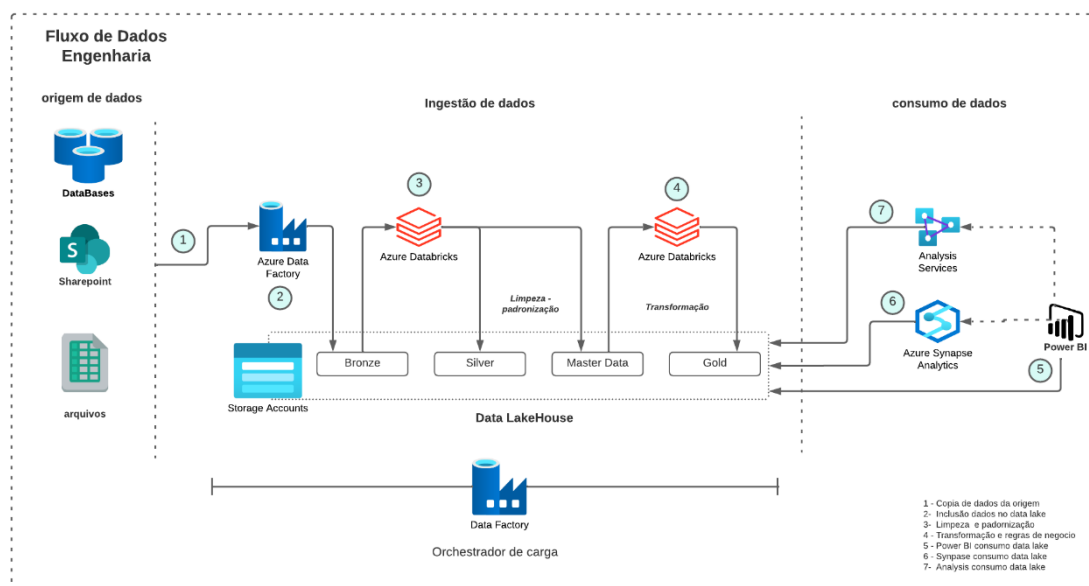


Engenharia CCR versão 1.0

Este documento tem como o objetivo auxiliar e orientar o processo de ingestão de dados na plataforma Azure CCR – Lab.

Visão geral

processo de ingestão de dados na plataforma:



Origem de dados

Os dados precisam ser analisados previamente e criado um mapa de dados inicial para realizar o início do processo de ingestão de dados.

Mapa de dados:

Eles podem ser de cunho técnico (nome da tabela, tipo de dado por coluna da tabela, quantidade de colunas da tabela, origem do dado, tipo de ingestão, entre outros) ou de cunho funcional [termo de negócio, glossário de termos de negócio, finalidade do dado, classificação PII (dado pessoal), descrição da regra de transformação (caso exista), etc.].

Exemplo:

	A	B	C	D	E	F
1	Tabelas					
2						
3	Database	Schema	Nome Tabela	Obs	LINK	SCRIPT COMENTARIO
4	SUATPRODS	AVINew	ADMC0B		ADMC0B	
5	SUATPRODS	AVINew	AREA_VEICULO		AREA_VEICULO	
6	SUATPRODS	AVINew	ARRECADADOR		ARRECADADOR	
7	SUATPRODS	AVINew	CATEGORIA		CATEGORIA	
8	SUATPRODS	AVINew	CONCESS		CONCESS	
9	SUATPRODS	AVINew	CONTROLE		CONTROLE	
10	SUATPRODS	AVINew	CONTROLE_MSG		CONTROLE_MSG	
11	SUATPRODS	AVINew	EFEITO_MOTIVO		EFEITO_MOTIVO	
12	SUATPRODS	AVINew	INTEGRA_LANCTO		INTEGRA_LANCTO	
13	SUATPRODS	AVINew	INTEGRA_RECEITA		INTEGRA_RECEITA	
14	SUATPRODS	AVINew	INTEGRA_RECEITA_DOC		INTEGRA_RECEITA_DOC	
15	SUATPRODS	AVINew	LBRANCA1		LBRANCA1	
16	SUATPRODS	AVINew	MARCA_VEICULO		MARCA_VEICULO	
17	SUATPRODS	AVINew	MOTIVO_DIFER		MOTIVO_DIFER	
18	SUATPRODS	AVINew	MOTIVO_IMAGEM		MOTIVO_IMAGEM	
19	SUATPRODS	AVINew	MOTIVO_LBRANCA		MOTIVO_LBRANCA	
20	SUATPRODS	AVINew	MUNICIPIO		MUNICIPIO	

