## Technical Test (Take Home Challenge)

The goal of this technical test is to showcase an end-to-end ELT pipeline from a data source to any data warehouse using Python, SQL, dbt (data-build tool) or Airflow, and data models to answer the following questions and provide some insights with them.

* Top 10 stores per transacted amount.
* Top 10 products sold.
* Average transacted amount per store typology and country.
* Percentage of transactions per device type.
* Average time for a store to perform its 5 first transactions.

**Deliverables:**

You have to **choose one** of the following options, the one that better fits your abilities.

* **Option 1:** Provide a project using Python, SQL & dbt ([dbt-core](https://handbook.gitlab.com/handbook/business-technology/data-team/platform/dbt-guide/)). With all the necessary files to run the dbt project. Answering all the questions using models from dbt. One model per question.
  + **Hints:**
    - [dbt-postgresql](https://medium.com/israeli-tech-radar/first-steps-with-dbt-over-postgres-db-f6b350bf4526)
    - [dbt-mysql](https://docs.getdbt.com/docs/core/connect-data-platform/mysql-setup)
* **Option 2:** Provide a project using Python, SQL & Airflow ([mwaa-local-env](https://github.com/aws/aws-mwaa-local-runner)) using Docker, the project should include the Dockerfile configuration to run it locally. You can answer the questions using SQL (*sql-alchemy package)*.
* **Option 3:** Provide a project using Python & SQL, provide the answers to the questions, and as a handicap, consume the UF value from the [Banco Central de Chile API](https://si3.bcentral.cl/estadisticas/Principal1/Web_Services/index.htm) and store it into a table, using *request* package, think of it a is will run daily to capture the UF daily value on each run.

Please share the source code, data models, scripts, developed to answer the above questions via public **git repository including a README** file explaining your assumptions, design and solution implementation details.

**Important:** The repository should contain all the elements to replicate the entire project on our side.

Assumptions:

* Each of our **customers** have one or multiple **stores**, which are physical locations where their business happens, those stores are classified by different typology.
* We also provide them with different types of **devices** (1 to 5), those devices are hardware needed to perform any kind of transactions, each of them is tied up to a specific store.
* A **transaction** is a payment made using the provided devices, currently our devices only handle payments made by card and in euros. Those transactions are made to pay for products sold inside the store, each **product** has a name and a SKU (stock keeping unit) which is unique.
* Every day, our customer’s customers walk into their favorite stores to buy products and pay for them using the devices that we provide.

We now want to target customers that will use our devices efficiently and benefit the most from our product. That’s why we need to answer the above questions to know which stores, products and devices are the most efficient and to know how long it takes for a store to adopt our devices.

In order to solve this problem, we have provided the following three datasets in csv format

* Stores
* Devices
* Transactions

\* The dataset supplied with this test contains only sample data. Your design and implementation should scale for larger volumes of data (millions to billions records)