March - 2023

Master

Data Analytics for Business

Master's Final Work

Internship Report

Big Data Framework Implemented in Cloud Azure

Sofia Pêga

ISEG Lisbon School of Economics & Management
Universidade de Lisboa

Master
Data Analytics for Business

Master's Final Work
Internship Report

Big Data Framework Implemented in Cloud Azure

Sofia Alegre Fernandes Pêga

SUPERVISION:
Prof. Dr. Jesualdo Cerqueira Fernandes
Dr. Flávio Romão

March - 2023

GLOSSARY

AMQP - Advanced Message Queuing Protocol
DDL - Data Definition Language
ETL - Extract, Transform, and Load
FTP - File Transfer Protocol
IAM – Identity and Access Management
IDE - integrated development environment
JMS - Java Message Service
MQTT - Message Queuing Telemetry Transport
RAM - Random Access Memory
SQL - Structured Query Language
STOMP - Simple/Streaming Text Oriented Message Protocol
TCP/IP - Transmission Control Protocol/Internet Protocol

ABSTRACT

As the technology boom of the last decades has led to a much higher availability of data, companies can leverage it to make better business decisions. There are tools and techniques available to work with large amounts of data but this report focuses on studying one particular tool: Microsoft Azure. With it, a framework for processing big data was implemented with the objective to explore the tools of one of the most popular cloud services and develop the most effective architecture maintaining cost-effectiveness and the restraints of the project.

The digital ecosystem of Microsoft Azure is extremely extensive and complex, so only the relevant concepts and tools were explored, namely Databricks, Synapse Analytics, Data Factory, WebJobs and Storage. During this process a greater understanding of Microsoft Azure elements was gained, both for their applications and limits.

After the exploration phase, the Architecture defined was implemented that included the entire of the big data processing lifecycle and, following the entry of the framework into production, multiple possible improvements were found that can be either implemented or further researched.

Keywords: Microsoft Azure, Big Data Framework, Data Lake

TABLE OF CONTENTS

TABLE OF FIGURES

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Jesualdo Fernandes, for his guidance and support.

Thank you to Link Redglue, Flávio Romão and all my team for this opportunity.

Finally, I am also grateful to all my family and friends, especially Rosanna Muller for all their encouragem

INTRODUCTION

The current technologic landscape has led to an explosion of data generated and collected by organizations. This data is arriving faster, in more diverse formats and in much bigger sizes than before, hence the name of "big data" (Cukier & Mayer-Schoenberger, 2013). This has fundamentally changed many organizations and created a new, growing business area to directly deal with this new world.

On the surface, it may appear as a clear advantage, but only if the data is effectively leveraged to become knowledge, otherwise it can cause damage to the state of an organization (Manyika et al., 2011). The reason for this is that valuable information can be understood incorrectly, and incorrect assumptions can be treated as truth, leading to mistakes in business decisions.

To avoid the threat of mishandled data, companies must implement frameworks to organize the data, which is a theorical structure that serves as a guideline to support the implementation of the architecture necessary to develop the project. In this case, framework is the theoretical concept, and the architecture are the tools used

Naturally, companies' goals, available resources, and the type of data they manage all influence the frameworks they use. As a result, a new architecture needs to be created whenever a new project is started. In this case, the goal is to build a big data process platform that will be developed on the cloud, most specifically, using Microsoft Azure services.

The cloud is a collection of software and servers accessible only over the internet, regardless of location, to offer applications and processing power to any user with internet access (Hugos & Hulitzky, 2010). The cloud has already been the choice of many companies (Manjoo, 2016), and one of the biggest cloud service providers on the market is Microsoft Azure (Reno, 2022).

This project will then offer an insight into some of Azure's resources, by developing a detailed data processing framework with Azure resources, when possible. As this is a complex process, with many concepts, the report focused on the tools and characteristics and behavior of the tools and not specifically on the code that will be written for the data transformation.

To do this, firstly a brief explanation of the concepts surrounding big data and processing will be given, as well as of the tools used to create the architecture. Finally, the description of how each of those tools were used in the architecture and how it may be improved postproduction.

1.1. The Company

In the current situation of the market, organizations have access to their data and the tools to leverage the data, but many are missing the individuals who possess the expertise and skills to effectively use the available resources. As there is a need in the market, there are those who fill it, in this case, organizations specialized in using those tools to harness the data's potential ("How Data Analytics Are Changing the Consulting Industry," 2018).

One such company is Link Redglue, whose core mission is to create technological solutions and products that enable organizations to derive genuine value from all dimensions of their data resources, shown in Figure 1. It focuses on three expertise areas, which are, data foundations, advanced analytics, and artificial intelligence. The basis of all the work is the data foundations since the company believes that continuous value can only be achieved with solid technical and financial foundations.

Only then can the other areas be implemented. advanced analytics, which is real time analytics to equip decision-makers to make data-driven decisions, and artificial intelligence, where Link Redglue focuses on empowering a company's best asset: its people, and elevate human intelligence with the tools available.

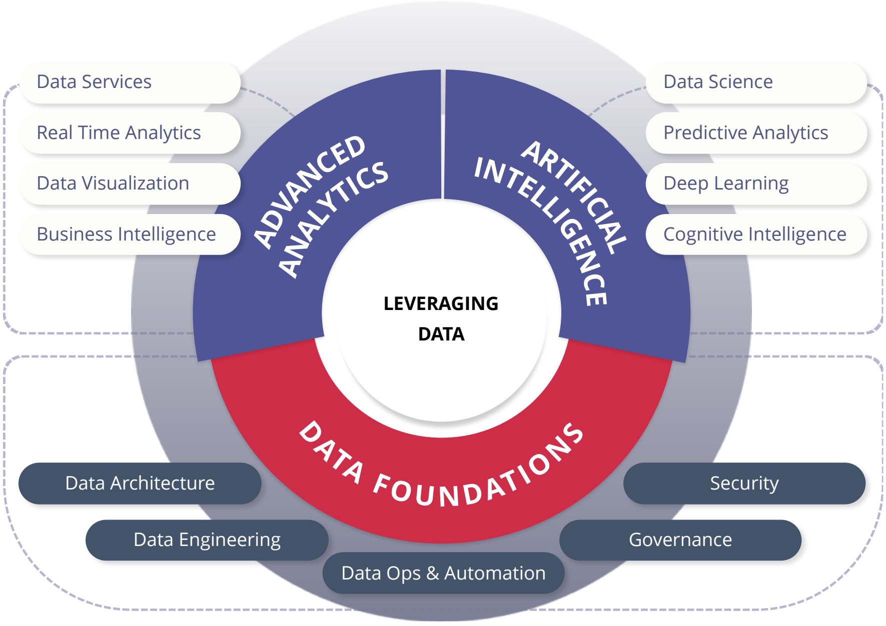


Figure 1 - Levels of Leveraging Data

Source: https://linkredglue.com/leveraging-data/

1.2. The Team

The team for this project was comprised of four professionals, one of them being the project manager, who is necessary to successfully manage project risks, find a balance between technical and managerial responsibilities, and handle interpersonal interactions within the project organization (Wilemon & Cicero, 1970).

Then, the Senior Consultant, a highly skilled team member, makes sure that the architecture is planned to meet the project's goals and satisfy the client's requirements. Additionally, the senior consultant advises the two junior data engineer analysts on how to implement the framework in the most effective manner possible, ensuring that the project is successfully completed, in the most efficient and correct manner.

1.3. Objectives

Microsoft Azure can be overwhelming to a new user trying to understand the plethora of services it provides, for example, Azure offers four different types of storage accounts, (Microsoft, 2023l) and seven different types of storage services (Microsoft, 2023i), each one designed for slightly different uses and data objects.

