# Workshop 1

Presented by:

Juan Camilo Buitrago Gonzalez

ETL

Teacher:

Javier Alejandro Vergara Zorrilla

Universidad Autónoma de Occidente

08/03/2024

# Introduction:

The purpose of this project is to perform an Extract, Transform, Load (ETL) process on the "candidates.csv" dataset. The dataset contains information about job candidates, and our goal is to enhance it by adding a new column called "hired", which will define whether a candidate was hired or not.

The ETL process consists of three main steps.

Extract: In this step, we will retrieve the data from the "candidates.csv" dataset, which contains various attributes such as first name, last name, email, application date, country, years of experience (YOE), seniority, technology skills, code challenge score, and technical interview score.

Transform: After extracting the data, we will perform transformations to enrich the data. The first transformation is to add a new column called "hired," which indicates whether a candidate was hired based on certain criteria, such as meeting a minimum score threshold in the code challenge and technical interview.

Loading: Once the data has been transformed, we will load it back into a database table, ready for further analysis.

# Technologies used:

The technologies used for this workshop were:

Python: The language used for the workshop

Jupyter notebook: The notebook platform used to make the EDA and other transformations.

Visual Studio Code: The chosen code editor for the workshop management and development

PostgreSQL: The database management system used for storing the candidates data

Power BI: The visualization platform used to make the dashboard.

## Architecture:

My workshop architecture is organized in a simple structure:

- src: This folder contains the two most important files, config.py and main.py.
    - Config.py is the file that contains all the functions necessary to make the program work.
    - Main.py is the file where I implemented all the functions from config.py and where the logic of the database and tables creation is located.
- Data: This folder contains the data I used to work in this workshop:
    - Candidates.csv is the csv that contains the candidate data; this is the data I worked with to create the tables.
    - Config.json is the file that contains all the data needed to connect to the database, such as user, password, db, host and port.
- Docs: This folder contains the two documents needed to explain this workshop:
    - Documentation is the file with all the explanations of the process in this workshop. (this file)
    - Dashboard.pdf is the visualization of the graphs I make in Power BI in pdf format.
- Notebooks: This folder contains the notebooks I used for some of the processes:
    - EDA.ipynb is the exploratory data analysis is used to analyze and investigate data sets and summarize their main characteristics, also using data visualization methods.
    - Transformation.ipynb is used to create a dictionary with all the technologies in the dataset and categorize them for better understanding and visualization.
- .gitignore: is the file with the names of the ignored files (files that are not committed to the github repository).
- README.md: This file contains the explanation of how to install this program and how it works.

# Implementation:

The implementation process begins by running the main.py file in the src folder. This file takes care of the creation of the database, the connection, and the creation of the "candidates" and "candidates_hired" tables, as well as the logic for adding a column called "hired".

Then you need to run the notebook called Transformation.ipynb to create a column called category_of_technology that contains the categorization of the technologies.

Finally, run the EDA.ipynb notebook, which contains the exploratory data analysis, which gives a description of what the data is, how it is structured, and some graphs to visualize what the data can tell us.

# Data information

I have 50k rows of data about candidates. The fields we will use are:

- First Name
- Last Name
- Email
- Country
- Application Date
- Yoe (years of experience)
- Seniority
- Technology
- Code Challenge Score
- Technical Interview

I consider a candidate HIRED when he has both scores greater than or equal to 7.

# Exploratory Data Analysis (EDA)

In this EDA we can know the next info of the data:

- Name of columns: after execute the main.py and transformation notebook, we have new columns, so these are the columns:

- First name
- Last name
- Email
- Application Date
- Country
- YOE (Years Of Experience)
- Seniority
- Technology
- Code Challenge Score
- Technical Interview Score
- Hired
- Category_of_technology

- The describe of the numeric columns: We can look at the describe returns descriptive statistics including: mean, median, max, min, std and counts for the numeric and date columns.

| | Application Date | YOE | Code Challenge Score | Technical Interview Score |
|---|---|---|---|---|
| count | 50000 | 50000.000000 | 50000.000000 | 50000.000000 |
| mean | 2020-04-03 23:04:14.592000 | 15.286980 | 4.996400 | 5.003880 |
| min | 2018-01-01 00:00:00 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 2019-02-17 00:00:00 | 8.000000 | 2.000000 | 2.000000 |
| 50% | 2020-04-06 00:00:00 | 15.000000 | 5.000000 | 5.000000 |
| 75% | 2021-05-21 00:00:00 | 23.000000 | 8.000000 | 8.000000 |
| max | 2022-07-04 00:00:00 | 30.000000 | 10.000000 | 10.000000 |
| std | NaN | 8.830652 | 3.166896 | 3.165082 |