	A	B	C	D	E	F
1	DETALHAMENTO TABELA					
2						
3	DATABASE	SUATPRODS				
4	SCHEMA	AVINew				
5	NOME_TABELA	AREA_VEICULO				
6						
7	ID_COLUNA	NOME_COLUNA	TIPO	TAMANHO	CHAVE	OBS
8	1	CDAREA_VEICULO	CHAR	5	Sim	
9	2	DCAREA_VEICULO	VARCHAR2	30		
10						
11						

Azure Data Lake

O data lake é um repositório único e centralizado onde você pode armazenar todos os seus dados, estruturados e não estruturados. O armazenamento de dados utilizado na plataforma é **Data Lake Storage Gen2**.

A arquitetura adotada é baseada na arquitetura medallion que descreve uma série de camadas de dados que denotam a qualidade dos dados armazenados no Lakehouse. Essa abordagem de várias camadas para criar uma única fonte confiável para produtos de dados corporativos. Essa arquitetura garante a atomicidade, consistência, isolamento e durabilidade à medida que os dados passam por várias camadas de validações e transformações antes de serem armazenados em um layout otimizado para análise eficiente.

Storage:

Name	Last modified	Public access level	Lease state
<input type="checkbox"/> bronze	12/14/2022, 10:11:58...	Private	Available ***
<input type="checkbox"/> config	12/14/2022, 10:12:24...	Private	Available ***
<input type="checkbox"/> gold	12/14/2022, 10:12:12...	Private	Available ***
<input type="checkbox"/> landing	12/14/2022, 10:12:34...	Private	Available ***
<input type="checkbox"/> masterdata	3/7/2023, 9:02:44 AM	Private	Available ***
<input type="checkbox"/> silver	12/14/2022, 10:12:06...	Private	Available ***

Estrutura dos diretórios/Containers

CONFIG - Camada para arquivos de configurações

LANDING - Camada para recepção de dados streaming

BRONZE - Camada para recepção de dados brutos e integração

SILVER - Camada para histórico, limpeza e dados validados

MASTER DATA - camada para consumo de dados e mínimas regras de negócio foco e self-service

GOLD - camada para consumo de dados com agregações, junções, filtragem e regras de negócio.

Subpastas

As subpastas dos containers serão organizadas por

Container/Tema/Projeto/

Authentication method: Access key (Switch to Azure AD User Account)

Location: bronze / RODOVIAS / SUAT

Search blobs by prefix (case-sensitive)	
Name	Modified
<input type="checkbox"/>  [..]	
<input type="checkbox"/>  AUTOBAN	
<input type="checkbox"/>  MSVIA	

É interessante tentar manter as estruturas parecidas ou semelhantes em todos os containers para facilitar a manutenção e entendimento dos projetos e sua respectivos temas.

OBS: Para criação de pasta é importante manter o padrão de **CAIXA ALTA** nos textos das pastas.

Azure Data Factory

É o serviço de integração de dados e ETL baseado em nuvem que lhe permite criar fluxos de trabalho orientados a dados para orquestrar a movimentação e a transformação de dados em escala.