This is the case with many of their products, and although it means that Azure offers so many solutions for varied problems, different tools accomplishing similar tasks can lead to confusion and less efficient choices. Nevertheless, the benefits that can be derived from the use of Azure tools greatly outweigh the challenges it brings, if used correctly.

By designing an infrastructure for Big Data analytics and processing using Microsoft Azure resources this report intends to allow for a better understanding of some of the relevant elements of the cloud provider.

2. LITERATURE REVIEW

The field of Big Data is relatively new, the term "Big Data" only gained momentum at the beginning of the century, due to this, there are many concepts that are not agreed upon by the scientific community. However, the impact and importance of understanding it and all of its elements are universally recognized. (Mayer-Schonberger & Cukier, 2013)

2.1. Big Data

Even though the concept of big data is challenging to narrow down, many describe it as data sets of a large size, that cannot be processed using traditional tools (Ohlhorst, 2012). To better characterize this concept, there are four commonly accepted dimensions of Big Data: Volume, represented by a large amount of data; Variety, because there are many more sources of data, in many formats, both structured and unstructured; Velocity, as data to be created in real-time, faster than ever and Value, which is the certainty that data has enough quality to derive knowledge from it (Iafrate, 2015).

Some authors also include veracity, making it 5 V's, which is "an indication of data integrity and the extent to which it can be trusted for analytical and decision-making purposes" (McNeely, 2015, p. 3).

Due to the technological and conceptual characteristics of this field, professionals need very specific skills, and in this area, there are three types of specialized professionals: data analysts, data engineers, and data scientists.

As with most concepts in this area, not all studies agree on the competencies of the data professionals but in general they are: data analysts, or business analysts, who focus on analyzing data and extracting insights from it, and data scientists, who use statistical and machine learning techniques to create models that can forecast future outcomes (Handfield, 2022). In contrast, data engineers are responsible for designing and maintaining the infrastructure that enables data to be processed (Mason, 2018).

2.2. Big Data Framework

The development of a big data framework is the construction of a structure to support the extraction of information from data. It can be split into three phases: data consumption, processing, and analysis or interpretation (Muan Sang et al., 2020).

The first phase must resolve the question of how to get the data into the necessary location. Even before working on any of it, data engineers must make sure they have access to the right data, on time. These two characteristics are necessary because if the data is not correct it will lead to misleading conclusions, or its value can be obsolete if it takes too long to be consumed.

Then, the data processing can be started, the first step is cleaning it to deal with errors or format incompatibilities and after this is concluded, the following transformations depend on the objectives of the project and working environment. This second phase is the most critical, requiring not only a thorough assessment of the available data to understand it but also the necessary business knowledge to carry out this phase. Any mistakes will negatively impact the output and may result in inaccurate data.

Finally, to get the knowledge and value to justify the cost and time of wrangling data, analysts must use visual or mathematical tools to present the information in a way that it can be used and understood by the final users. This is when information, which is the simple facts, is turned into knowledge, which is the understanding of a subject.

The design should not be done for a one-time operation, one of the main characteristics of big data is it is produced in a continuous and rapid manner. therefore, this process needs to be built to assure that it is mostly automatic.

2.3. Cloud

Cloud is an omnipresent reality in enterprise culture, the O'Reilly 2021 report on the use of the cloud by organizations found that 90% of respondents use the Cloud in some form, with the three cloud industry tycoons being Amazon Web Services, Microsoft Azure and Google Cloud, in this order (Loukides, 2021). This migration to the cloud is due to the many advantages it brings to organizations that are necessary to survive in the highly competitive technological landscape.

Azure defines the cloud as an ample collection of remote servers that "are designed to either store and manage data, run applications, or deliver content or a service" (Microsoft, 2023p) as opposed to physical servers or locally installed software. Although, some literature differentiates the terms "cloud" and "cloud computing", assigning the previous definition to the latter term, and the cloud is the entire internet (Rittinghouse & Ransome, 2009). In this work, both terms are used interchangeably.

The services provided in the cloud fall into three different categories Software-as-a-service (SaaS), Plataform-as-a-service (PaaS), and Infrastructure-as-a-Service (IaaS).

IaaS is the most basic type of cloud computing. Only the data center environment is provided, which is the ability to store and process data in the cloud. PaaS is the environment where developers can create applications, and SaaS provides the application, which are hosted in the cloud (Hugos & Hulitzky, 2010).

The main appeals of the cloud are the immediate access to the tools a project requires, without the need for costly and time-consuming investment in physical servers and the scalability, meaning businesses can augment and decrease the resources it consumes, with only the cost of what it uses (Sharma, 2018). Additionally, the cloud allows for an improvement in cooperation between all members of the company because users are given access to the same resources regardless of location or operating system.

Naturally, there are some limitations that organizations must be aware of: security and trust are significant worries (Indu et al., 2018) By design, data is kept in servers that are controlled by the cloud provider, not the company, and the only way to access it is through the cloud, therefore, cybersecurity is an important investment for cloud providers.

Organizations must also accept a loss of control in the infrastructure of their framework. Even though most cloud services are very flexible, there are always features that will be defined by the provider which must be accepted, for example VM sizes are limited according to region and it’s not possible to create custom VM sizes (Microsoft, 2023m).

2.4. Microsoft Azure

Microsoft Azure, the second largest cloud service in the market (Reno, 2022), is a platform with a collection of cloud services that provide an off-premises infrastructure to store, manage and leverage data created by Microsoft. It has characteristics similar to its competitors, namely, it is a pay-as-you-go service that offers a myriad of cloud applications, but its uniqueness makes it the best choice for certain situations.

The types of services Azure offers are cloud services from all different models: IaaS, PaaS, and SaaS. All these resources are created to help businesses to overcome their challenges through cooperation, using the latest cloud technology.

One of the characteristics that differentiate Microsoft Azure is the brand that created it: Microsoft, whose products dominate many markets. For example, Office 365, a collection of Microsoft’s most used tools, like Word and Excel. According to the company, in the second quarter of 2022, they had 56 million (Microsoft, 2023g) non-commercial subscribers for Office 365, likewise, in another domain, 90% of .NET developers surveyed use Microsoft Visual Studio as their tool of choice (Stack Overflow, 2020).

Many Microsoft Azure resources can integrate perfectly with other Microsoft products, notably, Azure offers complete integration with Microsoft 365 (Microsoft, 2023c). Consequently, companies that already use a large amount of Microsoft products may benefit from choosing the same brand Cloud infrastructure.

Microsoft Azure is also at the forefront of cloud security, a crucial functionality for most companies. It offers IAM (Identity & Access Management) which "guarantees security of identities and attributes of cloud users by ensuring that the right persons are allowed in the cloud systems" (Indu et al., 2018, p. 575). This implies that administrators give each user access to only what they need and logs of who uses each resource.

Furthermore, Microsoft has an interest in open-source projects (Campbell, 2020), and although not all Azure technologies are open-source, a big amount of its software and resources offers are. For example, Apache Spark, the language used normally in the Azure Databricks environment is an open-source language. This detail can be beneficial for a company to not be dependent on Azure, as it can be more easily migrated to other providers or on-premises infrastructure. These open-source tools also are inclined to be favored by developers because they have a bigger community working on them that provides documentation and explanations.

2.5. Data Lake Architecture

The data lake is a data repository allows to keep all data received as all raw data is saved, both in structured, unstructured, or semi-structured formats. In big data projects, this is essential because it makes it possible to derive insight from past and current data and do exploratory analysis when necessary.

Due to its nature, a strong and logical architecture is necessary to keep the value of the data, otherwise, data can become hard to find and understand. As a result, data lakes must have a logical structure. This structure and the names given to its elements can vary slightly but, in general, data is kept in three different layers, depending on its characteristics and processing stage: Raw, Processed, and Curated.

