2025-06-17 16:06:07.81	NULL
22	329591	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	23	Cávado	1	2025-06-17 16:06:07.81	NULL
23	329589	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	21	Região de Leiria	1	2025-06-17 16:06:07.81	NULL
24	329586	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts3	18	A. M. de Lisboa	1	2025-06-17 16:06:07.81	NULL
25	329565	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2017/2018	1	2025-06-17 16:06:07.81	NULL
26	329564	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2016/2017	1	2025-06-17 16:06:07.81	NULL
27	329566	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2018/2019	1	2025-06-17 16:06:07.81	NULL
28	329568	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2020/2021	1	2025-06-17 16:06:07.81	NULL
29	329567	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2019/2020	1	2025-06-17 16:06:07.81	NULL
30	329563	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2015/2016	1	2025-06-17 16:06:07.81	NULL
31	329606	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	unid...	57...	Número	1	2025-06-17 16:06:07.81	2025-06-18 13:16:15.72
32	329570	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2022/2023	1	2025-06-17 16:06:07.81	NULL
33	329605	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	recur...	56...	Rtec_quadros	1	2025-06-17 16:06:07.81	NULL
34	329569	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	ano_...	44...	2021/2022	1	2025-06-17 16:06:07.81	NULL
35	329573	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts1	1	Continente	1	2025-06-17 16:06:07.81	NULL
36	329580	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	506	Total	0	2025-06-17 16:06:07.81	2025-06-17 16:14:04.40
37	329577	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	6	Alentejo	1	2025-06-17 16:06:07.81	NULL
38	329579	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	8	Algarve	1	2025-06-17 16:06:07.81	NULL
39	329574	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts1	505	Total	0	2025-06-17 16:06:07.81	2025-06-17 16:14:03.95
40	329578	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	7	A. M. de Lisboa	1	2025-06-17 16:06:07.81	NULL
41	329571	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	natur...	56...	Público	0	2025-06-17 16:06:07.81	2025-06-17 16:33:44.92
42	329572	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	natur...	18...	Total (Púb+Priv)	1	2025-06-17 16:06:07.81	2025-06-17 16:33:45.73
43	329575	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	4	Norte	1	2025-06-17 16:06:07.81	NULL
44	329576	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	a	nuts2	5	Centro	1	2025-06-17 16:06:07.81	NULL
45	329648	68	[DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]	quadros	b	nuts3	507	Total	0	2025-06-17 16:06:07.81	2025-06-17 16:14:07.41

Table 5.7: Table PARAMS.DEFINICOES_FILTROS_Calculados_EXISTENTES

If it is necessary to remove a filter so that a specific value is not displayed, this can be achieved by setting the value of the "ativo" column to 0. Once deactivated, the system no longer considers that value in subsequent analyses. For example, if values associated with filter 31 should be excluded, setting "ativo" value to 0 removes it from visibility.

Finally, at the end, it will generate a view, this view contains a query that will realize these calculations, Figure 5.8.

```
ALTER VIEW [DW].[V_F_DGEEC_(QUADROS_QUADROS)_100_68] AS

                SELECT DISTINCT
                    ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) as rowid,
                    '(quadros/quadros)*100' as dataset,
                    a.[recurso],a.[nuts3],a.[ano_letivo],a.[unidade],a.[nuts2],a.[nuts1],
                    SUM((a.[valor]/b.[valor])*100) AS valor,
                    CAST(GETDATE() as date) as creationdate
                FROM [DW].[F_DGEEC_RTECPREBASICOSECUNDARIO] a
                INNER JOIN
                    ( SELECT dataset, ano_letivo, SUM(valor) as valor FROM [DW].[F_DGEEC_RTECPREBASICOSECUNDARIO]WHERE natureza NOT IN (5684) AND nuts1 NOT IN (505) AND nuts2 NOT IN (506) AND nuts3 NOT IN (507) GROUP BY dataset, ano_letivo) b
                ON a.[ano_letivo] = b.[ano_letivo]
                WHERE
                    a.[dataset] = 'quadros'
                    AND b.[dataset] = 'quadros'
                    AND b.[valor] IS NOT NULL AND b.[valor] != 0
                    AND a.[valor] IS NOT NULL AND a.[valor] != 0
                    AND (a.[natureza] NOT IN (5684) AND a.[nuts1] NOT IN (505) AND a.[nuts2] NOT IN (506) AND a.[nuts3] NOT IN (507))
                GROUP BY a.[recurso],a.[nuts3],a.[ano_letivo],a.[unidade],a.[nuts2],a.[nuts1]

GO
```

Figure 5.8: Query in View

And we can see the result in the column "valor", after the calculation, Table 5.8.

	rowid	dataset	recurso	nuts3	ano_letivo	unidade	nuts2	nuts1	valor	creationdate
1	1	(quadros/quadros)*100	56890	19	4478	5734	8	1	2.499391875456093408	2025-06-18
2	2	(quadros/quadros)*100	56890	16	4480	5734	6	1	3.328162459455648005	2025-06-18
3	3	(quadros/quadros)*100	56890	20	4482	5734	6	1	1.91069890020779484	2025-06-18
4	4	(quadros/quadros)*100	56890	32	4481	5734	6	1	1.657430730478589421	2025-06-18
5	5	(quadros/quadros)*100	56890	25	4483	5734	5	1	2.788222363625846011	2025-06-18
6	6	(quadros/quadros)*100	56890	34	4480	5734	5	1	3.929864147040849904	2025-06-18
7	7	(quadros/quadros)*100	56890	30	4478	5734	5	1	1.05509608367793724 2	2025-06-18
8	8	(quadros/quadros)*100	56890	15	4478	5734	5	1	2.435538798345901241	2025-06-18
9	9	(quadros/quadros)*100	56890	21	4481	5734	5	1	3.561712846347607053	2025-06-18
10	10	(quadros/quadros)*100	56890	28	4484	5734	5	1	0.566244520214320507	2025-06-18
11	11	(quadros/quadros)*100	56890	26	4483	5734	5	1	4.702955978480939434	2025-06-18
12	12	(quadros/quadros)*100	56890	27	4482	5734	4	1	2.914195935330191070	2025-06-18
13	13	(quadros/quadros)*100	56890	31	4484	5734	4	1	4.657817827569410619	2025-06-18
14	14	(quadros/quadros)*100	56890	12	4482	5734	4	1	0.587907353910090720	2025-06-18
15	15	(quadros/quadros)*100	56890	31	4477	5734	4	1	3.347436535589845694	2025-06-18
16	16	(quadros/quadros)*100	56890	14	4483	5734	4	1	17.174755596691155203	2025-06-18
17	17	(quadros/quadros)*100	56890	23	4480	5734	4	1	3.943966530343628073	2025-06-18
18	18	(quadros/quadros)*100	56890	35	4479	5734	4	1	5.177892379017554826	2025-06-18
19	19	(quadros/quadros)*100	56890	18	4483	5734	7	1	15.560826054260426910	2025-06-18
20	20	(quadros/quadros)*100	56890	11	4478	5734	6	1	0.839211870591097057	2025-06-18
21	21	(quadros/quadros)*100	56890	29	4479	5734	6	1	2.614991925821742981	2025-06-18
22	22	(quadros/quadros)*100	56890	33	4480	5734	5	1	2.467917077986179664	2025-06-18
23	23	(quadros/quadros)*100	56890	22	4480	5734	4	1	3.408075964838057632	2025-06-18

Table 5.8: View

5.2.2 Data Extraction and Transformation

In this folder, we have the notebooks of PySpark, as we can observe in the Figure 5.9, which are responsible for realizing the transformation, or in OECD and Eurostat cases for realizing the extraction and the transformation, that only contains the code used for the operation.

