

FACULTY OF ENGINEERING OF THE UNIVERSITY OF PORTO

A Framework for Integrating Heterogeneous Data Sources

Filipe de Moraes Teixeira Pinto



Master in Informatics and Computing Engineering

Supervisor: Bruno Miguel Carvalhido Lima

Second Supervisor: Vasco Ferreira Ribeiro

July 25, 2024

A Framework for Integrating Heterogeneous Data Sources

Filipe de Moraes Teixeira Pinto

Master in Informatics and Computing Engineering

Approved in oral examination by the committee

Chair: Prof. Dr. Ana Cristina Ramada Paiva

External Examiner: Prof. Dr. Miguel António Sousa Abrunhosa Brito

Supervisor: Prof. Dr. Bruno Miguel Carvalhido Lima

July 25, 2024

Resumo

A crescente variedade de dados desportivos provenientes de diversas fontes apresenta um desafio significativo para a análise desportiva. Atualmente, a análise desportiva não se limita a estatísticas padrão, mas estende-se a informações detalhadas sobre jogadores, táticas e até dados de ligas amadoras. Esta diversidade abrange desde o acompanhamento em tempo real do desempenho dos jogadores até estatísticas avançadas de equipas em competições menos conhecidas.

No entanto, as soluções existentes para a integração de dados muitas vezes não conseguem lidar eficientemente com a heterogeneidade desses dados. As ferramentas tradicionais carecem da flexibilidade necessária para se adaptar a diferentes formatos e estruturas de dados, potencialmente perdendo informações cruciais ou levando a interpretações imprecisas. Além disso, a constante evolução dos dados desportivos exige métodos de integração que possam adaptar-se rapidamente às novas necessidades analíticas.

Este trabalho propõe uma ferramenta para superar esses desafios. A abordagem visa integrar efetivamente os diversos dados desportivos, focando-se na adaptabilidade e versatilidade. As principais características incluem opções de Aceitar e Ignorar, permitindo a intervenção manual para garantir a integridade dos dados. Este equilíbrio entre precisão e flexibilidade é vital para gerir eficientemente uma variedade de conjuntos de dados, essencial para uma análise desportiva precisa e relevante.

A ferramenta opera através de um processo de configuração que mapeia regras e remove ou adiciona campos conforme necessário. Uma API automatiza essa transformação, verificando a disponibilidade e funcionalidade do fornecedor de dados, testando programaticamente o processo de transformação de dados e garantindo o correto tratamento dos pedidos de transformação de dados. Isto facilita fluxos de trabalho automatizados e a integração com outros sistemas.

A aplicabilidade prática desta ferramenta foi validada através de um caso de estudo em colaboração com a empresa zerozero.pt, um proeminente fornecedor de estatísticas desportivas. Este caso prático demonstra como a solução pode melhorar significativamente a eficiência e a precisão na integração de dados desportivos, mostrando a sua capacidade de lidar efetivamente com conjuntos de dados diversos e em evolução.

Abstract

The increasing variety of sports data from diverse sources presents a significant challenge for sports analysis. Currently, sports analysis is not limited to standard statistics but extends to detailed information about players, tactics, and even data from amateur leagues. This diversity ranges from real-time player performance tracking to advanced team statistics in lesser-known competitions.

However, existing solutions for data integration often need to handle the heterogeneity of these data efficiently. Traditional tools lack the flexibility to adapt to different data formats and structures, potentially losing crucial information or leading to inaccurate interpretations. Moreover, the constant evolution of sports data requires integration methods that can quickly adapt to new analytical needs.

This work proposes a framework to overcome these challenges. The approach aims to effectively integrate diverse sports data, focusing on adaptability and versatility. Key features include Accept and Ignore options, allowing manual intervention to ensure data integrity. This balance between precision and flexibility is vital for efficiently managing a variety of datasets, which is essential for accurate and relevant sports analysis.

The framework operates through a configuration process that maps rules and removes or adds fields as necessary. An API automates this transformation, verifying the availability and functionality of the data provider, testing the data transformation process programmatically, and ensuring the correct handling of data transformation requests. This facilitates automated workflows and integration with other systems.

The practical applicability of this framework was validated through a case study in collaboration with `zerozero.pt`, a prominent sports statistics provider. This practical case demonstrates how the solution can significantly improve efficiency and accuracy in sports data integration, showcasing its ability to effectively handle diverse and evolving datasets.

Agradecimentos

First and foremost, I would like to express my heartfelt gratitude to Professor Bruno Lima for his unwavering support and guidance throughout the development of this thesis.

A special word of thankfulness goes to the zerozero team, particularly Vasco Ribeiro, for their warm welcome and invaluable assistance during these last months.

I would also like to thank my family for their unwavering support and encouragement throughout my studies. In particular, I am deeply grateful to my parents, Maria and Rogério, who provided me with every opportunity to complete this work, and to my brother, Miguel, who was always there whenever I needed him.

A final thanks goes to Beatriz, who believed in me from the very beginning. Her unwavering support and encouragement have been a constant source of motivation, and I knew I could always count on her.

Filipe de Morais Teixeira Pinto

*“Believe in yourself and all that you are.
Know that there is something inside you that is greater than any obstacle.”*

Christian D. Larson

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.2	Goals	2
1.3	Document Structure	3
2	Background and State of the Art	4
2.1	Data	4
2.2	Big Data	5
2.2.1	MapReduce	8
2.2.2	Apache Hadoop	9
2.2.3	Apache Avro	11
2.2.4	Apache Spark	11
2.3	Data Integration	11
2.3.1	Business Intelligence	12
2.3.2	Data Warehouse	12
2.3.3	Enterprise Application / Information Integration	14
2.3.4	Data Migration	14
2.3.5	Master Data Management	14
2.4	Data Warehouse vs Databases	14
2.4.1	Databases	14
2.4.2	Data Warehouses	15
2.4.3	Core Differences	15
2.5	Data Integration Techniques	16
2.6	ETL	17
2.6.1	ETL Tools Categories	18
2.6.2	Popular ETL Tools	19
2.6.3	Comparative Analysis of ETL Tools	23
2.7	Existing Data Integration solutions	24
2.8	Data Mapping	26
2.8.1	Types of Data Mapping	28
2.8.2	Keys for Data Mapping	30
2.8.3	Data Mapping Challenges	31
2.8.4	Data Mapping Tools	32
2.8.5	Related Work	36
2.9	API	37
2.9.1	REST	37
2.9.2	API Management	39
2.10	Conclusion	40

3	PlayField	41
3.1	Requirements	41
3.1.1	User Interface and Interaction	41
3.1.2	Rule Management	43
3.1.3	Data Mapping and Rule Creation	44
3.1.4	Dropdown for Node Mapping	46
3.1.5	Rule Storage and Application	47
3.1.6	Exporting Transformed Data	47
3.1.7	Automation through API	48
3.2	Architecture	51
3.2.1	Technologies	52
3.2.2	Domain Model	52
3.2.3	Workflow	53
4	Validation	56
4.1	Experimental Validation	56
4.1.1	Iterative Feedback and Framework Enhancements	56
4.1.2	Testing with EnetPulse Files	57
4.1.3	Testing with Updated Versions	58
4.1.4	Enhancements Based on User Feedback	58
4.2	Use Case	59
4.2.1	Pre-Test Questionnaire	60
4.2.2	Task Execution	61
4.2.3	Post-Test Questionnaire	61
4.2.4	Detailed Analysis of Questionnaire Responses	64
4.2.5	Open-Ended Question	66
5	Conclusions and Future Work	69
5.1	Conclusions	69
5.2	Main Contributions	69
5.3	Future Work	70
	References	71
A	PlayField Validation	76
A.1	Validation Questionnaire	76

List of Figures

2.1	The three Vs of Big Data	6
2.2	Medium's explanation of the 5 V's of Big Data	7
2.3	Types of Data	7
2.4	MapReduce Execution	9
2.5	HDFS Architecture	10
2.6	Hadoop Architecture	10
2.7	Data Integration	12
2.8	Data Integration Areas	13
2.9	DW and Business Intelligence Users	13
2.10	ETL process	18
2.11	Fusco and Aversano proposed solution	26
2.12	KD SENSO-MERGER architecture	27
2.13	Data Mapping	29
3.1	PlayField Dropdowns	42
3.2	PlayField File Uploader	43
3.3	PlayField Rule Buttons	43
3.4	PlayField View Rules Pop-up	43
3.5	PlayField Edit Output Fields	44
3.6	PlayField Input and Output Nodes	45
3.7	PlayField Dropdown for Creating Rules	46
3.8	PlayField Export Button	48
3.9	PlayField API Endpoints	48
3.10	GET Providers Endpoint	49
3.11	POST Transform Endpoint Body	50
3.12	POST Transform Endpoint Response	50
3.13	PlayField Architecture	52
3.14	Domain Model	53
3.15	Execution Flow	54
4.1	Years of Experience in the Current Role	60
4.2	Experience of each user in Data Manipulation	61
4.3	SUS results from the questionnaire	63
4.4	Question 1 Results	64
4.5	Question 2 Results	64
4.6	Question 3 Results	65
4.7	Question 4 Results	65
4.8	Question 5 Results	65

4.9	Question 6 Results	66
4.10	Question 7 Results	66
4.11	Question 8 Results	66

List of Tables

2.1	Comparison of ETL Tools	24
-----	-----------------------------------	----

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BI	Business Intelligence
CDC	Change Data Capture
COBOL	COmmon Business Oriented Language
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
CSV	Comma Separated Values
DIF	Data Integration Framework
DTD	Document Type Definitions
DW	Data Warehouse
EA	Enterprise Application
ERP	Enterprise Resource Planning
ETL	Extract Transform Load
GaV	Global as View
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
KD	Knowledge Discovery
LaV	Local as View
MDM	Master Data Management
NLP	Natural Language Processing
OLAP	Online Analytical Processing
OLTP	Online Transactional Processing
PC	Personal Computer
PDF	Portable Document Format
REST	REpresentational State Transfer
RDF	Resource Description Framework
SAS	Statistical Analysis System
SaaS	Software as a Service
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SSIS	SQL Server Integration Services
UI	User Interface
URI	Uniform Resource Identifier
XML	Extensible Markup Language
YARN	Yet Another Resource Negotiator

Chapter 1

Introduction

This opening chapter presents an introductory overview of the thesis, detailing its context, underlying motivation, specific goals, and the document's structure. The aim is to furnish a comprehensive and lucid understanding of the thesis's objectives and lay the groundwork for the subsequent document sections.

1.1 Context and Motivation

The landscape of sports analytics has undergone a significant transformation, evolving from basic statistical analysis to a more comprehensive utilization of intricate data. This includes detailed metrics of individual player performances and a wealth of information from amateur leagues. In the past, football game statistics were primarily focused on the main leagues, encompassing basic data like goals scored, cards issued, and number of fouls. Nowadays, however, the scope of analysis has broadened remarkably. It now includes many detailed metrics such as player movement tracking, in-depth performance analysis across various leagues, including amateur and regional levels, and much more.

The expansion of the data scope in sports analytics highlights the increased complexity and diversity of information sources within the sports domain. Integrating these varied sources of data presents a unique challenge. Advanced methods are now required to ensure that the data remains coherent and valuable for sports analytics.

Today's sports industry generates and utilizes a wide range of data types, from traditional structured data to more dynamic and unstructured formats like JSON, XML, and CSV. Integrating such heterogeneous data sources is a technical challenge and a critical factor in deriving meaningful insights from sports data. Efficiently managing this diversity is pivotal for comprehensive analytics that can inform strategic decisions in sports management and athlete performance.

Traditional data integration methods in sports analytics often struggle to cope with modern sports data's volume, velocity, and variety. The limitations of these methods become apparent in scenarios requiring rapid processing and adaptation to changing data formats and sources. These

challenges highlight the need for a more robust and adaptable ETL (Extract, Transform, and Load) process tailored to the unique demands of sports data.

The present work was done in collaboration with [zerozero.pt](https://www.zerozero.pt)¹, which defines itself as the world's largest sports-related information database. [zerozero.pt](https://www.zerozero.pt) specializes in collecting, processing, and analyzing extensive sports data, delivering detailed insights into various sports—particularly football. Their large and varied dataset formed a reasonable basis for the development and testing of the framework created. The project, therefore, had to address specific needs and extraordinary complexities in the process of modern sports data integration by working very closely with [zerozero.pt](https://www.zerozero.pt) to make sure the solution approaches were effective and relevant.

1.2 Goals

As highlighted in the previous section, integrating diverse and complex sports data sources presents significant challenges in sports analytics. This thesis addresses these challenges by developing an innovative Data Mapping framework. The primary objective of this dissertation is to create a Data Mapping framework that effectively integrates heterogeneous sports data from various formats, including JSON, XML, and CSV, prevalent in sports analytics.

This thesis also aims to explore and develop data handling and processing techniques, mainly when data sources vary in structure and format. Accurate processing and integration of this data will substantially enhance the quality of analytics, leading to more informed decision-making in sports management.

To realize the goals mentioned above, this dissertation intends to accomplish the following specific goals:

1. Conduct an in-depth study of the relevant subjects in sports data integration and present the current state-of-the-art.
2. Analyze and refine data from a wide array of sports data feeds, focusing on specific leagues, teams, and performance metrics to ensure targeted and detailed integration.
3. Develop a comprehensive Data Mapping framework tailored for sports analytics, focusing on versatility and adaptability to different data formats.
4. Validate and evaluate the effectiveness of the Data Mapping framework through its practical application, utilizing data from a collaboration with [zerozero.pt](https://www.zerozero.pt). This process will involve assessing the framework developed in real-world scenarios to ensure its reliability and applicability in the field of sports analytics.
5. Enhance the overall accuracy and efficiency of sports data integration, contributing to advancements in sports analytics.

¹<https://www.zerozero.pt>

6. Publish the results in a scientific article to disseminate findings and contribute to the broader academic community.

1.3 Document Structure

Beyond the current chapter, the remainder of the document is divided as follows:

- Chapter 2 presents a state-of-the-art analysis, covering big data architectures, data warehouses, data integration, ETL processes, data mapping, and related work. It also explores their application in sports data integration.
- Chapter 3 details the PlayField framework, including its features, implementation, and architecture. It covers user interface and interaction, rule management, data mapping and rule creation, rule storage and application, and automation through API.
- Chapter 4 describes the validation process using a dual approach. Experimental validation tests the framework in a controlled environment, followed by a real-world case study with zerozero.pt to assess usability and effectiveness. Feedback from zerozero.pt's content managers and developers provided valuable insights.
- Lastly, chapter 5 concludes the thesis and discusses future work. Summarizes research findings, highlights the framework's effectiveness, and suggests enhancements such as advanced data mapping techniques, machine learning algorithms, and user interface improvements. Provides an overview of key milestones and steps undertaken during the research.

Chapter 2

Background and State of the Art

This chapter presents an in-depth exploration of the background and state-of-the-art related to the dissertation's theme.

The chapter is structured to methodically unfold the various layers of the topic, starting with the fundamental concept of data in the context of information technology, moving through the intricacies of Big Data, and delving into the essential frameworks and tools for Big Data processing. It encompasses a thorough data integration analysis, highlighting the significance of ETL tools in managing and synthesizing large volumes of data from diverse sources, and examines popular ETL tools, comparing their features and capabilities. Furthermore, the chapter explores data mapping, detailing its crucial role in data integration and the challenges and methodologies involved in aligning disparate data sources for accurate and efficient transformation.

2.1 Data

The Cambridge English Dictionary defines data in the context of IT (Information Technology) as "information in an electronic form that can be stored and processed by a computer" [9]. In the rapidly evolving field of technology, data could be considered information in various forms, such as documents, images, audio clips, software programs, or other types of data processed by a computer. This encompasses text, video, and audio and includes web and log activity records. Most of this data is unstructured, presenting unique challenges in its processing and analysis.

Data can be broadly categorized into two types: structured and unstructured. Structured data refers to information that is highly organized and easily searchable in databases. This type of data is typically stored in rows and columns, such as in spreadsheets or relational databases, following a predefined schema. Examples of structured data include transaction records in a financial database or sensor readings from an IoT (Internet of Things) device.

In contrast, unstructured data lacks a specific format or organization, making it more complex to process and analyze. This type of data does not fit neatly into traditional database tables and often includes free-form text, multimedia files, and social media posts. Examples of unstructured data include email messages, video files, audio recordings, and web pages.

The majority of the data generated today is unstructured, which presents unique challenges in terms of storage, processing, and analysis. Unlike structured data, which is neatly organized in databases and easily accessible through simple queries, unstructured data lacks a consistent format, making it difficult to store in traditional relational databases. This requires flexible storage solutions, such as NoSQL databases [66] or distributed file systems, that can handle various data formats and scales. Processing unstructured data is complex, often necessitating advanced techniques like natural language processing, machine learning, and computer vision to extract meaningful insights. The analysis of unstructured data also involves extensive preprocessing to clean and transform the data into a usable format.

The distinction between structured and unstructured data is crucial because it influences how data can be managed and utilized. Structured data is straightforward to query and analyze using traditional database management tools and SQL queries. On the other hand, unstructured data requires more advanced techniques, such as natural language processing (NLP) [31] and machine learning algorithms [44], to extract meaningful insights.

2.2 Big Data

Over the last few years, Big Data has been the "center of modern science and business," generated from diverse sources like online transactions, emails, videos, and social networking interactions. These data sets grow massively, making it difficult to manage and analyze using traditional database software tools [55].

One issue due to this rapid expansion was the absence of a clear definition of Big Data.

According to *Big Data: A Review* by Sagioglu et al. [55], Big data is a term defining massive data sets with large, varied, and complex structures, presents storage, analysis, and visualization challenges for further processing or results. Big Data is characterized by its three main components, Variety, Velocity, and Volume, and requires revolutionary steps forward from traditional data analysis as demonstrated in Figure 2.1.

- **Variety** This encompasses diverse data types and sources, both structured and unstructured, such as emails, images, videos, sensor data, PDFs, audio files, and social media content from platforms like Facebook, Twitter, LinkedIn, and YouTube. This diversity of data forms, ranging from text and multimedia to log files and sensor data, poses significant storage, mining, and analysis challenges.
- **Volume** to the enormous quantities of data generated from various sources, including individual and organizational activities, social media interactions, and machine-generated data. This vast volume of data, which used to be primarily user-generated, now encompasses a broader array of sources.
- **Velocity** describes the speed at which data is generated, processed, and made available for use. It encompasses the rapid flow of data from sources like business processes, machines,

networks, and human interactions, with examples such as the high-frequency trading data from stock exchanges [35].



Figure 2.1: The three Vs of Big Data

Even though there is consensus on the three V's, others have added more to their definition. These include:

- **Veracity:** IBM presented Veracity, emphasizing the challenges posed by data sources that are inherently unreliable, especially when using information obtained from social media. Despite this ambiguity, this data source includes a wealth of useful information. [29]
- **Value:** Oracle added Value as the fourth component of Big Data. The data obtained in its original form has a low value about its volume. However, evaluating vast amounts of data may achieve a high value. [16]
- **Variability and Veracity:** SAS introduced Variability and Veracity as two additional dimensions of big data as shown in Figure 2.2 Variability refers to the inconsistency in the data flow, marked by unpredictable changes and diverse trends, for example, social media trends or daily/seasonal peak data loads. Veracity refers to the quality of data. Because data comes from different sources, it can be difficult to link, match, clean, and transform. [57]

The types of data in big data encompass **structured**, **unstructured**, and **semi-structured** data described in Figure 2.3.

- **Structured Data** is organized in a formatted repository like a database, making it easily addressable for processing and analysis.
- **Unstructured Data**, which can be textual or non-textual, includes a variety of formats like PDFs, audio files, video files, and social media content.

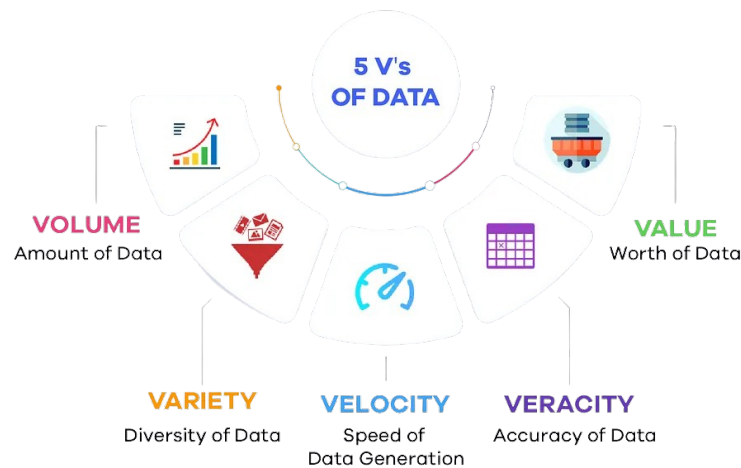


Figure 2.2: Medium's explanation of the 5 V's of Big Data [2]

- **Semi-structured Data**, although not as rigidly structured as traditional database data, possesses some organizational properties that make it amenable to database storage, as seen in formats like HTML, CSV, XML, JSON [35].

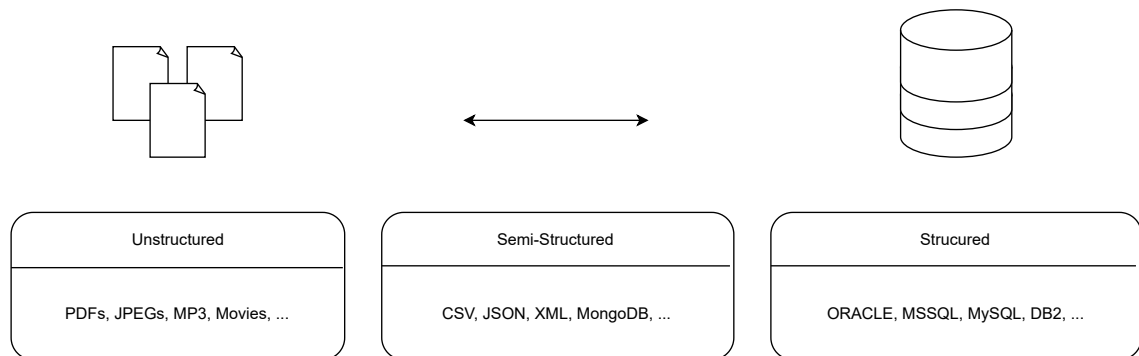


Figure 2.3: Types of Data

Big data sources include social media networks, mobile devices, sensor technology, and scientific instruments. Social media platforms generate vast amounts of data from likes, comments, posts, and uploads. Mobile devices, now capable of extensive tracking and data generation, contribute significantly to the volume of big data. Sensor networks, both active and passive, generate data through signal transmission and reception. Scientific instruments, such as satellites, produce large volumes of data through continuous monitoring and recording activities [35].