The raw or staging layer is where data lands after it is ingested, it has no or minimal transformations (Gorelik & Safari, 2019). Afterward, the processing or enriched layer, as the name explains "contains cleansed, enriched, and otherwise processed versions of the raw data" (Gorelik & Safari, 2019, p. 136). Finally, data is consumed from the Curated layer, where it is optimized for analysis and exploration.

2.6. Scrum

Often an overlooked element of a project, the management of the tasks and personnel is imperative to the success of any project. This stage consists in the use of organizational skills and business knowledge to organize the workload, balancing the time, financial and human constraints in order to deliver the agreed goals (Mantel et al., 2011).

There are many management frameworks that must be chosen in accordance with the requisites of the work. Scrum is a project management methodology that opposes traditional frameworks by utilizing observable facts and cooperation to make decisions. In general, teams iteratively observe the state of their project to examine what needs to be completed and what has been accomplished and together they decide on how to proceed (International Scrum Institute, 2019).

More specifically, the project is split into sprints of the same length and on the first day of each sprint, a Sprint Planning takes place, in which the team and the product owner who "represents the voice of the customer" (Pries & Quigley, 2010, p. 52) define tasks and assign to them a priority and a duration estimation.

In addition to this a daily meeting is meant to update the team in what is moved to the to or from the sprint backlog (the tasks that have not or have been completed), what tasks are active and what problems have been found.

3. TOOLS AND TECHNOLOGIES

The architecture explained in further sections makes use of a variety of tools, the majority of which can be used for many different objectives. In this next section they are explained, although, as some have varied functionalities and are highly customizable, only the relevant features will be detailed.

3.1. Azure Storage

In order to gather long-lasting knowledge from the process of wrangling data, it is necessary to retain it, this is the base of any framework: the data storage. Azure offers four types of storage accounts: Standard general-purpose v2, the most commonly used, and three types of Premium accounts which are designed for high-performance and high-transition applications (Microsoft, 2023l).

There are six Azure storage services, but due to the objective of this project, only the Blob storage will be discussed because it is optimized for large amounts of unstructured data (Microsoft, 2023j) and is the base of Azure Data Lake Storage Gen2 which is “set of capabilities dedicated to big data analytics, built on Azure Blob Storage” (Microsoft, 2023b).

According to Azure, the Data Lake Storage Gen2 improves on the Blob storage in areas of performance, management, and security. One of the improvements is the organization of files in a Hierarchical Namespace. This means that directories are organized similarly to a file computer system with folders (named containers) and subfolders (directories). Additionally, due to being derived from Blob storage, it also has the same benefits of scalability, since it allows for automatic augmenting of storage space and cost efficacy (Microsoft Azure, 2023).

3.2. Webjob

Another one of the resources that Microsoft Azure offers is the App Services: a Platform as a Service (PaaS) “for hosting web applications, REST APIs, and mobile back ends” (Microsoft, 2023a) in a variety of programming languages.

A WebJob is a component of an App Service that can run a script either continuously or when it is triggered (Microsoft, 2022b). It supports python and Java files, but most importantly, it is possible to develop a .NET Core console app in Microsoft’s Visual Studio and deploy it as a WebJob (Microsoft, 2022a), making development easier by allowing users to write their code with an IDE (integrated development environment) they are familiar with, where they can test their code before Implementing it.

The chosen software framework is .NET, which is made by Microsoft to be an open-source tool for the creation of multiple types of applications (Microsoft, 2023o), these apps can be written in C#(C sharp), F#(F sharp) and Visual Basic (Microsoft, 2023k). It has been ranked "as the #1 most-loved framework on the Stack Overflow Developer Survey for three years in a row (2019, 2020, and 2021)" (Microsoft, 2023q) because it is fast and it offers common and standard-use libraries, APIs and 300 000+ expansive packages to help development with the .NuGet package manager (nuget, 2023).

3.3. Apache Spark

Introduced to the public in 2014, Apache spark is an open-source big data processing framework that quickly became one of the most popular processing engines due to its speed and capacity to handle large amounts of data.

There are many reasons it is so often chosen as the engine of choice for data processing, firstly it "revolves around the concept of a resilient distributed dataset (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel" (Apache Software Fundation, n.d.), this results in faster processing times.

Also, "Spark supports in-memory computing, that enables it to query data much faster compared to disk-based engines such as Hadoop" (Ghaffar et al., 2015, p. 7). In-memory processing refers to a computing technique where data is loaded and processed entirely in the computer's RAM (Random Access Memory) and avoids "moving data from memory to CPU" (Ghose et al., 2019, p. 3:1), which can be time intensive.

Spark provides a wide range of high-level APIs in different programming languages, including Scala, Java, Python, and R (Apache Software Fundation, n.d.), allowing for users to benefit from cluster computing with standard programming languages. Overall, Apache Spark is a powerful and flexible tool for performing big data processing tasks that can scale to handle massive datasets and complex analytical workloads.

3.4. Databricks

Databricks is a platform based on Apache Spark for processing big data (Ghodsi, 2017), it is an open-source data wrangling engine that integrates multiple languages to allow faster, big data processing on computer clusters.

Databricks is composed of five main elements: the workspace, “that functions as an environment for your team to access Databricks assets” (Databricks, 2023) where the team collaborates in writing code, since it is where the notebooks are located, then, clusters a “set of computation resources and configurations” (Microsoft, 2023e), that are necessary to run code in Azure Databricks, and behave similarly to a virtual environment. Jobs, which are area automated tasks that can be configured on a cluster, and finally, Libraries, that are installed on a cluster and installed every time they are booted up.

Data Scientists and engineers can create notebooks in Databricks to connect to their storage account and afterwards process and transform data stored in Azure Storage, with Spark or other common tools such as SQL or Python.

3.5. Data Factory

The Azure Data Factory allows the user to create workflows to automate and organize big data processing and transformation (Microsoft, 2023q). These workflows are built with an intuitive drag-and-drop interface and defined as pipelines, which are “logical grouping of activities that performs a unit of work” (Microsoft, 2023h) that can extract, transform, and load data from various sources into destinations for analysis.

It can ingest data from a wide range of data sources, including some that are exterior to Microsoft Azure, it also has monitoring and alerting capabilities to help track and manage pipelines with ease.

3.6. Azure Synapse Analytics

Azure Synapse Analytics is a cloud-based data warehousing and analytics platform, it incorporates SQL and Spark technologies and allows integration with other Azure services. Significantly, files stored in delta tables in the data lake can be explored in Synapse SQL with no barrier between the two different technologies. Additionally, Synapse allows some of the same actions as Azure Data Factory, such as data ingestion and automated ETL processes (Microsoft, 2022d).

Users analyze data in different types of pools, dedicated or serverless, the latter, according to Microsoft “is a distributed data processing system, built for large-scale data and computational functions” (Microsoft, 2022c) it is automatically scaled and the user only pays for the resources they actually use, on the other hand Microsoft describes a dedicated SQL poll has a “analytics service that brings together enterprise data warehousing and Big Data analytics" (Microsoft, 2023n) whose size must be scaled manually.

3.7. Azure Devops Services

Azure DevOps is a cloud platform that offers a range of tools for software development and project management, it can be divided into DevOps Server and Services, the first is an on-premises application and the second, which will be used in this project, "provides a SaaS-based offering to manage the end-to-end DevOps life cycle" (K K, 2020, p. 10).

