WORKSHOP - 001

Paola Andrea Chaux Campo - 2220225

ETL (Extracción, Transformación y carga)

Docente:
Javier Alejandro Vergara

Ingeniería de Datos e Inteligencia Artificial
Facultad de Ingeniería
Universidad Autónoma De Occidente
2024

Documentation Workshop 001

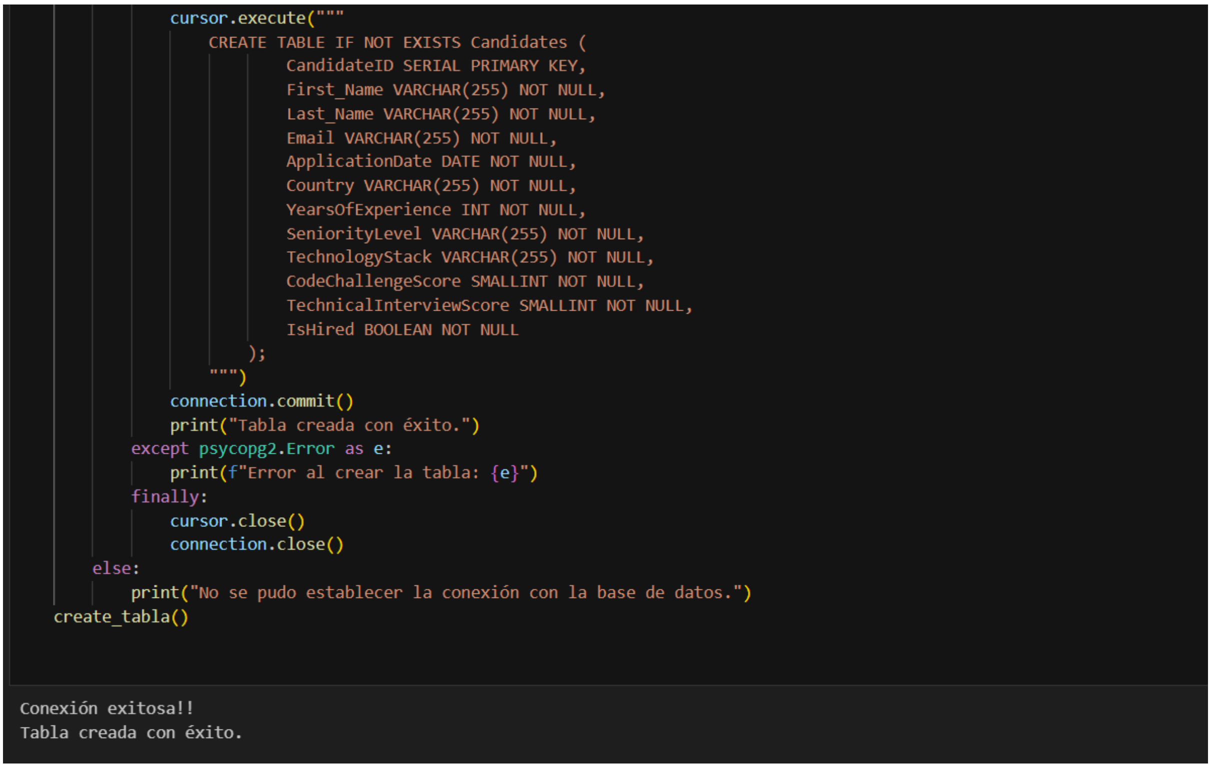
Step 1:

A connection to postgres was created, to establish the connection to the database, it is necessary to have a file named "db_config.json", that contains your database credentials in json format, a way to password and work safer. This file should include: "localhost" for the server address, "user" for the username, "password" for the password, and "database" for the specific database that you will use and you intend to access.

The challenge was done by importing modules: psycopg2 and pandas for the connection for its simplicity and ease of handling.

Step 2:

The table was created, creating a function declaring the columns and column types we will use. We also take into account a requirement of the challenge and create at once a column called IsHired to be efficient.