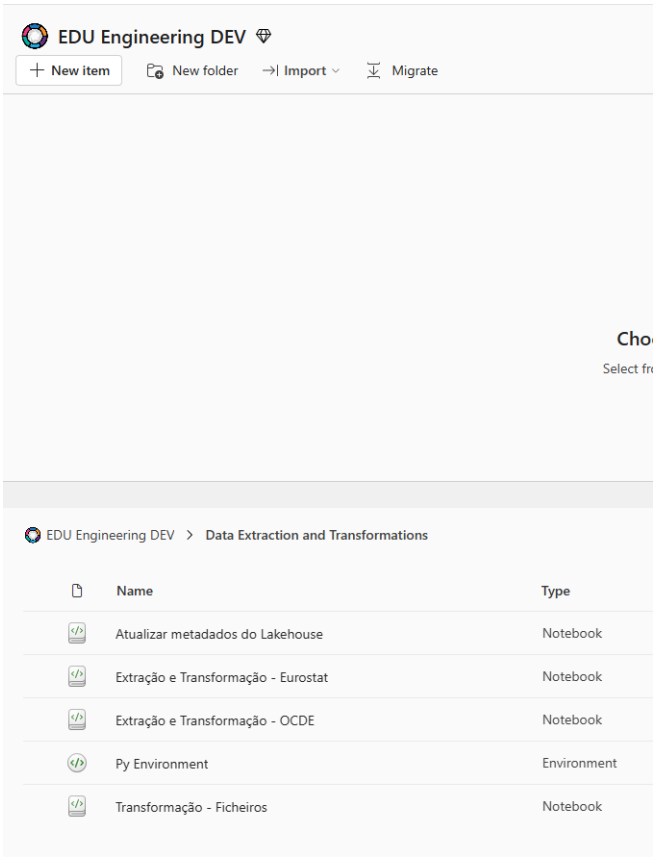


Figure 5.9: Data Extraction and Transformation

The code generally transforms the file's content into a Delta Table structure. On the other hand, in the OECD and Eurostat cases, it proceeds to the endpoint and extracts all the data, after which it materializes the transformed data into a Delta Table.
The following two subsections will present two of these notebooks.

Eurostat Extraction and Transformation

This process implements an automated extraction method and transformation of the data from the Eurostat API (endpoint) into the Fabric's platform.

```
1  from pyspark.sql import SparkSession
2  from pyspark.sql.functions import col, expr, monotonically_increasing_id
3  from pyspark.sql import functions as F
4  from notebookutils import mssparkutils
5  import os
6  import pandas as pd
7  import eurostat
8
9  # Initialize Spark Session (if not already done in Fabric)
10 spark = SparkSession.builder.getOrCreate()
```

Figure 5.10: Libraries Initialization Endpoint

The notebook begins with the initialization of some libraries, as shown in Figure 5.10. After that, we proceed by retrieving the dataset using the "get_data(dataset)" function, as illustrated in Figure 5.11. This function connects to the Eurostat API and retrieves the

specified dataset code.

```python
def get_data(dataset):

    # Get data

    dataframe = eurostat.get_data_df(dataset)

    # Rename 'geo\\TIME_PERIOD' column to 'geo'

    if 'geo\\TIME_PERIOD' != None:
        dataframe.rename(columns={'geo\\TIME_PERIOD': 'geo'}, inplace=True)

    # Add translations to acronyms

    dics = {param: dict(eurostat.get_dic(dataset, param, lang='en')) for param in eurostat.get_pars(dataset)}

    for param, dic in dics.items():
        if param == 'geo':
            # Replace with "Key - Value" format
            dataframe[param] = dataframe[param].map(lambda x: f"{x} - {dic[x]}" if x in dic else x)
        else:
            # Standard replacement
            dataframe[param] = dataframe[param].replace(dic)

    years = eurostat.get_par_values(dataset, 'TIME_PERIOD')

    pivot_columns = [column for column in dataframe.columns if column in years]

    if len(pivot_columns) > 0:
        # Create 'year' and 'valor' columns
        id_columns = [column for column in dataframe.columns if column not in pivot_columns]
        dataframe = dataframe.melt(id_vars=id_columns, value_vars=pivot_columns, var_name='year', value_name='valor')

    # Add 'dataset' column and row ID
    dataframe['dataset'] = dataset
    dataframe['rowid'] = range(1, len(dataframe) + 1)

    dataframe.columns = dataframe.columns.str.lower()

    return dataframe
```

Figure 5.11: Function get_data(dataset)

Once extracted, a different type of transformation will occur, starting with the modification of the "geo\\TIME_PERIOD" column to "geo", which creates a translation for the acronyms, creation of two rows, "year" and "valor", as shown in Figure 5.11, and finalizing by adding the "dataset" and "rowid" columns.

```python
def filters(unpivoted_dff, dataset):

    # Load filter definition tables
    df1 = spark.read.format("delta").load("Tables/PARAMS/DEFINICOES_FILTROS_EUROSTAT_EXISTENTES")
    df2 = spark.read.format("delta").load("Tables/PARAMS/DEFINICOES_COLUNAS_EUROSTAT_EXISTENTES")

    # Filter for active filters in df1 and df2 for the given dataset
    active_filters = df1.filter((df1['dataset'] == dataset) & (df1['ativo'] == 1))
    active_columns = df2.filter((df2['dataset'] == dataset) & (df2['ativo'] == 1)).select('filtro').distinct()

    if active_columns.count() > 0 and active_filters.count() > 0:

        # Step 1: Filter out rows with inactive columns (filtro)
        unpivoted_dff = unpivoted_dff.join(active_columns, on='filtro', how='inner')

        # Step 2: Identify rowids where any significado is inactive
        inactive_significados = df1.filter((df1['dataset'] == dataset) & (df1['ativo'] == 0))

        rowids_to_exclude = unpivoted_dff.join(
            inactive_significados,
            on=['filtro', 'significado'],
            how='inner'
        ).select('rowid').distinct()

        # Step 3: Exclude entire rows (rowid) with inactive significados
        if rowids_to_exclude.count() > 0:
            unpivoted_dff = unpivoted_dff.join(
                rowids_to_exclude,
                on='rowid',
                how='left_anti'
            )

        # Final count and output
        num_rows = unpivoted_dff.count()

        print(f'Number of rows after filtering: {num_rows}')

    return unpivoted_dff
```

Figure 5.12: Function filters(unpivoted_dff, dataset)

After the transformation part, the data will be loaded for the definition tables, as we can observe in Figure 5.12, and the next step involves unpivoting the dynamic columns, where the conversion of different columns into different rows will happen, following a standardized data model of EDU Fabric, as is pretended in the Figure 5.13.

```python
def unpivot_df(dataframe, dataset):

    # Read the dataframa into a Spark DataFrame
    df = spark.createDataFrame(dataframe)

    # Define the fixed columns
    fixed_columns = ['rowid', 'dataset', 'valor']

    # Dynamically identify the columns that need to be unpivoted
    dynamic_columns = [col for col in df.columns if col not in fixed_columns]

    # Unpivot the dynamic columns using Spark's 'melt' equivalent
    unpivot_expr = F.expr(
        f"stack({len(dynamic_columns)}, " + ", ".join([f"'{col}', `{col}`" for col in dynamic_columns]) + ")"
    ).alias("filtro", "significado")

    unpivoted_dff = df.select(fixed_columns + [unpivot_expr])
    num_rows = unpivoted_dff.count()
    print(f'Number of rows before filtering: {num_rows}')

    unpivoted_df = filters(unpivoted_dff,dataset)

    # Creating the lakehouse schema if it doesn't exist
    spark.sql(f"CREATE SCHEMA IF NOT EXISTS EUR ")

    # Write the unpivoted DataFrame to the lakehouse table using the correct schema and table name
    table_name = 'EUR.eur_' + dataset.lower()

    unpivoted_df.write.mode('overwrite').format("delta").saveAsTable(table_name)
    print(f"Data written to table: {table_name}")

    global lakehousetables
    lakehousetables.append(table_name)
```

Figure 5.13: Function unpivot_df(dataframe, dataset)

Finally, all the transformed data will be written into the Lakehouse, to the Silver Layer, in Delta Table format, creating or updating the final table dynamically, and the structure will follows the convention "EUR.eur_<dataset>", as shown in the Figure 5.13.
The entire process is parameterized so that the notebook can be run iteratively on a list of datasets simply by adjusting the datasets_list variable, as shown in the Figure 5.14. Upon running, the notebook provides a list of generated tables ready to be utilized further in the Fabric pipelines.

```python
global lakehousetables

datasets_list = datasets.split(',')

for dataset in datasets_list:

    # print(dataset)
    dataframe = get_data(dataset)
    unpivot_df(dataframe, dataset)

mssparkutils.notebook.exit(str(lakehousetables))
```

Figure 5.14: Function datasets_list

Files Transformation

This notebook implements an automated process that transforms the raw data that is in the Bronze Layer, specifically Excel and CSV files, into structured Delta Tables in the Silver Layer.