Azure DevOps Service has five main components: Boards, a project management tool, Repos (repositories) that allow teams to store and manage their code, Test Plans, a tool to track manual and automated tests, Artifacts, for library packets management and Pipelines which are to automate builds, tests and deployments to various platforms. Many of these components also integrates with other tools and services such as Visual Studio, Azure and GitHub (K K, 2020).

3.8. Data Origins

In this project, data will originate from two locations, a message broker, of the type ActiveMQ Artemis, created by the Apache foundation, and from an FTP server. ActiveMQ Artemis is an open-source message broker, written in Java, that is designed for the transaction of messages between different systems where the sender and the consumer are disconnected (Apache Software Foundation, n.d.).

The messages are sent to a topic or queue, then the consumer subscribes to it using a messaging protocol (a predefined way to exchange data between applications). In layman's terms, the message broker is the courier which receives the messages, and the messaging protocol is the way to open the door to reach those messages. The protocols currently supported by Artemis are JMS, STOMP, MQTT, and AMQP, the project will use Advanced Message Queuing Protocol (AMQP).

File Transfer Protocol (FTP) is the protocol to copy files from on host to another location, it is based on TCP/IP which are communication protocols between different servers or computers. FTP uses two different connections for data transfer and for control commands, making it more efficient than other transfer protocols (Forouzan & Fegan, 2002).

3.10 Power Bi

Power BI is a self-service business analytics service developed by Microsoft that allows users to connect to a multitude of data sources and create dashboards and reports to gather insight from their data. These reports can then be shared and access through the cloud.

4. DETAIL OF WORK

This big data framework was developed with the Microsoft Azure environment, taking advantage of the seamless integration between the various tools it offers to complete the process and to allow the team to work in cooperation and simultaneously. The project included the creation of a persistent architecture to gather and process data that will be stored in the tables of a relational model for end users to explore.

To start the process, there are initial actions that are standard for any project of this type, for instance, there were two environments, the production environment, and the development environment, the latter is where all the framework was designed and built, and the production was simply a copy of it: a duplicate of all the development resources.

Any debugging, improvement or other changes were first be tested in development, and when approved, they were be implemented in production. This report will describe the development environment as the creation of the production environment is simply a repetition of the same actions.

4.1. Project Planning And Management

Throughout the project a team needs a way to organize itself and its daily objectives, no matter the skill of its members. Therefore, a method of management is necessary, in this project, it is Scrum, “an iterative software engineering process to develop and deliver software” (International Scrum Institute, 2019, p. 13).

In this process, tasks were defined by and assigned by the Scrum team, in cooperation with each other the team divided the workload into smaller tasks, organizing them by priority. The project was divided into sprints, in this case, 6 sprints of two weeks as shown in figure 2, and in each new Sprint, tasks were assigned and defined, the previous Sprint was discussed, and the finished tasks were added to the Backlog. Each day, in the Daily Scrum, the team met to discuss advancements and problems in the development of the project.

Azure DevOps was the tool used to help with this job, specifically, the Boards help manage the state of the tasks, because of its visual, drag-and-drop interface for creating tasks, and with it, the team characterized the tasks as backlog, active, or resolved. These statuses change along the development of the project when new information is given during the daily meetings.

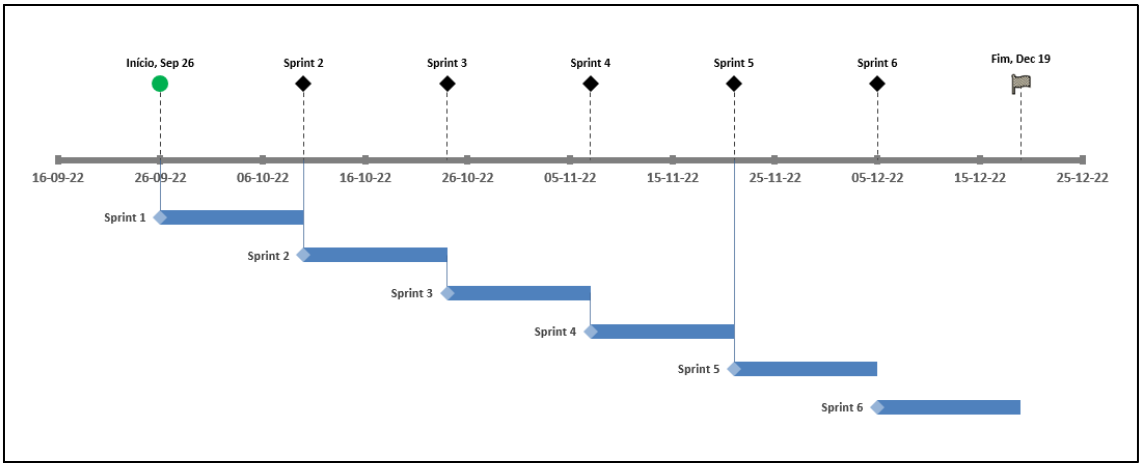


Figure 2 - Sprint Calendar

4.2. Architecture

As the provider of choice was Microsoft Azure, all the tools that can be chosen were Azure resources, when possible, or Microsoft products as it made integration between tools easier, as shown in the image below. Azure provides various resources, many of them with similar purposes, so, the choices on what to use in this specific project were made based on how simple the tools are to use and how well they apply to the specificity of the immutable elements of the project.

The data was gathered through an FTP server and a message broker, because these tools have different characteristics, they were ingested with different resources. Data factory already has a functionality to connect to FTP servers, and the tool was used further along the project, and, the fewer resources are used, the lower the cost can be and the simpler the architecture is. Azure has no direct connector to ArtemisMQ and it must therefore be done with a code that is continuously ingesting data, therefore, a WebJob was the choice to run the script.

All data was stored in a Data Lake Gen2, both in its raw and curated forms because, as mentioned before, it is a storage type specifically designed for big data analytics and it's a cost-effective solution allowing enterprises to store large amounts of data.

Treating, cleaning, and processing the data was done in Databricks, it is an organized way to process big data, the notebooks can be easily connected with the storage account, and it is optimized for spark programming, which is faster and gives many options of resources to program with. These notebooks were run periodically with pipelines created in Data Factory. Finally, to give access to end-users to the data a synapse analytics serverless pool is used to connect to applications outside of Azure, like Power BI, for visual data analysis. Figure 3 shows what was explained previously in a simplified manner.