Furthermore, Big Data should create additional value for the organization after processing. The applications of Big Data span various fields such as astronomy, genomics, and numerous sectors, including government finance and retail, offering deep insights to solve real problems [55].

As we delve deeper into the world of Big Data, it is essential to understand the underlying frameworks and methodologies that make it possible to process and analyze such vast and complex datasets effectively. Several significant advancements have revolutionized the way we handle large-scale data computations. Key among these are the MapReduce programming model, Apache Hadoop, Apache Avro, and Apache Spark. Each technology provides robust frameworks for efficiently processing and managing large datasets, addressing different aspects of the Big Data ecosystem. In the following subsections, we will explore these technologies in detail, discussing their components, execution processes, and applications in real-world scenarios.

2.2.1 MapReduce

MapReduce, a programming model developed by Google¹, offers a framework for processing and generating large datasets suitable for a wide range of real-world tasks. This model simplifies the handling of large-scale data computations, automating the parallelization across clusters of machines and managing machine failures, thus optimizing network and disk usage [14]. The MapReduce library requires users to define two functions: **map** and **reduce**.

- **Map**: processes input key/value pairs to produce intermediate key/value pairs.
- **Reduce**: merges these intermediate values, typically producing a smaller set of values.

The execution of **MapReduce** is depicted in Figure 2.4 and comprises the following steps:

1. The MapReduce library splits input files into M pieces, each typically 16-64MB, and starts many copies of the program across a cluster.
2. One of the nodes is the master. The master node assigns M map tasks, and R reduces tasks to idle worker nodes.
3. A worker who is assigned a map task reads the contents of the corresponding input split, parsing key/value pairs and passing them to the user-defined map function. Intermediate pairs are buffered in memory.
4. Buffered pairs are periodically written to the local disk partitioned into R regions. The locations of these pairs are reported to the master, which is responsible for forwarding this information to the reduce workers.
5. When a reduce worker receives information about these locations, reduce workers read the buffered data from the local disks of the map workers, sort it by intermediate keys, and group occurrences of the same key. This sort is needed because, typically, many different keys map to the same reduce tasks.
6. For each unique key, the reduce worker applies the reduce function and appends the output to a final output file.

¹<https://www.google.com/>

7. After all map and reduce tasks are completed, the master notifies the user program, and the MapReduce call returns.
8. The output is stored in R output files, one for each reduce task, and is typically used as input for subsequent MapReduce calls or other applications.

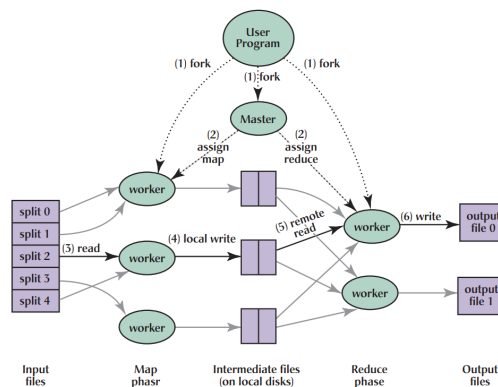


Figure 2.4: MapReduce Execution [14]

Different implementations of MapReduce are suitable for various environments, from small shared-memory machines to large clusters of networked devices. Google's implementation operates over large clusters of commodity PCs connected via Gigabit Ethernet, with each machine typically featuring dual-processor x86 processors and running Linux [14]. Fault tolerance is a critical aspect of MapReduce, as the library is designed for environments where machine failures are common. The master node in a MapReduce operation monitors the worker nodes and reallocates tasks as necessary to ensure continuity in the face of hardware failures.

MapReduce has seen widespread application within Google owing to its ease of use for programmers, including those without experience in parallel and distributed systems. The model's success also stems from its ability to express various problems, ranging from data generation for web search services to machine learning tasks. Its scalability to large clusters and efficient resource utilization make it suitable for handling Google's significant computational problems.

2.2.2 Apache Hadoop

Hadoop[49], an open-source framework, is designed to store and process large volumes of data efficiently and reliably across clusters of commodity hardware. Hadoop's architecture is built upon two main components: the **Hadoop Distributed File System (HDFS)** and the **MapReduce** programming model.

HDFS is a scalable file system that distributes data across various servers, achieving high data access speeds. Its design is inherently resilient, employing data replication across multiple nodes to ensure durability and fault tolerance. This distributed approach guarantees data safety in case of node failures and allows for seamless data storage expansion across numerous machines. The structural design of HDFS is depicted in Figure 2.5.

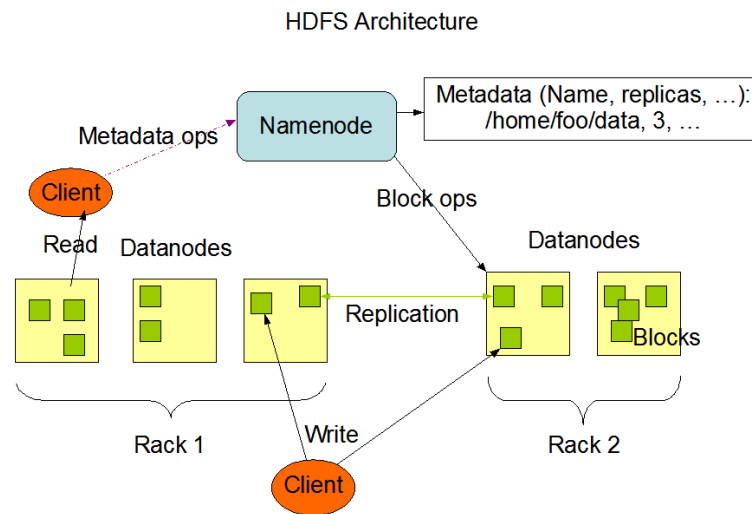


Figure 2.5: HDFS Architecture [1]

Hadoop's ecosystem also includes supplementary modules such as Hadoop Common, offering essential utilities for other Hadoop modules, and YARN (Yet Another Resource Negotiator), which is pivotal for job scheduling and resource management within the cluster. YARN's integration enhances Hadoop's capability, enabling a variety of data processing engines — from interactive SQL and real-time streaming to data science and batch processing — to efficiently manage data on a unified platform. Hadoop's architecture is presented in Figure 2.6.

Overall, Hadoop represents a significant advancement in big data processing, offering a robust, scalable, and cost-effective solution for managing the complexities of large-scale data analysis. Its ability to handle vast amounts of diverse data types and its fault tolerance make it a popular choice for organizations with big data challenges.

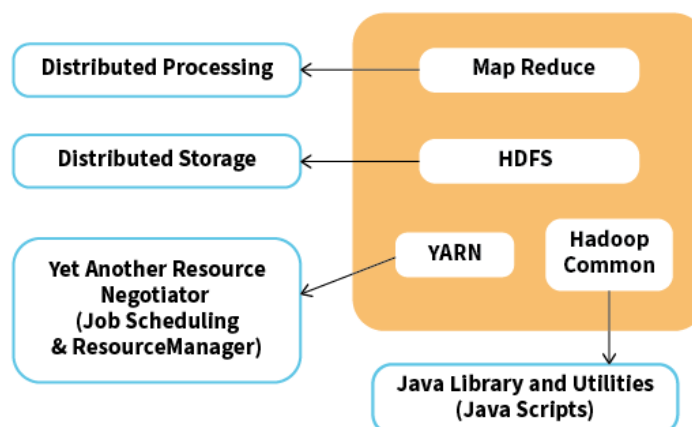


Figure 2.6: Hadoop Architecture [3]

2.2.3 Apache Avro

Apache Avro[69] is a data serialization system designed to support the application of large quantities of data exchange. It supports binary serialization forms and can conveniently quickly process large amounts of data. Avro relies on schemas to implement the data structure definition. One of the benefits of Apache Avro is its schema flexibility during data processing. While defining a schema is essential during the file writing phase, it's not required for reading. This flexibility stems from Avro's ability to supply the schema information during reading. Avro adeptly manages any schema variations that might occur between the writing and reading stages. To facilitate this, users must either consistently maintain attribute names or set default values for attributes exclusive to the reader's schema. Conversely, attributes unique to the writer's schema are disregarded during reading.

2.2.4 Apache Spark

Apache Spark[56] is a unified analytics engine that is open-source and well-known for its speed and powerful analytics features. Its cornerstone is Atop Spark Core, which provides essential activities like I/O operations and task scheduling via APIs for Java, Scala, Python, and R. Various modules are built on top of Spark Core to increase its capability. The DataFrames abstraction in Spark SQL offers a distributed query engine. Real-time data processing is possible using Spark Streaming. GraphX focuses on graph data processing, whereas MLlib provides machine-learning techniques for data analysis.

2.3 Data Integration

Data integration is a critical area in computer science, particularly for applications that require querying across multiple autonomous and heterogeneous data sources, as demonstrated in Figure 2.7. It is vital in large enterprises, scientific projects, government cooperation, and for adequate search quality across the vast data on the World Wide Web. The essence of data integration lies in combining data residing at different sources and providing a unified view to the user. This process involves overcoming challenges related to disparate data formats and sources, using various technologies and tools, particularly ETL tools, to ensure seamless integration [63]. This process involves several subproblems:

- **Structural Integration:** This refers to resolving structural heterogeneity, such as differences in data models, query languages, and protocols.
- **Semantic Integration:** It involves resolving semantic mismatches between schemata, where similar information may be represented differently.
- **Heterogeneity of Data Sources:** Each data source might have a different data model representation and may contain conflicting data.

- **Autonomy of Data Sources:** Data sources are independent, not specifically designed for integration, and can change unannounced.
- **Query Correctness and Performance:** Ensuring proper processing and performance of queries in the integrated system is essential.
- **Distribution:** It deals with the physical distribution of data sources and the system architecture's consideration for potential latency.

These challenges highlight the complexity of data integration in diverse systems and underline the importance of effective strategies and technologies in this field [43].

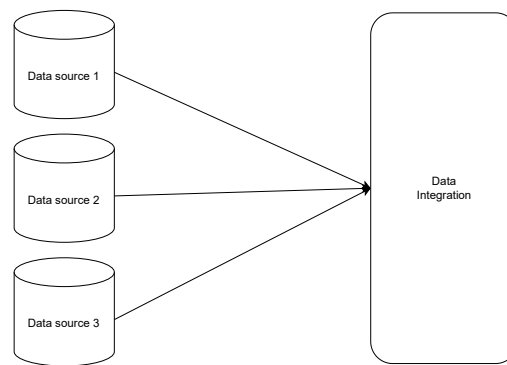


Figure 2.7: Data Integration

Data integration is an important area that covers other areas, i.e., sub-areas that include Business Intelligence (BI), Data warehouses, Enterprise Applications (EA), Data Migration and Data Visualization, as shown in Figure 2.8. The integrated data is mainly used to help businesses.

2.3.1 Business Intelligence

Business Intelligence involves transforming raw data into valuable insights to enhance business profitability. It focuses on analyzing customer behavior and market trends to identify opportunities or challenges. BI leverages historical data to facilitate decision-making in complex scenarios, providing a non-modifiable record of past operations. By offering detailed analyses of business processes, BI helps in drawing actionable conclusions to improve strategic outcomes [63].

2.3.2 Data Warehouse

A data warehouse [62], serves as a large-scale repository designed to aggregate and store critical data from multiple sources, acting as a unified database to facilitate consistent information management. The evolution of the data warehouse concept stemmed from the limitations of traditional databases, which were primarily designed to store data in megabytes or gigabytes. As data sizes grew, data warehouses emerged, capable of handling data in terabytes. They are particularly adept at storing historical data, enabling organizations to analyze existing data easily. This capability

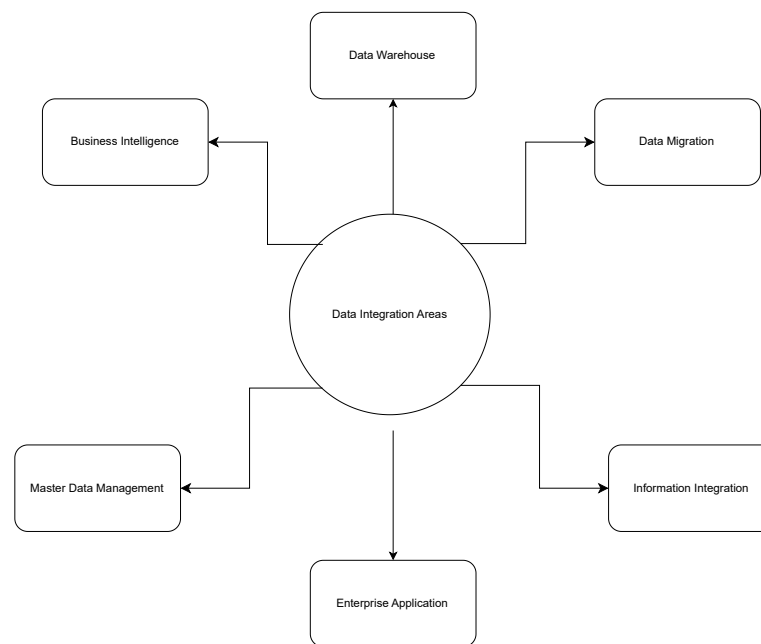


Figure 2.8: Data Integration Areas

of storing both historical and current data under a single framework renders a data warehouse an essential element of business intelligence (Figure 2.9). In a data warehouse, data is systematically organized into hierarchical groups known as dimensions and facts.

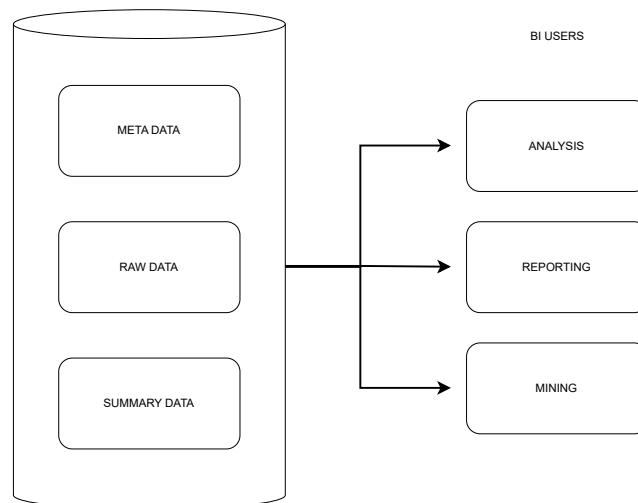


Figure 2.9: DW and Business Intelligence Users

According to [62], the benefits of data warehousing are:

1. Preserves historical data unavailable in original transaction systems.
2. Provides a unified organizational view by consolidating disparate data.
3. Enhances the overall quality of data compared to its initial form.

4. Organizes data from various sources into a coherent storage system, improving business understanding.
5. Supports informed decision-making by offering comprehensive data insights.
6. Optimizes data structure to boost retrieval and analysis efficiency.

2.3.3 Enterprise Application / Information Integration

An Enterprise Application is a comprehensive software system that offers business-oriented tools designed to enhance organizational efficiency. It incorporates specific business logic, facilitating restricted information exchange tailored to operational methods, thereby boosting productivity. Through the integration of various management systems, it enables the formation of departments and processes, fostering improved customer relationships and organizational effectiveness.[63]

2.3.4 Data Migration

Data migration involves the crucial task of moving information from one location to another, necessitating steps like preparation, extraction, and conversion of data. This process often employs ETL tools for the efficient relocation of data to a new storage site. A significant challenge in data migration is the potential for transferring mismatched data types, leading to issues with numbers, dates, sub-records, and various character sets, which may require specific encoding adjustments [63].

2.3.5 Master Data Management

Master data represents the unified information shared across an organization, encompassing critical data from both internal and external sources into a singular, comprehensive view. Master Data Management (MDM) focuses on overseeing this essential corporate information, such as product details—ID, name, price, and manufacturing date—that necessitate regular updates for accuracy. MDM facilitates improved data processing, enabling businesses to achieve greater operational success [63].

2.4 Data Warehouse vs Databases

Understanding the differences between databases and data warehouses is essential for comprehending their unique roles and functionalities in data management and analytics.

2.4.1 Databases

A database is a structured collection of data that is electronically stored and accessed. Databases are designed to manage and retrieve large amounts of data efficiently. They are primarily used for

Online Transactional Processing (OLTP) [11], which involves the day-to-day operations and transactions of an organization. Examples of databases include MySQL [18], PostgreSQL [47], Oracle Database [32], and Microsoft SQL Server [51]. The key characteristics of databases include:

- **Data Structure:** Data in databases is typically stored in two-dimensional tables (rows and columns) that are normalized to reduce redundancy and ensure data integrity.
- **Operations:** Databases are optimized for CRUD operations (Create, Read, Update, Delete) to support transactional consistency and quick query responses.
- **Use Cases:** Databases are commonly used for applications like customer relationship management (CRM), enterprise resource planning (ERP), and other systems requiring real-time data management and quick access.

2.4.2 Data Warehouses

A data warehouse, on the other hand, is a centralized repository designed specifically for storing, managing, and analyzing large volumes of historical data. Data warehouses are primarily used for Online Analytical Processing (OLAP) [7], which supports complex queries and data analysis. Examples of data warehouses include Amazon Redshift [33], Google BigQuery [23], and Snowflake [12]. The key characteristics of data warehouses include:

- **Data Structure:** Data in data warehouses is typically stored in multidimensional schemas, such as star or snowflake schemas, and is often denormalized to improve query performance.
- **Operations:** Data warehouses are optimized for read-heavy operations and complex queries, allowing for in-depth analysis, reporting, and data mining.
- **Use Cases:** Data warehouses are used for business intelligence (BI) applications, providing insights and supporting decision-making processes through comprehensive data analysis.

2.4.3 Core Differences

The core differences between databases and data warehouses are centered around their respective uses and data structuring methods:

- **Purpose:** Databases are designed for OLTP, supporting the day-to-day operations of an organization with quick transaction processing and data integrity. Data warehouses, in contrast, are built for OLAP, focusing on analyzing and querying large datasets to support strategic decision-making.
- **Data Structuring:** In databases, data is highly normalized to eliminate redundancy and ensure consistency, which helps in efficient transaction processing. In data warehouses, data is often denormalized to optimize read performance and enable faster query responses, accommodating the complex analytical needs.

- **Storage Methods:** Databases typically use two-dimensional tables for data storage, suitable for operational tasks. Data warehouses utilize multidimensional tables, which are better suited for extensive data analysis and multidimensional querying.
- **Optimization:** Databases are optimized for high-volume transactional operations, whereas data warehouses are optimized for high-volume analytical operations. This difference in optimization reflects their respective focus on transactional consistency and analytical efficiency.

In summary, while databases and data warehouses both play crucial roles in data management, their distinct functionalities and optimizations serve different purposes within an organization's data ecosystem. Databases handle real-time transactional data efficiently, whereas data warehouses support complex queries and data analysis, providing valuable insights for strategic decision-making [62].

2.5 Data Integration Techniques

There are many sophisticated ways of creating a unified view of data. Data integration techniques are used to consolidate data from different sources. There are several techniques employed for data integration [58, 64]:

- **Manual Data Integration:** This method manually collects and consolidates data from different sources. It is the most basic form of data integration but is often time-consuming and prone to errors.
- **Middleware Data Integration/Data Propagation:** This technique uses middleware software to mediate between different systems and facilitate the transfer and integration of data. It automates the process of moving data between systems.
- **Application-Based Integration/Data Propagation:** This approach integrates data at the application level. It involves modifying applications to communicate with each other directly, enabling them to share and synchronize data.
- **Uniform Access Integration/Data Federation:** Also known as data federation, this method provides a unified view of all data sources without moving or copying the data. It integrates data virtually from various sources, presenting it in a single, cohesive format.
- **Common Storage Integration/Data Warehousing:** In this approach, data from various sources is extracted, transformed, and loaded into a centralized data warehouse. This allows for unified data analysis and reporting.

Each of these techniques has its strengths and is chosen based on the specific requirements of the data integration task.

2.6 ETL

ETL stands for Extract, Transform, and Load, which are the three fundamental steps in the data integration process used to blend data from multiple sources into a data warehouse. ETL is critical for data warehousing, as it ensures that data is accurately extracted from various sources, transformed into a format suitable for analysis, and loaded into the data warehouse for reporting and analytics.

The ETL process, one of the most significant tasks in building a data warehouse, comprises three primary phases: **extraction, transformation, and loading** [20].

Extraction: The first phase of ETL involves extracting data from various source systems. This step requires careful management of different characteristics inherent to each data source, such as varying database management systems, operating systems, and communication protocols. The extraction process typically includes initial extraction, which populates the data warehouse with data from source systems, and incremental extraction or changed data capture (CDC), which refreshes the warehouse with new or modified data since the last extraction.

Methods in Extraction phase:

- **Full Extraction:** without any changes, it is being extracted completely.
- **Partial Extraction:** without update notification.
- **Partial Extraction:** with update notification.

Transformation: The second phase, transformation, involves cleaning and conforming the incoming data to ensure accuracy, completeness, consistency, and unambiguity. This step is crucial for defining the granularity of fact tables, dimension tables, and the overall data warehouse schema. The transformation includes data cleaning, integration, and handling slowly changing dimensions and factless fact tables. All transformation rules and resulting schemas are typically stored in a metadata repository.

ETL Transformation Methods:

- Multistage data transformation;
- Warehouse data transformation;

The primary transformations performed during this step are for verifying the data that will be placed in the warehouse. As outlined in *Data Integration in ETL Using Talend* by Sreemathy et al. [62], the key transformations executed are:

1. Cleaning
2. Filtering
3. Data Standardization
4. Data flow validation

5. Data threshold validation check
6. Row and Column transposing
7. Joining
8. Splitting
9. Sorting

Loading: The final phase, loading, involves writing the extracted and transformed data into the data warehouse's multidimensional structures, such as dimension tables and fact tables, which end users and applications access. This step is critical for making the processed data available for analysis and decision-making, as Figure 2.10 describes.