- Data types: In this cell we can see the types of data, in this case the most important for me are the new columns: "hired", this is the column that tells us what are the people that have been hired or not with a boolean data. Also we have the column "category_of_technology" that is for categorizing the column technology because there are a lot of types of technology, so here we have this new column for a data visualization more clear.

```
First Name                          object
Last Name                           object
Email                               object
Application Date            datetime64[ns]
Country                             object
YOE                                  int64
Seniority                           object
Technology                          object
Code Challenge Score                 int64
Technical Interview Score            int64
hired                                 bool
category_of_technology              object
dtype: object
```

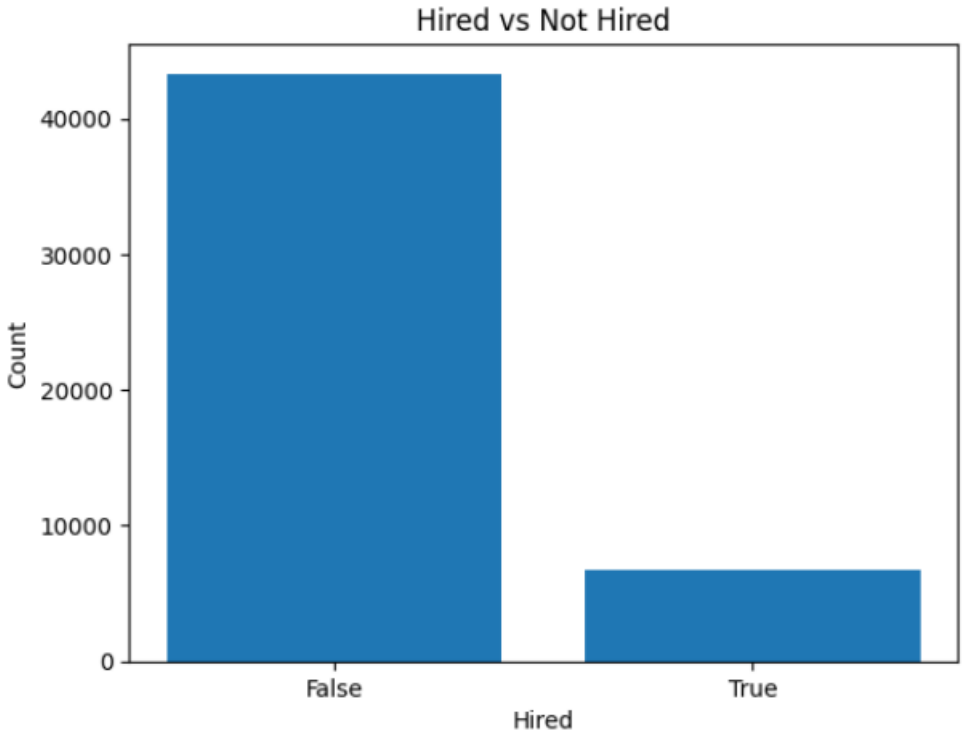
- Null values: We can see that the dataset does not have any null values.

```
First Name                    0
Last Name                     0
Email                         0
Application Date              0
Country                       0
YOE                           0
Seniority                     0
Technology                    0
Code Challenge Score          0
Technical Interview Score     0
hired                         0
category_of_technology        0
dtype: int64
```

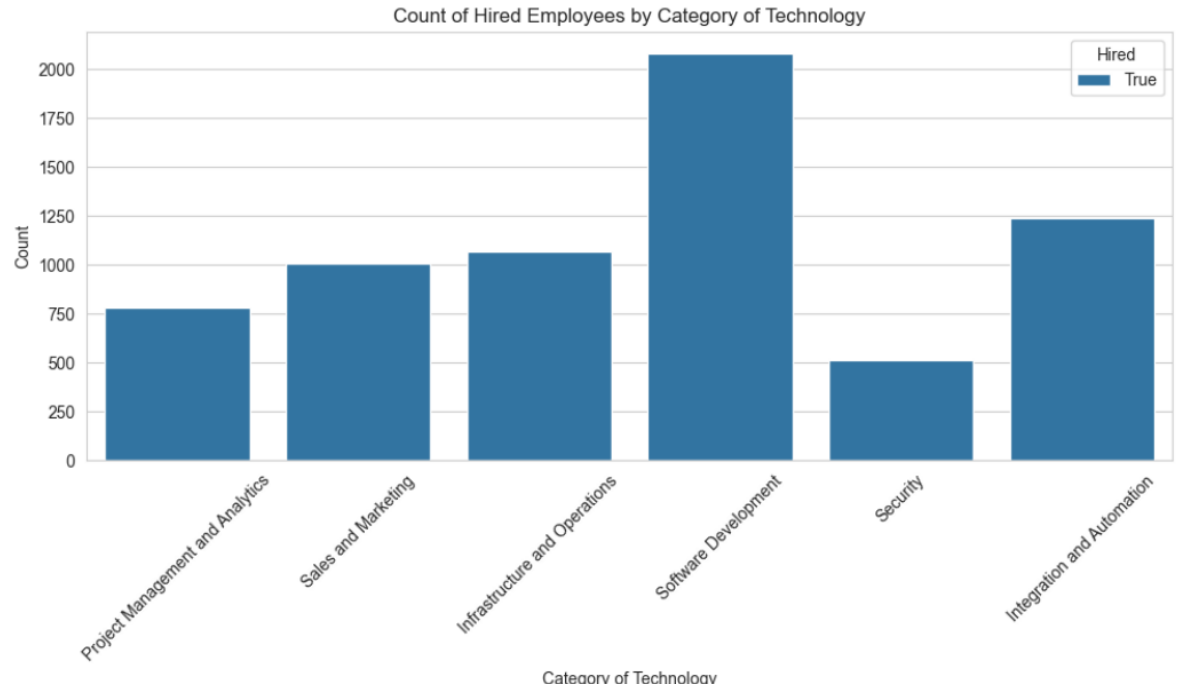
- Number of rows and columns:
  Number of rows: 50000
  Number of columns: 12