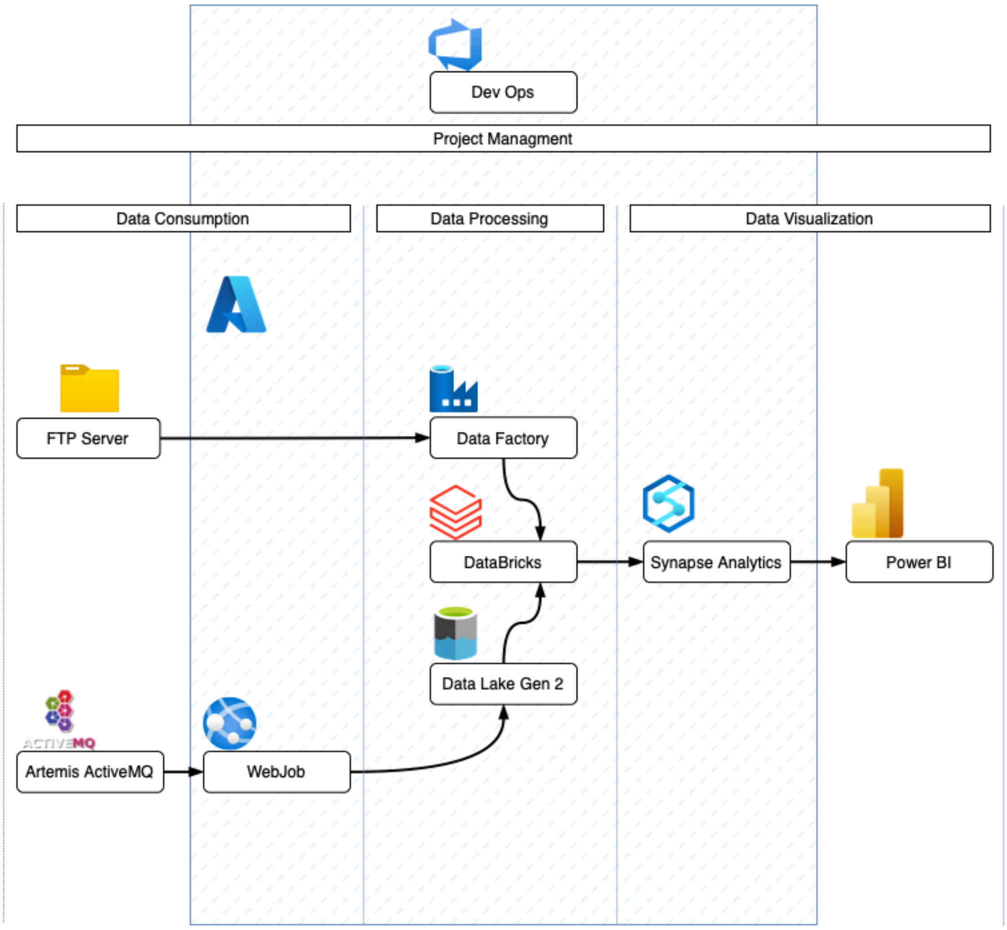


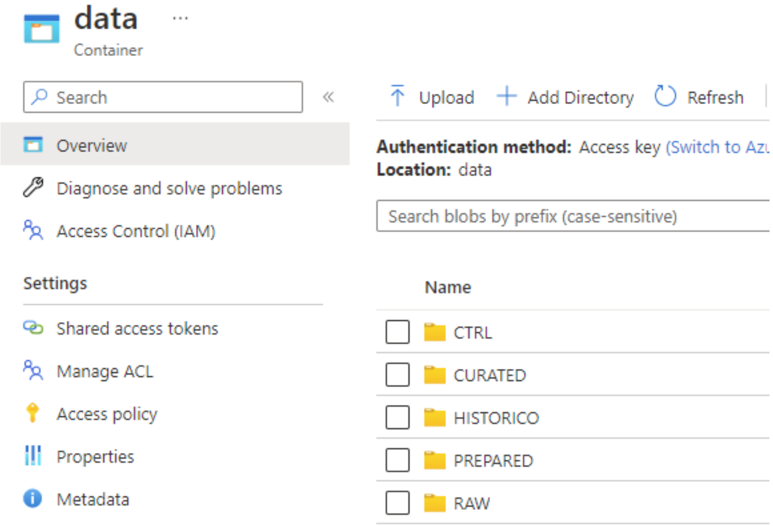
Figure 3 - Framework Architecture

4.3. Setting up the environment

All software offered in Azure is called resources, and it must be organized under “resource groups”, it can be compared to a general folder for a project on a computer, where all the instances of each application used were kept, so, all sources in the development environment were stored in the same resource group.

The Resources for each environment were stored under the same resource group, therefore, there were only two, one for development and another for production, as mentioned before they are copies of each other, unless a change is being tested in development.

After the resource group was created, the second most important is the storage account, which was set up as a Data Lake Gen2. This account only has one container for all data and this container data is distributed in five directories pictured in Figure 4: RAW, PROCESSED, CURATED, HISTORICO and CTRL. The first three have been previously explained, the HISTORICO is where the data gather before the start of the project is kept, and the CTRL is has the information for controlling pipeline processing.



Authentication method: Access key (Switch to Azu
Location: data

	Name
□	CTRL
□	CURATED
□	HISTORICO
□	PREPARED
□	RAW

Figure 4 - Storage Account

Then, another resource of the type Azure App Service was created and it is where the Web Job is hosted. Then, the Databricks and Data Factory were created, both need elements to be set up, the Databricks cluster and linked services in Data Factory, to access the FTP, server which will be explained in other sections. Finally, in Azure Synapse a serverless SQL pool was created.

4.4. Data consumption

In this project there are two sources of data, Apache ActiveMQ Artemis and an FTP server, this was defined by the client, and therefore cannot be replaced. The velocity of data coming from Apache Artemis is much higher than the one in the FTP server.

The Artemis connection was designed to be continuously sending near real-time data to storage and the FTP holds historical data and other types that are updated less frequently. Due to their different nature, the data in each of these mechanisms was moved to storage in different ways, to adapt to its peculiarities.

4.5. Connection to Artemis

As the messages coming from the broker arrive constantly, the first challenge was to send them to the RAW folder, for this, the process is more complicated than the connection to the FTP server.

The connection was made with a .NET script written in Visual Studio, the choice of IDE (integrated development environment) is due to the compatibility of Visual Studio and Microsoft Azure. The .NET framework is justified by the existence of client libraries both for connecting to azure storage and the message broker.

At first, the script connects to azure storage with the NuGet Azure storage client package (Microsoft, 2023d), then, it must connect to Apache Artemis with the AMQP protocol, also with a NuGet client. At this moment, the application is receiving the messages, and every five minutes, they are sent to the designated storage folder.

4.6. Copying from the FTP Server

As stated before, the FTP server keeps historical data and data that is updated daily or yearly. Both types of data are copied using Data Factory, with different pipelines.

As mentioned before, data factory has a functionality to directly connect to an FTP Server. Before creating the pipeline, a new linked service is created which is a connection between azure and an external resource, the server (Microsoft, 2023f), And with that, it is possible to create pipelines to copy the necessary data into specific folders defined by the user, as it can be shown in the image bellow.

To briefly explain figure 5, in this case, the pipeline reads the files inside a specific folder, then it reads the metadata to know when they were added and iterates through them to only select the ones whose date is greater than a variable with the timestamp with the last time the pipeline was run. Finally, a new value is set for that variable with the new timestamp. This is triggered yearly and daily.

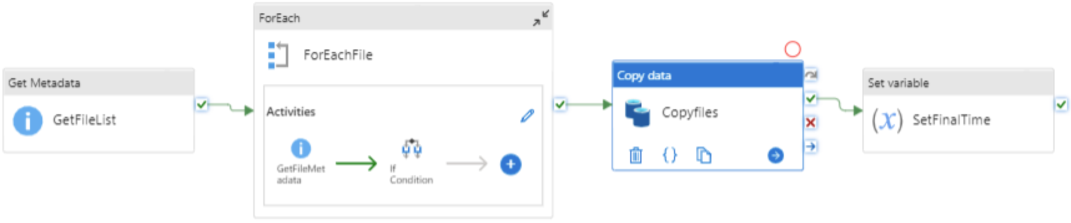


Figure 5 - Pipeline to copy historic data from FTP server

The pipeline for copying historic data is much simpler, as it is only a copy from folder task and triggered manually since it is done only once.

