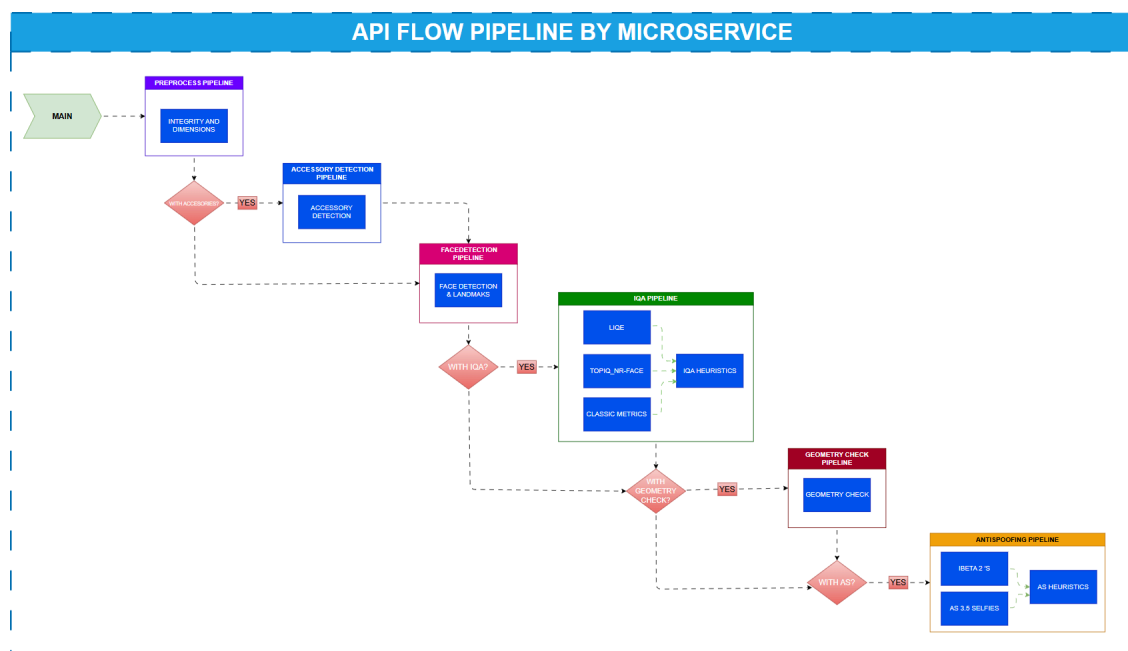


## ETL Final Workshop - Manuel Henao - [Github Repo](#)

**Problem Description:** The project aims to analyze the performance of a facial validation API using an ETL process. The API, built on a microservices architecture, includes face detection, accessory detection, image quality assessment, and Presentation Attack Detection (PAD) to prevent identity spoofing. Performance data is stored in a non-relational database. The primary objectives are to identify execution time bottlenecks of each microservice, analyze acceptance rates per client, and improve user experience by examining retry patterns.

**Context:** The facial validation API is designed to ensure secure and efficient identity verification by capturing two moments: one where the user is near the camera and another where the user is farther away. This process, known as liveness detection, aims to enhance the performance of the PAD microservice. Each microservice will have execution metrics for both images. The API comprises several microservices:

- Face Detection: Identifies and locates faces in images.
- Accessory Detection: Detects accessories like glasses or masks that might obscure the face.
- Image Quality Assessment: Evaluates the quality of the image to ensure it meets required standards using various methods.
- Presentation Attack Detection (PAD): Detects attempts to spoof the system using photos, videos, or masks.



Workflow Diagram of Facial Validation API by microservice

The API's performance data is stored in a non-relational database using MongoDB, capturing metrics such as execution times, acceptance rates, and retry counts for two attempts. The pipeline workflow executes the microservices in a serial manner. For example, if the accessory detection microservice detects an accessory, it will terminate the transaction without processing the subsequent microservices.