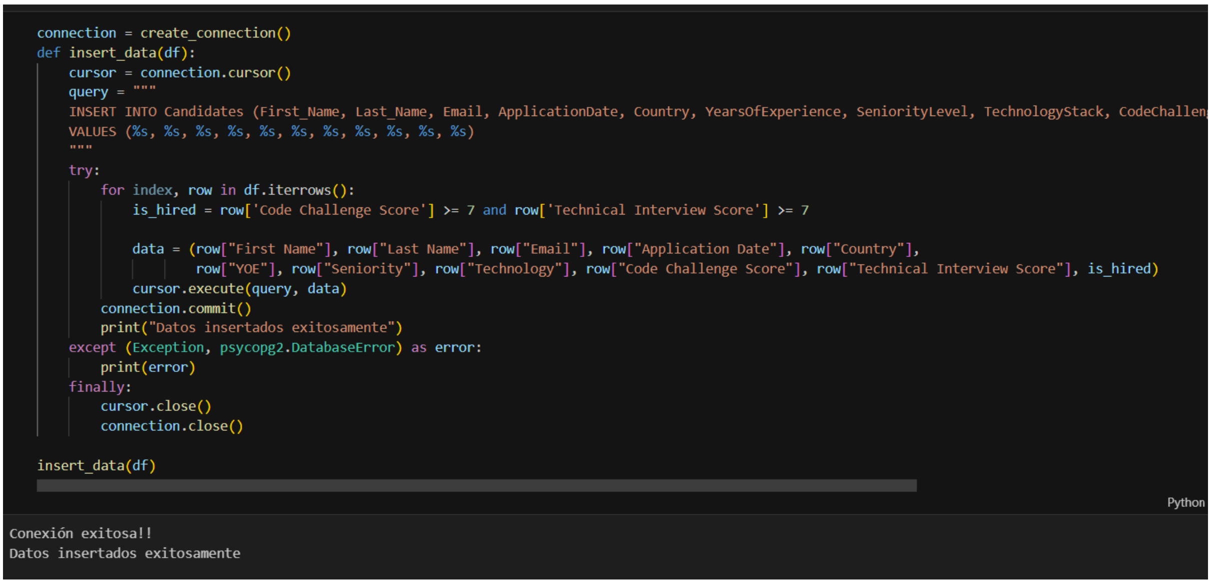
```
            cursor.execute("""
                CREATE TABLE IF NOT EXISTS Candidates (
                        CandidateID SERIAL PRIMARY KEY,
                        First_Name VARCHAR(255) NOT NULL,
                        Last_Name VARCHAR(255) NOT NULL,
                        Email VARCHAR(255) NOT NULL,
                        ApplicationDate DATE NOT NULL,
                        Country VARCHAR(255) NOT NULL,
                        YearsOfExperience INT NOT NULL,
                        SeniorityLevel VARCHAR(255) NOT NULL,
                        TechnologyStack VARCHAR(255) NOT NULL,
                        CodeChallengeScore SMALLINT NOT NULL,
                        TechnicalInterviewScore SMALLINT NOT NULL,
                        IsHired BOOLEAN NOT NULL
                    );
                """)
            connection.commit()
            print("Tabla creada con éxito.")
        except psycopg2.Error as e:
            print(f"Error al crear la tabla: {e}")
        finally:
            cursor.close()
            connection.close()
    else:
        print("No se pudo establecer la conexión con la base de datos.")
create_tabla()
```

```
Conexión exitosa!!
Tabla creada con éxito.
```

Step 3:

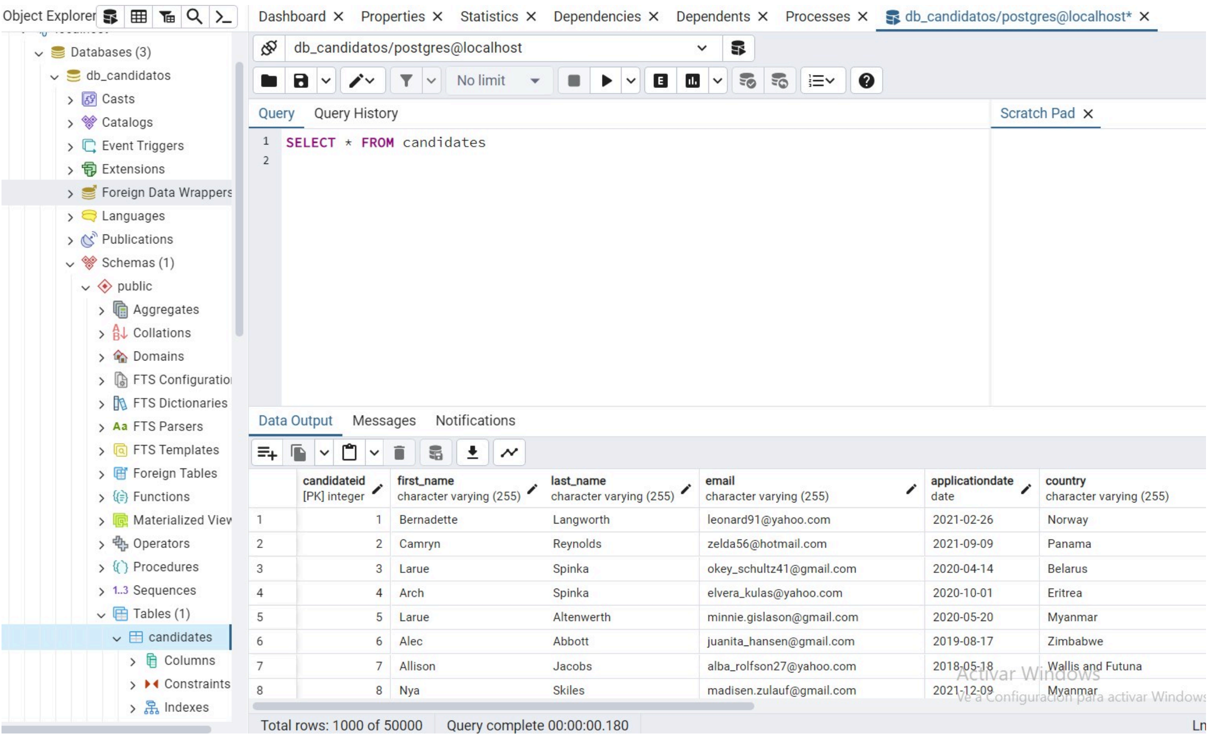
We inserted the data from the csv file after having read and put in a dataframe to the csv file, after the EDA and after having verified that no special cleaning was required because it had no null data or any problem caused we entered the data.

Here we also give the parameters to fill the IsHired column, to tell us if the candidate is hired or not, we designate it as boolean so it appears as TRUE or FALSE. The parameters would be: Code Challenge Score and Technical Interview Score both greater than 7.



```python
connection = create_connection()
def insert_data(df):
    cursor = connection.cursor()
    query = """
    INSERT INTO Candidates (First_Name, Last_Name, Email, ApplicationDate, Country, YearsOfExperience, SeniorityLevel, TechnologyStack, CodeChallen
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    try:
        for index, row in df.iterrows():
            is_hired = row['Code Challenge Score'] >= 7 and row['Technical Interview Score'] >= 7

            data = (row["First Name"], row["Last Name"], row["Email"], row["Application Date"], row["Country"],
                    row["YOE"], row["Seniority"], row["Technology"], row["Code Challenge Score"], row["Technical Interview Score"], is_hired)
            cursor.execute(query, data)
        connection.commit()
        print("Datos insertados exitosamente")
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        cursor.close()
        connection.close()

insert_data(df)
```

```
Conexión exitosa!!
Datos insertados exitosamente
```



```sql
SELECT * FROM candidates
```

	candidateid [PK] integer	first_name character varying (255)	last_name character varying (255)	email character varying (255)	applicationdate date	country character varying (255)
1	1	Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway
2	2	Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama
3	3	Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus
4	4	Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea
5	5	Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar
6	6	Alec	Abbott	juanita_hansen@gmail.com	2019-08-17	Zimbabwe
7	7	Allison	Jacobs	alba_rolfson27@yahoo.com	2018-05-18	Wallis and Futuna
8	8	Nya	Skiles	madisen.zulauf@gmail.com	2021-12-09	Myanmar