Types of Loading:

- Initial Load
- Incremental Load
- Full Refresh

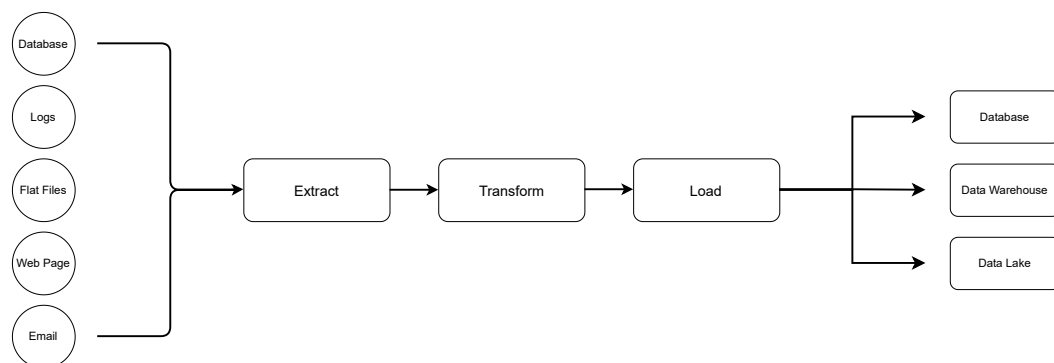


Figure 2.10: ETL process

2.6.1 ETL Tools Categories

The categorization of ETL tools provides insights into their diverse functionalities and use cases. These categories include **Batch Processing**, **Code-Based/Engine-Based**, **Cloud-Based**, **Open Source**, **GUI-Based**, **Real-Time**, and **NoSQL-Based** ETL tools. Each category has unique characteristics and is suited for specific data processing requirements, reflecting the evolution and versatility of ETL technologies in handling big data challenges [50].

- **Batch Processing:** handles data in batches, where the entire file is received, parsed, validated, cleaned, calculated, aggregated, and then delivered for further processing. This traditional approach is used by tools like IBM InfoSphere DataStage, SSIS, Informatica, and Oracle Data Integrator.

- **Code-Based/Engine-Based:** is written in universal programming languages like COBOL or C, whereas engine-based tools are proprietary solutions with unique data engines designed for performance. Oracle Warehouse Builder and SSIS are examples of code-based ETL tools.
- **Cloud-Based:** offers scalability and real-time streaming data processing, integrating with a growing number of data sources. Tools like Matillion, Blendo, Stitch, Fivetran, and Alouma represent this category.
- **Open Source:** preferred for their cost-effectiveness compared to commercial solutions, are popular among system integrators, departmental enterprise developers, and mid-market companies. Apache AirFlow, Apache Kafka, Apache NiFi, and Talend Open Studio are examples of open-source ETL tools.
- **GUI-Based:** provide an easy-to-use interface with drag-and-drop functionality for data loading and analysis. Popular GUI-based tools include Pentaho, Informatica, DataStage, and Abinitio.
- **Real-Time:** cater to the need for immediate data processing, changing the traditional ETL architecture. Tools like Alouma, Confluent, StreamSets, and Striim fall into this category.
- **NoSQL-Based:** With the rise of schema-less databases like MongoDB, NoSQL-based ETL tools have become essential for data integration. Tools that allow data integration with NoSQL databases include MongoSyphon, Transporter, Krawler, Panoply, Stitch, Talend Open Studio, and Pentaho.

2.6.2 Popular ETL Tools

This analysis critically evaluates ETL tools such as Fivetran, Dataddo, Hevo Data, Alouma, Talend, Informatica, IBM DataStage, and AWS Glue, based on features including real-time capabilities, user interfaces, cloud support, and scalability, among others. It aims to reveal each tool's strengths and limitations in meeting the project's nuanced requirements. Through this detailed examination, the necessity for a custom ETL solution, specifically designed to address the unique challenges and intricacies of the project, becomes evident.

2.6.2.1 Fivetran

Fivetran² is a cloud-based, automated data integration solution that allows businesses to extract and load data from disparate sources into a consolidated data warehouse. It streamlines the process of collecting data from several SaaS apps, databases, and other data sources, resulting in a unified and scalable solution for data consolidation and analysis. This service provides simple access to integrated data for business intelligence, reporting, and analytics needs with little user configuration and maintenance.

²<https://www.fivetran.com/>

2.6.2.2 Dataddo

Dataddo³ is a fully managed, no-code integration platform that syncs cloud-based services, dashboarding apps, data warehouses, and data lakes. It provides powerful and adaptable tools for automating data flow between systems, allowing for seamless data integration and transformation. Dataddo's user-friendly interface enables users to efficiently link and consolidate data, making it immediately available for analysis, visualization, and decision-making. This platform is especially beneficial for enterprises that want to leverage the potential of their data without the complications that come with data integration and administration.

As presented on their platform, the key benefits of Dataddo include:

- **250+ Connectors:** offers 250+ off-the-shelf connectors no matter your payment plan.
- **Highly Scalable and Future-Proof:** operates with any cloud-based tools you use now or in the future
- **Insights in Record Time:** enables swift data transfer from sources to destinations, facilitating immediate insights without necessitating a data warehouse
- **Continuous Dashboard Monitoring:** proactively monitors and maintains pipelines and manages all changes to the APIs of cloud services
- **SmartCache Storage:** stores historical data without needing a data warehouse.
- **Comprehensive Technical Support:** Offers extensive support throughout the rollout phases.
- **Backup and Data Migration:** Migrate data between warehouses or back it up to guarantee retention of completeness and quality.
- **Fully Managed and Maintenance-Free:** Manages API changes, monitors pipelines, and constructs new connectors, streamlining the integration process

2.6.2.3 Hevo Data

Hevo Data [4] is a no-code, automated data pipeline platform specializing in ETL processes, enabling seamless integration and transformation of data from diverse sources into a data warehouse. It offers a growing library of 150+ plug-and-play connectors, including all your SaaS applications, databases, file systems, and more.

It provides several features [4], including the following:

- **Multiple Workspaces within a Domain:** allows customers to create multiple workspaces with the same domain name.
- **Multi-region Support:** provides support for maintaining a single account across all Hevo regions, with a maximum of five workspaces

³<http://www.dataddo.com/>

- **ETL Pipelines with In-flight Data Formatting Capability:** provides a no-code ETL solution for data cleansing and preparation using Python code and Drag-and-Drop Transformations.
- **Draft Pipelines:** saves and resumes incomplete pipeline configurations.
- **Historical Data Sync:** fetches your historical data using the Recent Data First approach, getting you the latest Events first. Historical Data is all the data available in your Source at the time of creation of the Pipeline.
- **Flexible Data Replication Options:** Provides data replication methods that may be customized, including complete and incremental modes and scheduling options.
- **Sync from One or Multiple Databases:** allows loading data from multiple databases in the source.
- **Data Deduplication:** deduplicates data based on primary keys to ensure unique records in the destination.
- **Skip and Include Objects:** provides object-level control over data input from sources, including the ability to skip or include objects as needed.
- **Load New Tables with the Same Pipeline:** automatically ingest from any new table created in the Source or any deleted table that is re-created post-Pipeline creation.
- **Smart Assist:** is the prompt, preemptive, and smart assistance built into the product that provides complete visibility and control over your data while helping you to minimize costs.
- **Usage-based Pricing:** offers a variety of subscription plans.
- **Observability and Monitoring:** offers insights into the various aspects of data replication, including latency and speed of data, event failures and usage details.
- **Recoverability:** helps recover from source and destination problems, maintaining data integrity and continuity.

2.6.2.4 Alooma

Alooma is a platform that offers real-time data streaming and utilizes cloud and code engines for data manipulation. It can capture, convert, and store vast quantities of data from various sources and streams. This functionality enables various applications such as analyzing historical records to enhance sales processes, dynamically altering prices and inventories, integrating machine learning and AI for predictive modeling, and developing new revenue streams. These features make Alooma a powerful tool for data integration and analytics [52].

2.6.2.5 Talend

Talend⁴ is an open-source tool for data integration, offering a comprehensive suite of tools and resources for data integration, management, and business process integration [61].

Talend Open Studio⁵ is today's most open, creative, and powerful data integration solution. It can also provide an integration suite, on-demand services, an open profiler, and data quality. Business modeling, real-time debugging, robust execution, graphical development, and metadata-driven design and execution are just a few of the features [62].

2.6.2.6 Informatica

Founded in 1993 in California, Informatica⁶ is a renowned tool for collecting and retrieving data from diverse sources. It provides many features, including Data Integration, Data Security, Data Quality, and Data Management. Its ETL product, Informatica PowerCenter, is known for its transaction throttling ability, platform-independent architecture, and the capability to reverse-engineer mappings into reusable templates. It has sophisticated features such as a Metadata Manager for data flow mapping, performance optimization via parallel processing and load balancing, real-time data transformation, and support for custom transformations written in C and Java [52, 48].

2.6.2.7 IBM DataStage

IBM DataStage⁷ is a well-known ETL technology for integrating data from diverse business systems. It makes use of a powerful parallel framework that is available both on-premises and in the cloud. This adaptable platform supports the Enterprise network and provides full metadata management. DataStage manages heterogeneous data sources successfully, including large-scale data at rest (as in Hadoop-based systems) and data in motion (stream-based) across distributed and mainframe platforms [6].

2.6.2.8 AWS Glue

AWS Glue⁸ is a fully managed cloud-based ETL solution that reads and loads data sources in various formats. AWS Glue is useful in IoT and Big Data since it offers event-driven ETL pipelines and automated execution as new data enters. It is usual for IoT devices to transfer produced data in CSV, parquet, or JSON forms to a central storage location [65].

AWS Glue also simplifies application development, machine learning, and analytics by allowing discovery, preparation, and merging of data. It permits data engineers and ETL developers to create, run, and monitor ETL workflows visually in AWS Glue Studio. Data scientists and analysts may use AWS Glue DataBrew to visually enhance, clean, and standardize data, while application

⁴<https://www.talend.com/>

⁵<https://www.talend.com/products/talend-open-studio/>

⁶https://docs.informatica.com/pt_pt/

⁷<https://www.ibm.com/products/datastage>

⁸<https://aws.amazon.com/pt/glue/>

developers can use AWS Glue Elastic Views to integrate and replicate data across various stores using SQL [52].

2.6.3 Comparative Analysis of ETL Tools

In this section, we will conduct a comparative analysis of the eight ETL tools previously described. This comparison is based on a range of specific features, such as Cloud Support, Parallel Processing, and Real-Time Integration, among others. In Table 2.1, 'Y' indicates the presence of a feature in the ETL tool, and 'N' is its absence.

We established key features for a successful ETL solution based on internet sources. [52, 48] The key features are:

- Accept/Ignore Mechanisms (A/I);
- Real-Time Integration (RI);
- Real-Time Analysis (RA);
- Web-based UI (UI);
- Cloud Support (C);
- Non-RDBMS Connections (N);
- Metadata Management (M);
- Automation (A);
- Horizontal Scalability (S);
- Parallel Processing (P);
- Data Transformation (T);
- Large Volume Performance (VP);
- Human workflow of Error Handling (HE);
- Join Multiple Sources (MS);
- Data Partitioning (DP);

According to the comparison of eight ETL tools shown in the table, it is clear that none of the tools contain all of the mentioned functionalities, revealing a gap in the existing ETL tool offerings. Considering this, the proposed solution seeks to solve these gaps. The framework under development will have extensive functionality, drawing influence from technologies like Fivetran and Dataddo. The choice to be inspired by Fivetran and Dataddo in designing our ETL solution is based on these platforms' distinct features. Fivetran's powerful automation, broad data integration

Feature	Informatica	IBM DataStage	AWS Glue	Talend	Alooma	Hevo Data	Fivetran	Dataddo
A/I	N	N	N	N	N	N	N	N
T	Y	Y	Y	Y	Y	Y	Y	Y
RI	Y	Y	Y	Y	Y	Y	Y	Y
RA	Y	Y	Y	Y	Y	Y	Y	Y
UI	Y	Y	Y	Y	N	Y	Y	Y
C	Y	Y	Y	Y	Y	Y	Y	Y
S	N	Y	N	N	N	Y	Y	Y
N	N	Y	Y	N	N	N	Y	Y
P	Y	Y	N	Y	N	N	N	Y
M	Y	Y	Y	N	N	Y	Y	Y
A	N	Y	Y	N	N	Y	Y	Y
VP	Y	Y	Y	Y	N	Y	Y	Y
HE	N	N	N	N	N	N	N	N
MS	Y	Y	Y	Y	Y	Y	Y	Y
DP	Y	Y	N	Y	Y	Y	Y	Y

Table 2.1: Comparison of ETL Tools

capabilities, and Dataddo’s flexibility and scalability provide an excellent structure for developing a complete and efficient ETL framework. This approach guarantees that the proposed solution covers a broad range of functionality and incorporates the strengths and best practices of the well-established ETL solutions analyzed. The goal is to provide a more adaptable and powerful ETL framework that can address a greater range of data integration and processing requirements.

2.7 Existing Data Integration solutions

The mediator architecture, along with data warehousing, represents two prominent strategies for integrating heterogeneous information sources. In a typical mediator setup, data remains within its original sources, each described by local schemas. A global schema, serving as a unified query entry point, is then crafted and mapped to these local schemas. The connection between the mediator and the data sources is facilitated by wrappers. Two primary methodologies are recognized in this context: Global-as-View (GAV) and Local-as-View (LAV). In GAV, the global schema is envisioned as a compilation of views over the source data. In contrast, LAV conceives local schemas as views of an independently defined global schema [41].

A somewhat parallel approach, not explicitly labeled as a mediator system, was developed by **Adams et al. (2000)** [5] to amalgamate legacy databases within the Boeing corporation. This system utilizes a knowledge-based framework with an inference engine to integrate various databases, serving as a unified data model. This setup requires technically adept personnel to establish and align the global ontology with the existing knowledge representations from legacy databases. While the authors recognized the need for adaptability in incorporating new information sources, their study primarily focused on pre-existing legacy databases without delving into the integration of additional data sources.

The manual crafting of schemas and mappings can become a significant bottleneck in the integration process, highlighting the need for automation. **Reynaud et al. (2001)** [54] investigated the automatic integration of heterogeneous XML data sources through a mediator architecture. Their

system, SAMAG, identifies mappings between Document Type Definitions (DTDs) based on semantic and structural similarities, utilizing WordNet to establish semantic connections between DTD terms through relationships like synonymy, hyponymy, and meronymy (**Fellbaum, 1998**) [22].

Fusco and Aversano (2020) [28] introduced a Data Integration Framework (DIF) that utilizes both GaV and LaV paradigms for semantic integration of heterogeneous data sources as represented in Figure 2.11. The GaV approach defines the global schema as a view over the local schemas, meaning the global schema is derived from the existing schemas of the individual data sources. Conversely, LaV defines local schemas as views over the global schema, implying that each local schema is a perspective of the overarching global schema. These methodologies are enhanced through ontologies, which provide a structured framework of the domain knowledge, enabling different systems to understand the data within the context of its semantics. Ontologies are critical in this framework as they facilitate the alignment and merging of data from different sources by providing a common vocabulary and set of relationships within a specific domain. This semantic layer allows for a more nuanced and meaningful integration of data, which is particularly relevant to the ETL processes outlined in this thesis, where data from diverse sports databases must be harmonized and integrated.

This approach aligns closely with my proposed solution, emphasizing the utilization of semantic technologies to enhance the integration and interoperability of data from heterogeneous sources. By incorporating a similar methodology, my ETL framework benefits from the structured, semantic understanding of ontologies, mirroring the flexibility and depth of data integration seen in Fusco and Aversano's work [28]. This alignment validates the chosen direction for semantic integration in my thesis. It underscores the potential for advanced data harmonization strategies in sports analytics, leveraging our approaches' conceptual and technical synergies.

In a similar vein, the **Knowledge Discovery (KD) SENSO-MERGER (2024)** architecture [34] introduces a novel approach to semantic integration of heterogeneous data sources, including structured and unstructured data. This architecture addresses the complexities of knowledge extraction, data integrity, and scalability in data integration processes. By employing a plugin-based architecture that leverages natural language processing and ontologies for knowledge discovery, KD SENSO-MERGER offers a dynamic solution that complements the objectives of my ETL framework. Such a methodology enhances data's semantic understanding and interoperability, which is pivotal for integrating diverse sports databases. This synergy underscores the relevance of semantic technologies in advancing data harmonization strategies, as depicted in Figure 2.12.

The KD SENSO-MERGER architecture and my proposed ETL framework utilize semantic technologies for data integration, yet they cater to different needs. KD SENSO-MERGER's strength lies in its ability to process structured and unstructured data, leveraging natural language processing for knowledge extraction [34]. In contrast, my solution focuses on integrating structured sports data, prioritizing semantic consistency through ontologies. While KD SENSO-MERGER offers scalability across various data types, my framework is designed specifically for the sports analytics domain, demonstrating the adaptability of semantic integration techniques to

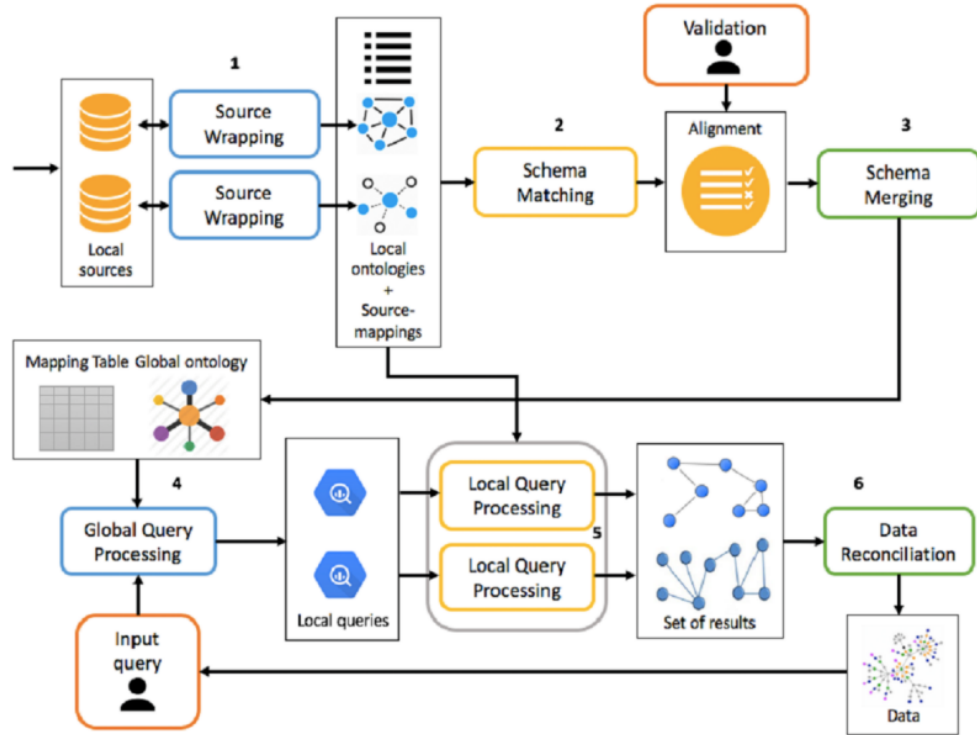


Figure 2.11: Fusco and Aversano solution [28]

specialized fields.

The **"Obi-Wan" system (2020)** [8] integrates heterogeneous data sources using a GLAV-based mediator system, focusing on semantic integration through SPARQL (SPARQL Protocol and RDF Query Language) queries. It uniquely combines data and ontologies, leveraging RDF (Resource Description Framework) models for enhanced query capabilities. This approach represents a significant advancement in handling complex data integration challenges, providing a highly flexible and expressive framework. Comparatively, my proposed ETL framework emphasizes ontological mappings and data transformation processes tailored to sports analytics. While "Obi-Wan" focuses on maximizing query-answering capabilities through an RDF-based integration model, my framework targets the specific needs of sports data integration, offering specialized solutions for this domain. Both systems showcase the power of semantic technologies in enhancing data interoperability, yet they cater to different aspects and requirements of data integration.

2.8 Data Mapping

Data mapping is a critical process in data integration that involves matching data fields from one database system to corresponding fields in another system. This process effectively establishes a bridge that allows for the accurate translation and transfer of information [71]. At its core, data

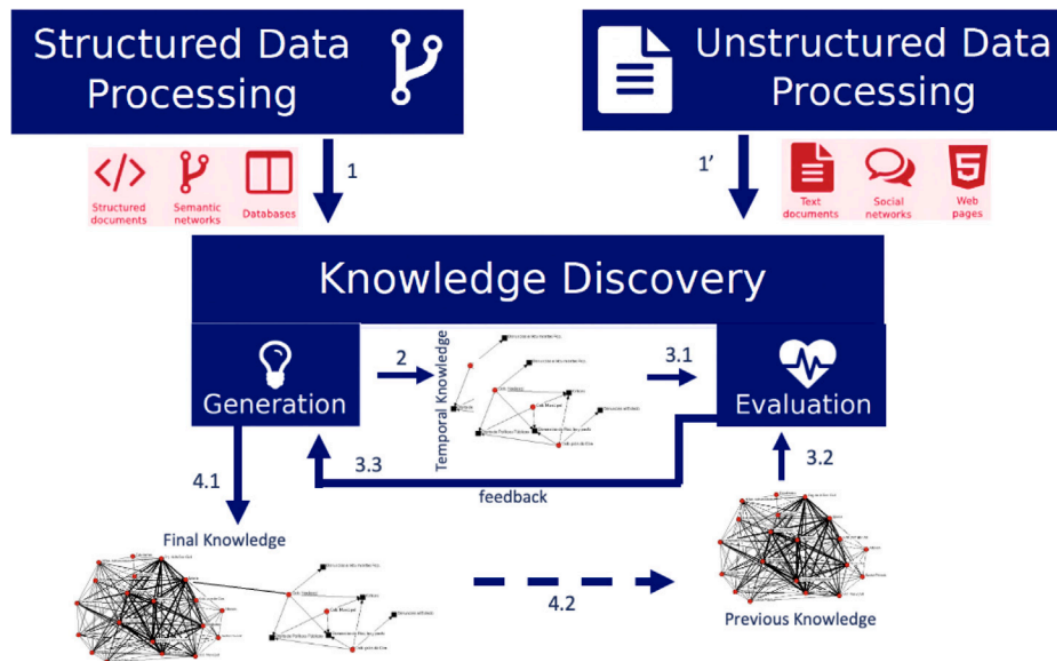


Figure 2.12: KD SENSΟ-MERGER architecture[34]

mapping aligns the diverse syntax and semantic properties of data, transforming and interpreting it so that it is coherent and usefully integrated into new, consolidated analytical environments.

Data mapping is essential for several reasons:

- **Data Integration:** It ensures that data from various sources can be combined and used together in a cohesive manner. By mapping data fields accurately, organizations can integrate disparate datasets, enabling comprehensive analysis and reporting.
- **Data Migration:** When moving data from one system to another, data mapping ensures that data is correctly transferred without loss or misinterpretation. This is crucial during system upgrades, consolidations, or migrations to new platforms.
- **Data Transformation:** Data mapping helps in transforming data into a suitable format for analysis. It aligns different data structures and formats, making it possible to perform complex transformations that prepare data for use in a data warehouse or other analytical systems.