Pipelines

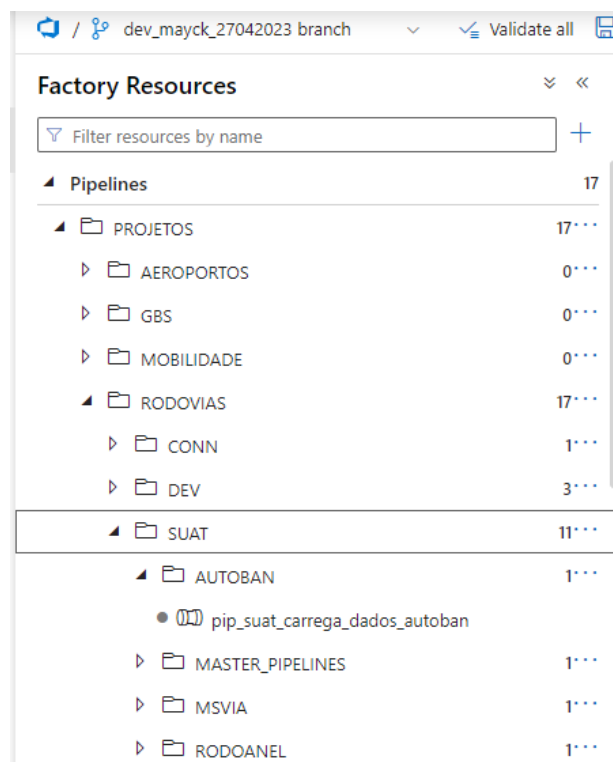
Um pipeline é um agrupamento lógico de atividades que realiza uma unidade de trabalho. Para desenvolvimento dos pipelines será adotado alguns padrões de nomenclatura e organizações de pastas.

Nomenclatura

- pip_sistema_livre
 - pip_master_sistema_livre (pipeline que executa outros pipelines)
- ex: pip_suat_carrega_tabela

PASTAS

PROJETOS/TEMA/SISTEMA



Atividades

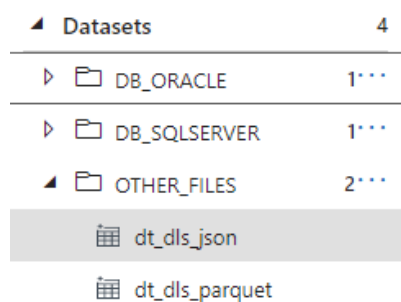
As atividades representam uma etapa de processamento em um pipeline. O Data Factory dá suporte a três tipos de atividades: atividades de movimentação de dados, atividades de transformação de dados e atividades de controle.

Por exemplo, você pode usar uma atividade de cópia para copiar dados de um repositório de dados para outro.

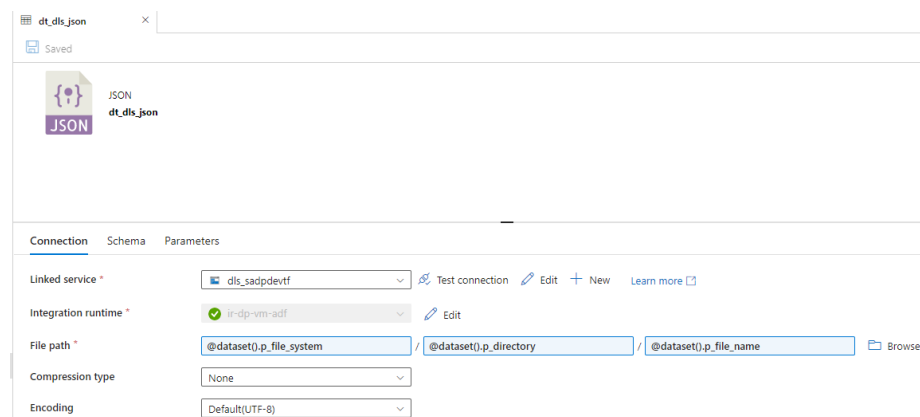
Datasets

Os Datasets representam as estruturas de dados nos repositórios de dados.

Para criação de Datasets usar parâmetros preferencialmente assim podemos facilitar a utilizações para diversos usuários.



Dataset Com exemplo de parâmetro:



Importante sempre alinhar com o Dev lead para usar o padrão de nomenclaturas

- dt_tipoconexão

Linked services

Linked services são como cadeias de conexão, que definem as informações de conexão necessárias para que o serviço se conecte a recursos externos.

Para criação de Linked services usar parâmetros preferencialmente assim podemos facilitar a utilizações para diversos usuários.

Linked services







Linked service defines the connection information to a data store or compute. [Learn more](#)

New

Filter by name

Annotations : Any

Showing 1 - 6 of 6 items

Name	Type	Related	Annotations
 databricks_dp_dev_eng	Azure Databricks	14	
 dls_sadpdevtf	Azure Data Lake Storage Gen2	2	
 kv_dp_dev_adf_tf	Azure Key Vault	0	
 kv_spa_dev	Azure Key Vault	0	
 ls_db_oracle_generic	Oracle	1	
 ls_db_sqlserver_generic	SQL server	1	

Importante sempre alinhar com o Dev lead para usar o padrão de nomenclaturas

- ls_tipoconexão

Data Flow

Não é previsto uso de dataflow pois utilizamos o databricks para execução das transformações de dados.