**Dataset Description:** The dataset includes various performance metrics for each microservice. Below is a model card for the variables of interest ([Staging Table](#)):

## Process

The ETL Workshop will be broken down into 3 main stages

1. **Extract Stage:** Involves the collection performance data from the API's non-relational database.
  - a. Due to sensible data the raw data which is stored in a non-relational db in a json format some features were omitted and some others were anonymized because clients transactions behavior were involved. In order to achieve that as part of the ETL process the first stage called **000\_anonymized.py** anonymizes and omits raw semi-structured data into a tabular format represented in a CSV file with the features mentioned in the model card.
  - b. Following the ETL process the second stage is loading the data as part of the staging part that executed by **001\_staging.ipynb** stages the raw data (after being anonymized) in a relational database.
2. **Transform Stage:** It covers the cleaning, transformation, and preprocessing of the data to ensure consistency, accuracy, and relevant insights. The transformation stage is executed by **002\_eda\_transform.ipynb**. This notebook covers the EDA of the staging table obtained and performs the transformation workflow to discover insights in the data that can describe API performance. It aims to help identify possible time bottlenecks of each microservice, analyze acceptance rates per client, and improve user experience by examining retry patterns. These transformations are described below:
  - **Transform Aspect Ratio:** Determines the aspect ratio of an image based on its width and height. It returns the following categories ('SQUARE', 'PORTRAIT', 'LANDSCAPE')
  - **Transform Compare Aspect Ratio:** This process compares the outputs of aspect ratio transformations and returns a binary category (True, False). Under normal API execution, this value is expected to be true. If false, it may indicate an anomaly with the input images used in the transaction, potentially signaling a presentation attack attempt.
  - **Transform Quality Image:** Determines the quality of an image based on its resolution. It returns the following categories ('8K', '4K', 'ULTRAHD', 'FULLHD', 'HD', 'XGA', 'SD', 'QVGA', 'LOWERSCALES')
  - **Transform Compare Quality Image:** This process compares the outputs of image quality transformations and returns a binary category (True, False). Under normal API execution, this value is expected to be true. If false, it may indicate an anomaly with the input images used in the transaction, potentially signaling a presentation attack attempt.
  - **Transform Date by Range:** This transformation converts a date column into multiple date-related columns. It returns the data divided by year, month, day, day of the week, hour, and minute. Additionally, it maps the month and day of the week into human-readable formats.
  - **Transform Missing Values related with Time:** Due to the sequential execution of the pipeline, some microservices may not be executed if an event is triggered in a previous microservice. For example, if the image contains accessories, the subsequent microservice will not be executed. To analyze the total execution time per microservice without affecting overall performance, rows with missing values will be filled with 0 seconds.
  - **Transform Missing Values related with metrics and probabilities:** Due to the sequential execution of the pipeline, some microservices may not be executed if an event is triggered in a previous microservice. For example, if the image contains accessories, the subsequent microservice will not be executed. The metrics and probabilities cover numerical scores with different ranges. In this case, filling a value with 0 could affect subsequent analysis to establish a possible image quality

assessment threshold for accepting or denying a transaction. Therefore, missing values will be filled with -1 for columns related to probabilities and -2 for columns related to scores.