The process begins with identifying the relationships between the source and the destination data fields, often involving complex logic and transformations to ensure the maintained integrity and usability of the data (Yaddow, 2019)[71]. This careful orchestration of data requires expertise in the structure and semantics of the data being mapped, as well as the processes involved in data integrity and cleaning.

Furthermore, data mapping is not a static process. It evolves with changes in the data sources and the integration requirements. As datasets become more complex and new types of data become available (for example, the inclusion of real-time biometric data in sports analytics), this process becomes more dynamic, demanding adaptive data mapping frameworks capable of handling structured and unstructured data (Fletcher, 2005)[27].

In the context of integration, data mapping serves multiple purposes: It can ensure the uniformity required for comparative analyses, facilitate the merging of datasets to enrich data-driven insights, and also underpin the automated features of modern data integration tools, which are increasingly reliant on sophisticated data mapping algorithms to handle the vast volume and variety of contemporary datasets (Yaddow, 2019), (Fletcher, 2005)[71, 27].

As part of the data mapping process, metadata—which includes information about the structure, definitions, and formats of the data—is also critical. Metadata informs the data mapping process and can enhance data discovery and interoperability across various platforms and applications, thus broadening the scope and applicability of the integrated datasets (Tan et al., 2008)[67].

Engagement with data governance teams at the beginning stages of data mapping is vital. These stakeholders can provide crucial insights into the specific needs and nuances of the data that are specific to sports analytics, helping ensure that the data mapping accurately captures the complex relationships and hierarchies present within the data, from player statistics to team performance metrics (Yaddow, 2019)[71].

Data mapping is the linchpin to successful data integration, providing a systematic approach to reconciling heterogeneous data sources. Addressing both the theoretical and practical aspects of data mapping is essential to develop robust frameworks that can handle the intricacies of sports data, drive insightful analyses, and support the strategic decision-making processes within sports organizations (Yaddow, 2019)[71].

2.8.1 Types of Data Mapping

As illustrated by their comprehensive guide [30], the primary types of data mapping include:

- **Manual Data Mapping** This is the most traditional method where data mappings are crafted by individuals who understand the data content, context, and business rules applied to the data during the transformation process. In manual mapping, each field from the source system must be identified and matched to the corresponding field in the target system manually. The mapper defines how each piece of data will be cleaned, transformed, and loaded from the source to the destination. While manual mapping provides a high level of accuracy and is tailored to specific business needs, it is labor-intensive, slow, and can be susceptible to human error, particularly with complex or extensive datasets.
- **Semi-automated Data Mapping** uses software tools to assist in the data mapping process. These tools typically analyze the data schemas of the source and target systems and propose mappings based on naming conventions, data types, and other metadata. They may also suggest transformations for data normalization. A human must then review these suggestions to

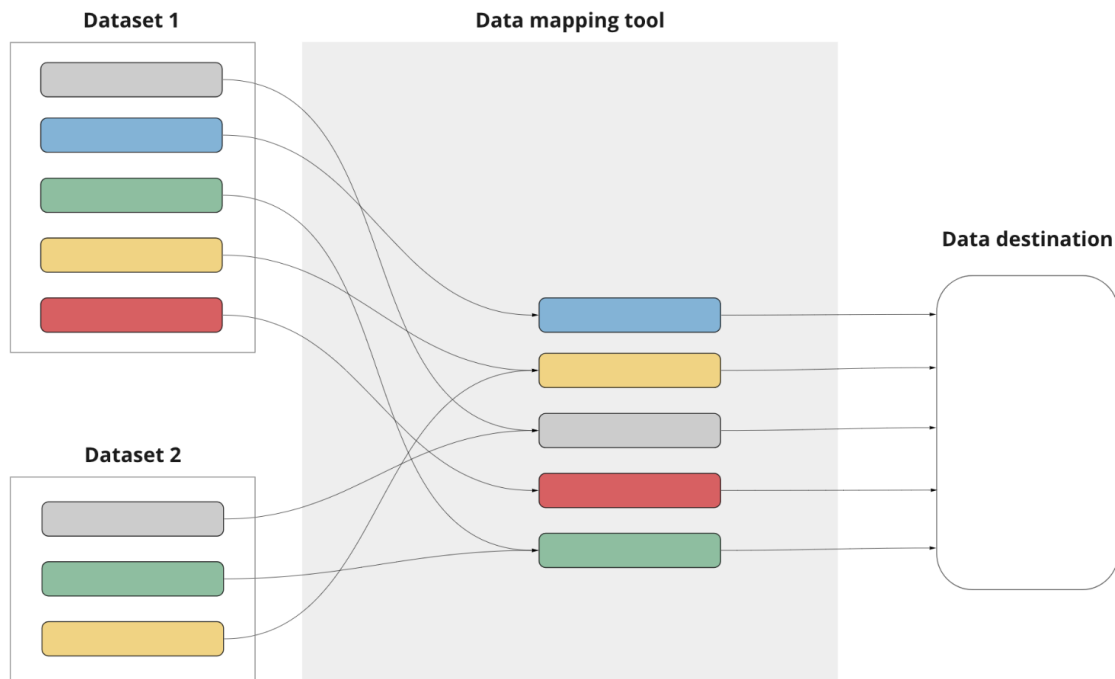


Figure 2.13: Data Mapping[17]

ensure they are correct and make sense in the context of the business logic. Semi-automated methods can accelerate the data mapping process and reduce human error, but they still rely on skilled technicians to review and approve the final mappings. These solutions often work well for medium-scale projects where some automation offers efficiency, but human expertise is still required to ensure data quality and consistency.

- **Automated Data Mapping**, often powered by artificial intelligence and machine learning algorithms, represents the cutting edge in mapping technology. These solutions can process large volumes of data, identify complex patterns, and establish mappings at a speed unattainable by human mappers. Automated tools can also learn from previous mappings, thereby becoming more efficient over time. However, while automated mapping is extremely powerful for handling substantial and straightforward data sets, it might lack the nuanced understanding of unique or complex business rules that govern the data in specific contexts. Automated tools are most valuable when dealing with structured, standard data with clear and consistent transformation rules.
- **Schema Mapping**: Deals with aligning the structure of the datasets, which includes the relationships between tables and columns in databases, XML schemas, or any other structured formats. It is a blueprint that defines how data from the source schema is transformed and loaded into the target schema. The process involves resolving structural differences and may necessitate complex transformations, especially if the source and target adhere to different data models (e.g., converting from relational to non-relational models). Schema mappings

are essential when integrating or migrating between databases where data organization or interpretation may differ fundamentally. They ensure that relationships between entities are maintained and that the integrity of the data is preserved after the mapping is implemented. Due to the potential complexity involved in schema mapping, especially when dealing with heterogeneous systems or legacy data, sophisticated tools and experienced data architects can be required to map one schema accurately to another.

2.8.2 Keys for Data Mapping

Effective data mapping is a detailed and careful process that requires several key considerations to ensure success, such as:

- **Thorough Understanding of Source and Target Systems:** Comprehensively understand the source and target data structures, types, and qualities. This includes not only the formats and fields but also the business context in which the data operates [59].
- **Data Quality Assessment:** Assess the quality of the data before mapping. This means checking for accuracy, completeness, consistency, and relevance. Poor-quality source data can lead to poor-quality target data, regardless of how effective the mapping process is [53, 42].
- **Clear Definition of Data Mapping Rules:** Develop clear, consistent rules for how each piece of data will be mapped, transformed, and, if necessary, converted or formatted. These rules must be well documented [40].
- **Use of Appropriate Data Mapping Tools:** Choose the right tools to support the data mapping process. This can range from simple spreadsheet software to sophisticated ETL tools, depending on the complexity of the task [37].
- **Human Expertise and Intervention:** Ensure enough domain expertise is involved in the process, especially for complex or nuanced datasets. Even with semi-automated or automated processes, human review is crucial [13].
- **Validation and Testing:** Implement a rigorous process for validating and testing the mapped data. This should involve checking that the data meets the required business needs and that all transformations function as expected [39].
- **Scalability and Flexibility:** Design mapping processes that are flexible and scalable to accommodate changes in the data, source systems, and business requirements [15].
- **Compliance and Security:** Consider regulatory compliance requirements, especially data privacy and protection. Ensure that data mapping adheres to these requirements [70].
- **Error Handling and Logging:** Have mechanisms in place for error handling and logging to quickly identify, report, and address any issues during the data mapping process [68].

- **Continuous Review and Improvement:** Make sure to continuously review and update the data mappings as necessary to cater to evolving business needs and to incorporate feedback from ongoing analyses or changes in the source/target systems. Remember, these keys must work harmoniously throughout the data mapping process to ensure that the resulting integrated data system operates effectively and benefits the organization [19].

2.8.3 Data Mapping Challenges

Data mapping encounters several critical challenges that can hinder its effectiveness and efficiency [36], such as:

- **Complex, Manual Processes:** Traditional data mapping often involves manual procedures, which can be time-consuming and error-prone. The complexity of modern data environments exacerbates this issue, requiring meticulous attention to detail. Automating these processes is vital for improving accuracy and efficiency. Manual data mapping requires extensive human intervention, making it prone to errors and inconsistencies. Each data element must be mapped individually, which becomes increasingly difficult as data sets grow in size and complexity. Human error in manual mapping can lead to significant data quality issues, necessitating rework, and additional validation steps. Automation can mitigate these challenges by streamlining the mapping process. Automated data mapping tools can quickly identify and match data elements based on predefined rules and machine learning algorithms. These tools not only reduce the risk of errors but also significantly speed up the mapping process, allowing IT teams to focus on more strategic tasks.
- **Data Diversity:** The increasing diversity of data types, sources, and formats presents a significant challenge. Organizations must handle structured, semi-structured, and unstructured data, each requiring different mapping techniques. Additionally, big data's volume, velocity, variety, and veracity (the four "Vs") demand robust solutions to ensure data consistency and quality. Structured data, such as databases and spreadsheets, are relatively straightforward to map. However, semi-structured data, like JSON files and XML documents, and unstructured data, such as text files and multimedia, require more sophisticated mapping strategies. Each data type has unique characteristics and challenges, necessitating tailored approaches for effective mapping. Furthermore, the sheer volume of data generated at high velocities complicates the mapping process. Ensuring data veracity—accuracy and reliability—requires robust data quality management practices integrated into the mapping process. Advanced tools that handle diverse data types and large volumes can help maintain data integrity and consistency.
- **Performance Issues:** Inefficient data mapping can lead to performance bottlenecks, increasing processing times and operational costs. Inaccurate mappings can misinterpret data, affecting downstream analytics and decision-making processes. Implementing intelligent mapping tools that optimize data flow and transformation is crucial to mitigate these issues.

Performance issues often arise from poorly designed mapping rules that do not account for the complexities of data relationships and dependencies. These inefficiencies can cause delays in data processing, impacting the timeliness and relevance of the data. Moreover, performance bottlenecks can escalate operational costs due to increased resource utilization and processing time. Intelligent mapping tools leverage advanced algorithms to optimize data transformations and flow. These tools can dynamically adjust mappings based on data characteristics and usage patterns, ensuring efficient data processing. By minimizing performance bottlenecks, organizations can enhance their data integration efforts' overall efficiency and effectiveness.

- **Trust and Transparency:** Ensuring end-to-end visibility and traceability in data mapping processes is essential for building trust within an organization. Teams need to be confident in the accuracy and reliability of their data transformations. Lack of transparency can lead to skepticism and reduced trust in the data, impacting overall data governance and compliance efforts. Transparency in data mapping involves providing clear documentation and lineage of data transformations. This traceability allows stakeholders to understand how data is transformed and integrated across different systems. With this visibility, ensuring data accuracy and compliance with regulatory requirements becomes easier. Building trust requires robust data governance frameworks that include comprehensive documentation of mapping processes, regular audits, and validation checks. Transparent data mapping practices help organizations maintain data integrity, comply with regulatory standards, and foster a culture of trust in data-driven decision-making.

2.8.4 Data Mapping Tools

2.8.4.1 Talend⁹

Features:

- **Open-Source:** Talend offers a powerful open-source version that is freely available, along with more feature-rich enterprise editions.
- **Graphical Interface:** The user-friendly graphical interface allows for drag-and-drop job design, making it accessible to users with varying levels of technical expertise.
- **ETL Capabilities:** Provides comprehensive ETL functionalities, supporting complex data transformations and workflows.
- **Connectors:** Features an extensive range of connectors for databases, cloud services, enterprise applications, and more, including Salesforce, AWS, and Hadoop.
- **Data Transformation:** Advanced data transformation tools include filtering, joining, aggregating, and cleansing data, as well as support for complex business rules.

⁹<https://www.talend.com/products/data-integration/>

- **Scalability:** Talend scales effectively from small projects to large enterprise implementations, supporting parallel processing and big data integration.
- **Real-Time and Batch Processing:** Supports both real-time and batch data processing, making it versatile for different use cases.

Best for

- **Complex Data Integration:** Ideal for projects requiring sophisticated data integration and transformation.
- **Diverse Data Sources:** Suitable for enterprises dealing with a wide variety of data sources and needing robust connectivity options.
- **Customizable Solutions:** Organizations looking for a flexible and customizable data integration tool.

2.8.4.2 Informatica PowerCenter¹⁰**Features:**

- **High Scalability:** Designed to handle large data volumes and complex data integration scenarios efficiently.
- **Extensive Transformations:** Provides a broad range of data transformation options, including data aggregation, sorting, joining, and more.
- **Metadata Management:** Advanced metadata management features help maintain data lineage and governance, ensuring data accuracy and compliance.
- **Data Quality:** Integrated data quality tools ensure that data is accurate, complete, and consistent across the enterprise.
- **Big Data Support:** Supports integration with big data platforms like Hadoop, enabling processing and transformation of large datasets.
- **Real-Time Data Integration:** Capable of real-time data integration for time-sensitive data processing and analytics.

Best for

- **Large Enterprises:** Ideal for large organizations with significant data integration needs and complex data environments.
- **Data Governance:** Enterprises requiring strong data governance, quality management, and compliance capabilities.
- **High-Performance Needs:** Organizations that need to handle large volumes of data with high performance and scalability.

¹⁰<https://www.informatica.com/products/data-integration/powercenter.html>

2.8.4.3 Microsoft SQL Server Integration Services (SSIS)¹¹

Features:

- **ETL Capabilities:** Comprehensive ETL functionalities to extract, transform, and load data efficiently.
- **Integration with SQL Server:** Seamless integration with Microsoft SQL Server, making it a natural choice for organizations already using SQL Server.
- **Batch Processing:** Efficient batch data processing capabilities, suitable for handling large datasets.
- **Scriptable Components:** Allows for customization of tasks through script components, supporting both VBScript and C#.
- **User-Friendly:** An intuitive interface with drag-and-drop features simplifies the creation and management of ETL workflows.
- **Data Warehousing:** Supports data warehousing solutions, making it easier to integrate and manage data across the enterprise.

Best for

- **Microsoft Ecosystem:** Organizations already using Microsoft SQL Server, seeking a tightly integrated ETL solution.
- **Cost-Effective ETL:** Businesses looking for a cost-effective ETL tool with robust capabilities.
- **Batch Processing Needs:** Ideal for environments where batch processing of data is a primary requirement.

2.8.4.4 Apache Nifi¹²

Features:

- **Open-Source:** Free to use, with a strong community support and frequent updates.
- **Real-Time Data Ingestion:** Capable of handling real-time data flows and streaming data, making it suitable for dynamic data environments.
- **User-Friendly Interface:** A visually intuitive interface that allows users to design data flows using drag-and-drop components.

¹¹<https://docs.microsoft.com/en-us/sql/integration-services/>

¹²<https://nifi.apache.org/>

- **Processor Library:** An extensive library of processors for various data operations, including data ingestion, transformation, routing, and more.
- **Scalability:** Scales well for large data volumes and can be deployed in clustered environments for high availability and performance.
- **Data Provenance:** Built-in data provenance features track data as it flows through the system, ensuring traceability and auditing.

Best for

- **Real-Time Data Integration:** Ideal for projects that require real-time data processing, such as IoT data flows.
- **Flexible and Customizable:** Organizations looking for a flexible and customizable open-source data integration tool.
- **Data Flow Management:** Businesses needing robust data flow management and monitoring capabilities.

2.8.4.5 MuleSoft Anypoint Platform¹³**Features:**

- **API-Led Connectivity:** Emphasizes API management and connectivity, enabling seamless integration of applications and data sources.
- **Extensive Connectors:** Offers a wide variety of pre-built connectors for various applications, databases, and cloud services.
- **Cloud Integration:** Native support for cloud-based data integration, making it easy to connect cloud and on-premises applications.
- **Real-Time Integration:** Supports real-time data processing and integration, enabling timely and accurate data flows.
- **Unified Platform:** Combines data integration, API management, and analytics in one cohesive platform.
- **Developer-Friendly:** Provides tools and resources for developers to create, manage, and deploy APIs and integrations efficiently.

Best for

- **API Management:** Businesses requiring robust API management capabilities alongside data integration.

¹³<https://www.mulesoft.com/platform/enterprise-integration>

- **Cloud-First Strategy:** Organizations looking for a comprehensive platform that supports both cloud and on-premises integrations.
- **Scalable Solutions:** Enterprises needing a scalable and flexible integration solution that can grow with their needs.

2.8.5 Related Work

Yaddow (2019) emphasizes that data mapping is crucial for data integration. He explains that data mapping serves as a bridge between different database systems, enabling the accurate transfer and transformation of data. This process involves aligning the syntax (format and structure) and semantics (meaning) of data from different sources to ensure that the integrated data is coherent and useful in analytical environments. This alignment is essential for maintaining the integrity and utility of the data when it is used for analysis or reporting [71].

Fletcher (2005) discusses the evolving nature of data mapping. As data sources change over time and new types of data, such as real-time biometric data, become important for analytics, data mapping frameworks must adapt. This adaptation involves handling both structured data (like databases) and unstructured data (like text files or sensor data). Fletcher highlights the necessity for data mapping frameworks to be flexible and capable of managing these diverse data types effectively [27].

Tan et al. (2008) explore how metadata is critical in data mapping. Metadata includes information about the structure, definitions, and formats of data, and it plays a significant role in enhancing data discovery and interoperability across different platforms and applications. By providing detailed descriptions of data elements, metadata informs the data mapping process, making it easier to integrate datasets from various sources and ensuring that they can be effectively used together [67].

Silvers (2012) [59] outlines several keys to successful data mapping.

These include:

- **Understanding source and target systems:** Knowing the details of where data is coming from and where it is going.
- **Data quality assessment:** Ensuring that the data being mapped is of high quality.
- **Defining data mapping rules:** Clearly specifying how data should be transformed and transferred
- **Using appropriate tools:** Selecting the right tools for the data mapping task. Silvers also emphasizes the importance of human expertise, thorough validation and testing, scalability (ability to handle growing amounts of data), compliance with regulations, and continuous improvement of the data mapping process.

Redman (2008) and Loshin (2010) focus on the importance of managing data quality in the context of data mapping. They argue that poor-quality source data can lead to inaccurate target

data, no matter how well the mapping process is designed. Therefore, ensuring high-quality data from the start is critical for successful data mapping. This includes identifying and correcting errors in the data before it is mapped [53, 42].

Kimball (2013) delves into the technical details of defining data mapping rules and implementing robust data validation and testing processes. He stresses the need for clear documentation of data mappings and rigorous testing to ensure they meet business needs and function as intended. This includes creating detailed documentation for each step of the mapping process and conducting thorough tests to validate the accuracy and completeness of the mapped data [40].

Inmon (2010) discusses the importance of choosing the right data mapping tools. Depending on the complexity of the task, these tools can range from simple spreadsheet software to sophisticated ETL (Extract, Transform, Load) tools. Inmon highlights that selecting the right tools is crucial for effective data mapping. He also emphasizes the role of data governance in maintaining data integrity and compliance with regulations, ensuring that data mapping processes are well-managed and adhere to required standards [37].

Davenport (2007) and Eckerson (2011) underscore the importance of human expertise and ongoing review in the data mapping process. They argue that even with advanced automated tools, human intervention is essential for handling complex or nuanced datasets. Continuous review and improvement of the data mapping process are necessary to adapt to evolving business needs and to incorporate feedback from ongoing analyses. This ensures that the data mapping process remains accurate and relevant over time [13, 19].

2.9 API

In the realm of software development, an Application Programming Interface (API) is a crucial component that serves as a bridge between different software programs. It allows these programs to communicate with each other by defining a set of rules and protocols. This communication is essential for building complex software systems where different components must interact and share data seamlessly and efficiently. APIs are fundamental in modern software architecture, providing the necessary interfaces for web services, data exchange, and application functionality integration [46]. APIs facilitate interaction between applications and operating systems, enabling remote procedure calls, such as Java's remote method invocation. They also serve in libraries and frameworks, offering language bindings to translate functionalities across programming languages. API architectures are categorized into REST, JSON-RPC/XML-RPC, and SOAP, each with distinct communication protocols and data exchange formats to support various application needs.

2.9.1 REST

Roy Fielding introduced Representational State Transfer (REST) in his 2000 Ph.D. dissertation [26], presenting it as an architectural style for designing networked applications. REST emphasizes a stateless communication protocol, typically using HTTP (Hypertext Transfer Protocol), where each request from a client to a server contains all the information needed to understand

and complete the request. This approach simplifies the architecture of web services, making them more scalable, performant, and easy to integrate with various web technologies.

RESTful APIs, following the principles of Representational State Transfer (REST), have become a fundamental part of modern web services due to their simplicity, scalability, and flexibility.

The primary HTTP methods used in RESTful services include GET, POST, PUT, and DELETE, each serving a distinct role in resource manipulation:

- **GET:** This method is used to retrieve information from the specified server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
- **POST:** This method is used to send data to the server. For example, customer information, file upload, etc. using an HTML form. The POST method is often used when submitting form data or uploading a file. It sends data to the server to create or update a resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.
- **PUT:** This method replaces all current representations of the target resource with the uploaded content. It is used to update existing data or create a new resource at a specific URI, in cases where the resource URI is known by the client.
- **DELETE:** This method removes all current representations of the target resource given by a URI.

REST imposes a set of constraints on the architecture of web services, which, when adhered to, enable systems to be more performant, reliable, and scalable.