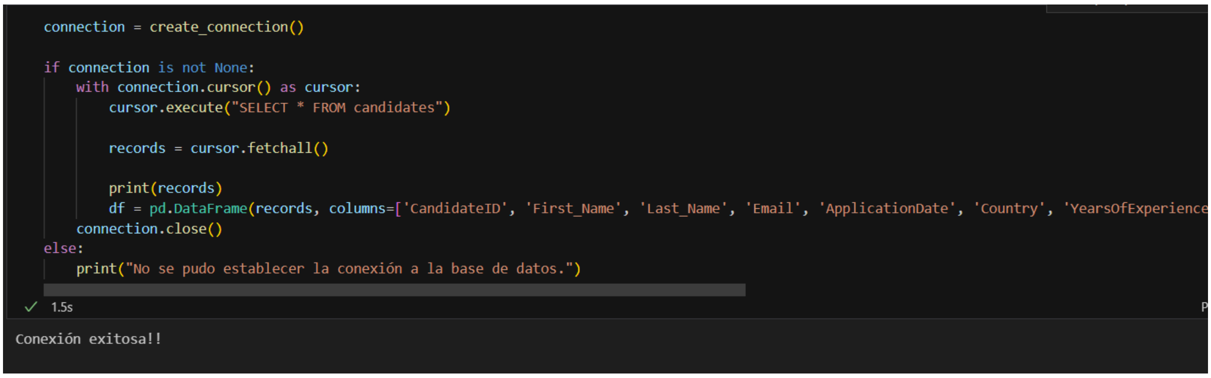
Total rows: 1000 of 50000 Query complete 00:00:00.180

Step 4:

EDA, the respective exploratory data analysis was carried out and the following conclusions were reached:

Something important to clarify was that the db_connection.ipynb file was converted to a .py file to make use of the functions that were determined in this file and thus be able to access more easily to the function to establish the connection to the database.

Now, we have a direct dataframe from the database.

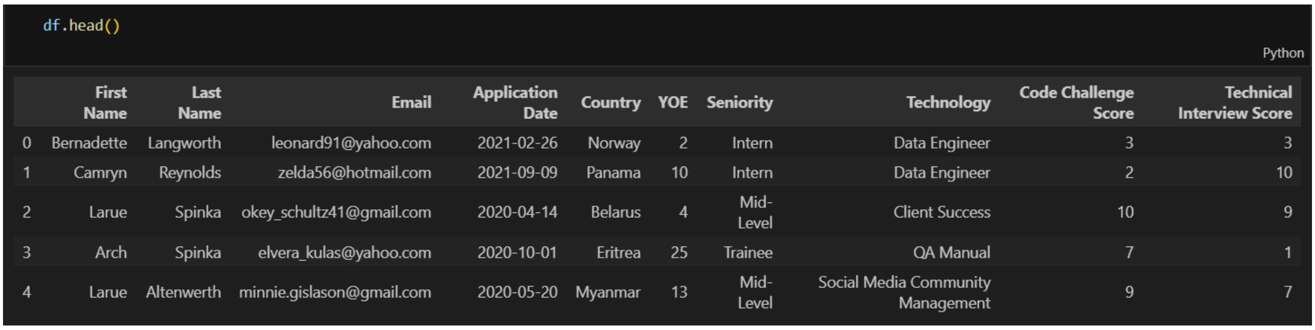


```
connection = create_connection()

if connection is not None:
    with connection.cursor() as cursor:
        cursor.execute("SELECT * FROM candidates")

        records = cursor.fetchall()

        print(records)
        df = pd.DataFrame(records, columns=['CandidateID', 'First_Name', 'Last_Name', 'Email', 'ApplicationDate', 'Country', 'YearsOfExperience
    connection.close()
else:
    print("No se pudo establecer la conexión a la base de datos.")
```

✓ 1.5s

```
Conexión exitosa!!
```

Examine the dimensions of the dataframe to ensure it contains the expected data volume and if it is corroborated to be of the expected size (50,000 rows, 10 columns).