Integration Runtime

O IR (Integration Runtime) é a infraestrutura de computação usada pelo Data Factory.

É utilizado um IR Self Hosted para execução dos pipelines (regra de conectividade e segurança).

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment. [Learn more](#)

+ New Refresh						
Filter by name						
Showing 1 - 2 of 2 items						
Name	Type	Sub-type	Status	Related	Region	Version
 AutoResolveIntegrationR...	Azure	Public	Running	0	Auto Resolve	---
 ir-dp-vm-adf	Self-Hosted	---	Running	4	---	5.27.8466.1

Como padrão de execução utilizamos **ir-dp-vm-adf**.

TRIGGER

Trigger determina quando uma execução de pipeline precisa ser inicializada.

Nomenclatura:

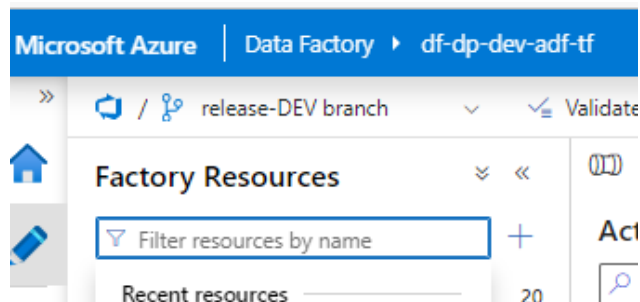
- TGR_MASTER_SISTEMA_PERIODO_OPCIONAL (pipeline que executa outros pipelines)

-TGR_SISTEMA_PERIODO_OPCIONAL

Importante sempre alinhar com o Dev lead para usar o padrão de nomenclaturas

Branch

Para utilização do Data Factory é obrigatório a utilização de Branch. Todas as branch devem ser criadas a partir da RELEASE -DEV Branch

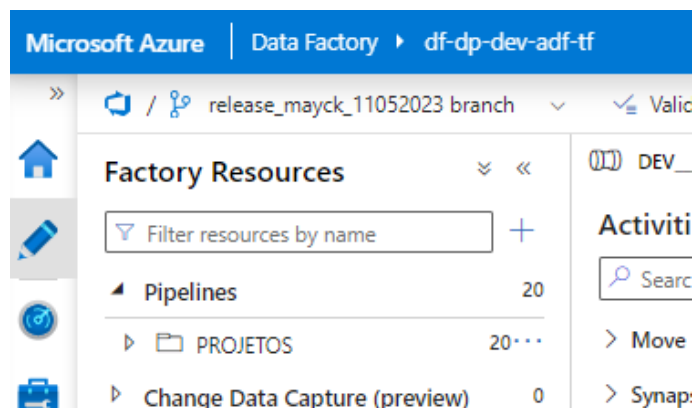


A criação das Branch deve seguir o padrão de nomenclaturas

Nomenclatura:

Brach_nome_data_livre

Ex: release_mayck_01012023



Todos os processos devem ser feitos pull request para inclusão na master e solicitar aprovação dos respectivos responsáveis.

Azure DataBricks

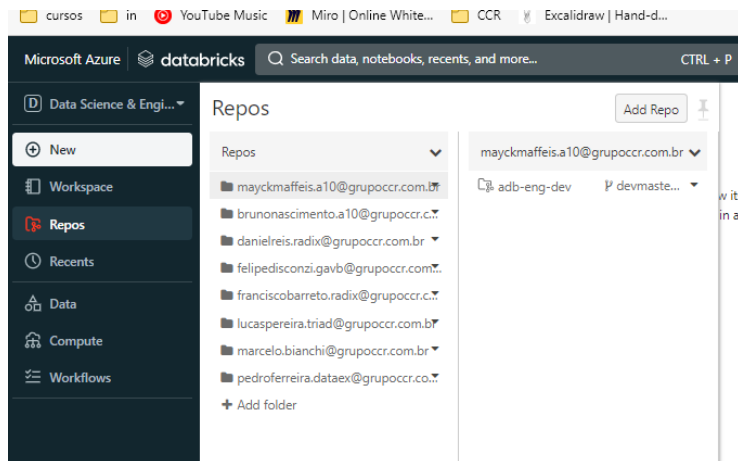
A Plataforma do Azure Databricks Lakehouse fornece um conjunto unificado de ferramentas para criar, implantar, compartilhar e manter soluções de dados. O Databricks implanta clusters de computação efêmeros usando recursos de nuvem na conta para processar e armazenar dados no armazenamento de objetos e outros serviços integrados controlados