These constraints are as follows:

- **Client-server architecture managed through HTTP:** RESTful APIs use HTTP for communication between clients and servers, which are distinct entities. The client initiates requests to perform actions (e.g., retrieve, update, or delete resources), and the server processes these requests and returns responses. This separation allows for independent evolution of client and server technology stacks [25].
- **Stateless communication:** Each request from the client to the server must contain all the information the server needs to fulfill the request. The server does not store any session information about the client between requests. This statelessness ensures that each request can be understood in isolation, which simplifies server design and improves scalability [24].
- **Caching:** RESTful APIs are designed to support caching at various levels. Responses from the server can be explicitly marked as cacheable or non-cacheable, which helps to reduce client-server interactions and improve the efficiency and scalability of the application by reusing previously fetched resources [25].

- **Uniform interface:** RESTful APIs standardize interactions between clients and servers through a consistent interface, streamlining and modularizing the system for independent evolution. Fundamental concepts include identifying resources in requests, resource representation in communications, self-explanatory messages, and leveraging hypermedia for navigating application states [24].
- **Layered system:** REST allows for an architecture composed of layers, each with a specific function (e.g., load balancing, security enforcement). This layering increases the system's scalability by enabling load distribution across various servers and enhances security through encapsulation of systems [25].
- **Code on demand (optional):** allows servers to enhance or customize client capabilities by sending executable code (e.g., JavaScript) for the client to execute. This feature, while optional, allows for a more interactive and adaptable client experience [10].

2.9.2 API Management

API Management encompasses the strategies and tools used to control and monitor the APIs an organization exposes both internally and externally. As outlined in *Continuous API Management* by Medjaoui et al. [46], it involves creating and maintaining APIs and their documentation, security, and performance analysis. Essentially, the continued balance of these three elements - **scope, scale, and standards** - powers a healthy, growing API management program.

2.9.2.1 Scope

The scope of API Management extends beyond mere technical aspects, including strategic planning for API exposure, determining the target audience (internal developers, partners, or external customers), and aligning APIs with business goals. This strategic component is essential for ensuring that APIs serve the broader objectives of the organization, facilitating seamless integration and interaction among different software systems and services [45].

2.9.2.2 Scale

As an organization grows, so does its API ecosystem. Managing the scale involves ensuring that the API infrastructure can handle increased loads, providing consistent performance, and facilitating the integration of many services and data sources. The growth of the API ecosystem necessitates robust management practices to ensure scalability and reliability of API services [60].

2.9.2.3 Standards

Adhering to standards is crucial in API Management. This includes following industry best practices for RESTful API design, security protocols like OAuth, and data exchange formats like JSON or XML. Standardization ensures interoperability, easier maintenance, and better compliance with

regulatory requirements. The establishment of API design guidelines and the adoption of standardized practices are pivotal for the consistent development and deployment of APIs across different platforms and systems [21].

In summary, effective API Management is indispensable for harnessing the full capabilities of APIs, requiring a strategic approach to manage its scope, scale, and compliance with standards. These elements are foundational to a thriving API ecosystem, enabling organizations to achieve their digital transformation objectives and maintain a competitive edge in the digital marketplace.

2.10 Conclusion

The current data integration solutions are advanced and comprehensive but do not entirely address the issue of converting different sports data into a standard format. These solutions often fall short in handling diverse and complex data sources, integrating real-time data efficiently, and creating precise mappings to maintain data consistency and integrity.

Existing methods are limited in their ability to harmonize various data structures and formats, making it challenging to achieve seamless integration of sports data. Although these solutions offer a range of functionalities, they do not fully meet the specific needs of sports data integration, particularly in terms of flexibility and precision required for effective data transformation.

To address these challenges, exploring advanced data mapping techniques and automation tools is crucial. Utilizing some of the ideas and concepts from the current state-of-the-art solutions, the focus should be on developing a framework that can harmonize different data structures and formats more effectively. This approach would ensure accurate and efficient integration of diverse data sources, providing a standardized dataset ready for integration into the final database.

By leveraging the strengths of existing technologies and addressing their limitations, it is possible to create a robust framework for data integration in the sports domain. This framework would enhance the overall effectiveness of data integration efforts, ensuring that diverse data sources are accurately and efficiently transformed into a consistent format, ready for various applications.

Chapter 3

PlayField

This chapter outlines the PlayField framework, detailing its features and implementation. Leveraging various technologies and libraries, the PlayField framework provides a robust interface for data interaction and visualization. This chapter offers a comprehensive overview of each core feature of the framework, highlighting its functionality and contribution to the overall system.

The discussion begins with a detailed examination of the framework’s requirements in Section 3.1, divided into five key subsections: user interface and interaction, rule management, data mapping and rule creation, rule storage and application, and automation through API. Each subsection will explore specific features designed to facilitate efficient data handling, transformation, and visualization, ensuring a seamless user experience and meeting the needs of various stakeholders.

Section 3.2 will delve into PlayField architecture and overall design. We will discuss the framework’s key components, technologies, and how they integrate to facilitate efficient data transformation and integration. This examination will illustrate the robustness and versatility of the PlayField framework in handling complex data sets and maintaining data integrity in real-world applications.

3.1 Requirements

The PlayField framework encompasses several core areas designed to facilitate efficient data handling, transformation, and visualization. These areas collectively provide a seamless user experience and ensure the framework meets the needs of various stakeholders. The features are grouped into specific categories within each area, providing detailed descriptions of how they function and contribute to the overall framework:

3.1.1 User Interface and Interaction

The user interface (UI) of PlayField is designed to be intuitive and user-friendly, enabling users to perform complex data integration tasks with ease. Streamlit[38], the Python library used to build the UI, allows for rapid development and deployment of interactive web applications.

3.1.1.1 Dropdowns Menus for Data Selection

The PlayField framework starts with a user-friendly interface featuring dropdown menus for selecting the type of input data and the corresponding data provider, as shown in Figure 3.1. This intuitive design allows users to easily choose from predefined data types and providers or add new ones:

- **Type of Information Dropdown:** Users can select the type of information they are dealing with, such as match feeds, competitions, or odds data. This helps in filtering the data and applying the correct transformation rules.
- **Provider Selection Dropdown:** Allows users to select an existing provider or add a new one using the "Add New Provider" option. When adding a new provider, a text input field appears where users can enter the provider's name. Upon submission, the new provider is saved in the database and becomes the pre-selected option in the dropdown.

The screenshot shows a dark-themed web interface. At the top, it says "Welcome to PlayField" with a gear icon. Below this, there are two dropdown menus. The first is labeled "Select the type of info" and has "Match Feed Info" selected. The second is labeled "Select the name of the provider" and has "New provider" selected. Below the second dropdown, there is a text input field labeled "Name of provider". At the bottom left of the form area is a "Submit" button.

Figure 3.1: PlayField Dropdowns

3.1.1.2 File Uploader

After selecting the type of info of the file and the corresponding provider, users can upload the input file they wish to convert, as shown in Figure 3.2. This functionality ensures that the framework can handle various file types and sizes, making it adaptable to different data sources.

- **File Uploader:** The file uploader component allows users to drag and drop files or select files from their computer. Supported file types include CSV, JSON, and XML, among others text files. This flexibility is essential for dealing with diverse data formats from different providers.



Figure 3.2: PlayField File Uploader

3.1.2 Rule Management

Effective rule management is a fundamental aspect of the PlayField framework, enabling users to specify the transformation of data from its original format to the standardized format required by the target database (refer to Figure 3.3).



Figure 3.3: PlayField Rule Buttons

3.1.2.1 Preview and Edit Rules

The framework provides robust rule management features to ensure accurate data transformation:

- **Preview Rules:** A button that opens a pop-up, as indicated in Figure 3.4, displaying all existing rules for the selected provider. Users can review these rules and delete any that are no longer needed. This feature is crucial for maintaining data integrity and ensuring that outdated or incorrect rules do not affect the transformation process. Note that rules are specific to each provider, and users can only preview the rules associated with the currently selected provider.

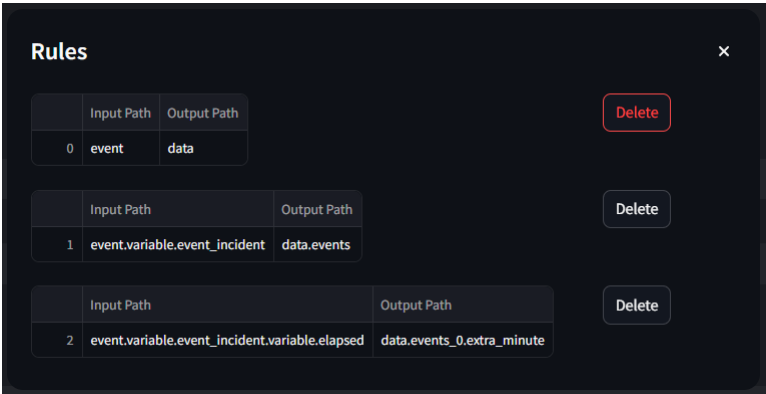


Figure 3.4: PlayField View Rules Pop-up

- **Edit Output Fields:** This feature allows users to see the important/mandatory fields in the output file. By toggling checkboxes, users can include or exclude fields, tailoring the output to their needs as shown in Figure 3.5. This customization ensures that only relevant data is included in the final output, optimizing both storage and processing efficiency.



Figure 3.5: PlayField Edit Output Fields

3.1.3 Data Mapping and Rule Creation

The data mapping and rule creation process is a critical step in transforming the input data into a standardized output format. This subsection explains how users can map fields and create rules using the PlayField framework.

3.1.3.1 Input and Output Nodes

The interface displays two columns, as demonstrated in Figure 3.6, representing the input and output nodes with their respective fields. Users can map these fields to create rules that define how data from the input should be transformed to fit the output structure.

- **Input Nodes:** These are the fields present in the uploaded input file. They represent the raw data from the provider, which needs to be transformed.
- **Output Nodes:** These are the fields required in the final output file. They represent the standardized format used by the target database.

Input Node: data	Output Node: data
<pre> { "league_id" : 2 "referee_id" : 18671 "localteam_id" : 3468 "visitorteam_id" : 503 "winner_team_id" : 3468 "weather_report" : {...} "commentaries" : true "attendance" : 76579 "pitch" : "Good" "details" : NULL "neutral_venue" : false "winning_odds_calculated" : true "formations" : {...} "scores" : {...} "time" : {...} "coaches" : {...} "standings" : {...} "assistants" : {...} "leg" : "2/2" "colors" : {...} "deleted" : false "is_placeholder" : false "lineup" : {...} "bench" : {...} "stats" : {...} "comments" : {...} "referee" : {...} "localCoach" : {...} </pre>	<pre> { "sport_id" : "1" "state_id" : "18996932" "name" : "Shakhtar Donetsk vs Porto" "starting_at" : "2023-09-19 19:00:00" "result_info" : "Porto won after full-time." } </pre>

Figure 3.6: PlayField Input and Output Nodes

3.1.4 Dropdown for Node Mapping

Each column contains dropdowns with the names of the nodes, enabling users to select and match nodes from the input to the output, as observed in Figure 3.7. This mapping process is crucial for establishing a clear and accurate data transformation path.

- **Dropdown for Input Nodes:** Users can select the appropriate input field from a dropdown list.
- **Dropdown for Output Nodes:** Users can select the corresponding output field from a dropdown list.

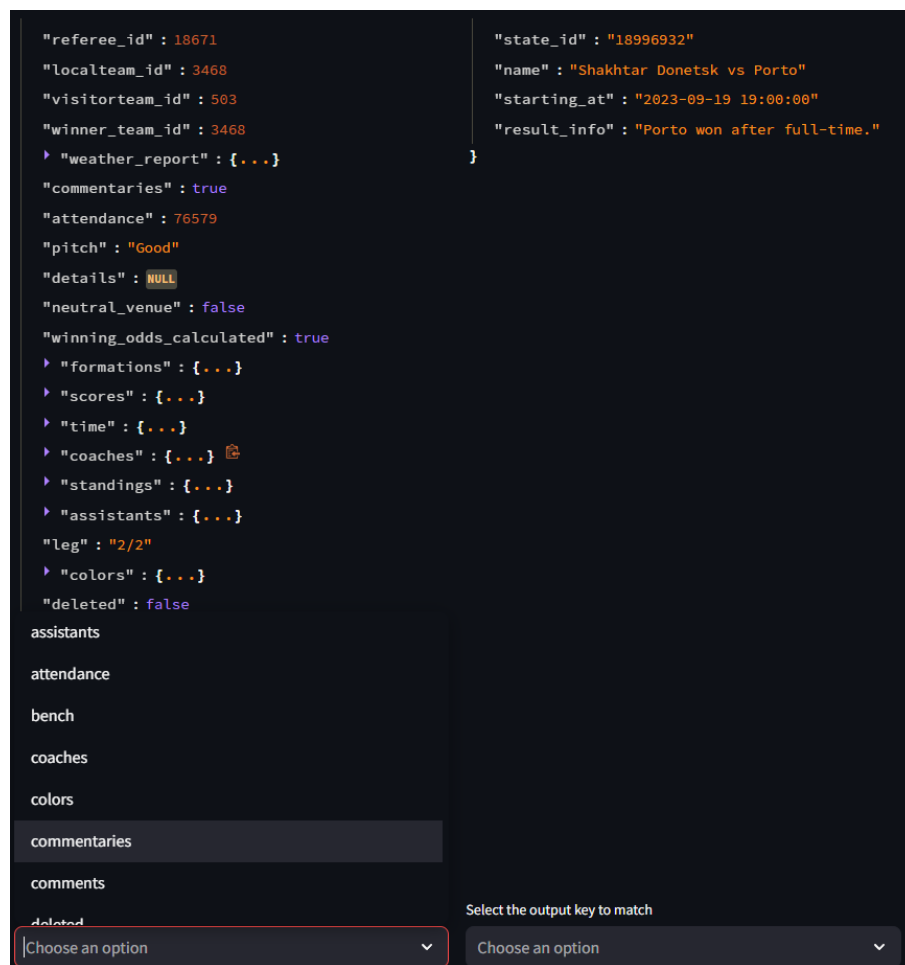


Figure 3.7: PlayField Dropdown for Creating Rules

3.1.5 Rule Storage and Application

3.1.5.1 Database Storage

Once rules are created, they are stored in the PostgreSQL¹ database. This storage mechanism ensures that all transformations are recorded and can be reused or modified as needed.

- **Rule Storage:** Rules are stored in a structured format, enabling quick retrieval and application during the data transformation process.

3.1.5.2 Exact Match and Hierarchical Rule Creation

The framework supports exact matches between input and output fields and hierarchical rule creation for nested structures like arrays and dictionaries. This flexibility is essential for handling complex data formats and ensuring that all relevant information is accurately transformed.

- **Exact Match:** Ensures that fields with identical names are mapped directly, simplifying the rule creation process.
- **Hierarchical Rule Creation:** Allows users to define rules for nested structures, such as arrays within dictionaries. This is particularly useful for complex data sets that contain multiple levels of nested information.

3.1.6 Exporting Transformed Data

After the rules are applied, users can export the transformed input data into a standardized format, typically a JSON file. This export feature ensures that the data is ready for integration into the target database or other systems.

- **Export Button:** A button labeled "Export Transformed Input" triggers the export process, converting the transformed data into a JSON file as shown in Figure 3.8.

¹<https://www.postgresql.org>

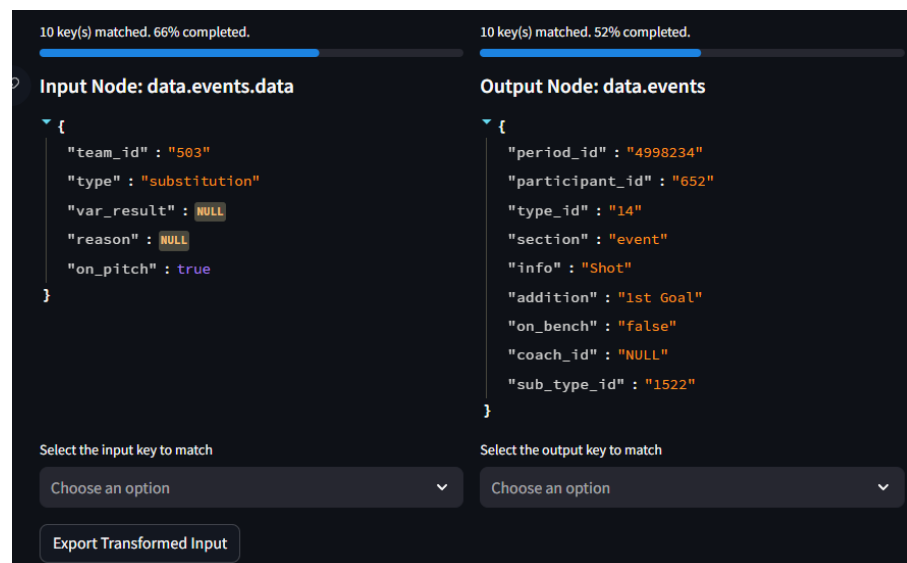


Figure 3.8: PlayField Export Button

3.1.7 Automation through API

To further enhance the framework's capabilities, an API was developed to automate the data integration process. This subsection describes the API endpoints, as presented in Figure 3.9 and their functionalities.

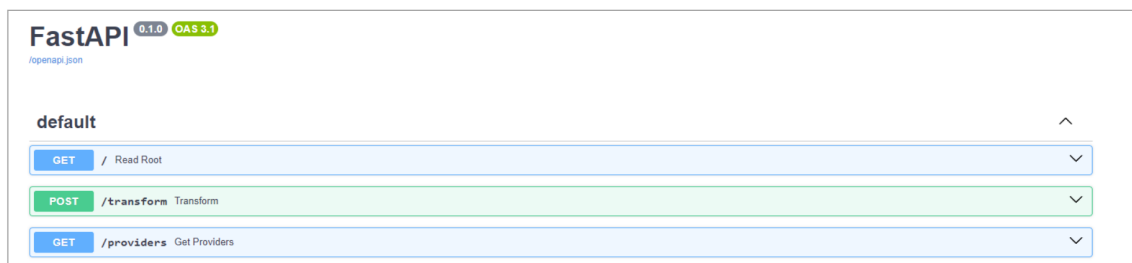


Figure 3.9: PlayField API Endpoints

3.1.7.1 API Endpoints

The PlayField framework includes two API endpoints to facilitate automated data transformation and integration:

- **List Providers Endpoint:** Returns a list of all registered providers, allowing users to see which providers are available for data transformation, as illustrated in Figure 3.10

Endpoint: /api/providers

Method: GET

Response: List of providers in JSON format

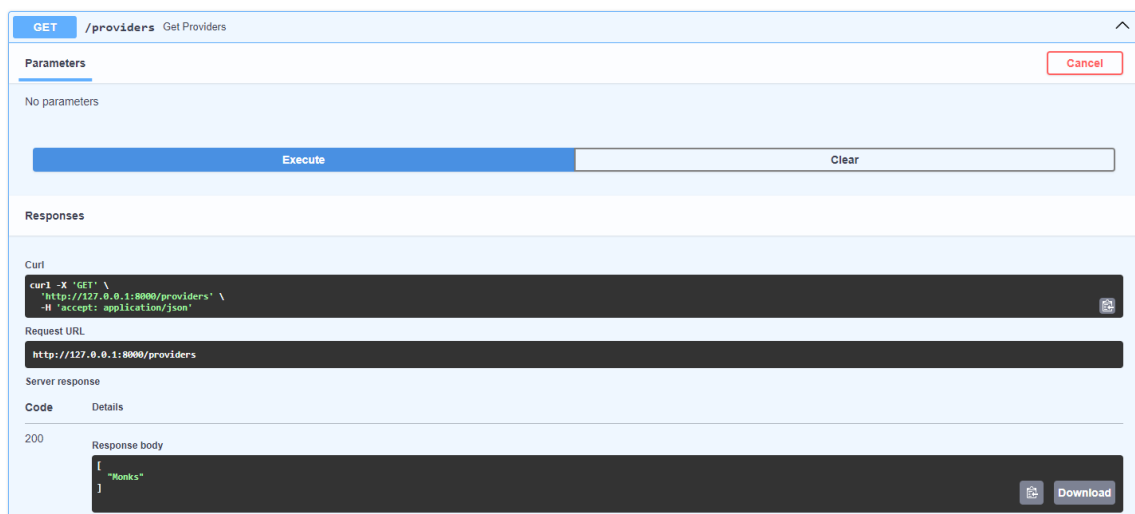


Figure 3.10: GET Providers Endpoint

Example API Call:

```
curl -X GET http://playfield.api/providers
```

- **Transform Input Endpoint:** This endpoint accepts the provider's name and the input file in the request body, returning the transformed data. This allows for seamless integration with external systems and automated workflows as shown in Figures 3.11 and 3.12.

```
Endpoint: /api/transform
Method: POST
Parameters: provider_name, input_json
Response: Transformed data in JSON format
```

Example API Call:

```
curl -X POST -F "provider_name=ProviderB"
-F "input_json=@player_stats.json"
http://playfield.api/transform
```


3.2 Architecture

To provide a comprehensive understanding of the PlayField framework, this section delves into its technical architecture, the technologies employed, and the overall data flow. The PlayField framework is constructed on a modular architecture designed to ensure scalability, flexibility, and maintainability, as shown in Figure 3.13. This section will detail the critical components of the framework, including the frontend, backend, database, and API, highlighting their roles and interactions. We will explore the specific technologies utilized, such as Streamlit for the frontend and PostgreSQL for the database, and examine the domain model that defines the core entities and relationships within the system. Furthermore, the workflow subsection will present a detailed step-by-step process of data transformation and integration, showcasing how the framework efficiently manages and processes complex data sets. Through this detailed examination, the architecture section aims to illustrate the robustness and versatility of the PlayField framework in real-world applications. The key components include:

- **Frontend:** Developed using Streamlit, providing an interactive and responsive user interface.
- **Backend:** Implemented in Python, handling the logic for data transformation and rule management.
- **Database:** PostgreSQL, used for storing rules, providers, and the input and output data.
- **API:** RESTful API built with FastAPI², enabling automated data processing and integration.