- **Transform Getting Total Time by Microservice:** Total time in ms and queue time per each microservice for near and far images are sum in order to get a total in ms by each microservice. As output the following columns are the result the original or individual time executions are dropped by this totals
    - 'total\_time\_ms\_accessory\_detector'
    - 'total\_time\_ms\_face\_detector'
    - 'total\_time\_ms\_liqe'
    - 'total\_time\_ms\_topiq'
    - 'total\_time\_ms\_classic\_metrics'
    - 'total\_time\_ms\_as\_35\_selfies'
    - 'total\_time\_ms\_ibeta2\_crops'
    - 'total\_time\_ms\_ibeta2\_full'
    - 'total\_time\_ms\_ibeta2\_clip'
    - 'total\_time\_ms\_geometry\_check'
  - **Transform Getting Overall Other Time Microservice:** Using the 'Transform Getting Total Time by Microservice' as input and the 'total\_time\_ms' column, the remaining time, which encompasses all execution times not returned by the API payload, is estimated. These times include, for example, transmission time between microservices, preprocessing times, image upload time to cloud storage, and time for insertion into the database.
  - **Transform Getting Transactions without retries:** As part of the acceptance rates per client analysis, the column 'procesoConvenioGuid' represents the unique identifier of the transaction. Since each client has a different number of maximum retries for transactions per day, the final acceptance rate per client depends on the last response of each 'procesoConvenioGuid'. The final API response for each case depends on the columns 'summary\_status' and 'summary\_descr', which specify which microservice was triggered.
  - **Dataset Description:** After the dataset transformations the model card for the variables is updated below ([Transform Table](#)):
3. **Load Stage:** Load the transformed data into an analytical database for further analysis those DBs are etl\_transform\_table\_final\_workshop and etl\_transform\_table\_final\_workshop, both of them ready to use for visualization purpose , the first one more related to overall API performance considering transaction retries and the second one focuses on the acceptance rate per clients, due to the fact it considers only the last transaction.

## Evidences

Extract Stage Evidence after executing **001\_staging.ipynb**

Loading / Reading Staging Table

```
D> db_table_name = "etl_staging_table_final_workshop"
with engine.connect() as conn:
    of_staging = ps.read_sql(f"SELECT * FROM {db_table_name}", conn)

print(of_staging.info())
print(of_staging.describe())
of_staging.head()
```

```
[1]: RE Open VCL Staging in Data Manager Python
```

```
class 'pandas.core.frame.DataFrame'
RangeIndex: 14654 entries, 0 to 14653
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype
---  ---
 0   api_hash            14654 non-null  object
 1   user_name           14654 non-null  object
 2   file_token          14654 non-null  object
 3   service             14654 non-null  object
 4   summary_code        14654 non-null  int64
 5   summary_status      14654 non-null  object
 6   summary_desc        14654 non-null  object
 7   http_code           14654 non-null  int64
 8   total_time_ms       14654 non-null  int64
 9   date               14654 non-null  object
10   procesoConvenioGuid 14654 non-null  object
11   convenio            14654 non-null  object
12   documento           14654 non-null  object
13   img_size_near_img_width  14654 non-null  int64
14   img_size_near_img_height 14654 non-null  int64
15   img_size_far_img_width  14654 non-null  int64
16   img_size_far_img_height 14654 non-null  int64
17   score_like_near_img  13593 non-null  float64
18   score_like_far_img   13593 non-null  float64
19   score_topis_near_img  13593 non-null  float64
...
75%      1.000000
max      101.000000

[8 rows x 19 columns]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

	api_hash	user_name	file_token	service	summary_code	summary_status	summary_desc	http_code	total_time_ms	date
0	02bc5e7	user_name_1	HLqBPI2M...	main	1400	ok	antispoofing OK	200	2204	2025-01-08T22:10:36.542487-05:00
1	02bc5e7	user_name_1	DQmVCFHA	main	1400	ok	antispoofing OK	200	1837	2025-01-08T22:14:12.635331-05:00
2	02bc5e7	user_name_1	SRlpp7e4Q	main	1400	ok	antispoofing OK	200	1936	2025-01-08T22:16:10.768124-05:00
3	02bc5e7	user_name_1	CaISZGg2...	main	1400	ok	antispoofing OK	200	1640	2025-01-08T22:16:14.887930-05:00
4	02bc5e7	user_name_1	7RUJ4Frag	main	302	error	blur/haze	200	919	2025-01-08T22:28:01.054463-05:00

## pgAdmin4 query access

pgAdmin 4

Welcome etl\_db\_final\_workshop/postgres@PostgreSQL 17\*

etl\_db\_final\_workshop/postgres@PostgreSQL 17

Query

