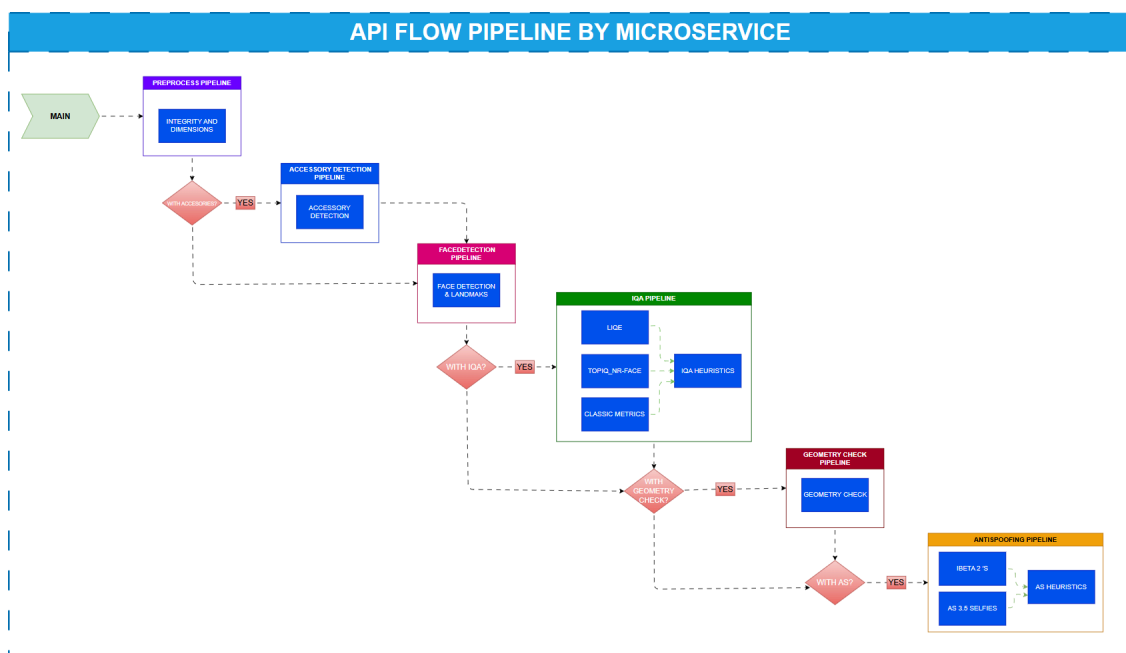


ETL Final Workshop - Manuel Henao - [Github Repo](#)

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_desc', 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

Evidences

Extract Stage Evidence after executing 001_staging.ipynb

```
Loading / Reading Staging Table

> <
db.table_name = "tbl_staging_table_final_workshop"
with engine.connect() as conn:
    of_staging = pd.read_sql(f"SELECT * FROM {db.table_name}", conn)

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

```
[1] ✓ 55s Open (f)Staging in Data Wrangler Python
```

```
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34654 entries, 0 to 34653
Data columns (total 79 columns):
 #   Column                                Non-Null Count  Dtype  Dtype
---  --
 0   api_hash                             34654 non-null object object
 1   user_name                            34654 non-null object object
 2   file_token                           34654 non-null object object
 3   service                             34654 non-null object object
 4   summary_code                        34654 non-null int64  int64
 5   summary_status                      34654 non-null object object
 6   summary_desc                        34654 non-null object object
 7   http_code                           34654 non-null int64  int64
 8   total_time_ms                       34654 non-null int64  int64
 9   date                                34654 non-null object object
10   procesoConvenioGuid                34654 non-null object object
11   convenio                           34654 non-null object object
12   documento                           34654 non-null object object
13   img_size_near_img_width            34654 non-null int64  int64
14   img_size_near_img_height           34654 non-null int64  int64
15   img_size_far_img_width              34654 non-null int64  int64
16   img_size_far_img_height             34654 non-null int64  int64
17   score_like_near_img                 13593 non-null float64 float64
18   score_like_far_img                  11915 non-null float64 float64
19   score_cross_near_img                13593 non-null float64 float64
...
77%      1.000000
max      101.000000