- Duplicated values: There are not duplicated values.

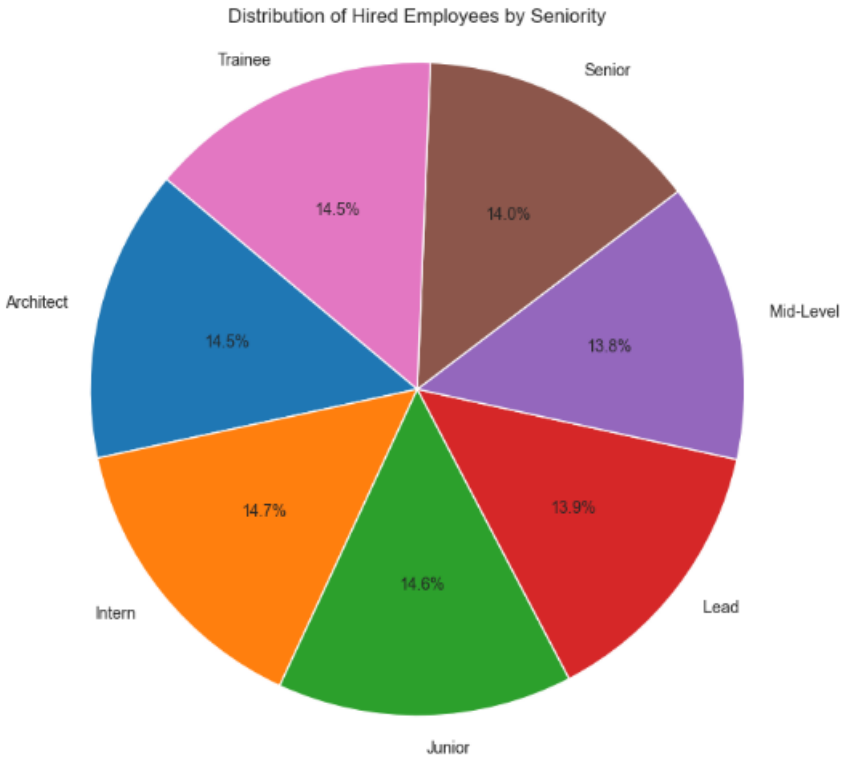
## Graphics




- Here we can see that the 'not hired' are more than the 'hired', this could tell us that being 'hired' is harder than being 'not hired'.



- Here we can see that the most 'hired' are the 'Software Development', this can tell us that the category of technology with the most possibilities to get 'hired' is the 'Software Development'.




- In this graph we have the distribution of hired by seniority, in this case we do not have a big difference between seniorities, but it is interesting that "Junior" has more hired than "Senior", which is a greater level than "Junior".

## Conclusions:

- This was a great challenge to improve my programming skills, I know that I have some skills that I need to improve if I want to be a good Data and IA engineer, but I think I know that I did this workshop with my actual skills and with this I can noticed what are my pros and my weakness.
- About the workshop: In this dataset I have some conclusions.
    - The 2022 has not the enough data to get conclusions, so this year is not taking account in the analysis.
    - I noticed that in a senior vs junior taking account the count of hired, the junior candidates are the most hired than senior candidates, this is

curious because senior is a level of software development bigger than junior.

- The quantity of not hired candidates is very large, this can demonstrate the lower percentage of hired candidates in comparison with the total candidates in this range of time.