Para utilização do Databricks será adotado alguns padrões de desenvolvimento para facilitar a manutenção e controle dos processos de ingestão de dados

Repo

Os desenvolvimentos de projetos serão necessários criar uma Branch no repo do databricks no seu usuário para os notebooks ficarem atrelados ao git.

CCR devops acessar o projeto - **ccr-gbs-labinov-dataproject-eng**. Utilizar o **repo adb-eng-dev** (esse é o repo de notebooks)



Caso não tenha acesso ao devops no projeto **ccr-gbs-labinov-dataproject-eng**, solicitar ao arquiteto devops ou arquiteto de dados da dataplataforma.

Após adicionar repo criar a branch baseada na **release-dev**

Branch

Para utilização do Data Bricks é obrigatório a utilização de Branch. Todas as branches devem ser criadas a partir da RELEASE-DEV Branch

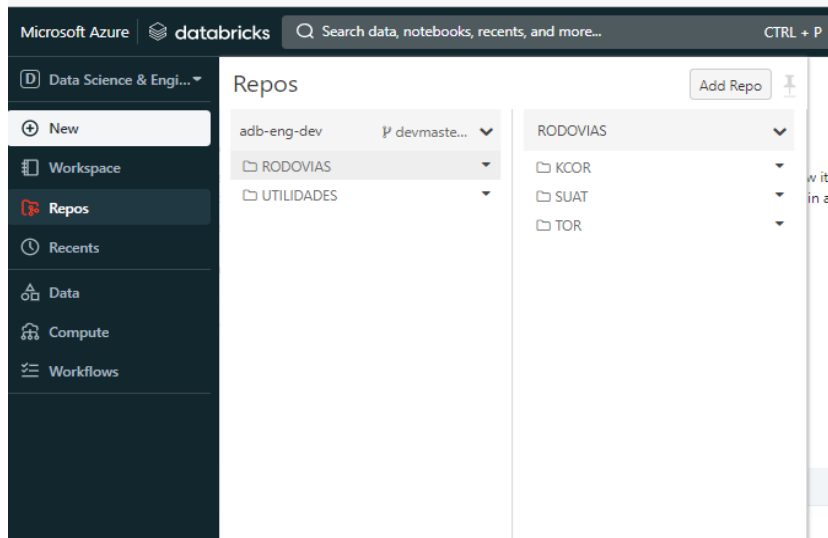
Nomenclatura:

Branch_nome_data_livre

Ex: release_mayck_01012023

Pastas

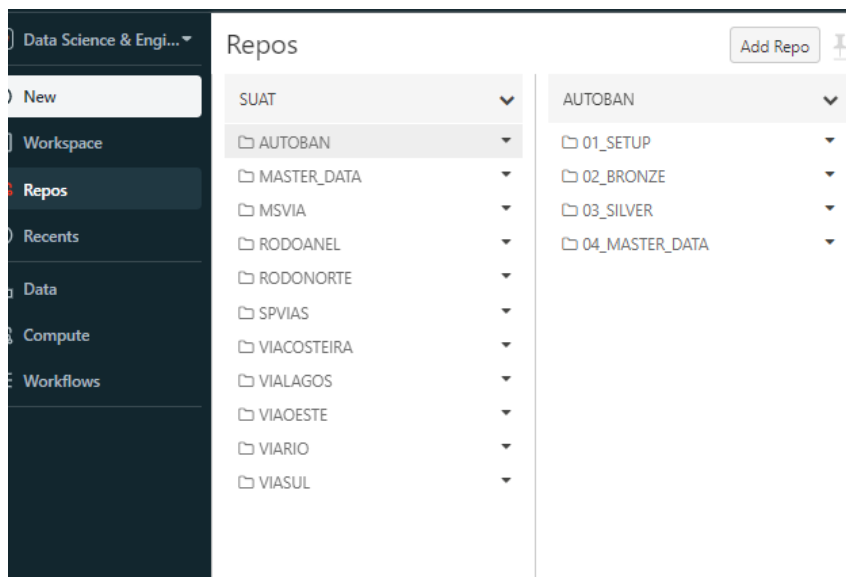
PROJETOS/TEMA/SISTEMA



Organização notebooks

Usar pastas para facilitar a localização dos notebooks

SETUP/ BRONZE/SILVER /MASTERDATA/GOLD



TABELAS

Para criação das tabelas no databricks é importante seguir alguns padrões.