²<https://fastapi.tiangolo.com>

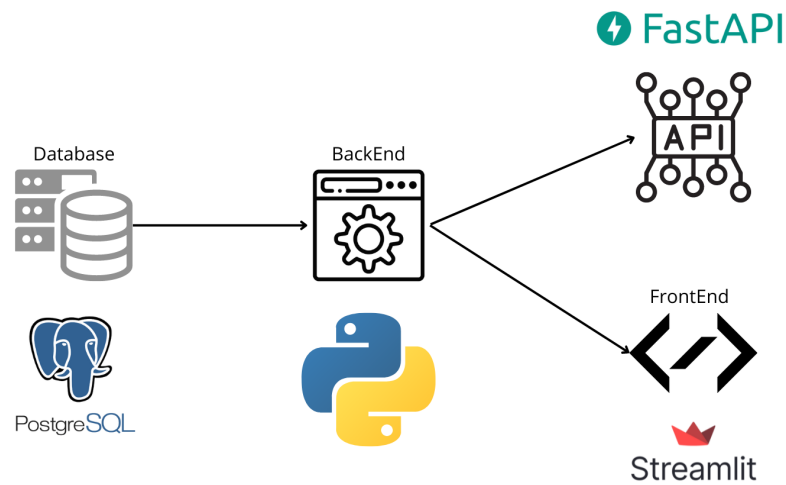


Figure 3.13: PlayField Architecture

3.2.1 Technologies

- **Streamlit:** Provides a simple way to create interactive web applications in Python. It is used for the frontend of the PlayField framework.
- **Python:** The primary programming language used for the backend logic and data processing.
- **PostgreSQL:** A powerful relational database system used to store rules, providers, and the input and output data.
- **FastAPI:** modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.

3.2.2 Domain Model

The domain model provides a conceptual representation of the key entities and their relationships within the PlayField framework as shown in Figure 3.14. This model is crucial for understanding how data is structured and interacts across different components.

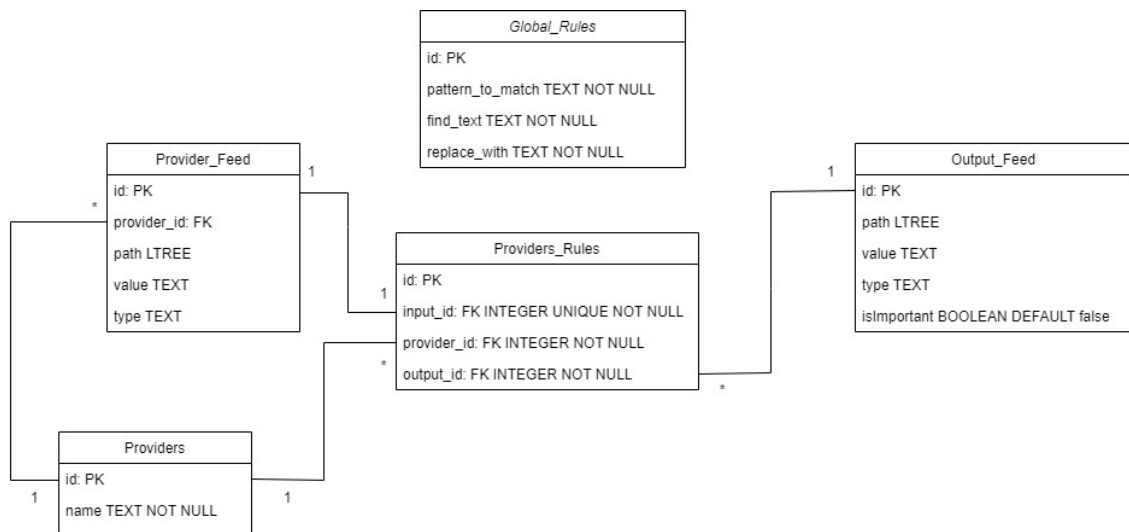


Figure 3.14: PlayField Domain Model

The domain model consists of the following entities:

- **Global_Rules:** Represents the global transformation rules that apply across all providers.
- **Providers:** Contains information about each data provider.
- **Provider_Feed:** Stores the raw data feeds from providers.
- **Providers_Rules:** Defines the mapping rules for transforming provider-specific data to a standardized format.
- **Output_Feed:** Contains the standardized format.

3.2.3 Workflow

The PlayField framework follows a structured workflow to ensure efficient and accurate data transformation. This workflow is visually represented in Figure 3.15, providing a step-by-step guide from accessing the application to exporting the transformed data. Below is a detailed description of each step involved in the process:

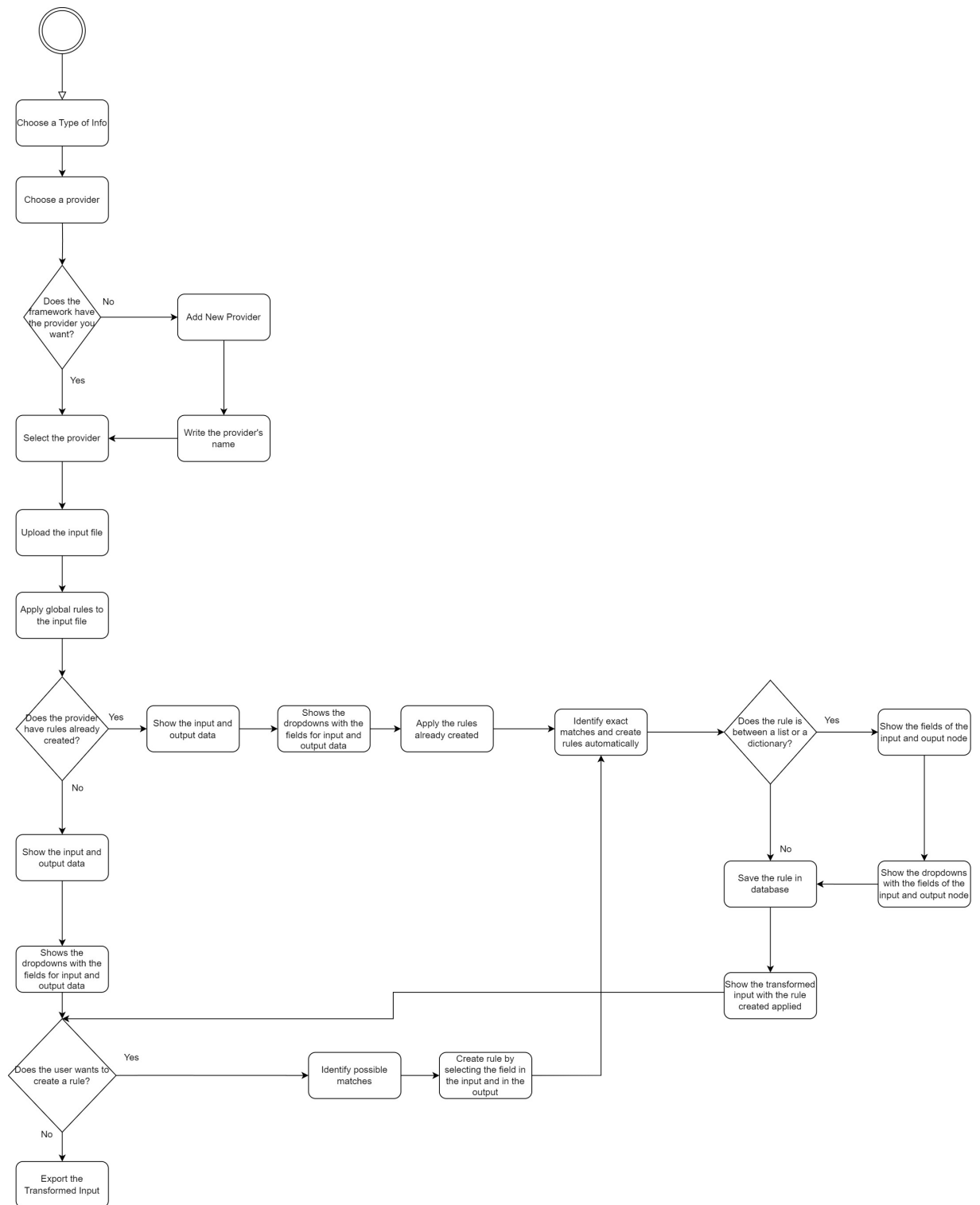


Figure 3.15: Execution Flow of the PlayField Framework

1. **Access the PlayField Interface:** The user opens the PlayField application built with Streamlit.
2. **Select Type of Info:** The user selects the type of data to be processed (e.g., player statistics, match results, team data) from the dropdown menu.
3. **Select Provider:** The user selects the data provider from the dropdown menu. If the provider is not listed, the user can add a new provider by selecting the “Add New Provider” option and entering the provider’s name.
4. **Upload Input File:** The user uploads the input file (CSV, JSON, XML, etc.) via the file uploader component. This file contains the raw data from the selected provider.
5. **Define Transformation Rules:** The user defines transformation rules through the interactive user interface. This involves mapping input fields to the corresponding output fields.
6. **Set Hierarchical Rules:** If the input data contains nested structures (e.g., arrays within dictionaries), the user defines hierarchical rules to ensure all nested data is correctly transformed.
7. **Automatic Rule Saving:** As the user defines rules, they are automatically saved in the PostgreSQL database. This ensures that all rules are recorded and can be retrieved or modified as needed.
8. **Initiate Data Transformation:** The backend processes the input data according to the defined rules. The system applies each rule to transform the raw data into a standardized format previously defined.
9. **Export Transformed Data:** Once the data transformation is complete, the user can export the transformed data. This can be done in two ways: **Download as JSON File:** The user clicks the "Export Transformed Input" button to download the transformed data as a JSON file or through the **API:** The user can use the *transform* endpoint to retrieve the transformed data programmatically.

Chapter 4

Validation

The validation of any framework is a crucial step to ensure its effectiveness, usability, and relevance to the end-users. This chapter presents a comprehensive validation process for the data transformation and mapping framework. To thoroughly test and validate the framework, we implemented a dual approach.

First, we conducted an experimental validation to rigorously test the framework’s functionalities and performance in a controlled environment. This initial phase allowed us to identify and address potential issues, ensuring the framework’s robustness and reliability.

Following the experimental validation, we conducted an extensive case study with zerozero.pt. This real-world application aimed to validate the framework’s usability and effectiveness in a practical, operational environment. By having zerozero.pt’s content managers and software developers test the framework and provide feedback through a detailed questionnaire, we gathered valuable insights on its performance under actual usage conditions.

4.1 Experimental Validation

To rigorously evaluate the robustness and adaptability of the framework, a comprehensive series of tests were conducted using data from multiple providers, including various iterations and updated versions of the same providers. This approach was designed to thoroughly assess the framework’s ability to accurately process and transform diverse input file formats and structures. By introducing data from both different sources and newer versions of existing sources, we aimed to observe and analyze how effectively the framework manages variations in data schemas, ensures consistency in transformation processes, and maintains overall data integrity across disparate datasets. This systematic testing methodology was crucial in validating the framework’s reliability and versatility in real-world applications.

4.1.1 Iterative Feedback and Framework Enhancements

Initially, the framework was designed to display the raw input and output files, presenting every field of both files in their entirety. Consequently, a corresponding text input was provided for the

input and output files. To create a transformation rule, users had to manually write the full path of the fields they wished to map. For instance, mapping the field *event.event_incident_detail.minute* from the input file to *data.events_0.minute* in the output file required the user to specify these paths explicitly. This approach, however, proved to be exceedingly complex and not user-friendly, particularly for the intended end users—non-technical content managers. The requirement to manually input field paths was both tedious and error-prone, making the framework impractical for those without technical expertise. Feedback gathered through a series of detailed meetings with the supervisors that highlighted these issues extensively. Supervisors pointed out that the process was overly complicated, prone to user error, and largely inaccessible for content managers who typically lack programming knowledge. These meetings provided critical insights into the practical challenges faced by users and underscored the need for a more intuitive solution. The feedback emphasized that, for the framework to be beneficial, it needed to be redesigned to cater to the capabilities and expectations of non-technical users. The initial complexity not only hindered efficiency but also discouraged adoption among the very users the framework was intended to help. In response to this constructive feedback, significant changes were made to the framework to enhance its usability and accessibility. We shifted from a raw field display and manual path entry system to a more intuitive, user-friendly interface that facilitates more straightforward mapping and rule creation. This involved the development of dropdown menus and visual aids to help users select and map fields without needing to know or input the full paths. This adjustment aimed to simplify the rule creation process, making it more manageable for non-technical users and significantly improving the overall user experience. By addressing these critical usability concerns, the framework became more aligned with the needs of its primary users, ensuring that content managers could effectively utilize the tool without extensive technical training. This transformation was pivotal in enhancing the framework’s practical utility and user satisfaction.

4.1.2 Testing with EnetPulse Files

To further validate the framework’s enhancements, we first employed EnetPulse¹ input file to create and refine transformation rules. The primary objective in this phase was to ensure that the output generated by the framework accurately reflected the desired target structure. We tested the framework’s initial rule-setting and transformation capabilities by mapping fields from the EnetPulse input file to the specified output fields.

During this phase, the framework was tasked with transforming the EnetPulse data to align with the predefined output schema. This process involved several key steps:

1. **Initial Rule Creation:** Using the EnetPulse input file, we manually defined transformation rules to map the input fields to the corresponding output fields. This included both simple and complex hierarchical mappings.

¹ A leading sports data provider (enetpulse.com/pt)

2. **Feedback and Iteration:** Based on user feedback, we identified areas of complexity and usability issues. Users found the manual rule creation process to be cumbersome and challenging, especially for non-technical content managers.
3. **Automation Enhancement:** To address these issues, we introduced an automation feature within the framework. This feature enabled the automatic creation of rules by scanning through each field of the input and output files. If fields had matching names, the framework would automatically create the corresponding rule.

The automation feature significantly streamlined the rule creation process, making it more efficient and user-friendly. By automating the identification and mapping of fields with identical names, we reduced the manual effort required and minimized the potential for errors.

4.1.3 Testing with Updated Versions

Once the automated rule creation feature was functioning effectively, we conducted further tests using newer versions of data from the same provider. Specifically, we tested with an updated version of the Monks², a fast, reliable, and cost-effective Sports Data Provider file. This allowed us to evaluate the framework's ability to adapt to changes in data schemas and maintain consistency in transformation processes.

The testing process involved the following steps:

1. **Creating New Rules:** We created new transformation rules for the updated version of the Monks file. Given that it was from the same provider, the framework automatically generated a significant number of rules by identifying fields with matching names.
2. **Validation of Automated Rules:** The framework's automation feature was rigorously tested with the updated Monks file. The system automatically identified and created rules for fields with matching names, ensuring that the new data version was accurately transformed into the desired output format.
3. **Analysis and Refinement:** We analyzed the transformed output to identify any discrepancies or issues. Feedback from this phase was used to further refine the automation feature, ensuring that it could handle a variety of data structures and updates effectively.

4.1.4 Enhancements Based on User Feedback

During the initial development phase, the framework only allowed users to create rules between fields at the same depth level. This limitation became apparent through informal feedback from potential users. They highlighted that such restrictions significantly hindered the flexibility needed for complex data transformations.

The following steps were taken to address this issue:

²<https://www.sportmonks.com>

1. **Identification of the Limitation:** Feedback from supervisors indicated that the initial framework design, which restricted rule creation to fields at the same depth level, was insufficient to meet the requirements of the framework. This was particularly problematic for non-technical users who needed to map fields across different hierarchical levels within the data.
2. **Implementation of Depth-Agnostic Rule Creation:** In response to the feedback, the framework was enhanced to support the creation of rules between fields at different depths. This was achieved by allowing users to select any field from the input file and map it to any field in the output file, regardless of their respective hierarchical levels.
3. **Dropdown for Invalid Rules:** To facilitate this feature, the framework now generates additional dropdown menus when an invalid rule is detected. These dropdowns appear next to the list or dictionary fields, enabling users to navigate through the hierarchical structure until a valid match is found. This iterative process continues until the user selects a valid corresponding field.
4. **Validation and Testing:** The enhanced framework was tested with various data sets to ensure that the new feature worked seamlessly. Users were able to create rules between fields at different depths, significantly improving the framework's usability and flexibility.
5. **Feedback and Refinement:** Continuous feedback was gathered from users to identify any remaining pain points. This iterative process of refinement ensured that the framework could handle a wide range of data transformation scenarios efficiently.

By incorporating user feedback and addressing the limitations of the initial design, the framework was significantly improved. The ability to create rules between fields at different depths not only enhanced its flexibility but also made it more accessible to non-technical users, thereby broadening its applicability and effectiveness in real-world data integration tasks.

4.2 Use Case

To assess the importance and usability of the framework in transforming input files to a common structure, we conducted an evaluation with real users at zerozero.pt, specifically targeting two groups: content managers and software developers. This rigorous validation process was designed to ensure the framework's effectiveness and usability in real-world scenarios, providing valuable insights and confirming its suitability for practical applications. For the validation purpose, we reached out to seven members of the content management department and six software developers from zerozero.pt to participate in this validation phase.

Initially, we provided a detailed explanation of the main concept of the framework. This introduction was crucial for ensuring that all participants had a clear understanding of the framework's objectives and functionalities.

4.2.1 Pre-Test Questionnaire

The pre-test questionnaire aimed to collect baseline information about the participants to understand the demographic and professional background of the users. This section was essential for contextualizing the feedback and understanding the diversity of the participant pool.

- **Language Selection:** Participants were given the option to select their preferred language for the questionnaire (Portuguese or English) to ensure clarity and comfort in responding.
- **Consent:** Participants had to agree to provide personal information and details about their background. This consent was crucial for ethical research practices.
- **Demographic Information:** Basic information such as gender and age was collected to categorize the participants.
- **Professional Background:** Participants were asked about their current role, years of experience, as illustrated in Figure 4.1, and their level of expertise in data manipulation, as observed in Figure 4.2. These details helped in understanding the user profiles and their relevance to the framework.

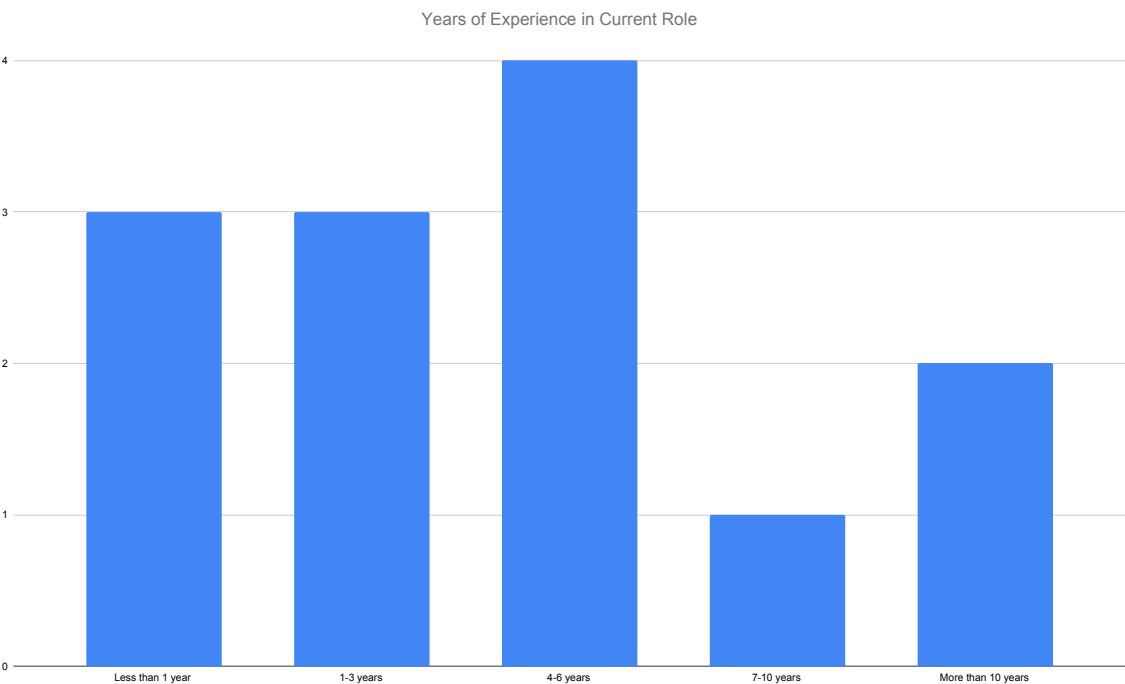


Figure 4.1: Years of Experience in the Current Role

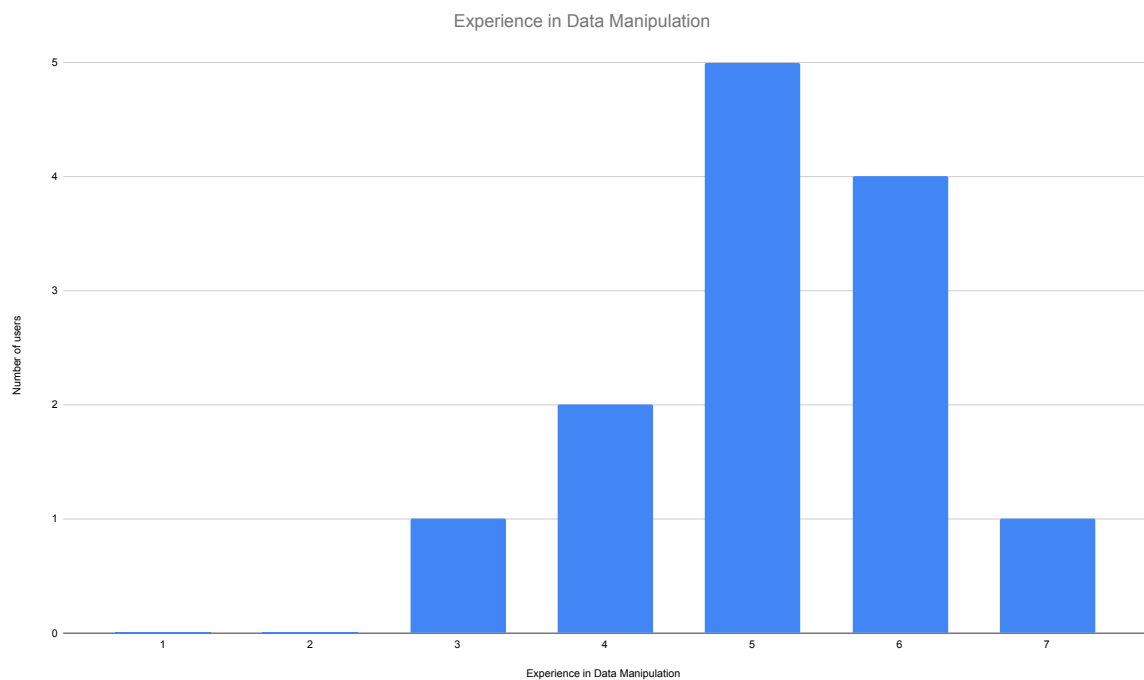


Figure 4.2: Experience of each user in Data Manipulation

4.2.2 Task Execution

This section focused on practical tasks that participants had to perform using the framework. The tasks were designed to evaluate the ease of use, intuitiveness, and effectiveness of the framework.