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, expr, monotonically_increasing_id
from pyspark.sql import functions as F
from notebookutils import mssparkutils
import os
import pandas as pd

# Initialize Spark Session (if not already done in Fabric)
spark = SparkSession.builder.getOrCreate()
```

Figure 5.15: Libraries Initialization Files

First, the notebook initializes a Spark session and imports the necessary libraries, as is shown in the Figure 5.15, to process different file types and convert them into a normalized unpivoted format.

```python
def unpivot_xlsx_file(file_path, source_folder, file_name):
    # Read the Excel file into a Pandas DataFrame
    df = pd.read_excel(file_path)

    # Add a row number (Identity Column) using the DataFrame's index
    df['rowid'] = range(1, len(df) + 1)

    # Define the fixed columns
    fixed_columns = ['rowid', 'dataset', 'valor']

    # Dynamically identify the columns that need to be unpivoted
    dynamic_columns = [col for col in df.columns if col not in fixed_columns]

    # Unpivot the dynamic columns using Pandas' melt function
    unpivoted_df = pd.melt(df, id_vars=fixed_columns, value_vars=dynamic_columns,
                           var_name='filtro', value_name='significado')

    # Disable Arrow optimization
    spark.conf.set("spark.sql.execution.arrow.pyspark.enabled", "false")

    # Convert the Pandas DataFrame to a PySpark DataFrame
    unpivoted_df_spark = spark.createDataFrame(unpivoted_df)

    # Creating the lakehouse schema if it doesn't exist
    spark.sql(f"CREATE SCHEMA IF NOT EXISTS {source_folder}")

    # Write the unpivoted DataFrame to the lakehouse table using the correct schema and table name
    table_name = source_folder + '.' + file_name.lower()

    unpivoted_df_spark.write.mode('overwrite').format("delta").saveAsTable(table_name)
    print(f"Data written to table: {table_name}")

    global lakehousetables
    lakehousetables.append(table_name)
```

Figure 5.16: Function unpivot_xlsx_file(file_path, source_folder, file_name)

There are two different core transformation functions defined. The first one, called "unpivot_xlsx_file()", is for Excel data, as can be observed in Figure 5.16. This function imports an Excel file into a DataFrame and adds a unique "rowid" to each row. It specifies fixed rowid, dataset, and valor columns, and dynamically selects and unpivots the remaining columns by invoking the Pandas melt() function. It defines two generic fields, filtro and significado, that accommodate the EDU Fabric tabular model. The generated dataset is

then written out in Delta format to the Lakehouse in the proper schema and table name.

```python
def unpivot_csv_file(file_path, source_folder, file_name):

    global lakehousetables

    # Write the unpivoted DataFrame to the lakehouse table using the correct schema and table name
    table_name = source_folder + '.' + file_name.lower()

    # Read the CSV file into a Spark DataFrame
    df = spark.read.option("header", True).option("delimiter", ";").option("encoding", "UTF-8").csv(file_path)

    # Add a row number (Identity Column) using monotonically_increasing_id
    df = df.withColumn("rowid", monotonically_increasing_id())

    # Rename all columns to lowercase
    df = df.toDF(*[col.lower() for col in df.columns])

    # If the file which is being iterated is not from the LABELS folder, and does not start with 'BF_LB'
    if file_name[:5] != 'BF_LB' and source_folder != 'LABELS':

        # Define the fixed columns
        fixed_columns = ['rowid', 'dataset', 'valor']

        # Dynamically identify the columns that need to be unpivoted
        dynamic_columns = [col for col in df.columns if col not in fixed_columns]

        # Unpivot the dynamic columns using Spark's 'melt' equivalent
        unpivot_expr = F.expr(
            f"stack({len(dynamic_columns)}, " + ", ".join([f"'{col}', `{col}`" for col in dynamic_columns]) + ")"
        ).alias("filtro", "significado")

        df = df.select(fixed_columns + [unpivot_expr])

        lakehousetables.append(table_name)

    # If the file is from the LABELS folder but isn't the BF_LB_LABLES.csv file, the process should return it's mentioning
    elif source_folder == 'LABELS' and file_name != 'BF_LB_LABELS':

        lakehousetables.append(table_name)

    # Creating the lakehouse schema if it doesn't exist
    spark.sql(f"CREATE SCHEMA IF NOT EXISTS {source_folder}")

    df.write.mode('overwrite').format("delta").saveAsTable(table_name)
    print(f"Data written to table: {table_name}")
```

Figure 5.17: Function unpivot_csv_file(file_path, source_folder, file_name)

In Figure 5.17 we have the other function, "unpivot_csv_file()", which loads a CSV file as a Spark DataFrame, with the correct encoding and delimiters. It generates a unique "rowid" using Spark's "onotonically_increasing_id()" method and all column names converted to lowercase for normalization. In non-classification files, it dynamically drops columns into the filtro and significado fields using Spark expressions. The table is then saved as a Delta Table to the indicated Lakehouse schema.

```python
def process_lakehouse_files(search_directories_path, locate_file_path, filenames, id_sistema):

    # Passing the filenames string to dataframe array
    # filesdf = spark.createDataFrame([(filenames,)], ["files"])

    # # Convert the comma-separated string to an array
    # filesarray = filesdf.withColumn("filesarray", F.split(F.col("files"), ","))


    # List all source folders in the lakehouse directory
    source_folders = os.listdir(search_directories_path)

    for source_folder in source_folders:

        print('Searching folder ' + source_folder + '...')
        readyfolder = os.listdir(search_directories_path + '/' + source_folder + '/' + 'Ready/')

        for readyfile in readyfolder:
            file_path = locate_file_path + '/' + source_folder + '/' + 'Ready/' + readyfile
            table_path = locate_file_path + '/' + source_folder + '/' + 'Ready/' + source_folder + '.'

            if (readyfile in filenames) or (id_sistema == 2 and readyfile == 'BF_LB_LABELS.csv'):

                if readyfile.endswith('.csv'):  # Only process CSV files
                    file_name = readyfile.replace('.csv', '')  # File name without extension
                    table_path = table_path + file_name

                    # Unpivot the CSV file and write to the appropriate table in the correct schema
                    print(f'Processing file: {file_path} for table {file_name}')
                    unpivot_csv_file(file_path, source_folder, file_name)

                elif readyfile.endswith('.xlsx'):  # Only process XLSX files
                    file_name = readyfile.replace('.xlsx', '')  # File name without extension
                    table_path = table_path + file_name

