**Proof of Concept Guide**

**New DB Migration**

Module: DB Migration

Process: New SQL DB Acquisition

Subprocess:

Version: 1.0

Content Index

[1. Main Goal 3](#_Toc57754130)

[2. Scope 3](#_Toc57754131)

[3. Data Sources 3](#_Toc57754132)

[3.1. CSV Files 3](#_Toc57754133)

[3.2. Functional Description 3](#_Toc57754136)

[4. Data Destination 4](#_Toc57754137)

[4.1. Azure SQL Database 4](#_Toc57754138)

[4.1.1. Conexiones 4](#_Toc57754139)

[4.1.2. Esquemas 4](#_Toc57754140)

[4.1.3. Tablas 4](#_Toc57754141)

[6. Execution Cicle 4](#_Toc57754143)

[8. Risk Management 4](#_Toc57754145)

[11. PoC Authoring 4](#_Toc57754148)

[12. Version History 4](#_Toc57754149)

# Main goal

The main goal for this project is to migrate/consolidate a new Database, and to migrate all the hired employees information star schema into that Database.

# Scope

The resources that will be used for this Project will be base on Azure cloud services:

* Azure Data Factory for extracting, transforming and loading the Data into the new Database.
* Azure Databricks to perform further preparation of the Data with Python.
* Azure SQL Database to store the new data.
* Azure DataLake Gen2 to store the staging thata while it is being curated.
* GitHub, for code versioning and backup.
* Azure DevOps, for releasing the pipeline into Production.

# Data Sources

# CSV Files

The source csv files will be located in an Azure blob storage for retrieval. That storage will include a lifecycle policy, and will go into Archive mode in 6 months from being uploaded.

# Functional Description

The proposed solution involves using Azure Data Factory to create and manage an automated pipeline. There are other Azure services involved in the process, as stated in these functional steps:

1. An Azure DataFactory instance will be created for development and another will be set up for production mode.
2. A DataLake Storage Account Gen2 will be used as the staging environment. It will also be a folder for each environment.
3. From DataFactory, the CSV files will be initially from a blob storage and loaded into a staging folder, for data preparation.
4. An Azure Databricks instance will then be called for applying data rules and validations with Pyspark
5. Databricks will Split the data to separate records that are not aligned with the API rules, and generate a new file to be sent for revision.
6. From DataFactory a REST API will be called to send the information into the Azure SQL Database.

To be consider in the process: The DataFactory instance will be linked to GitHub for code versioning/backup, and Azure DevOps will be used to send the pipelines for deployment.

# Data Destination

# Azure SQL Database

The destination of the Data will be an Azure SQL instance, where the data will be ready for consumption.

# Execution Cicle

On demand (non Schedule defined yet)

# Risk management

* The SQL Database will be Geo-replicated for disaster recovery
* All the credentials will be stored in an Azure Key Vault for security purposes.
* DataFactory Pipelines will have alerts for any failures in the processing of the data.

# PoC authoring

|  |  |  |  |
| --- | --- | --- | --- |
| Data Engineering Team | | | |
| Prepared by | Ezequiel Fernandez Navone | Fecha | 08/09/2022 |
| Approved by |  | Fecha |  |

# Version history

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 08/09/2022 | 1.0 | Ezequiel Fernandez Navone | Initial PoC version |
|  |  |  |  |
|  |  |  |  |