- **Task 1:** Participants were required to integrate and manipulate data from a specified provider using the framework. The objectives included ensuring proper mapping and transformation of data fields and refining the dataset by removing unnecessary fields.
- **Task 2:** After the initial data manipulation, participants were asked to automate the transformation process using the provided API. This task aimed to test the framework's ability to handle automated workflows and integration with other systems.

The tasks were designed to be completed within a total 20-minute timeframe, ensuring that they were challenging yet achievable. Participants were encouraged to solve the tasks independently to maintain an unbiased evaluation.

4.2.3 Post-Test Questionnaire

Following the task execution, the post-test questionnaire aimed to gather detailed feedback on the framework's usability and overall user experience.

- **Usability Evaluation:** Participants rated their agreement with various statements about the system's usability on a scale from 'Strongly Disagree' to 'Strongly Agree.' The statements

covered aspects such as the intuitiveness of the solution, the need for technical support, the complexity of the solution, and the learning curve.

- **Feedback on Specific Features:** Participants provided their subjective evaluations on whether the solution could accelerate the data collection process and whether they would recommend the solution to colleagues.
- **Additional Comments:** An open-ended section allowed participants to share any additional feedback or comments about their experience with the framework. This section was vital for capturing qualitative data that could provide deeper insights into user perceptions and potential areas for improvement.

The usability evaluation statements used were and the overall results are presented in Figure 4.3:

1. I think that the solution is intuitive.
2. I think that I would need the support of a technical person to be able to use this solution.
3. I think that I would like to use this solution in my day-to-day work.
4. I found the solution unnecessarily complex.
5. I would imagine that most people would learn to use this solution very quickly.
6. I think this solution is able to accelerate the process of data collection.
7. I think that the learning curve for this solution is smooth and easy to navigate.
8. I would recommend this solution to a colleague.

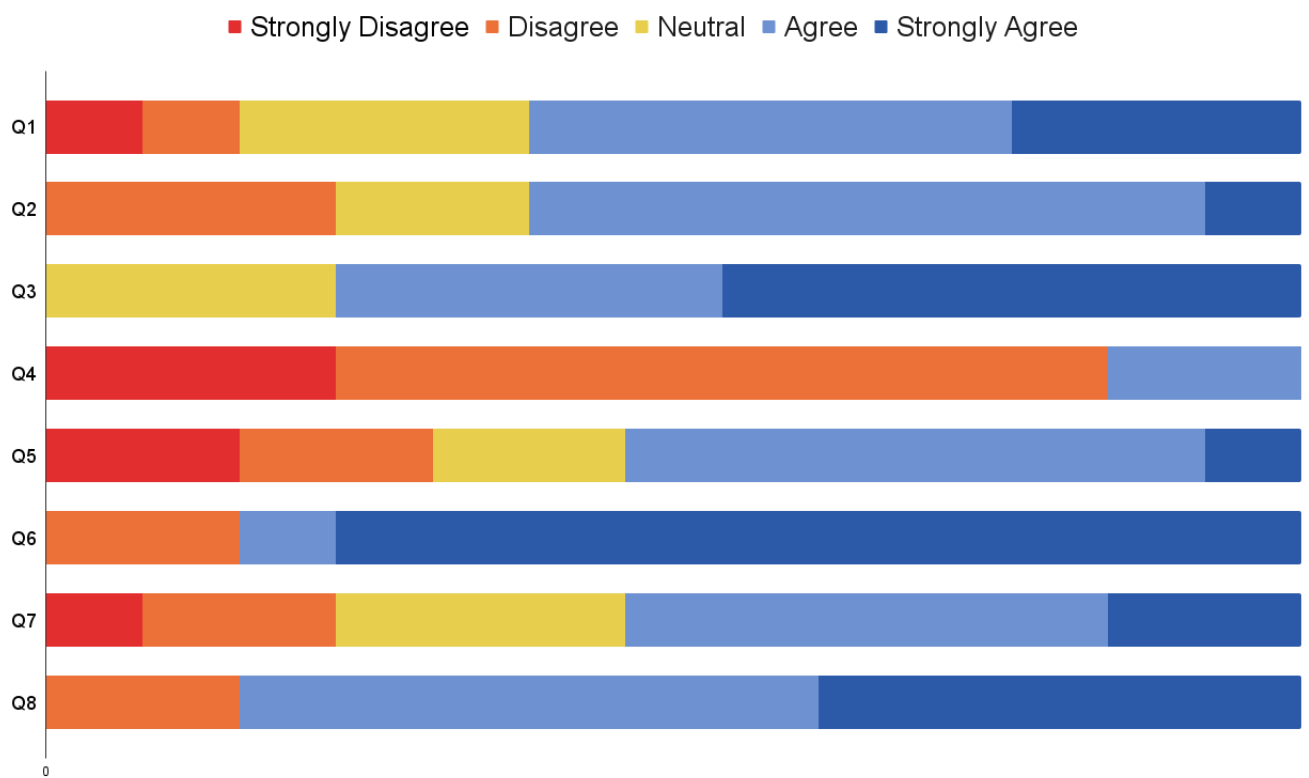


Figure 4.3: SUS results from the questionnaire

4.2.4 Detailed Analysis of Questionnaire Responses

The following section provide detailed insights into the responses for each question. This breakdown helps identify specific areas of strength and areas that may require further improvement within the framework.

For Question 1: A majority (61.5%) of the respondents agree or strongly agree that the solution is intuitive, suggesting that the framework is generally perceived as user-friendly. However, 23.1% are neutral, and 15.4% disagree/strongly disagree, indicating room for improvement in intuitiveness, as presented in Figure 4.4.

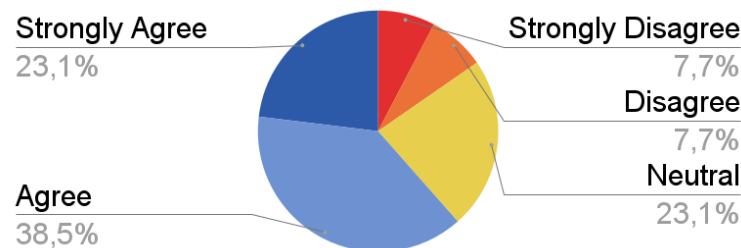


Figure 4.4: Question 1 Results

For Question 2: Half of the respondents (61.5%) agree or strongly agree that they would need technical support to use the solution, indicating that the framework may require a certain level of technical proficiency. Meanwhile, 15.4% are neutral, and 23.1% disagree, showing that some users feel confident using the framework without additional help, as observed in Figure 4.5.

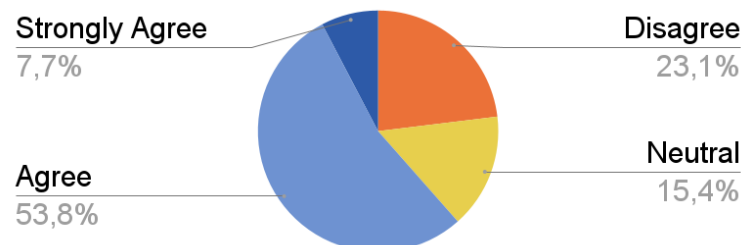


Figure 4.5: Question 2 Results

For Question 3: A significant majority (77%) of respondents agree or strongly agree that they would like to use the solution in their daily work, indicating a positive perception of the framework's utility. Only 23% are neutral, with no negative responses, suggesting overall acceptance, as demonstrated in Figure 4.6.

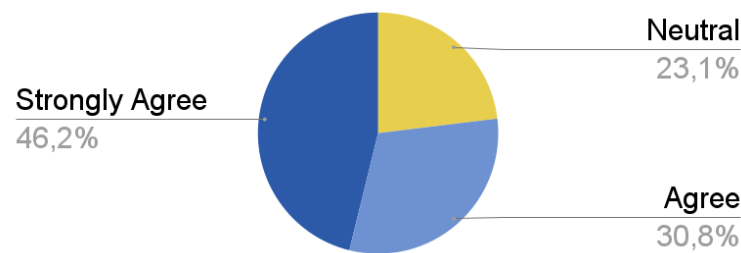


Figure 4.6: Question 3 Results

For Question 4: A significant majority (84.6%) of respondents disagree or strongly disagree that the solution is unnecessarily complex, indicating that the complexity level of the framework is generally acceptable. However, 15.4% agree, suggesting there is still some room to simplify certain aspects, as depicted in Figure 4.7.

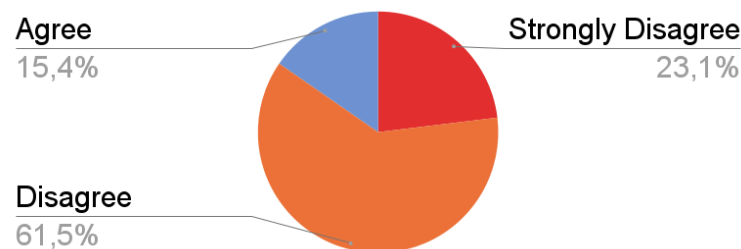


Figure 4.7: Question 4 Results

For Question 5: Nearly half of the respondents (53.8%) agree or strongly agree that most people would learn to use the solution quickly, while 15.4% are neutral and another 30.8% disagree/strongly disagree, as illustrated in Figure 4.8.

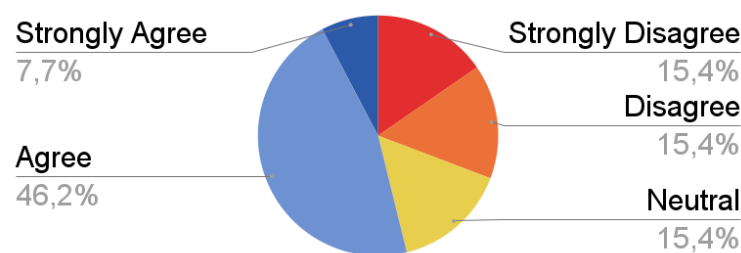


Figure 4.8: Question 5 Results

For Question 6: An overwhelming majority (76.9%) of respondents strongly agree that the solution accelerates data collection, with no negative responses, as observed in Figure 4.9. This indicates a strong consensus on its efficiency in this aspect, highlighting the framework's ability to significantly improve productivity.

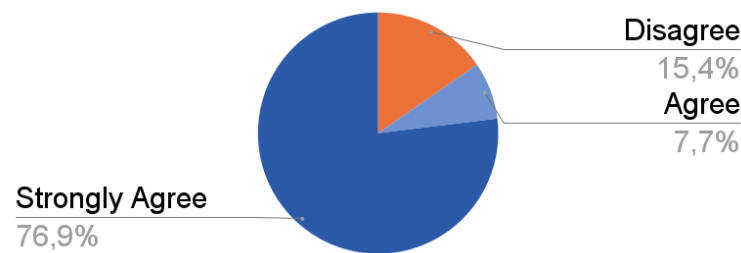


Figure 4.9: Question 6 Results

For Question 7: A notable portion (53.8%) of respondents agree or strongly agree that the learning curve is smooth, but 23.1% are neutral and another 23.1% disagree/strongly disagree, as outlined in Figure 4.10. This suggests that while many find it easy to learn, there is still a significant portion of users who might struggle initially.

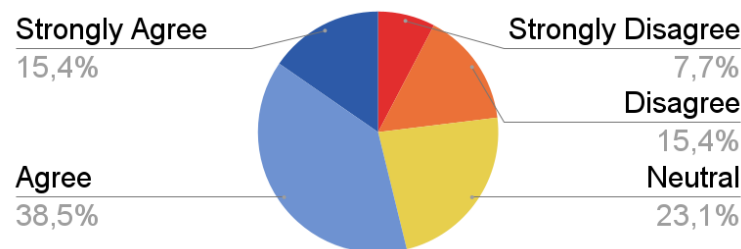


Figure 4.10: Question 7 Results

For Question 8: A majority (84.6%) of respondents agree or strongly agree that they would recommend the solution to a colleague, indicating a high level of satisfaction and endorsement among users. Only 15.4% disagree, reflecting overall positive feedback, as indicated in Figure 4.11.

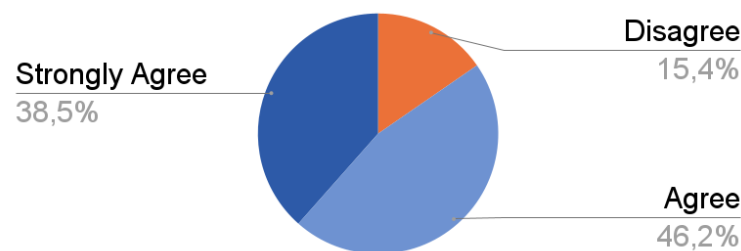


Figure 4.11: Question 8 Results

4.2.5 Open-Ended Question

In addition to the specific questions asked in the survey, an open-ended question was included to gather any additional feedback from users. This allowed users to provide more detailed and nuanced insights into their experiences and suggestions for improvement. Many users responded

to this open-ended question, providing critical feedback. 9 out of 13 participants offered their thoughts. This open-ended feedback was instrumental in identifying both strengths and areas needing enhancement, ensuring a comprehensive understanding of user needs and preferences.

Overall, the feedback indicates that the PlayField framework is effective, user-friendly, and valuable for its intended purpose. Users have recognized its potential to significantly enhance the efficiency and effectiveness of data transformation processes, particularly within the operational context of zerozero.pt. However, the validation process also highlighted several limitations and areas for improvement that need to be addressed to achieve even higher levels of user satisfaction and effectiveness.

One major strength of the PlayField framework is its ability to accelerate the data collection process. Feedback such as "the tool fits perfectly in terms of accelerating and facilitating processes with thousands of pieces of information" emphasizes that the tool effectively streamlines workflows, thereby improving productivity. Users found the framework to be an excellent starting point for facilitating data integration tasks, with comments like "the solution is a great start for a task that can be significantly simplified with the use of this project."

Another strength is the general intuitiveness of the framework, as some users appreciated its ease of use, highlighted by feedback stating, "I found the initiative interesting and, to a certain extent, intuitive." The framework's automated rule creation feature was particularly well-received, enhancing its efficiency and reducing the manual effort required from users.

However, the framework is not entirely suitable for standard non-technical workers. Feedback indicated that the interface and the need for substantial technical knowledge, such as understanding JSON structures, pose significant challenges for content managers. Comments like "Not for the standard non-technical worker!" underscore this issue, suggesting that the framework requires more user-friendly enhancements.

Users provided valuable suggestions for improving the interface to enhance user-friendliness. They recommended better visual indicators for mandatory fields, visibility of already mapped nodes with visual markers, and simplified navigation. While feedback such as "The solution is a bit complex regarding the interface and could be improved... It should be more user-friendly" highlights areas for enhancement, the framework's overall usability remains strong.

To address potential errors, such as unintentional rule matches due to miss-clicks, users suggested implementing confirmation steps for rule creation and deletion, and highlighting recent changes to the data. Feedback like "It would be safer if there was a confirmation button for a match rule" underscores these points without overshadowing the framework's effectiveness.

Additionally, users proposed adding search functionality and a dictionary of expressions to make the tool easier to navigate and understand, particularly for less technically proficient users. Suggestions such as "Search fields in various parts of the tool or even a 'dictionary' of expressions" aim to enhance usability further.

For large and complex JSON files, users suggested visually highlighting recent changes made by the last created rule to help understand the impact of transformations more clearly. Feedback

indicating, "... it is important if the last modifications made to the transformed input... were somehow highlighted," reflects this need.

Moreover, introducing a confirmation dialog for deleting rules and better error handling mechanisms would improve the overall user experience and prevent accidental data loss, as highlighted by feedback like "Add confirmation on Delete of rules."

The PlayField framework demonstrates considerable promise and utility, particularly in enhancing the efficiency of data collection and transformation processes. Addressing its limitations by enhancing user intuitiveness, simplifying complex aspects, and improving support resources will be crucial for maximizing its effectiveness and user satisfaction. By regularly engaging with users and iteratively improving the framework based on their feedback, PlayField can continue to be a valuable tool for data transformation and integration tasks.

In summary, this two-pronged validation strategy—combining experimental testing with real-world application—ensures a thorough and credible framework evaluation. This rigorous validation confirms that the framework is not only technically sound but also practical and beneficial for end-users in real-world scenarios.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The landscape of sports analytics has evolved significantly, transitioning from basic statistical analysis to a more comprehensive use of intricate data, including detailed metrics of player performance and extensive information from amateur leagues. This expansion has introduced increased complexity and diversity in data sources, necessitating advanced integration methods to maintain coherence and value in sports analytics. Traditional data integration methods often fall short in handling the volume, velocity, and variety of modern sports data, highlighting the need for a robust and adaptable ETL process tailored to these unique demands.

In response to the identified problems, a new framework was developed to address these challenges effectively. This framework is based on advanced data mapping techniques, ensuring accurate and efficient data transformation and integration. Despite its limitations, the framework is already being actively utilized at zerozero.pt, describing itself as the most extensive database of sports-related information in the world. This implementation at zerozero.pt demonstrates the practical impact and value of the framework to the company. The real-world application not only underscores the framework's relevance and effectiveness but also highlights its capability to address complex data integration challenges in a professional setting.

The developed framework was subjected to rigorous testing and validation at [zerozero](https://zerozero.pt), where the use case validation occurred. This validation process highlighted the framework's effectiveness and robustness in various data integration scenarios, demonstrating its ability to facilitate the integration of different data sources while improving the quality and consistency of the processed data. The results obtained from using this framework at [zerozero](https://zerozero.pt) were impressive, indicating its potential for application in production environments and various business areas.

5.2 Main Contributions

This thesis has made several key contributions to the field of data integration:

- **Validated Framework:** Development and rigorous validation of a robust data integration framework, demonstrating its effectiveness and practical applicability in real-world scenarios.
- **Publication:** Preparation and submission of a scientific article to a prestigious Data Engineering conference, enhancing the visibility and impact of the research.
- **Deepened Knowledge:** Significant advancements in the understanding of heterogeneous data mapping, addressing both theoretical and practical challenges.
- **Real-World Application:** Successful collaboration with zerozero.pt, showcasing the framework's ability to handle complex data integration tasks in a real-world scenario.

Additionally, the work carried out in this thesis provided a deeper understanding of emerging methodologies and technologies in the field of data integration. The lessons learned, and best practices documented throughout the framework development will serve as a foundation for future work and research in the area.

In summary, this thesis achieved its objectives by developing and validating an innovative framework for data integration, significantly contributing to the state of the art and promoting practical and theoretical advancements in the field. The contributions of this research will be valuable to academia and industry, facilitating the development of more efficient and effective solutions to data integration challenges.

5.3 Future Work

This thesis provides a solid foundation for further advancements in data integration. Future work should focus on enhancing the tool's capabilities by improving data mapping techniques through advanced machine learning algorithms. These algorithms can automatically discover relationships between fields in input and output files and generate mapping rules, significantly reducing manual effort.

By developing a machine learning model trained on user-defined mapping rules, the tool can learn and refine its accuracy over time. This model will enable the automatic creation of precise mapping rules for new datasets, streamlining the data mapping process and handling more complex integration tasks. Additionally, this will facilitate quicker integration of new data sources, enhancing the tool's adaptability and efficiency.

Furthermore, developing sophisticated data transformation algorithms and enhancing the user interface for easier workflow management will improve user experience. Continuous user feedback and collaborative research will be essential in refining and optimizing the tool to meet evolving data integration needs. This approach ensures that the tool remains valuable for academic research and industry applications, driving continuous improvement and innovation.

References

- [1] Apache hadoop. <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>. Accessed: 20-12-2023.
- [2] Big data explained: The 5v's of data. https://medium.com/@get_excelsior/big-data-explained-the-5v-s-of-data-ae80cbe8ded1. Accessed: 20-12-2023.
- [3] Hadoop architecture. <https://www.interviewbit.com/blog/hadoop-architecture/>. Accessed: 10-01-2024.
- [4] Hevo features. https://docs.hevodata.com/introduction/hevo-features/?utm_source=docs_sidebar. [Online; accessed 30-December-2023].
- [5] Thomas Adams, James Dullea, Peter Clark, Suryanarayana Sripada, and Thomas Barrett. Semantic integration of heterogeneous information sources using a knowledge-based system. In *Proc 5th Int Conf on CS and Informatics (CS&I'2000)*, 2000.
- [6] Md. Badiuzzaman Biplob, Galib Ahasan Sheraji, and Shahidul Islam Khan. Comparison of different extraction transformation and loading tools for data warehousing. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISSET)*, pages 262–267, 2018.
- [7] Galih Budianto. Data warehouse modeling using online analytical processing approach. *Jurnal Ilmiah Informatika Dan Ilmu Komputer (JIMA-ILKOM)*, 1(1):7–13, 2022.
- [8] Maxime Buron, François Goasdoué, Ioana Manolescu, and Marie-Laure Mugnier. Obi-wan: ontology-based rdf integration of heterogeneous data. *Proceedings of the VLDB Endowment*, 13(12):2933–2936, 2020.
- [9] Cambridge University Press. Data - cambridge dictionary. <https://dictionary.cambridge.org/dictionary/english/data>, Accessed 2023. [Online; accessed 20-December-2023].
- [10] Antonio Carzaniga, Gian Pietro Picco, and Giovanni Vigna. Is code still moving around? looking back at a decade of code mobility. In *29th International Conference on Software Engineering (ICSE'07 Companion)*, pages 9–20. IEEE, 2007.
- [11] Fábio André Castanheira Luís Coelho. *Towards a transactional and analytical data management system for Big Data*. PhD thesis, Universidade do Minho (Portugal), 2018.