                    # Unpivot the XLSX file and write to the appropriate table in the correct schema
                    print(f'Processing file: {file_path} for table {file_name}')
                    unpivot_xlsx_file(file_path, source_folder, file_name)
```

Figure 5.18: Function process_lakehouse_files(search_directories_path, locate_file_path, filenames, id_sistema)

The main orchestration function, process_lakehouse_files(), as is in Figure 5.18, scans the given input directories, going through subdirectories and picking up the ones marked as "Ready" for processing. Each filename is then compared against a static list (ficheiros), and the appropriate transformation logic is applied depending on the file type. Specific files are treated specially: those in the "LABELS" directory or with filenames starting with "BF_LB" are marked and processed separately to deal with their unique structure. All processed files are dynamically written to Delta Tables in the Lakehouse based on a naming convention related to their source folder name and file name.

5.2.3 Orchestration

Inside the folder of the Orchestration, we have three options, as shown in Figure 5.19, but only the last 2 were essential for us, the "0 - Orquestração Geral", and the "EDUSTAT".

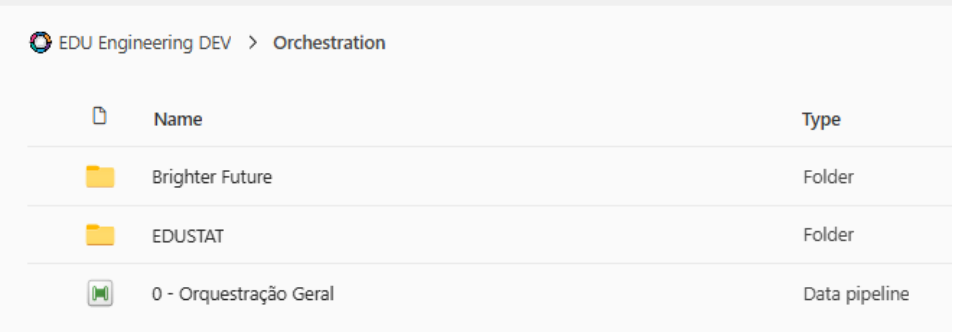


Figure 5.19: Orchestration

These two options will be presented in more detail below.

0 - General Orchestration (0 - Orquestração Geral)

In Figure 5.20 we have the structure of the most essential orchestrator, which is the principal pipeline, and is the one that runs when the user, in the back office, wants to run the ETL. That is one parameter sent to the user when he runs, and that is the indicators, separated by commas.

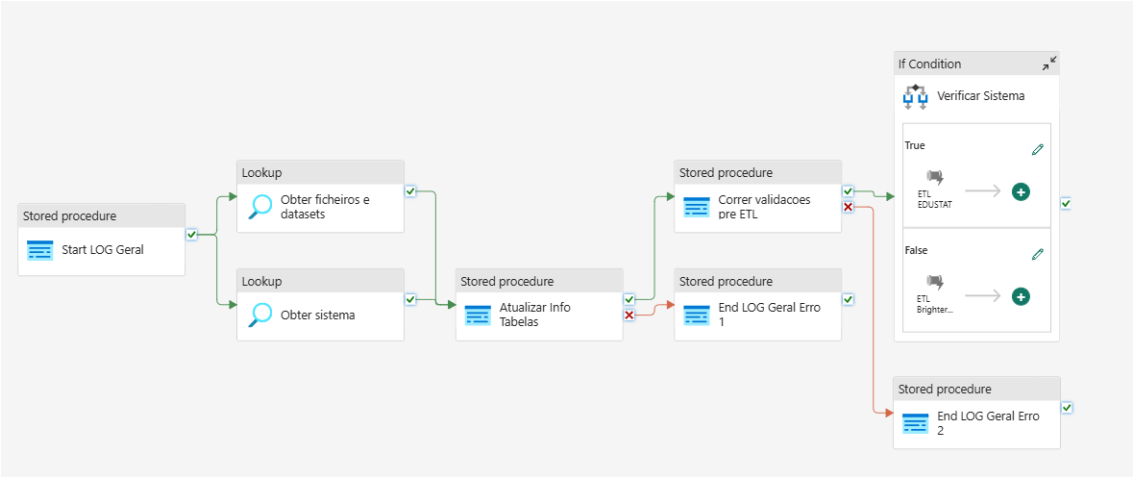


Figure 5.20: 0 - Orquestração Geral

After that, two lookups run. The "Obter Sistema" lookup takes every single indicator sent by the user and determines which one is their system. In the case that the indicators are from different systems, if some of them are from Brighter Future and the others are from the Edustat mix, the system returns an error. It does not work because the pipeline only works for one system at a time, so the user needs to run it once per system. The other lookup will check which files and datasets are associated with each indicator, in this way, the process knows that it needs to run this file.
It is possible to run tasks from Eurostat and files simultaneously in the same ETL. This lookup will take a distinct approach to the files and datasets that will be extracted.
The next step will insert the "tabela" column, whose name will be. In case one of the results of this process is an error, an automatic error message is sent. In case no error is obtained, the "Correr validacoes pre ETL" step is run, with thirteen validations, every single one to check if the user made any error or not, in the population process of the fields. In case of a mistake, one message in portuguese is sent with the pipeline and the error. In case everything is right, the system runs the last step, and that is an if step, depending on the system that the user ran, if it is the EDUSTAT, the answer is true and runs the "ETL EDUSTAT", otherwise the answer is false. It runs the "ETL Brighter Future".

EDUSTAT

All the pipelines that exist in the EDUSTAT folder are in the Figure 5.21, the number on the left of the name shows us how deep they are, being the pipelines children of the general pipeline, the pipeline 0 invokes the 1, the 1 invokes the 2, and so on.

Pipeline 0 serves as the header of EDUSTAT, as illustrated in Figure 5.22. It initiates the process by invoking other pipelines, specifically the level 1 pipeline, which sends the datasets

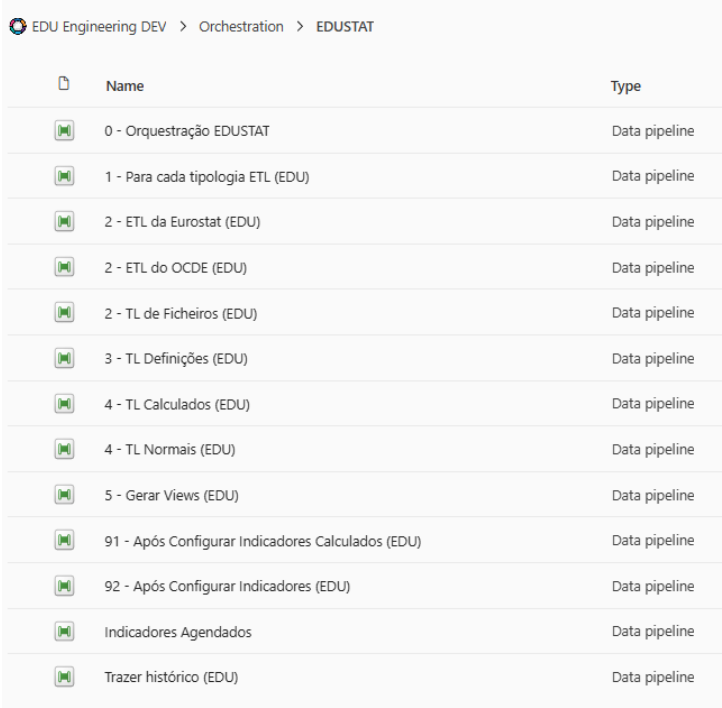


Figure 5.21: EDUSTAT Orchestration Pipelines

to be extracted.
This pipeline is the first and main one in the entire process, serving as the heart of the process, and it outlines the steps to follow for populating the EDUSTAT tables.

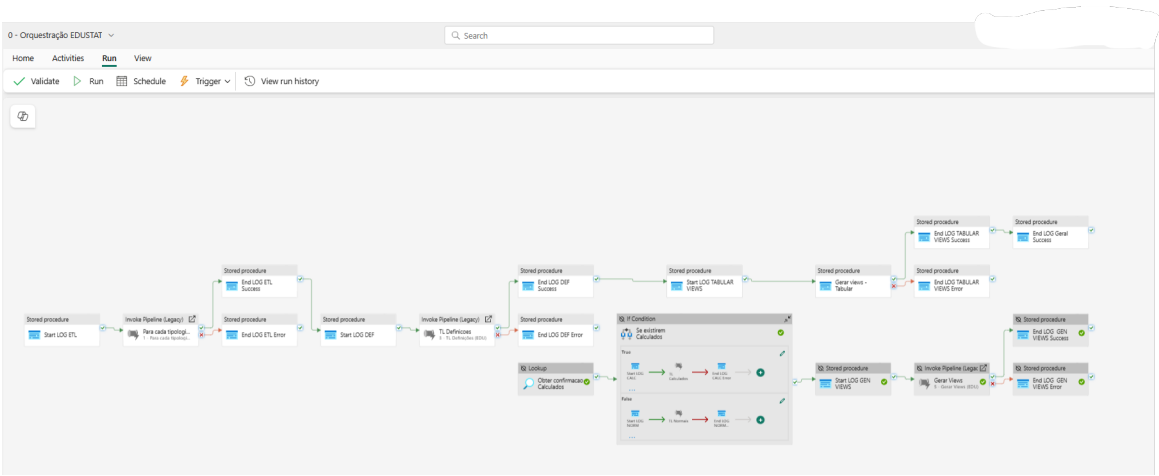


Figure 5.22: 0 - EDUSTAT Orchestration

Pipeline 1, Figure 5.23, starts with a loop, that is a for each, and will iterate over each tipology. If the field of a file is empty, it means it comes from an endpoint, and that is the first step of the loop; if it is a file, it runs the second step of the loop; otherwise, if it comes from an endpoint, and it is from the Eurostat it will run the third step, or if it is from the OECD runs the last step of the loop. After that process, it will run more three steps, the first one is to insert the NUTS, the second one is to update the PARAMS of existing

columns, and the last one is to insert the totals.
The term "NUTS" is a Portuguese term for "Nomenclatura das Unidades Territoriais para Fins Estatísticos", or in English "Nomenclature of Territorial Units for Statistical Purposes", used by the EUROSTAT, and are the regional divisions existing in all member states of the European Union. For Portugal, there are three different NUTS types: NUTS 1, NUTS 2, and NUTS 3.
The NUTS 1 is composed of three units, corresponding to the mainland and each of the Autonomous Regions of the Azores and Madeira. The NUTS 2 is composed of nine units, seven of which are on the mainland and the territories of the Autonomous Regions of the Azores and Madeira. Moreover, the NUTS 3 is composed of 26 units, 24 of which are located on the mainland, and two in the Autonomous Regions of the Azores and Madeira, corresponding to the Intermunicipal Entities [Wikipedia n.d.]. This information was obtained from the "NUTS de Portugal".

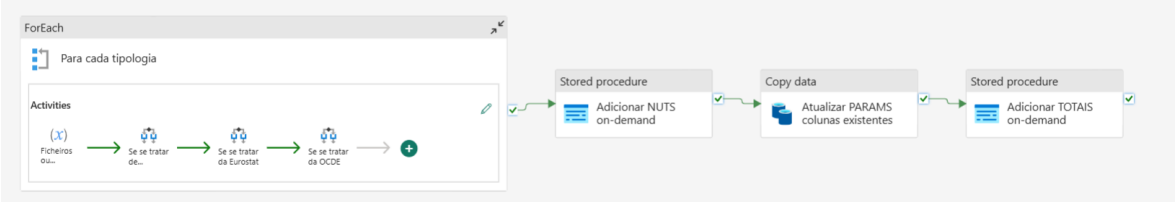


Figure 5.23: 1 - For each ETL typology (EDU)

We have three level 2 pipelines: two for realizing the ETL pipelines for the endpoints, and the other is a TL pipeline for the files, which we will analyze.