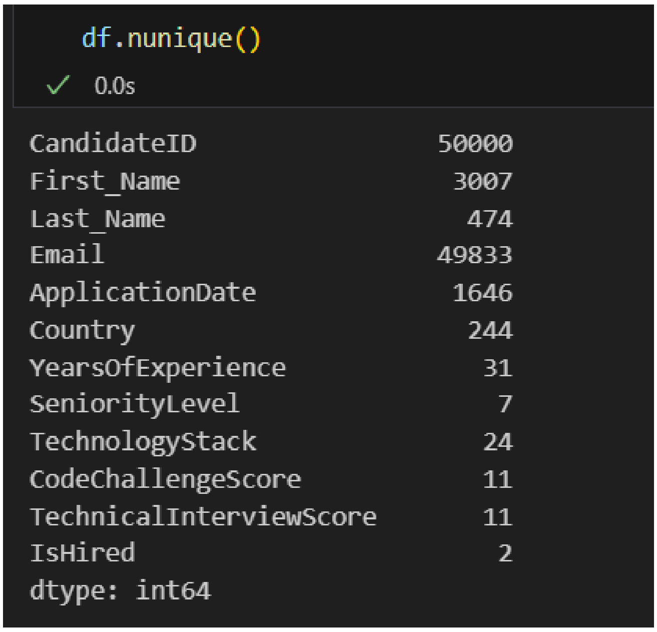


```
df.head()
```

Python

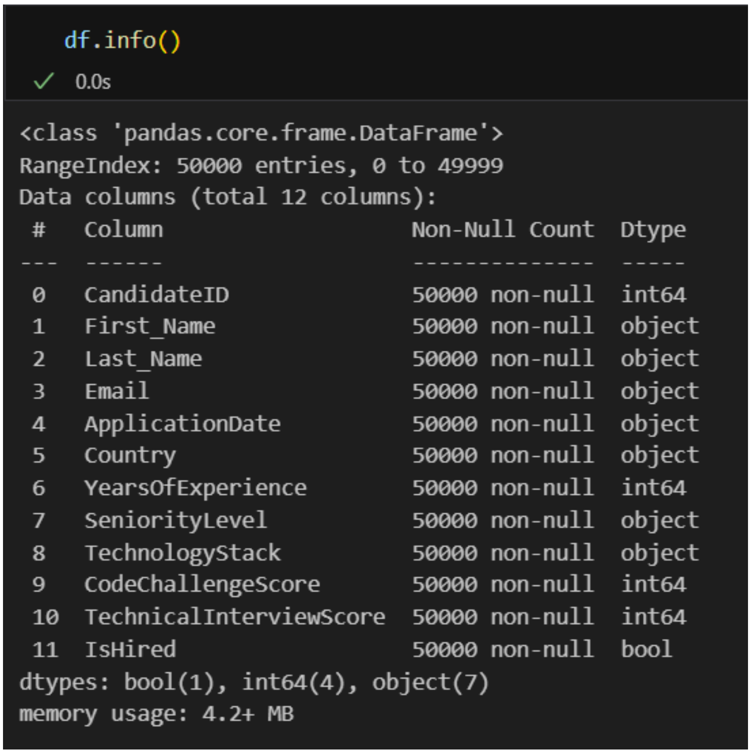
	First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code Challenge Score	Technical Interview Score
0	Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	3	3
1	Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	2	10
2	Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	10	9
3	Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	7	1
4	Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	9	7

Here is a visualization of the dataframe loaded with some data of the total.



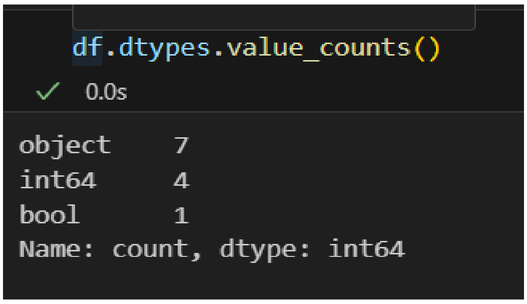
```
df.nunique()
✓ 0.0s

CandidateID                 50000
First_Name                   3007
Last_Name                     474
Email                       49833
ApplicationDate              1646
Country                       244
YearsOfExperience              31
SeniorityLevel                  7
TechnologyStack                24
CodeChallengeScore             11
TechnicalInterviewScore        11
IsHired                         2
dtype: int64
```