- [12] Benoit Dageville, Thierry Cruanes, Marcin Zukowski, Vadim Antonov, Artin Avanes, Jon Bock, Jonathan Claybaugh, Daniel Engovatov, Martin Hentschel, Jiansheng Huang, et al. The snowflake elastic data warehouse. In *Proceedings of the 2016 International Conference on Management of Data*, pages 215–226, 2016.
- [13] Thomas H Davenport and Jeanne G Harris. *Competing on Analytics: The New Science of Winning*. Harvard Business Review Press, 2007.
- [14] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [15] Barry Devlin. *Data Warehouse: From Architecture to Implementation*. Addison-Wesley Professional, 1997.
- [16] Jean-Pierre Dijcks. Oracle: Big data for the enterprise. Technical report, Oracle Corporation, October 2011.
- [17] Sean Dougherty. Your guide to data mapping. *Funnel.io Blog*, 2022. Published: August 4, 2022, Last updated: April 2, 2024.
- [18] Paul DuBois. *MySQL*. Addison-Wesley, 2013.
- [19] Wayne W Eckerson. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. John Wiley & Sons, 2011.
- [20] Shaker H. Ali El-Sappagh, Abdeltawab M. Ahmed Hendawi, and Ali Hamed El Bastawissy. A proposed model for data warehouse etl processes. *Journal of King Saud University - Computer and Information Sciences*, 23(2):91–104, 2011.
- [21] I. Engelbrecht and H. Steyn. Does tdwg need an api design guideline? *Biodiversity Information Science and Standards*, 2021.
- [22] Christiane Fellbaum. *WordNet: An electronic lexical database*. MIT press, 1998.
- [23] Sérgio Fernandes and Jorge Bernardino. What is bigquery? In *Proceedings of the 19th International Database Engineering & Applications Symposium*, pages 202–203, 2015.
- [24] R. Fielding and R. Taylor. Principled design of the modern web architecture. *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, pages 407–416, 2000.
- [25] R. Fielding and R. Taylor. Principled design of the modern web architecture. *ACM Trans. Internet Techn.*, 2:115–150, 2002.
- [26] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [27] George P. Fletcher. The data mapping problem: Algorithmic and logical characterizations. 01 2005.
- [28] Giuseppe Fusco and Lerina Aversano. An approach for semantic integration of heterogeneous data sources. *PeerJ Computer Science*, 6:e254, 2020.
- [29] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144, 2015.

- [30] Nikola Gemes. What is data mapping in etl and how does it work? *Whatagraph Blog*, 2023. Published: February 24, 2023.
- [31] Farhad Soleimanian Gharehchopogh and Zeinab Abbasi Khalifelu. Analysis and evaluation of unstructured data: text mining versus natural language processing. In *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–4. IEEE, 2011.
- [32] Rick Greenwald, Robert Stackowiak, and Jonathan Stern. *Oracle essentials: Oracle database 12c*. " O'Reilly Media, Inc.", 2013.
- [33] Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, and Vidhya Srinivasan. Amazon redshift and the case for simpler data warehouses. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1917–1923, 2015.
- [34] Yoan Gutiérrez, José I. Abreu Salas, Andrés Montoyo, Rafael Muñoz, and Suilan Estévez-Velarde. Kd senso-merger: An architecture for semantic integration of heterogeneous data. *Engineering Applications of Artificial Intelligence*, 132:107854, 2024.
- [35] Shaikh Abdul Hannan. An overview on big data and hadoop. *International Journal of Computer Applications*, 154(10), 2016.
- [36] Informatica. What is data mapping? <https://www.informatica.com/resources/articles/data-mapping.html>, 2024. Accessed: 2024-05-31.
- [37] William H Inmon, Derek Strauss, and Genia Neushloss. *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Morgan Kaufmann, 2010.
- [38] Mohammad Khorasani, Mohamed Abdou, and Javier Hernández Fernández. Getting started with streamlit. In *Web Application Development with Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework*, pages 1–30. Springer, 2022.
- [39] Ralph Kimball and Joe Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley & Sons, 2004.
- [40] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons, 2013.
- [41] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246, 2002.
- [42] David Loshin. *Master Data Management*. Morgan Kaufmann, 2010.
- [43] Marek Macura. Integration of data from heterogeneous sources using etl technology. *Computer Science*, 15:109–132, 2014.
- [44] Neha Mangla and Priya Rathod. Unstructured data analysis and processing using big data tool-hive and machine learning algorithm linear regression. *Int. J. Comput. Eng. Technol.*, 9(2):61–73, 2018.

- [45] Max Mathijssen, Michiel Overeem, and S. Jansen. Source data for the focus area maturity model for api management. *ArXiv*, abs/2007.10611, 2020.
- [46] Mehdi Medjaoui, Erik Wilde, Ronnie Mitra, and Mike Amundsen. *Continuous API management*. " O'Reilly Media, Inc.", 2021.
- [47] Bruce Momjian. *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley New York, 2001.
- [48] Rajendrani Mukherjee and Pragma Kar. A comparative review of data warehousing etl tools with new trends and industry insight. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 943–948, 2017.
- [49] Jyoti Nandimath, Ekata Banerjee, Ankur Patil, Pratima Kakade, Saumitra Vaidya, and Divyansh Chaturvedi. Big data analysis using apache hadoop. In *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, pages 700–703, 2013.
- [50] Monika Patel and Dhiren B. Patel. Progressive growth of etl tools: A literature review of past to equip future. In Vijay Singh Rathore, Nilanjan Dey, Vincenzo Piuri, Rosalina Babo, Zdzislaw Polkowski, and João Manuel R. S. Tavares, editors, *Rising Threats in Expert Applications and Solutions*, pages 389–398, Singapore, 2021. Springer Singapore.
- [51] Dusan Petkovic. *Microsoft Sql Server 2008 A Beginner'S Guide*. McGraw-Hill Osborne Media, 2008.
- [52] Asma Qaiser, Muhamamd Umer Farooq, Syed Muhammad Nabeel Mustafa, and Nazia Abrar. Comparative analysis of etl tools in big data analytics. *Pakistan Journal of Engineering and Technology*, 6(1):7–12, 2023.
- [53] Thomas C. Redman. *Data Driven: Profiting from Your Most Important Business Asset*. Harvard Business Review Press, 2008.
- [54] Chantal Reynaud, J-P Sirot, and Dan Vodislav. Semantic integration of xml heterogeneous data sources. In *Proceedings 2001 International Database Engineering and Applications Symposium*, pages 199–208. IEEE, 2001.
- [55] Seref Sagiroglu and Duygu Sinanc. Big data: A review. *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, pages 42–47, 2013.
- [56] Salman Salloum, Ruslan Dautov, Xiaojun Chen, Patrick Xiaogang Peng, and Joshua Zhexue Huang. Big data analytics on apache spark. *International Journal of Data Science and Analytics*, 1(3):145–164, Nov 2016.
- [57] SAS Institute Inc. What is big data? https://www.sas.com/en_ae/insights/big-data/what-is-big-data.html, Accessed 2024. [Online; accessed 20-December-2023].
- [58] Wael Shehab, Sherin M ElGokhy, and ElSayed Sallam. Rohdip: resource oriented heterogeneous data integration platform. *International Journal of Advanced Computer Science and Applications*, 7(9), 2016.
- [59] Elizabeth M. Silvers. *The Data Asset: How Smart Companies Govern Their Data for Business Success*. John Wiley & Sons, 2012.

- [60] J. Simon. Apis, the glue under the hood. looking for the “api economy”. 23:489–508, 2021.
- [61] J Sreemathy, R Brindha, M Selva Nagalakshmi, N Suvekha, N Karthick Ragul, and M Praveennandha. Overview of etl tools and talend-data integration. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1650–1654, 2021.
- [62] J. Sreemathy, Infant Joseph V., S. Nisha, Chaaru Prabha I., and Gokula Priya R.M. Data integration in etl using talend. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1444–1448, 2020.
- [63] J Sreemathy, K Naveen Durai, E Lakshmi Priya, R Deebika, K Suganthi, and PT Aisshwarya. Data integration and etl: A theoretical perspective. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1655–1660, 2021.
- [64] J Sreemathy, K Naveen Durai, E Lakshmi Priya, R Deebika, K Suganthi, and PT Aisshwarya. Data integration and etl: A theoretical perspective. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1655–1660, 2021.
- [65] Geno Stefanov. Analysis of cloud based etl in the era of iot and big data. In *Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE)*, pages 198–202. International Conference on Application of Information and Communication . . . , 2019.
- [66] Christof Strauch, Ultra-Large Scale Sites, and Walter Kriha. Nosql databases. *Lecture Notes, Stuttgart Media University*, 20(24):79, 2011.
- [67] Qingzhao Tan, Prasenjit Mitra, and C. Lee Giles. Metadata extraction and indexing for map search in web documents. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, page 1367–1368, New York, NY, USA, 2008. Association for Computing Machinery.
- [68] Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. Conceptual modeling for etl processes. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, pages 14–21. ACM, 2002.
- [69] Deepak Vohra. *Apache Avro*, pages 303–323. Apress, Berkeley, CA, 2016.
- [70] Paul Westerman and Paul Westerman. *Data Warehousing: Using the Wal-Mart Model*. Morgan Kaufmann, 2001.
- [71] Wayne Yaddow. The process of data mapping for data integration projects. 10 2019.

Appendix A

PlayField Validation

A.1 Validation Questionnaire

PlayField - User Experience

* Indica uma pergunta obrigatória

1. [PT] Selecione a linguagem em que quer responder ao formulário *

[EN] Select the language in which you want to respond to the form

Marcar apenas uma oval.

- ☐ Português Pular para a pergunta 14
- ☐ Inglês Pular para a pergunta 2

English Questionnaire

Welcome to the PlayField - User Experience

The primary aim of this form is to gather your opinion on a tool designed for centralizing multiple sports feeds of sports information in a structured manner.

In the initial sections of the form, we collect personal information along with details about your background to characterize the types of people who will use the tool.

In this study, all the data collected will remain anonymous. This data may be utilized for presentations at conferences, academic events, similar gatherings, and scientific publications.

By proceeding, you voluntarily agree to participate in this questionnaire and consent to provide the necessary information. You understand that your responses will be used for research and improvement purposes. You are aware that your participation is entirely voluntary, and you can withdraw at any point without providing a reason.

2. Do you consent to provide personal information and details about your background to characterize the types of people who will use the tool for the purpose of this study? *

Marque todas que se aplicam.

☐ I agree with the terms.

3. Are you willing to participate in the study, understanding that your responses and interactions with the tool will be recorded and analyzed for research purposes? *

Marque todas que se aplicam.

☐ I agree with the terms.

4. Do you agree to share your subjective evaluation of the system's usability using the System Usability Scale (SUS), acknowledging that your feedback will be used for research and improvement purposes? *

Marque todas que se aplicam.

☐ I agree with the terms.

[1] Pre-Test Questionnaire

In this questionnaire, we will only collect basic information about you for statistical purposes only. All information you provide will remain anonymous and none of it will be sold or shared with third-parties.

5. Gender *

Marcar apenas uma oval.

- ☐ Male
- ☐ Female
- ☐ Rather not say

6. Age *

7. Role *

Marcar apenas uma oval.

- ☐ Content Manager
- ☐ Software Developer
- ☐ Outro: _____

8. Years of Experience in Current Role *

Marcar apenas uma oval.

- ☐ Less than 1 year
- ☐ 1-3 years
- ☐ 4-6 years
- ☐ 7-10 years
- ☐ More than 10 years

9. How would you classify your experience and/or knowledge in current role in the field of **data manipulation**? *

Marcar apenas uma oval.

1 2 3 4 5 6 7

I have ☐ ☐ ☐ ☐ ☐ ☐ ☐ I have extensive knowledge and experience on the topic.

[2] Tasks (20 minute limit)

(wait in this section for further instructions from the Administrator)

During the execution of these tasks, you are asked to:

- During the session, familiarize yourself with the objectives and functionalities of the solution.
- Complete the solution tasks within the allocated 20-minute timeframe.

Important Reminders

- Try to interpret and solve the tasks on your own. If you feel stuck or lost, the administrator can provide assistance, but you should avoid it whenever possible.
- A "guide" on how to interpret the solution will not be provided to maintain an unbiased evaluation of the prototype's ease of use and effectiveness.
- Tasks are intentionally designed to be unbiased, ensuring fairness in evaluating different versions of the solution.

Task 1

This task involves interacting with the PlayField framework to integrate and manipulate data from a specified provider. The aim is to seamlessly integrate the provider's data into the framework, ensure proper mapping and transformation of data fields, and ultimately refine the dataset by removing unnecessary fields and creating specific rules to align the data structure. This process ensures the data is correctly processed and formatted for further use or analysis.

Please do the Task 1 present in the file

Task 2

After completing Task 1, where rules were created and fields were removed, the framework provides an API to automate this transformation. The aim is to verify the availability and functionality of the data provider through the API, test the data transformation process programmatically, and automate the transformation of the inputs. By doing so, we ensure that the system can handle requests for data transformation correctly and return the expected results, facilitating automated workflows and integration with other systems.

Please do the Tasks 2 present in the file

[3] Post-Test Questionnaire

Your inputs from this questionnaire will help to understand how well the solution meets your expectations and how user-friendly it is. Please approach this test with an open mind, providing honest and constructive feedback regarding the usability of the solution.

10. Please rate your agreement with the following statements regarding the system usability on a scale from 'Strongly Disagree' to 'Strongly Agree'.

*

Marcar apenas uma oval por linha.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I think that the solution is intuitive	☐	☐	☐	☐	☐
I think that I would need the support of a technical person to be able to use this solution	☐	☐	☐	☐	☐
I think that I would like to use this solution in my day-to-day work	☐	☐	☐	☐	☐
I found the solution unnecessarily complex	☐	☐	☐	☐	☐
I would imagine that most people would learn to use this solution very quickly	☐	☐	☐	☐	☐
I think this solution is able to accelerate the process of data collection	☐	☐	☐	☐	☐
I think that the learning curve for this solution is smooth and easy to navigate	☐	☐	☐	☐	☐
I would recommend this solution to a colleague	☐	☐	☐	☐	☐

11. Do you have any additional comments or feedback about the solution? Feel free to share it with us, any feedback is greatly appreciated!

12. Do you want to be contacted if need it? *

Marcar apenas uma oval.

- ☐ Yes *Pular para a pergunta 13*
- ☐ No

Email

13. Email *

Questionário Português

Bem-vindo ao PlayField - User Experience

O **objetivo** principal deste formulário é recolher a sua opinião sobre uma ferramenta desenvolvida para centralizar múltiplos feeds desportivos de informações de desporto de forma estruturada.

Nas seções iniciais do formulário, recolhemos informações pessoais, juntamente com detalhes sobre o seu histórico, para caracterizar os tipos de pessoas que irão utilizar a ferramenta.

Neste estudo, todos os dados recolhidos permanecerão anónimos. Esses dados poderão ser utilizados em apresentações em conferências, eventos académicos, encontros similares e publicações científicas.

Ao prosseguir, concorda voluntariamente em participar neste questionário e consente em fornecer as informações necessárias. Compreende que as suas respostas serão utilizadas para fins de pesquisa e melhoria. Está ciente de que a sua participação é inteiramente voluntária e que pode retirar-se a qualquer momento sem fornecer um motivo.

14. **Consente em fornecer informações pessoais e detalhes sobre o seu ackground para caracterizar os tipos de pessoas que utilizarão a ferramenta para fins deste estudo?** *

Marcar apenas uma oval.

- ☐ Concordo com os termos.

15. **Está disposto a participar do estudo, compreendendo que as suas respostas e interações com a framework serão gravadas e analisadas para fins de pesquisa?** *

Marcar apenas uma oval.

- ☐ Concordo com os termos.

16. **Concorda em partilhar a sua avaliação subjetiva da usabilidade do sistema baseado na System Usability Scale (SUS), reconhecendo que o seu feedback será utilizado para fins de pesquisa e melhoria?** *

Marcar apenas uma oval.

- ☐ Concordo com os termos.

[1] Pre-Test Questionário

Neste questionário, apenas recolheremos informações básicas sobre si para fins estatísticos. Todas as informações fornecidas permanecerão anónimas e nenhuma delas será vendida ou partilhada com terceiros.

17. Género *

Marcar apenas uma oval.

- ☐ Masculino
- ☐ Feminino
- ☐ Prefiro não dizer

18. Idade *

19. Função *

Marcar apenas uma oval.

- ☐ Gestor de Conteúdos
- ☐ Desenvolvedor de Software
- ☐ Outro: _____

20. Anos de Experiência *

Marcar apenas uma oval.

- ☐ Menos de 1 ano
- ☐ 1-3 anos
- ☐ 4-6 anos
- ☐ 7-10 anos
- ☐ Mais de 10 anos

21. Como classificaria a sua experiência e/ou conhecimento na função atual no campo da **manipulação de dados**? *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Não ☐ ☐ ☐ ☐ ☐ ☐ ☐ Tenho bom conhecimento e experiência sobre o tema

[2] Tasks (limite de 20 minutos)

(Aguarde nesta seção por mais instruções do Administrador)

Durante a execução destas tarefas, você deve:

- Familiarizar-se com os objetivos e funcionalidades da solução durante a sessão.
- Completar as tarefas da solução dentro do prazo de 20 minutos.

Notas Importantes:

- Tente interpretar e resolver as tarefas por conta própria. Se você se sentir perdido ou bloqueado, o administrador pode fornecer assistência, mas deve ser evitado sempre que possível.
- Um "guia" sobre como interpretar a solução não será fornecido para manter uma avaliação imparcial da facilidade de uso e da eficácia da solução.
- As tarefas são projetadas intencionalmente para serem imparciais, garantindo a justiça na avaliação de diferentes versões da solução.

Tarefa 1

Esta tarefa envolve a interação com a framework PlayField para integrar e manipular dados de um fornecedor específicos. O objetivo é integrar os dados do fornecedor na framework, garantir o mapeamento adequado e a transformação dos campos e, por fim, refinar o conjunto de dados removendo campos desnecessários e criando regras específicas para alinhar a estrutura dos dados. Este processo garante que os dados sejam processados e formatados corretamente para uso ou análise posterior.

Por favor, execute a Tarefa 1 presente no ficheiro fornecido

Tarefa 2

Após completar a Tarefa 1, onde foram criadas regras e eliminados campos, a framework providencia uma API para automatizar esta transformação. O objetivo é verificar a disponibilidade e funcionalidade do fornecedor de dados através da API, testar o processo de transformação de dados programaticamente e automatizar a transformação dos ficheiros de input. Ao fazer isso, garantimos que o sistema pode lidar corretamente com solicitações de transformação de dados e retornar os resultados esperados, facilitando fluxos de trabalho automatizados e a integração com outros sistemas.

Por favor, execute as Tarefas 2 presentes no ficheiro fornecido

[3] Questionário Pós-Teste

As suas respostas a este questionário ajudarão a entender quão bem a solução atende às expectativas e quão intuitiva é para o usuário. Por favor, aborde este questionário com mente aberta, fornecendo feedback honesto e construtivo sobre a usabilidade da solução.

22. Por favor, avalie a sua concordância com as seguintes afirmações sobre a usabilidade do sistema numa escala * de 'Discordo Totalmente' a 'Concordo Totalmente'.

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
Acho que a solução é intuitiva	☐	☐	☐	☐	☐
Acho que precisaria do apoio de uma pessoa técnica para conseguir usar esta solução	☐	☐	☐	☐	☐
Acho que gostaria de usar esta solução no meu trabalho do dia-a-dia	☐	☐	☐	☐	☐
Achei a solução desnecessariamente complexa	☐	☐	☐	☐	☐
Imagino que a maioria das pessoas aprenderia a usar esta solução muito rapidamente	☐	☐	☐	☐	☐
Acho que esta solução é capaz de acelerar o processo de coleção de dados	☐	☐	☐	☐	☐
Acho que a curva de aprendizagem para esta solução é suave e fácil de percorrer	☐	☐	☐	☐	☐
Recomendaria esta solução a um colega	☐	☐	☐	☐	☐

23. Tem algum comentário ou feedback adicional sobre a solução? Sinta-se à vontade para o compartilhar, qualquer feedback é muito importante!

24. Gostaria de ser contactado caso seja necessário? *

Marcar apenas uma oval.

☐ Sim Pular para a pergunta 13

☐ Não

Tasks

Task 1

1. **Open the framework:** Go to the [framework](#)
2. **Select a type of info:** Select "Match Feed Info"
3. **Select the name of the provider:** If **Enetpulse** is available, select it. If not, choose "Add New Provider," give it the name **Enetpulse**, and submit.
4. **Upload the input file:** Upload an Enetpulse input file, accordingly, with the provider
5. **Explore the input and output file:** Explore the data from the input and output file. The nodes, the values...etc
6. **Create the first rule:** Create a rule between **event** and **data**
7. **Remove some fields:** Remove the field **data.id**, **season_id**, **stage_id** and **aggregate_id**
8. **Observe the changes in the output**
9. **Create another rule:** Create a rule between **event_incident** and **events**
10. **Create another rule:** Create a rule between **elapsed** and **minute**
11. **Create another rule:** Create a rule between **elapsed_plus** and **extra_minute**
12. **Create another rule:** Create a rule between **comment** and **info**
13. **Observe the changes in the output**
14. **Delete the rule:** Delete the rule between **elapsed_plus** and **extra_minute**
15. **Create another rule:** Create a rule between **event.variable.event_incident.event_incident_detail.participant_id** and **data.events.participant_id**
16. **Export the output file**

Task 2

1. **Use API:** Go to the [API docs](#)
2. **Try to use GET /providers endpoint**
3. **Test the POST /transform endpoint:** In the body, insert the name of the provider, and in input_json, the input file of the accordingly provider

Tarefas

Tarefa 1

1. **Abra a ferramenta:** Abra a [framework](#)
2. **Selecione um tipo de informação:** Selecione "Match Feed Info"
3. **Selecione o nome do fornecedor:** Se **Enetpulse** estiver disponível, selecione-o. Caso contrário, escolha "Add New Provider," dê-lhe o nome **Enetpulse**, e submeta.
4. **Dê upload ao ficheiro de input:** Upload um ficheiro de input da Enetpulse, de acordo com o fornecedor.
5. **Explore o ficheiro de input e de output:** Explore os dados do ficheiro de input e output. Os nós, valores... etc
6. **Crie a primeira regra:** Crie uma regra entre **event** e **data**
7. **Remova alguns campos:** Remova os campos **data.id**, **season_id**, **stage_id** e **aggregate_id**
8. **Observe as alterações no output**
9. **Crie outra regra:** Crie uma regra entre **event_incident** e **events**
10. **Crie outra regra:** Crie uma regra entre **elapsed** e **minute**
11. **Crie outra regra:** Crie uma regra entre **elapsed_plus** e **extra_minute**
12. **Crie outra regra:** Crie uma regra entre **comment** e **info**
13. **Observe as alterações no output**
14. **Elimine a regra:** Elimine a regra entre **elapsed_plus** e **extra_minute**
15. **Crie outra regra:** Crie uma regra entre **event.variable.event_incident.event_incident_detail.participant_id** e **data.events.participant_id**
16. **Exporte o ficheiro de saída**

Tarefa 2

1. **Use a API:** Vá para [API docs](#)
2. **Tente usar o endpoint GET /providers**
3. **Teste o endpoint POST /transform:** No body, insira o nome do fornecedor e, em input_json o ficheiro de input do fornecedor correspondente