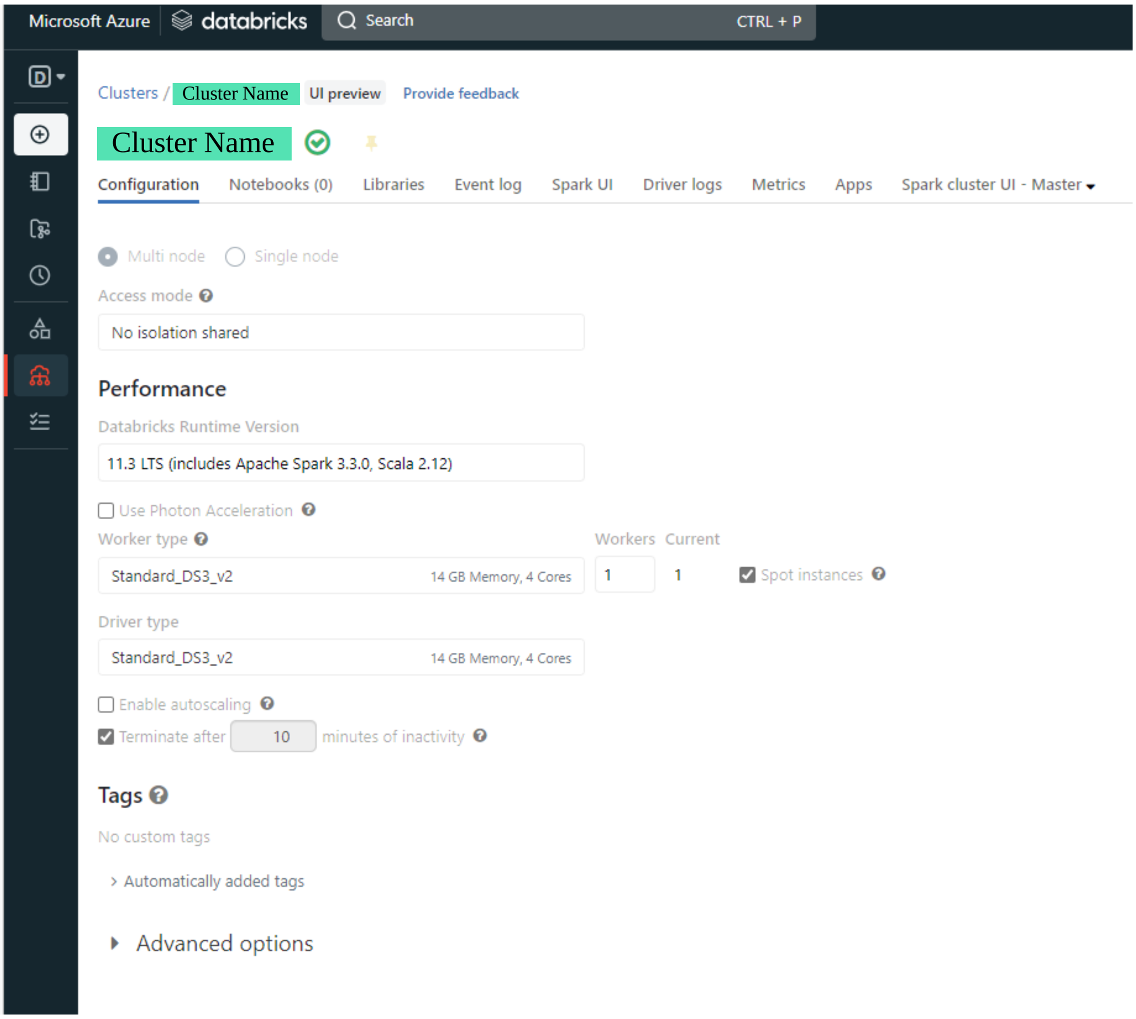
5. DATA PROCESSING

5.1. Setting up the Cluster

When raw data is entering into storage, it is finally possible to transform it into information. This is by far, the most complex part of the process, with many steps, but, before any line of code is written, the characteristics of the cluster must be defined, these are dependent on the details of the project.

When creating a cluster, it already has default values for every configuration, in the case of this project a few details are important to change, access mode must allow all users with access to this node to be able to use the cluster as this is a team project and many worked in the same environment.

The runtime version was kept at the latest available at the time of creation, unless it is a beta version. The worker type is standard, since it is not necessary an extremely big processing capacity, these details were subject to change in the duration of the development of the project, since only during that time it is possible to see if the configuration fits the necessities of the project. In the case of this project that was not necessary.



Cluster Name

● Multi node ○ Single node

Access mode: No isolation shared

Performance

Databricks Runtime Version: 11.3 LTS (includes Apache Spark 3.3.0, Scala 2.12)

□ Use Photon Acceleration

Worker type		Workers	Current	
Standard_DS3_v2	14 GB Memory, 4 Cores	1	1	☑ Spot instances

Driver type	
Standard_DS3_v2	14 GB Memory, 4 Cores

□ Enable autoscaling

☑ Terminate after 10 minutes of inactivity

Tags

No custom tags

› Automatically added tags

▸ Advanced options

Figure 6 - Cluster characteristics

After the cluster was deployed, with the characteristics shown in figure 6, all necessary libraries were installed directly in the cluster, this is similar to a programming environment, so the libraries will be installed when the cluster in booted up and all notebooks can use them.

5.2. Databricks Notebooks

Before writing in the notebooks, organization is a requisite to build a valuable framework, so the notebooks were distributed into 8 sections pictured in figure 7:

In the COMMON_HEADER, notebooks are called at before all other notebooks, because its where a library of common functions and variables are defined and its where the connection with the storage account is done so that it is possible to read, move and delete files and data in the data lake.

DDL is where the tables and defined are defined, and then created, these notebooks are only run once. There are subfolders is this case for the "prepared" model and "curated" tables, even if they are the exact same, they are duplicated in the two models. In the RAW folder each notebook ingests one type of data that can come in varied formats and then that data is formatted into delta tables with minimal transformations to the data itself. This is done to make processing faster since delta tables are faster to read and write in the next steps.

The notebooks in the PROCESSED folder are a continuation of work of the RAW folder, the needed transformations are applied, and data is inserted into the prepared model. Finally, data is copied from the processed layer to the curated, these scripts are written in the CURATED folder.

To process historic data, there is a separate folder, the HISTORICO, this is necessary both for organizational purposes and because many times historical data come in different formats, or it has already been processed, therefore the notebooks read the files, transform them and insert them directly into the processed model.

ON_FAIL is an additional folder where notebooks are only run when there is an error in the pipelines, this will be explained further along. Finally, EXPLORATORY, is where developers can test any code or do analysis on the models, it is, as the name shows, a place to store scripts that explore the environment or changes to be made.

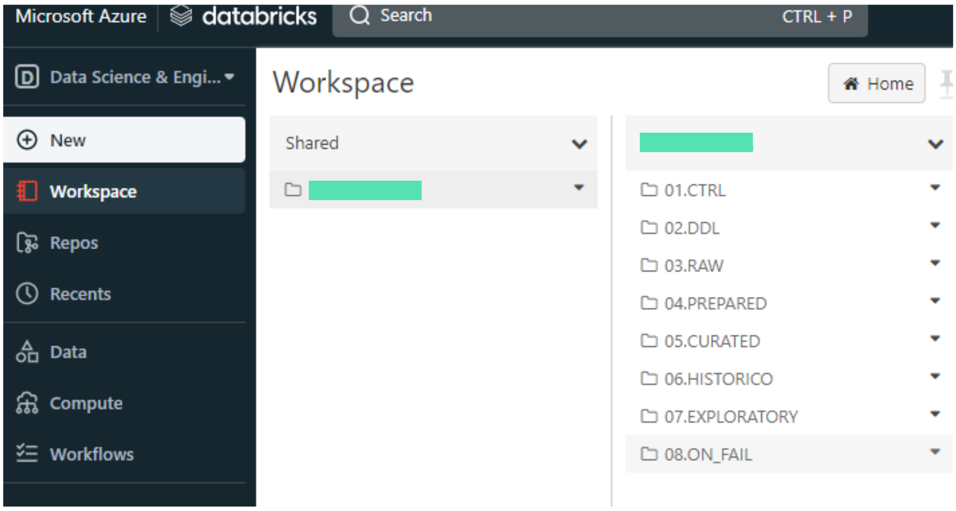


Figure 7 - Databricks Workspace

r

5.3. Pipelines in Data Factory

With all the notebooks ready to run, it is possible to automate the process, with the drag and drop interface of Azure Data Factory. One pipeline for each notebook is created, those are in turn aggregated in a parent pipeline for each type of data, in sequence of raw, prepared and curated. Because of the interdependence between them, a new notebook is run only when the other was successful, if there is a failure, its logged with timestamp and a trigger id, and the notebook will try to process it again, if not successful the parent pipeline is stopped completely, and the log of the unsuccessful run is kept in a table stored in the CTRL folder. An example diagram is shown below for better clarification.