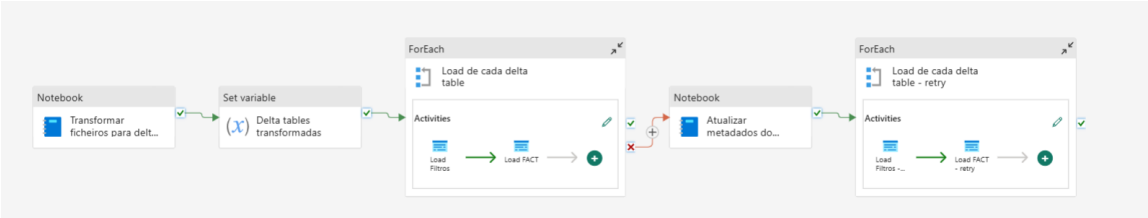


Figure 5.24: 2 - TL Files (EDU)

Figure 5.24 shows us the pipeline of the Transform and Load of the files. First, the parameters, dataset, or file are sent to the notebook, where it will run and transform the data in the file into a Delta Table. After that, the system inserts the name of the file and runs the T into the DW, where it first inserts the filters, which include those that do not exist, and automatically creates a new ID. After that, the fact load starts, which begins by deleting existing tables and creating a new one in the DW, already associated with the filters (all procedures are located in EDUHOUSE, within the DW schema).
Both steps forward were just a repetition of the back steps, because sometimes some errors are not the fault of the process. However, if there is an error with the Fabric, the back office will receive the error. It will report it to the team. This happens frequently, especially in the files, due to the incorrect insertion of data, as some columns are mandatory but do not contain any data.
Backing into the pipeline 0, Figure 5.22, runs one step that will call one pipeline level 3, called "TL Definicoes", that was a set of procedures, as shown in Figure 5.25, that will feed all the existing definitions tables, for example the calculated tables, previously presented.

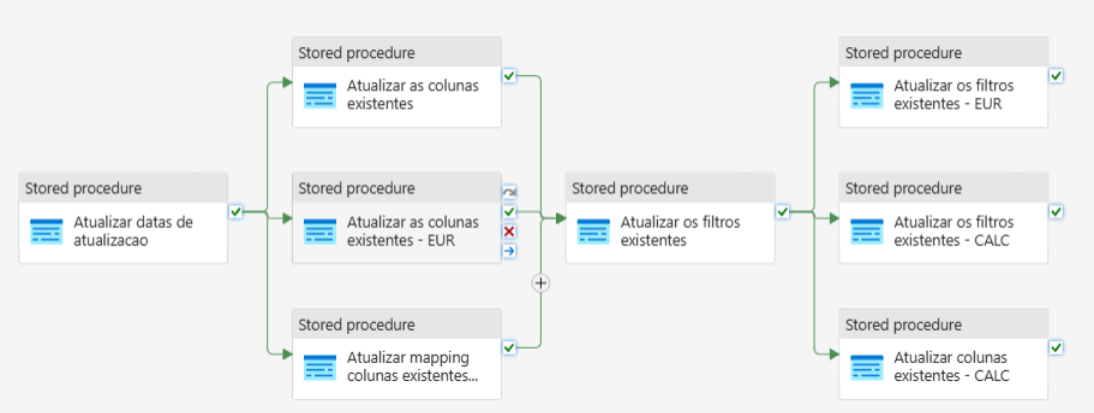


Figure 5.25: 3 - TL Definitions (EDU)

There are two essential tables, the "DEFINICOES_COLUNAS_EXISTENTES", Table 5.9, which will have the final product, and this table only has the indicators that go to the website, the "ativo" column means whether I want to show or not the value (the users can efficiently inactivate these columns, need to change the value to 0), generating a final indicator view, this is an improvement implementation, that will improve the website, suggested by the team, and not one of Fabric's characteristics.
These definitions will populate the entire dataset.

	ID	indicador	tabela	dataset	coluna	ativo	comparavel	creationdate	alterdate
1	2162	23	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-24 16:16:23.10	NULL
2	2163	23	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-24 16:16:23.10	NULL
3	2055	25	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	ano_letivo	1	0	2025-01-22 10:01:06.71	NULL
4	2056	25	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	unidade	1	0	2025-01-22 10:01:06.71	NULL
5	2189	9	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-28 12:25:45.18	NULL
6	2190	9	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-28 12:25:45.18	NULL
7	2309	27	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 14:48:21.76	2025-02-24 11:16:41.68
8	2197	10	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 11:26:50.43	NULL
9	2198	10	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-30 11:26:50.43	NULL
10	2201	11	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 11:26:50.43	NULL
11	2202	11	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-30 11:26:50.43	NULL
12	2203	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 11:26:50.43	NULL
13	2204	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-30 11:26:50.43	NULL
14	2205	25	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 11:26:50.43	NULL
15	2206	25	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-30 11:26:50.43	NULL
16	2209	7	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts1	1	0	2025-01-30 11:26:50.43	NULL
17	2210	7	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nuts2	1	0	2025-01-30 11:26:50.43	NULL
18	2069	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	ano_letivo	1	0	2025-01-22 10:14:05.51	NULL
19	2070	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	faixa_etaria	1	0	2025-01-22 10:14:05.51	NULL
20	2071	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	genero	1	0	2025-01-22 10:14:05.51	NULL
21	2072	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	habilitacoes_academicas	1	0	2025-01-22 10:14:05.51	NULL
22	2073	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	nacionalidade_estrangeira	1	0	2025-01-22 10:14:05.51	NULL
23	2074	24	[DW].[F_DGEEC_DOCENTESSUPERIOR]	DocentesSup	natureza	1	0	2025-01-22 10:14:05.51	NULL

Table 5.9: Table PARAMS.DEFINICOES_COLUNAS_EXISTENTES

The other important definition table was the "DEFINICOES_FILTROS_EXISTESTES", Table 5.10 was equal to the other table, but in this case, for the filters, where the inactivation of the filters can also occur, a group by is not used.

	ID	indicador	tabela	dataset	coluna	filtro	padrao	cor	ativo	creationdate	alterdate
1	261713	3	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	7	0	#df6298	1	2025-02-20 18:27:13.49	NULL
2	255392	4	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	7	0	#df6298	1	2025-02-10 17:43:35.76	NULL
3	265291	19	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	7	0	#df6298	1	2025-02-25 17:22:34.62	NULL
4	274726	468	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	7	0	#df6298	1	2025-02-27 15:41:51.50	NULL
5	274960	486	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	7	0	#df6298	1	2025-03-06 11:51:25.54	NULL
6	265288	19	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	4	0	#a762df	1	2025-02-25 17:22:34.62	NULL
7	274957	486	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	4	0	#a762df	1	2025-03-06 11:51:25.54	NULL
8	274723	468	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	4	0	#a762df	1	2025-02-27 15:41:51.50	NULL
9	261710	3	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	4	0	#a762df	1	2025-02-20 18:27:13.49	NULL
10	255389	4	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	4	0	#a762df	1	2025-02-10 17:43:35.76	NULL
11	255391	4	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	6	0	#c4df62	1	2025-02-10 17:43:35.76	NULL
12	261712	3	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	6	0	#c4df62	1	2025-02-20 18:27:13.49	NULL