[8 rows x 69 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	fb2da7	user_name_1	200a7201g	main	1400	ok	ansuoposimg OK	200	2224	2025-01-08T22:10:39.542487-0500
1	fb2da7	user_name_1	200a7201g	main	1400	ok	ansuoposimg OK	200	1837	2025-01-08T22:14:12.853951-0500
2	fb2da7	user_name_1	5Fapp7a4Q	main	1400	ok	ansuoposimg OK	200	1958	2025-01-08T22:16:10.768124-0500
3	fb2da7	user_name_1	Cx22g2p2g	main	1400	ok	ansuoposimg OK	200	1840	2025-01-08T22:16:14.807950-0500
4	fb2da7	user_name_1	7Pnqg7a5g	main	1001	error	hour:hour	200	879	2025-01-08T22:20:07.054480-0500

pgAdmin4 query access

pgAdmin 4

Welcome etl_db_final_workshop/postgres@PostgreSQL 17*

etl_db_final_workshop/postgres@PostgreSQL 17

Query Query History

Scratch Pad x

1 SELECT * FROM etl_staging_table_final_workshop;

2

Data Output Messages Notifications

Showing rows: 1 to 10000 Page No: 1 of 2

	api_hash	user_name	file_token	service	summary_code	summary_status	summary_desc	http_code	total_time_ms	date	procesoConvenioGuid	convenio	documento
	text	text	text	text	integer	text	text	integer	integer	text	text	text	text
1	02bc5e7	user_name_1	HLqbP2GM...	main	1400	ok	antispoofting OK	200	2204	2025-01-08T22:10:36.542467-05:00	procesoConvenioGuid_1	convenio_1	documento_1
2	02bc5e7	user_name_1	DQktwYCP...	main	1400	ok	antispoofting OK	200	1837	2025-01-08T22:14:12.653531-05:00	procesoConvenioGuid_2	convenio_1	documento_2
3	02bc5e7	user_name_1	SRlppj7e4Q	main	1400	ok	antispoofting OK	200	1936	2025-01-08T22:16:10.768124-05:00	procesoConvenioGuid_2	convenio_1	documento_2
4	02bc5e7	user_name_1	Csl5ZGg2...	main	1400	ok	antispoofting OK	200	1640	2025-01-08T22:16:14.887930-05:00	procesoConvenioGuid_2	convenio_1	documento_2
5	02bc5e7	user_name_1	7RUq4FLag	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	wyZZWD...	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	sWkm2qh...	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	XIZzVUa7Q	main	406	error	non-frontal	200	692	2025-01-08T22:41:38.050918-05:00	procesoConvenioGuid_4	convenio_1	documento_4
9	02bc5e7	user_name_1	zIDYISIG4g	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	j8aEbuYlg	main	203	error	hat	200	572	2025-01-08T22:49:46.404009-05:00	procesoConvenioGuid_5	convenio_1	documento_5
11	02bc5e7	user_name_1	RnMwhL...	main	1400	ok	antispoofting OK	200	1615	2025-01-08T22:49:57.402437-05:00	procesoConvenioGuid_6	convenio_1	documento_5
12	02bc5e7	user_name_1	R3aypPO...	main	1400	ok	antispoofting 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	okxtq3upQ	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	clEvGwqr...	main	1400	ok	antispoofting OK	200	1662	2025-01-08T22:52:49.485790-05:00	procesoConvenioGuid_7	convenio_1	documento_3
16	02bc5e7	user_name_1	KAXqDRL...	main	1400	ok	antispoofting OK	200	1630	2025-01-08T22:53:09.765574-05:00	procesoConvenioGuid_7	convenio_1	documento_3
17	02bc5e7	user_name_1	X3ltaMAd...	main	1400	ok	antispoofting OK	200	1667	2025-01-08T22:54:52.561405-05:00	procesoConvenioGuid_8	convenio_1	documento_3
18	02bc5e7	user_name_1	6-Lvmex4...	main	1400	ok	antispoofting 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](#)