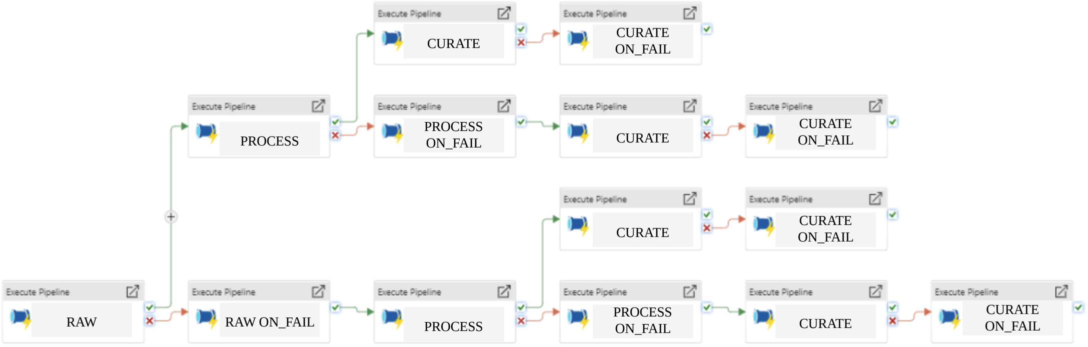


Figure 8 - Example of a data processing pipeline

With the pipelines constructed, a schedule for each was created, these are run every minute or hour depending on the velocity the data comes into storage.

5.4. Data Visualization and End-User access

When the pipelines were concluded and functioning, the final tables could start receiving data and its possible can start obtaining knowledge from all that information available. Databricks is designed for data analysis, but the objective is to allow users without a data analytics background to be able to visualize or query the data.

Synapse Analytics is the best way to allow final user access to the database, there external tables and views were created in a serverless SQL pool from the curated data in the storage account. The choice of a serverless SQL pool is due to being less costly, as data is not stored in synapse and automatically scaled up or down depending on use, which is necessary because the amount of data is constantly increasing.

To visualize the data, Power BI was the tool of choice, as it uses a drag-and-click interface to create simple dashboards and it allows more experienced user to develop more complex queries, additionally, as a Microsoft product there is a direct way to connect it to Azure Synapse to read the tables created there.

6. DISCUSSION

Implementing this framework was not straightforward, but many of the choices made proved to be beneficial for the accomplishment of the project. With this experience Microsoft Azure justifies its place on the market, in part because of the clear documentation of all of its tools, concepts, as well as the integration between most of its services. Although some changes had to be done to circumvent the restrictions of the cloud resources and, similar to any endeavor, some improvements can be studied and applied.

The simplicity stated earlier is evidenced by how easy it is for a user to generate resources. To create tools, users fill out a form that is standardized to the greatest extent between all the resources. The characteristics are chosen mostly with drop-down lists and most of the choices are explained within the form, or an external link is given. This made it much easier to deploy very complex applications.

Related to this point of comprehensibility, Microsoft's massive library of learning resources was critical to be able to accomplish this project, not only for general concepts but also for detailed explanations of almost every process related to setting up and using the resources, even integration with each other.

Last but not least, the management tools used were extremely beneficial to team cooperation and individual goals. Because SCRUM forces constant communication between the team members, every daily sprint issues were exposed and the team shares knowledge and details, that were previously unnoticed. The fact that the teams define their actions reduced wasted time.

As it is common, the architecture was not maintained during the development fase, one of the most time-consuming challenges was: receiving the data from the message broker to the storage account, this is the only moment that a product that is not made by Microsoft is used (apart from the FTP), which is a very negative point to an azure-based infrastructure, that it is very difficult to stray from the Microsoft brand, this is called a vendor lock-in.

The specific problem was that connection to Artemis ActiveMQ queue is made through a TCP or SSH connection, but as the Web Job that made this action was hosted in an Azure App Service, there needed to be another step, rerouting the host and port of the broker into an azure port that can be accessed by the Web Job, that was done through an Azure Virtual Network, which is a tool that allows for the message broker and azure to communicate. Although this solves the issue, it implies a higher cost and a lot of time was spent to reach this conclusion.

Another problem that arose due to this architecture, was the undermined costs until the end of development, Azure provides a guide of prices, but it is impossible to know the costs of the resources fully, and sometimes, it’s difficult to understand the specific source of the high costs.

For example, the scripts in Databricks influenced the cost of the storage account, as there were some scripts that moved data in storage between different folders, these costs were extremely high, and they were defined as “storage account costs”, it took some discussion to understand that was actually costs actioned by Databricks notebooks. This type of fluctuating costs makes it necessary to monitor costs after the architecture is in production and change it if necessary.

Finally, there were some restrictions: Spark, as a programming language, alongside the use of cloud storage, has some restrictions, not all libraries that are available for python are also available for PySpark, for example. Also, Spark has very few load functions (Apache Sofware Foundation, n.d.), this means that any formats that are not text, JSON, of excel/csv files must be read as binary files, and then decoded. It makes the script a bit more complex and time-consuming, but this is not something that can’t be solved, for most types of files.

There is one main area that should be improved in this process in the future, with the current technology available.

The provision of the curated data to the end-user should be faster, it is, as mentioned before, done through synapse serverless pool, but that tool is very often slow, because the tables are not stored in synapse but in the data lake, and every query is passed through synapse from Databricks to Power BI, and there is no caching allowed.

A dedicated SQL pool is not a viable alternative because it does not support creating tables from delta formats, which is the type of format used in the delta lake, therefore, further research is necessary to find a better connector between the delta lake and exterior end-user applications.

7. CONCLUSION

As a whole, this report has provided an overview of an Azure-based big data framework and explored Microsoft Azure's capability for handling large amounts of data and to clarify how this cloud may help businesses. It showed the innumerous capabilities of some of the most commonly used tools, although there are still many that can be explored, and even in the resources used, their capabilities are far beyond what was explored.

The clear division of the data lake architecture into raw, processed, and curated was essential to keep the data organized. Furthermore, the research and understanding about the necessary tools was vital to be able to implement the architecture.

Logically, this architecture was not perfect, and it was restricted by what Azure allows: a company needs to be prepared for a costly endeavor, since the initial cost may be high because it can only know the prices after the framework is in production. Although, it is possible to constantly strive to improve the efficiency will alterations postproduction.

This project shows that business should chose Microsoft Azure for their big data analysis, provided they have a good understanding of the tools they chose, and which ones fit their needs the best.

REFERENCES

Apache Software Foundation. (n.d.). *ActiveMQ Artemis Documentation*. Retrieved February 28, 2023, from https://activemq.apache.org/components/artemis/documentation/latest/preface.html

Apache Software Foundation. (n.d.). *Overview*. Spark 3.3.2 Documentation. Retrieved February 28, 2023, from https://spark.apache.org/docs/latest/

Apache Sofware Foundation. (n.d.). *Generic Load/Save Functions*. Spark 3.3.2 Documentation. Retrieved February 27, 2023, from https://spark.apache.org/docs/latest/sql-data-sources-load-save-functions.html

Campbell, R. (2020). When Open Source Came to Microsoft. *CODE Magazine*, 22–27. https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=aps&AN=145425549&lang=pt-br&site=ehost-live&custid=s6058661

Databricks. (2023). *Databricks concepts*. Databricks Documentation. https://docs.databricks.com/getting-started/concepts.html

Forouzan, B. A., & Fegan, S. C. (2002). *TCP/IP Protocol Suite* (2nd ed.). McGraw-Hill Higher Education.