13	274725	468	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	6	0	#c4df62	1	2025-02-27 15:41:51.50	NULL
14	274959	486	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	6	0	#c4df62	1	2025-03-06 11:51:25.54	NULL
15	265290	19	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	6	0	#c4df62	1	2025-02-25 17:22:34.62	NULL
16	255393	4	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	8	0	#62dddf	1	2025-02-10 17:43:35.76	NULL
17	265292	19	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	8	0	#62dddf	1	2025-02-25 17:22:34.62	NULL
18	261714	3	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	8	0	#62dddf	1	2025-02-20 18:27:13.49	NULL
19	274727	468	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	8	0	#62dddf	1	2025-02-27 15:41:51.50	NULL
20	274961	486	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	8	0	#62dddf	1	2025-03-06 11:51:25.54	NULL
21	255394	4	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	9	0	#00ada1	1	2025-02-10 17:43:35.76	NULL
22	261715	3	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	9	0	#00ada1	1	2025-02-20 18:27:13.49	NULL
23	274962	486	[DW].[F_DGEEC_DOCENTESPREBASICOSECUNDARIO]	DocentesPBS	nuts2	9	0	#00ada1	1	2025-03-06 11:51:25.54	NULL

Table 5.10: Table PARAMS.DEFINICOES_FILTROS_EXISTENTES

After all this process, the generation of the views into the tabular form happens, which will originate the Power BIs, which are generated by the procedures that were well defined in "EDUSITE.V", Figure 5.26, these views are the ones that are implemented on the site and realize the aggregations.

Figure 5.26: EDUSITE.VIEWS

We have the "TABULAR.V", Figure 5.27, generates all the columns in Power BI, one of the implementations, made at these tables, is that instead of appearing the ID column or the number, because we do not know what a simple number means, it does a JOIN, which is in the tabular schema. For each column that has an ID, it does a JOIN with the table "DW.DIM_FILTROS", by ID, which was generated automatically. Returns text instead of an ID, and it is a view that is dynamic, and has the advantage that when the users go to the back office and update the value "nfd", the Power BI, when it is refreshing, will take the new definition.

Figure 5.27: TABULAR.VIEWS

The pipelines with prefixes 91 and 92 were created solely to run the calculated tables, allowing users to update them in the back office. Sometimes, users want to process only part of the ETL, including the generation of the calculated indicator, rather than the entire ETL.

The pipeline with the prefix 91 will perform the TL of the calculated data and generate the views at the end. Pipeline 92 is designed to create only the views. For example, when a filter is applied, instead of running the entire ETL, it only generates the views for the website.

Chapter 6

Test and Evaluation of the Solution

This chapter is dedicated to this specific task of measuring the effect of the proposed solution by carrying out comparative testing between the baseline setup, and actual, on Azure and the new setup on Fabric. The goal is to verify the enhancements caused by the migration in two key areas: performance and cost reduction.
To quantify performance, the identical data processing tasks — namely the execution of a common SQL query on Eurostat datasets — were performed in both environments, execution times were quantified and compared to determine the gains in processing speed and efficiency.
Cost estimation apart, numbers were taken from Fabric's official pricing calculator, available on Azure, "Microsoft Fabric - Pricing", [Microsoft n.d.(c)], and for Azure's official pricing calculator, available too on Azure, "Pricing - Azure SQL Database Single Database", [Microsoft n.d.(a)], but in this case the price was saw just for the Database. The Fabric's source provides up-to-date and clear pricing details based on Capacity Units (CUs), making it easier to determine the costs incurred for hosting the solution in the new environment compared to the Azure-based infrastructure, whose costs are typically broken down into compute, storage, and orchestration services.
Together, the tests create a concrete foundation to validate the architectural decisions in this project and offer an understanding of the practical benefits of adopting Fabric.

6.1 Test Environment Setup

To ensure that there is an impartial and controlled comparison between Fabric and Azure, tests were conducted on the same technical and functional level. Both setups were configured to process the same source of data — Eurostat indicators — and the same transformation and modeling logic.
The performance evaluation is focused on the execution time of a single SQL query that generated multiple Eurostat indicators — an ordinary and reproducible task for both architectures. The execution times were taken for typical system load to ensure the measurements are as homogenous as possible.
For cost estimation, the Azure environment is considered under pay-as-you-go for usage of Data Factory, SQL compute units, and storage. Fabric pricing is estimated based on the official pricing page, which suggests cost in terms of Capacity Units (CUs) per hour of activity.
All tests were performed on comparable data and within the same project size, enabling results to be interpreted as valid measures of real-world performance and monetary impact.

6.2 Performance Comparison

To compare the performance difference between Fabric and Azure, a sample SQL query that is applied to generate several Eurostat indicators was executed on both platforms. The given query, executed daily, involves transformations and aggregations on a normalized Silver Layer schema. By comparing the total time it takes for this operation in both platforms, one can put a measure on the actual impact of the architectural transition on processing efficiency.

6.2.1 Azure Execution Results

To realize this analysis, we used this SQL code:

```
SELECT DISTINCT processid,
                Title,
                Fonte,
                Indicadores,
                DATEDIFF(MINUTE, StartDate, EndDate) AS DuracaoEmMinutos
FROM [LOGS].[Refresh_Overview]
WHERE Title LIKE '%Eurostat%'
        AND Indicadores IS NOT NULL
        AND IsSuccess = 1
ORDER BY processid DESC
```

Listing 6.1: Azure Query for Eurostat refresh logs

In this code, we can obtain the process identifier, title, source, affected indicator, and duration.
The table obtained from the code is presented in Table 6.1.

	processid	Title	Fonte	Indicadores	DuracaoEmMinutos
1	554	Eurostat E	Eurostat	121	15
2	552	Eurostat E	Eurostat	121	6
3	219	Eurostat E	Eurostat	139,141	1
4	129	Eurostat E	Eurostat	259,344	186185
5	127	Eurostat E	Eurostat	259,344	187634
6	119	Eurostat E	Eurostat	259,344	188021
7	117	Eurostat E	Eurostat	259,344	188894
8	115	Eurostat E	Eurostat	259,344	189235
9	109	Eurostat E	Eurostat	184	257255
10	68	Eurostat E	Eurostat	114	278199
11	58	Eurostat E	Eurostat	107	285829
12	55	Eurostat E	Eurostat	107	287194
13	52	Eurostat E	Eurostat	107	287360
14	47	Eurostat E	Eurostat	107	288823
15	41	Eurostat E	Eurostat	53	289999
16	12	Eurostat E	Eurostat	53	369243
17	3	Eurostat ETL - Manual	Eurostat	357	369406
18	2	Eurostat ETL - Manual	Eurostat	357	369408

Table 6.1: Table Azure Indicators Duration

In this table, we can observe that most of the processes were only the E (Extraction) part, some of them had more than one indicator, but to calculate the time of the indicator, we can do the mean of the indicator, to see the difference between the Azure and the Fabric.

6.2.2 Fabric Execution Results

To realize this analysis, we used this SQL code, and we did it for the same indicators, to obtain and realize the comparison between both indicators and processes:

```
SELECT   executionid ,
         indicadores ,
         step ,
         duracao
FROM  [DW] . [LOGS]
WHERE
     (
          indicadores LIKE '%107%'
         OR indicadores LIKE '%114%'
         OR indicadores LIKE '%121%'
         OR indicadores LIKE '%139%'
         OR indicadores LIKE '%141%'
         OR indicadores LIKE '%184%'
         OR indicadores LIKE '%259%'
         OR indicadores LIKE '%344%'
         OR indicadores LIKE '%357%'
         OR indicadores LIKE '%53%'
     )
     AND resultado = 'Sucesso'
     AND step = 'ETL'
ORDER BY 1 DESC
```

Listing 6.2: Fabric Query for Eurostat refresh logs

By this code, we will obtain the execution identifier, the indicator, the step name, and the duration of the process.
The Title in the last process corresponds to the step in this table.
The table obtained from the code is presented in Table 6.2.

	executionid	indicadores	step	duracao
1	GER#932	259	ETL	00:09:47.00
2	GER#790	344,366	ETL	00:16:14.00
3	GER#614	448,449,450,451,452,453	ETL	00:03:10.00
4	GER#3532	53,110,113,114,115	ETL	00:13:49.00
5	GER#3495	107,122	ETL	00:12:26.00
6	GER#312	399,400,401,107,103,110	ETL	00:13:27.00
7	GER#2091	130,131,132,133,134,135,136,137,138,139	ETL	00:15:08.00

Table 6.2: Table Fabric Indicators Duration

In this case, the step of the process was the entire ETL (Extraction, Transform, and Load).

6.2.3 Comparison Analysis

To illustrate the difference in performance between Azure and Fabric more clearly, we focus on one example with Indicator 259, shared by the execution logs. In Fabric, the entire ETL process (Extraction, Transformation, and Loading) for this indicator is merely 9 minutes and 47 seconds.
In another way, in the Azure environment, the execution time is much longer. If we consider the "processid" 129, which relates to the E (Extraction) step of Indicator 259 and Indicator

344, the total captured time amounts to approximately 186185 minutes, as we can observe in the Table 6.1. Even assuming we cut the time for each of these indicators in half, we are getting an estimate of 93092 minutes and 30 seconds for the E step of Indicator 259 alone, without even reaching the subsequent T and L phases.
This difference highlights the gigantic performance difference between the two systems. While Azure relies on a decoupled architecture where extract operations are likely to introduce latency from service orchestration as well as data transfer, Fabric has a tightly coupled pipeline with significantly lower latency and better throughput, leveraging native Lakehouse architecture, optimized Delta processing, and reduced orchestration.
Thus, performance monitored verifies that migration into Microsoft Fabric not only reduces complexity but also shows orders of magnitude acceleration in processing speed, thereby making it significantly better suited for scalable and time-sensitive analytical workloads.

6.3 Cost Comparison

This section contrasts Microsoft Azure SQL Database and Microsoft Fabric cost structures in terms of scalability, pricing models, and available capacity. The comparison is supported by official Microsoft pricing tables and is followed by a general discussion of how they compare.

6.3.1 Microsoft Azure SQL Database Pricing

The pricing model for Microsoft Azure SQL Database relies on the vCore framework, with prices determined by the virtual cores and the proportionate memory allocation. It starts at 2 vCore (10.2 GB of memory) for €254.591/month on the pay-as-you-go plan, as we can observe on the Table 6.3, obtained from the "Pricing - Azure SQL Database Single Database", [Microsoft n.d.(a)], and scales up to 80 vCores and 396 GB of memory.

vCORE	Memory (GB)	Pay as you go	1 year reserved capacity [1]	3 year reserved capacity [1]
2	10.2	€254.591/month	€165.469/month ~35% savings	€114.558/month ~55% savings
4	20.4	€509.182/month	€330.938/month ~35% savings	€229.115/month ~55% savings
6	30.6	€763.772/month	€496.407/month ~35% savings	€343.672/month ~55% savings
8	40.8	€1,018.363/month	€661.876/month ~35% savings	€458.230/month ~55% savings
10	51	€1,272.953/month	€827.344/month ~35% savings	€572.787/month ~55% savings
12	61.2	€1,527.544/month	€992.813/month ~35% savings	€687.344/month ~55% savings
14	71.4	€1,782.134/month	€1,158.282/month ~35% savings	€801.901/month ~55% savings
16	81.6	€2,036.725/month	€1,323.751/month ~35% savings	€916.459/month ~55% savings
18	91.8	€2,291.315/month	€1,489.220/month ~35% savings	€1,031.016/month ~55% savings
20	102	€2,545.906/month	€1,654.688/month ~35% savings	€1,145.573/month ~55% savings
24	122.4	€3,055.087/month	€1,985.626/month ~35% savings	€1,374.688/month ~55% savings
32	163.2	€4,073.449/month	€2,647.501/month	€1,832.917/month

Table 6.3: Table Costs Estimation Azure

Even with such flexibility, Azure SQL Database is scalable only up to 80 vCores, which might be sufficient for transactional workloads or less-intensive analytics but might be a bottleneck for highly large-scale processing compared to Microsoft Fabric.

6.3.2 Microsoft Fabric Pricing

Microsoft Fabric uses a Capacity Unit (CU) model and offers much greater scalability than Azure SQL Database. It begins at F2 (2 CUs) for €278.759/month on pay-as-you-go subscription, as we can observe on the Table 6.4, obtained from the "Microsoft Fabric - Pricing", [Microsoft n.d.(c)], to F2048 (2,048 CUs) for €285,449.165/month.

SKU	Capacity unit (CU)	Pay-as-you-go	Reservation
F2	2	€278.759/month	€165.763/month ~41% savings
F4	4	€557.518/month	€331.526/month ~41% savings
F8	8	€1,115.036/month	€663.051/month ~41% savings
F16	16	€2,230.072/month	€1,326.102/month ~41% savings
F32	32	€4,460.144/month	€2,652.203/month ~41% savings
F64	64	€8,920.287/month	€5,304.405/month ~41% savings
F128	128	€17,840.573/month	€10,608.809/month ~41% savings
F256	256	€35,681.146/month	€21,217.618/month ~41% savings
F512	512	€71,362.292/month	€42,435.236/month ~41% savings
F1024	1,024	€142,724.583/month	€84,870.471/month ~41% savings
F2048	2,048	€285,449.165/month	€169,740.942/month ~41% savings

Table 6.4: Table Costs Estimation Fabric

The Capacity Unit (CU) model offers extremely high compute capability, making Fabric extremely well-suited for big data analytics, AI/ML, and high-frequency data pipelines. CU is not equivalent to GB, but rather a combined measure unit of compute, memory, and IO throughput of the capacity provided by the Fabric.

6.3.3 Overall Analysis

By analyzing the data we had, we can create a table, Table 6.5, to compare both systems.

Feature/ Metric	**Microsoft Azure SQL Database**	**Microsoft Fabric**
Smallest Unit	2 vCores, 10.2 GB,	F-2, 2 CUs
Largest Unit	80 vCores, 396 GB	F2048, 2048 CUs
Pay-as-you-go	Yes	Yes
Scalability	Limited to 80 vCores	Extremly high, 2048 CUs
Best Suited For	Transactional / Moderate Analytics	Large-Scale Analytics / High Compute Workloads

Table 6.5: Table of Comparison

The Microsoft Fabric and the Microsoft Azure SQL Database accommodate flexible pricing, pay-as-you-go, and reserved capacity discounts. However, the CU-based design of Fabric enables much greater scalability and processing power, removing most of the performance

limitations of Azure SQL Database.
For analytical mid-sized workloads or transactional applications, Azure SQL Database remains an extremely optimized and cost-effective solution, particularly for companies with structured database workloads that do not require high compute intensities.
Conversely, Fabric provides a higher-performance, future-proof environment to scale straightforwardly into thousands of capacity units. This positions it particularly well in big data analytics, machine learning pipelines, and real-time processing application scenarios, where enormous throughput and flexibility of processing matter most.

Chapter 7

Conclusion

In this chapter, we will talk about the biggest contributions of the migration, and also about the achieved results with the migration of the infrastructure of EDUSTAT from Azure to Fabric. It begins by analysing the objectives defined in Chapter 1, and also the ones achieved. After discussing the limitations of the project and the future work that will need to be done, the chapter will present the final appreciation, highlighting the overall value of Fabric for EDULOG, in particular for the project EDUSTAT.

7.1 Objectives Achieved

The main objective of this dissertation is to relocate the technological foundation of EDUSTAT, one infrastructure of EDULOG, from Azure to Fabric without hampering operations in any way, as discussed on Chapter 1, was fully achieved. Migration was successfully performed, with current services being reconfigured, data being transferred securely, and new features being deployed to maintain continuity of operations and optimize utilization of resources. The result was a smooth transition that not only preserved operational integrity but also enhanced the resilience and efficiency of the platform.
The secondary objectives were also met. The dissertation comprehensively analyzed the effect of the migration on infrastructure performance and saw empirical testing reflecting stark improvements in scalability and processing performance. Execution times were reduced by orders of magnitude, and ETL processes that took days or hours in Azure took minutes in Fabric. The cost of operation analysis had verified significant differences between the two platforms, highlighted by the way Fabric's capacity model on consumption guarantees more scalable and potentially lower long-term costs when integrated with reserved capacity. Finally, the dissertation had presented a methodological replication model for migration, reporting the step-by-step procedure carried out. This model can be utilized as a blueprint for future infrastructure transitions not only at EDULOG but even other organizations with similar modernization challenges.
Other than the desired aims, the project also introduced new functions unplanned initially but that benefited the platform nonetheless. They included parametrization of indicators by way of definition tables and automating ETL processes, which promoted greater flexibility, reduced developer intervention, and improved long-term capabilities of the platform in accommodating evolving requirements.

7.2 Limitations and Future Work

There were no significant technical limitations within the migration process. The requirements outlined earlier were effectively addressed by Microsoft Fabric's unified approach. That said, like any other analytical platform, it will require maintenance for it to remain effective.
Moving ahead, attention will need to be paid to maintenance and continual development. In particular, this work would need to ensure the architecture and infrastructure continue to align with new releases from Microsoft, perform continued monitoring of performance for possible scaling, and add new functionalities as user needs change. Additional opportunities to extend the existing platform analytics and predictive AI insights and real-time data processing can be pursued to enhance EDULOG's ability to supply timely and dependable public statistics.

7.3 Final Appreciation

This dissertaions shows taht the Fabric is a strategic driver of analytical modernization. By consolidating ingestion, transformation, storage, modeling, and visualization into a single environment, Fabric offers unequivocal advantages over the siloed Azure architecture. Its consolidated, capacity-based model ensures greater performance, simplifies governance, and provides scalability in terms beyond the previous architecture. The empirical experimentation carried out within this study confirmed these advantages with overwhelming evidence: processes that were hours or even days long in Fabric were all executed in minutes in Fabric. At the same time, the cost analysis indicated that Fabric's pricing model, particularly with its reserved capacity, is an unambiguous and financially sound method of long-term adoption which facilitates organizations to plan ahead better and simplify their utilization of resources. For EDUSTAT project specifically, this migration represents a quantum leap in the evolution of its analytical capability. Not only is the new platform quicker, but it is also more stable and versatile, able to incorporate heterogeneous public data sources transparently and publish official statistics with unprecedented speed and consistency. With the introduction of the standardized tables structures in the Silver Layer, and the configurational flexibility of the Gold Layer allow EDULOG to respond more easly to new reporting demands. These innovation give to EDUSTAT a resilient, and strong future-proof architecture that will support business continuity and innovation.
In essence, Fabric has provided EDULOG with more than technological upgradation, it has provided EDULOG with the foundations of an agile, scalable, and strategically led analytical platform. With this transformation, EDULOG is now better positioned not only to meet existing challenges but to capture upcoming opportunities with assurance, to continue to enhance its capacity for generating insight, further evidence-based decision-making, and making ever more valuable contributions to society.

Bibliography

Alouffi, Bader et al. (Apr. 2021). "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies". In: *IEEE Access* 9, pp. 57792–57807. url: https://ieeexplore.ieee.org/abstract/document/9404177.

Fehling, Christoph et al. (Dec. 2013). "Service Migration Patterns – Decision Support and Best Practices for the Migration of Existing Service-Based Applications to Cloud Environments". In: *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, pp. 9–16. url: https://ieeexplore.ieee.org/abstract/document/6717278.

Jamshidi, Pooyan, Aakash Ahmad, and Claus Pahl (Oct. 2013). "Cloud migration research: a systematic review". In: *IEEE transactions on cloud computing* 1.2, pp. 142–157. url: https://ieeexplore.ieee.org/abstract/document/6624108.

Lopes, Fábio Rafael Santos (Nov. 2023). "Lakehouse Data Architecture: Data as a First-Class Citizen within an Organization". In: pp. 1–70. url: https://www.proquest.com/openview/fbeea7e4f2e70b939f43341bb8d22d70/1?pq-origsite=gscholar&cbl=2026366&diss=y.

Microsoft (n.d.[a]). *Azure SQL Database pricing*. https://azure.microsoft.com/en-us/pricing/details/azure-sql-database/single/. Accessed: 2025-08-15.

– (n.d.[b]). *CI/CD for pipelines in Data Factory in Microsoft Fabric*. https://learn.microsoft.com/en-us/fabric/data-factory/cicd-pipelines. Accessed: 2025-08-30.

– (n.d.[c]). *Microsoft Fabric Price*. https://azure.microsoft.com/en-us/pricing/details/microsoft-fabric/. Accessed: 2025-08-15.

– (n.d.[d]). *Understand medallion lakehouse architecture for Microsoft Fabric with OneLake*. https://learn.microsoft.com/en-us/fabric/onelake/onelake-medallion-lakehouse-architecture. Accessed: 2025-08-30.

Sabiri, Khadija et al. (Nov. 2015). "Towards a cloud migration framework". In: *2015 Third World Conference on Complex Systems (WCCS)*, pp. 1–6. url: https://ieeexplore.ieee.org/abstract/document/7483315.

Thoutam, Mahesh (Oct. 2024). "MIGRATING ON-PREM DATA WAREHOUSING TO MICROSOFT FABRIC: LESSONS LEARNED AND BEST PRACTICES". In: *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET)* 15.5, pp. 682–693. url: https://mylib.in/index.php/IJCET/article/view/IJCET_15_05_063.

Trofimov, Slava (n.d.). *Understand medallion lakehouse architecture for Microsoft Fabric with OneLake*. https://learn.microsoft.com/en-us/fabric/onelake/onelake-medallion-lakehouse-architecture. Accessed: 2025-08-24.

Wikipedia (n.d.). *NUTS de Portugal*. https://pt.wikipedia.org/wiki/NUTS_de_Portugal. Accessed: 2025-09-14.