Tabelas precisam ser criadas sempre apontando para o repositório do datalake utilizando o padrão DELTA e Iniciando com sigla TBL

Exemplo:

-Tabela logica:

NOMESchema.TBL_SISTEMA_NOMETABELA

-Tabela física / delta table:

Caminho datalake **/TBL_NOME_TABELA**

Exemplo:

cria tabela

```
1 sql
2 DROP TABLE IF EXISTS bronze_autoban.tbl_suat_area_veiculo;
3
4 CREATE OR REPLACE TABLE bronze_autoban.tbl_suat_area_veiculo
5 (
6     CDAREA_VEICULO STRING
7     ,DCAREA_VEICULO STRING
8     ,DATA_CARGA TIMESTAMP
9 )
10 USING DELTA
11 LOCATION '/mnt/bronze/RODOVIAS/SUAT/AUTOBAN/AREA_VEICULO/TBL_AREA_VEICULO/'
```

Data Explorer

bronze_autoban

bronze_autoban.tbl_suat_categoria

Columns

Sample Data

Details

Permissions

History

CD_CATEGORIA	CD_CATEGORIA	INDENOS	FLMARRA	CDPAIS	CDCONCESS	CD_CATEG_BASICA	CD_CATEG_ADM	CD_CATEG_PC	CD_CATEG_CORRECAO	FLCONSIGD_CP	FLCONSIGD_TRAFEGO	NOORDEM	TPVEICULO	TPCATEGORIA	FLVARIAB_BNO	IDCATEG_MSG
13	20	2	2	0618	00259	02	02	02		S	S	13	2	P	N	2
14	30	3	2	0618	00259	02	03	03		S	S	14	2	P	N	3
15	40	4	2	0618	00259	02	04	04		S	S	15	2	P	N	4
01	AUTO	2	1	0618	00259	01	01			S	S	1	1	L	N	1
02	20	2	2	0618	00259	02	02			S	S	2	3	P	N	2
03	30	3	2	0618	00259	02	03			S	S	3	3	P	N	3
04	40	4	2	0618	00259	02	04			S	S	4	3	P	N	4
05	50	5	2	0618	00259	02	05			S	S	5	3	P	N	5
06	60	6	2	0618	00259	02	06			S	S	6	3	P	N	6
07	70	7	0	0618	00259	02	61			S	S	7	3	P	S	61
08	80	8	0	0618	00259	02	62			S	S	8	3	P	S	62
09	90	9	0	0618	00259	02	63			S	S	9	3	P	S	63
10	MOTO	0	1	0618	00259	01	09	09		S	S	10	0	L	N	
11	35	3	1	0618	00259	01	07	07		S	S	11	1	L	N	7
12	45	4	1	0618	00259	01	08	08		S	S	12	1	L	N	8
20	CAT 20	0	0	0618	00259	02	64			S	S	16	3	P	S	
16	50	5	2	0618	00259	02	05	05		S	S	5	3	P	N	5
17	60	6	2	0618	00259	02	06	06		S	S	6	3	P	N	6

Obs: sempre criar tabelas apontando para o datalake não criar tabelas internas no databricks.