Ghaffar, A., Tariq, S. &, Soomro, R., Shoro, A. G., & Tariq, &. (2015). Big Data Analysis: Ap Spark Perspective. *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, *15*.

Ghodsi, A. (2017). *Introducing Azure Databricks*. Databricks. https://www.databricks.com/blog/2017/11/15/introducing-azure-databricks.html

Ghose, S., Boroumand, A., Kim, J. S., Gómez-Luna, J., & Mutlu, O. (2019). Processing-in-memory: A workload-driven perspective. *IBM Journal of Research and Development*, *63*(6), 3:1-3:19. https://doi.org/10.1147/JRD.2019.2934048

Gorelik, A., & Safari, an O. M. Company. (2019). *The Enterprise Big Data Lake*.

Handfield, R. (2022). *Data Analyst, Data Scientist, and Data Engineer: What's the Difference?*

How data analytics are changing the consulting industry. (2018). *Consultancy.Uk*. https://www.consultancy.uk/news/18522/how-data-analytics-are-changing-the-consulting-industry

Hugos, M. H., & Hulitzky, D. (2010). *Business in the Cloud : What Every Business Needs to Know about Cloud Computing*. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=624431

Iafrate, F. (2015). *From Big Data to Smart Data*. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=1964138

Indu, I., Anand, P. M. R., & Bhaskar, V. (2018). Identity and access management in cloud environment: Mechanisms and challenges. *Engineering Science and Technology, an International Journal*, *21*(4), 574–588. https://doi.org/10.1016/J.JESTCH.2018.05.010

International Scrum Institute. (2019). *The Scrum Framework Framework Training Book*. www.scrum-institute.org

K K, A. (2020). Azure DevOps for Web Developers. In *Azure DevOps for Web Developers*. Apress. https://doi.org/10.1007/978-1-4842-6412-6

Loukides, M. (2021). *The Cloud in 2021: Adoption Continues*. O'Reilly. https://www.oreilly.com/radar/the-cloud-in-2021-adoption-continues/

Manjoo, F. (2016). Tech's 'Frightful 5' Will Dominate Digital Life for Foreseeable Future - The New York Times. *The New York Times*. https://www.nytimes.com/2016/01/21/technology/techs-frightful-5-will-dominate-digital-life-for-foreseeable-future.html

Mantel, S. J., Meredith, J. R., Shafer, S. M., & Sutton, M. M. (2011). *Project Managment in Practice*.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition, and productivity*. www.mckinsey.com/mgi.

Mason, R. (2018). Changing Paradigms of Technical Skills for Data Engineers. *Issues in Informing Science and Information Technology*, *15*. https://doi.org/10.28945/4033

Mayer-Schonberger, V., & Cukier, K. (2013). *Big Data: A Revolution That Will Transform How We Live, Work, and Think - PDFDrive.com*.

McNeely, C. L. (2015). Prospects and Challenges in the Information Society. *Journal of the Washington Academy of Sciences*, *101*(3), 1–10. https://www.jstor.org/stable/jwashacadscie.101.3.1

Microsoft. (2022a). *Develop and deploy WebJobs using Visual Studio - Azure App Service*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/app-service/webjobs-dotnet-deploy-vs

Microsoft. (2022b). *Run background tasks with WebJobs - Azure App Service*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/app-service/webjobs-create

Microsoft. (2022c). *Serverless SQL pool - Azure Synapse Analytics*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/on-demand-workspace-overview

Microsoft. (2022d). *What is Azure Synapse Analytics? - Azure Synapse Analytics*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Microsoft. (2023a). *App Service overview*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/app-service/overview

Microsoft. (2023b). *Azure Data Lake Storage Gen2 Introduction - Azure Storage*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

Microsoft. (2023c). *Azure integration with Microsoft 365*. Microsoft Learn. https://learn.microsoft.com/en-us/microsoft-365/enterprise/azure-integration?view=o365-worldwide

Microsoft. (2023d). *Azure Storage SDK for .NET - Azure for .NET Developers*. Microsoft Learn. https://learn.microsoft.com/en-us/dotnet/api/overview/azure/storage?view=azure-dotnet

Microsoft. (2023e). *Clusters - Azure Databricks*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/databricks/clusters/

Microsoft. (2023f). *Copy data from an FTP server - Azure Data Factory & Azure Synapse*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/data-factory/connector-ftp?tabs=data-factory

Microsoft. (2023g). *Earnings Release FY22 Q2*. Microsoft. https://www.microsoft.com/en-us/Investor/earnings/FY-2022-Q2/press-release-webcast

Microsoft. (2023h). *Introduction to Azure Data Factory - Azure Data Factory*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/data-factory/introduction

Microsoft. (2023i). *Introduction to Azure Storage - Cloud storage on Azure*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction

Microsoft. (2023j). *Introduction to Blob (object) Storage - Azure Storage*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

Microsoft. (2023k). *.NET programming languages*. Microsoft .NET. https://dotnet.microsoft.com/en-us/languages

Microsoft. (2023l). *Storage account overview - Azure Storage*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview#types-of-storage-accounts

Microsoft. (2023m). *VM sizes - Azure Virtual Machines*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/virtual-machines/sizes

Microsoft. (2023n). *What is dedicated SQL pool (formerly SQL DW)? - Azure Synapse Analytics*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is?context=%2Fazure%2Fsynapse-analytics%2Fcontext%2Fcontext

Microsoft. (2023o). *What is .NET?* https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet

Microsoft. (2023p). *What is the Cloud - Definition*. Microsoft Azure. https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-the-cloud

Microsoft. (2023q). *Why choose the .NET*. Microsoft .NET. https://dotnet.microsoft.com/en-us/platform/why-choose-dotnet

Microsoft Azure. (2023). *Azure Storage Blobs Pricing*. Azure Pricing. https://azure.microsoft.com/en-us/pricing/details/storage/blobs/#overview

Muan Sang, G., Xu, L., & de Vrieze, P. (2020). *A Reference Architecture for Big Data Systems*.

nuget. (2023). *NuGet Gallery*. https://www.nuget.org/

Ohlhorst, F. J. (2012). *Big Data Analytics : Turning Big Data into Big Money*. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=821833

Pries, K. H., & Quigley, J. M. (2010). *Scrum Project Management*. Taylor & Francis Group. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=589930

Reno. (2022). Cloud Infrastructure Services Market. *Synergy Research Group*. https://www.srgresearch.com/articles/q3-cloud-spending-up-over-11-billion-from-2021-despite-major-headwinds-google-increases-its-market-share

Rittinghouse, J. W., & Ransome, J. F. (2009). *Cloud Computing : Implementation, Management, and Security*. Taylor & Francis Group. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=472836

Sharma, V. (2018). *The Cloud-Based Demand-Driven Supply Chain*. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/iseg/detail.action?docID=5592835

Stack Overflow. (2020). Developer Survey 2020. In *2020*. https://insights.stackoverflow.com/survey/2020#development-environments-and-tools

Wilemon, D. L., & Cicero, J. P. (1970). The Project Manager--Anomalies and Ambiguities. *The Academy of Management Journal*, *13*(3), 269–282. https://doi.org/10.2307/254964