```
1 SELECT * FROM etl_staging_table_final_workshop;
```

Data Output Messages Notifications

	api_hash	user_name	file_token	service	summary_code	summary_status	summary_desc	http_code	total_time_ms	date	procesoConvenioGuid	convenio	documento
1	02bc5e7	user_name_1	HLqBPI2M...	main	1400	ok	antispoofing OK	200	2204	2025-01-08T22:10:36.542487-05:00	procesoConvenioGuid_1	convenio_1	documento_1
2	02bc5e7	user_name_1	DQmVCFHA	main	1400	ok	antispoofing OK	200	1837	2025-01-08T22:14:12.635331-05:00	procesoConvenioGuid_2	convenio_1	documento_2
3	02bc5e7	user_name_1	SRlpp7e4Q	main	1400	ok	antispoofing OK	200	1936	2025-01-08T22:16:10.768124-05:00	procesoConvenioGuid_2	convenio_1	documento_2
4	02bc5e7	user_name_1	CaISZGg2...	main	1400	ok	antispoofing OK	200	1640	2025-01-08T22:16:14.887930-05:00	procesoConvenioGuid_2	convenio_1	documento_2
5	02bc5e7	user_name_1	7RUJ4Frag	main	302	error	blur/haze	200	919	2025-01-08T22:28:01.054463-05:00	procesoConvenioGuid_3	convenio_1	documento_3
6	02bc5e7	user_name_1	wyZ2WD...	main	302	error	blur/haze	200	763	2025-01-08T22:28:13.094088-05:00	procesoConvenioGuid_3	convenio_1	documento_3
7	02bc5e7	user_name_1	sWKmZq...	main	302	error	blur/haze	200	699	2025-01-08T22:28:26.157333-05:00	procesoConvenioGuid_3	convenio_1	documento_3
8	02bc5e7	user_name_1	XIZvUaTQ	main	406	error	non-frontal	200	692	2025-01-08T22:41:28.050918-05:00	procesoConvenioGuid_4	convenio_1	documento_4