Mount

Mount point - **"/mnt/"**

```
1 %fs
2 ls /mnt|
```

Table

path

name

size

modificationTime

1	dbfs:/mnt/bronze/	bronze/	0	0
2	dbfs:/mnt/config/	config/	0	0
3	dbfs:/mnt/gold/	gold/	0	0
4	dbfs:/mnt/landing/	landing/	0	1679579601000
5	dbfs:/mnt/masterdata/	masterdata/	0	0
6	dbfs:/mnt/silver/	silver/	0	0

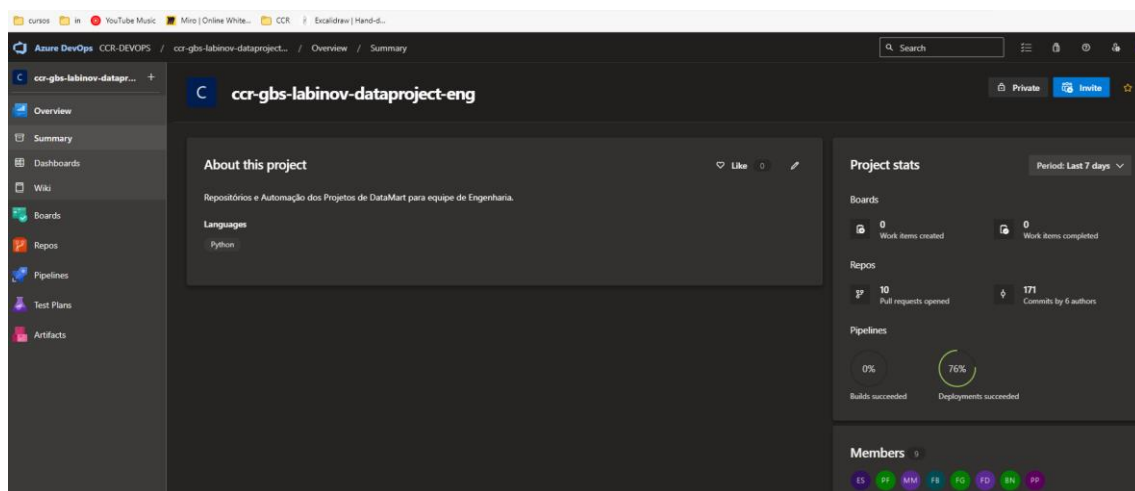
Cluster

Preferencialmente utilizar o cluster de acordo com o tema do projeto e alinhado com arquiteto de dados da plataforma.

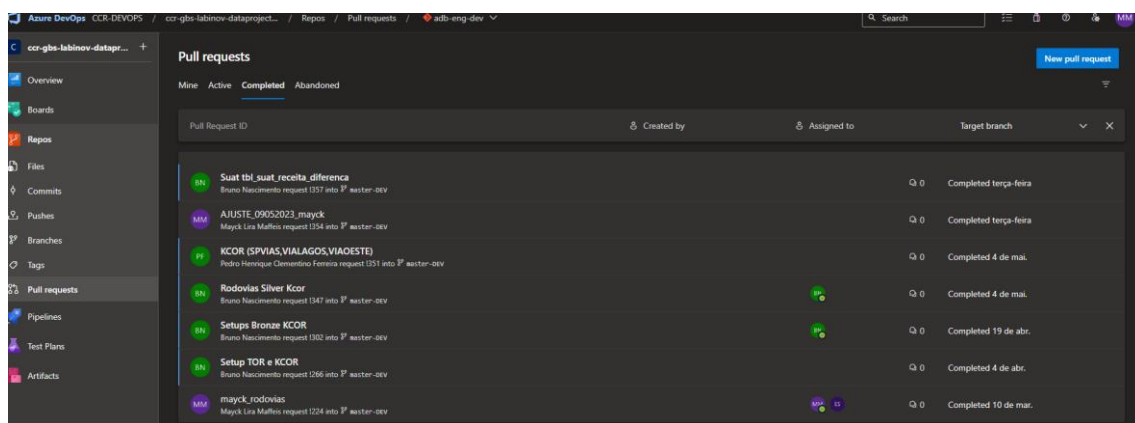
DEVOPS

Todos os engenheiros de dados estarão centralizados no devops de engenharia

<https://dev.azure.com/CCR-DEVOPS/ccr-gbs-labinov-dataproject-eng>



Para subida de datafactory e databricks será necessário a criação de pull request para aprovação de pipelines e notebooks para ambientes de DEV/HML/PROD .



As pull requests precisam ser alinhadas e validas pelos arquitetos devops e dados. Na fase de envio do projeto de *hml* para *prod* será adicionado a validação dos pipelines e notebooks o respectivo PM do projeto.

Azure Synapse

Próximos passos

Azure Purview