9	02bc5e7	user_name_1	ztDfYSIG4g	main	302	error	blur/haze	200	925	2025-01-08T22:45:42.796236-05:00	procesoConvenioGuid_5	convenio_1	documento_3
10	02bc5e7	user_name_1	j8aBvYIy	main	203	error	hat	200	572	2025-01-08T22:49:46.404009-05:00	procesoConvenioGuid_6	convenio_1	documento_5
11	02bc5e7	user_name_1	RnMwhL...	main	1400	ok	antispoofing OK	200	1615	2025-01-08T22:49:57.402437-05:00	procesoConvenioGuid_6	convenio_1	documento_5
12	02bc5e7	user_name_1	RR3aypQO...	main	1400	ok	antispoofing OK	200	1730	2025-01-08T22:50:00.843492-05:00	procesoConvenioGuid_6	convenio_1	documento_5
13	02bc5e7	user_name_1	HZDscx6H...	main	302	error	blur/haze	200	805	2025-01-08T22:51:03.447437-05:00	procesoConvenioGuid_7	convenio_1	documento_3
14	02bc5e7	user_name_1	oXtq3upQ	main	302	error	blur/haze	200	795	2025-01-08T22:51:40.923717-05:00	procesoConvenioGuid_7	convenio_1	documento_3
15	02bc5e7	user_name_1	c1eVQw...	main	1400	ok	antispoofing OK	200	1662	2025-01-08T22:52:49.485790-05:00	procesoConvenioGuid_7	convenio_1	documento_3
16	02bc5e7	user_name_1	KAXvJDRL...	main	1400	ok	antispoofing OK	200	1630	2025-01-08T22:53:09.765574-05:00	procesoConvenioGuid_7	convenio_1	documento_3
17	02bc5e7	user_name_1	X3l8MAdc...	main	1400	ok	antispoofing OK	200	1667	2025-01-08T22:54:52.561405-05:00	procesoConvenioGuid_8	convenio_1	documento_3
18	02bc5e7	user_name_1	6-LVmx4...	main	1400	ok	antispoofing OK	200	1654	2025-01-08T22:54:55.947408-05:00	procesoConvenioGuid_8	convenio_1	documento_3

Total rows: 14654 Query complete 00:00:00.596 CRLF Ln 2, Col 1

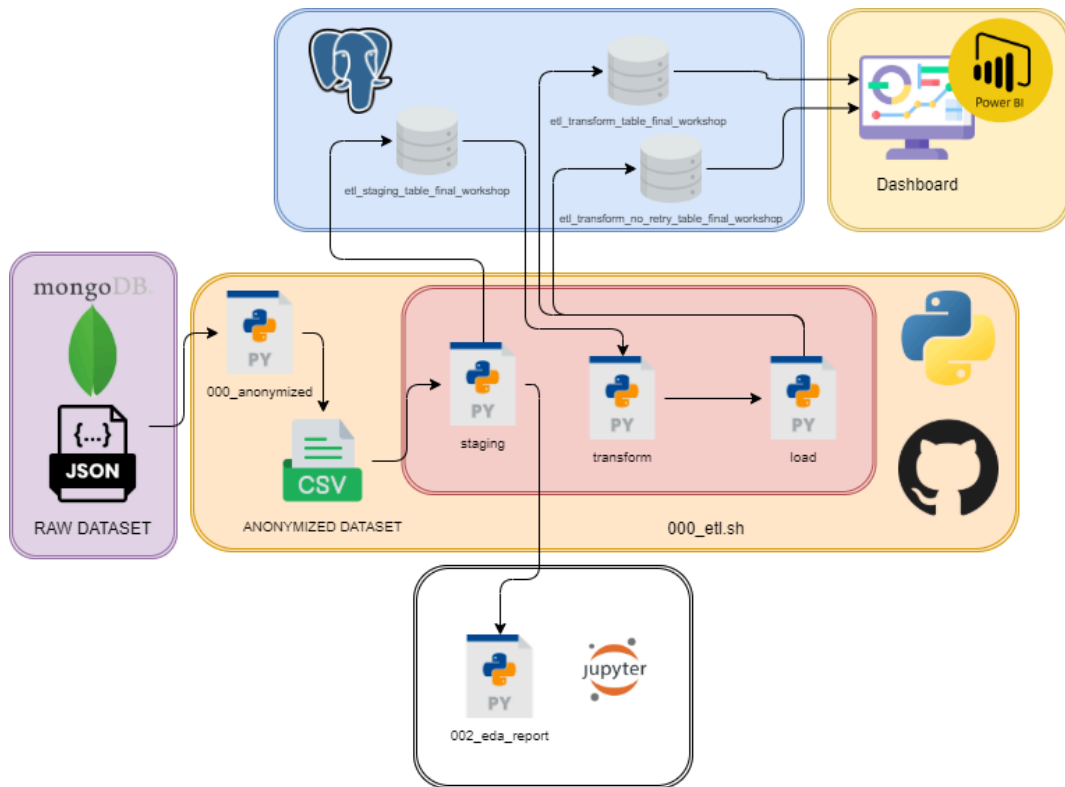
Transform Stage Evidence after executing **002\_eda\_transform.ipynb** also the EDA is available in this EDA report also executing **002\_eda\_report.ipynb** the EDA reports for:

- [EDA etl staging table final workshop](#)
- [EDA etl transform table final workshop](#)
- [EDA etl transform no retries table final workshop](#)

## ETL WORKFLOW

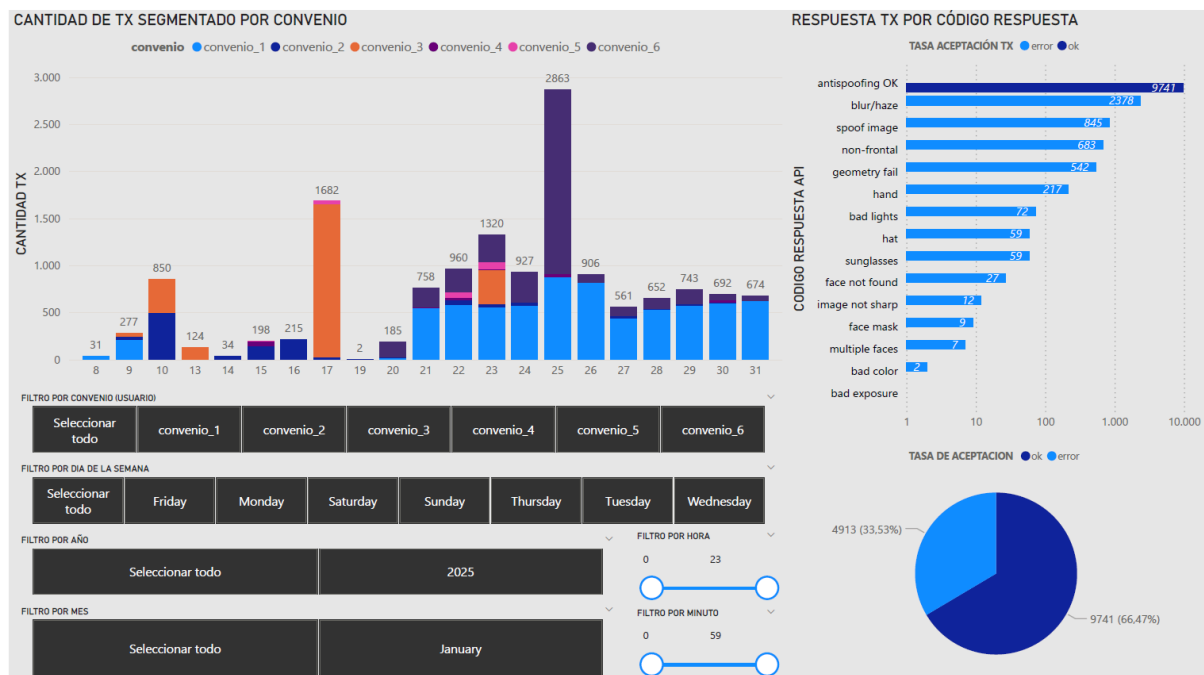
The ETL workflow is executed running the bash script 000\_etl.sh it launches a python script 000\_etl.py where all stages described before were automatically launched, it executes the staging, transform and load steps saving the output db ready to visualize are etl\_transform\_table\_final\_workshop and etl\_transform\_table\_final\_workshop

The



## DASHBOARD TABS

### USER CLIENT INSIGHTS TAB

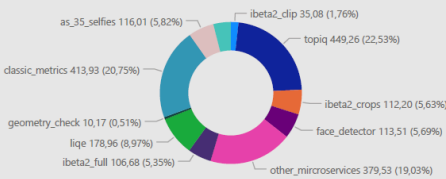


## MICROSERVICE API TIME EXECUTION INSIGHTS TAB

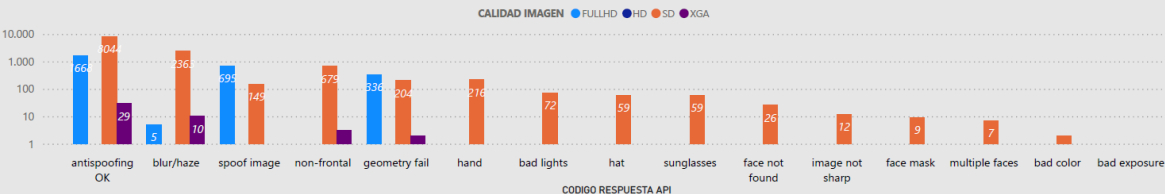
PROMEDIO TIEMPO EJECUCION POR MICROSERVICIO

MICROSERVICIOS

- ibeta2\_clip
- topiq
- ibeta2\_crops
- face\_detector
- other\_microservices
- ibeta2\_full
- liqe



RESPUESTA TX POR CÓDIGO RESPUESTA SEGMENTADO POR CALIDAD DE IMAGEN



RESPUESTA TX POR CÓDIGO RESPUESTA SEGMENTADO POR RELACIÓN DE ASPECTO DE IMAGEN