Now, we can see the unique values for columns, we have values that are repeated mostly in the categorical or selection columns where the people are included.



```
df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   CandidateID              50000 non-null  int64 
 1   First_Name               50000 non-null  object
 2   Last_Name                50000 non-null  object
 3   Email                    50000 non-null  object
 4   ApplicationDate          50000 non-null  object
 5   Country                  50000 non-null  object
 6   YearsOfExperience        50000 non-null  int64 
 7   SeniorityLevel           50000 non-null  object
 8   TechnologyStack          50000 non-null  object
 9   CodeChallengeScore       50000 non-null  int64 
 10  TechnicalInterviewScore  50000 non-null  int64 
 11  IsHired                  50000 non-null  bool  
dtypes: bool(1), int64(4), object(7)
memory usage: 4.2+ MB
```

Gives us a result where we see a concise summary of the dataframe, if there are null or empty values, number of data by columns and the type of data.

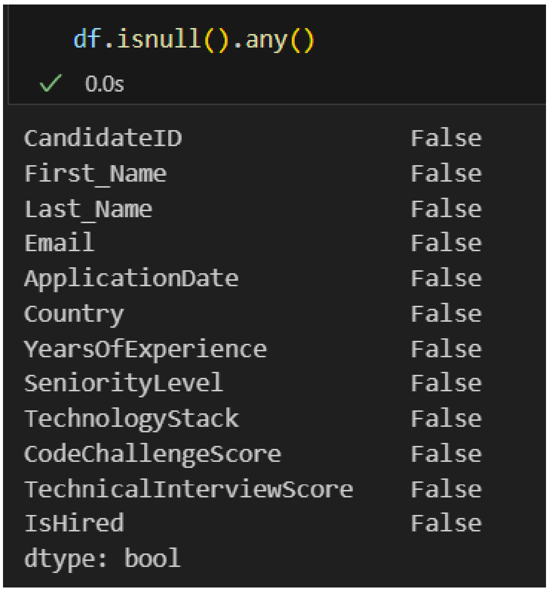


```
df.dtypes.value_counts()
```

✓ 0.0s

```
object    7
int64     4
bool      1
Name: count, dtype: int64
```

We can see that the number of columns with object type are 7 in total, int64 type are 3 and bool 1.

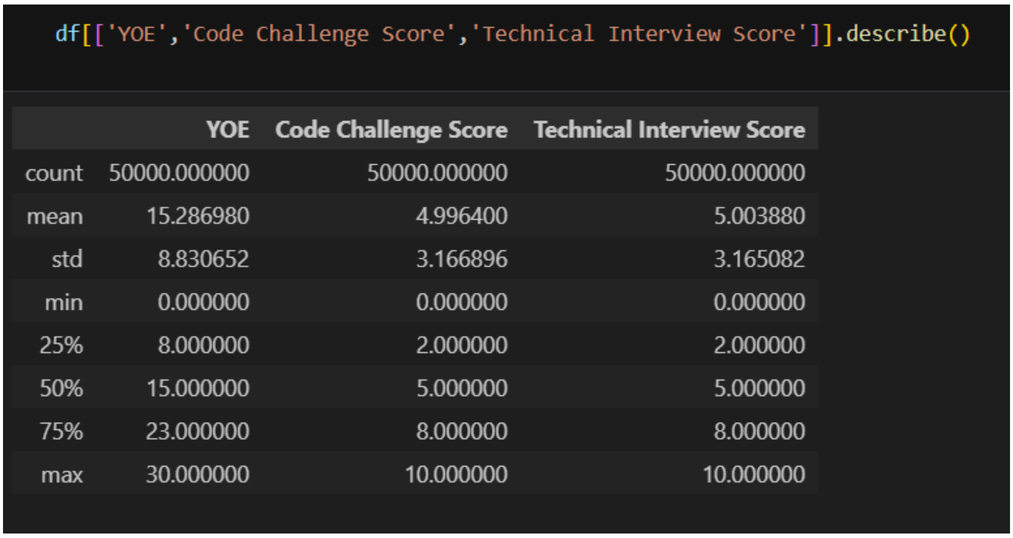


```
df.isnull().any()
```

✓ 0.0s

```
CandidateID               False
First_Name                False
Last_Name                 False
Email                     False
ApplicationDate           False
Country                   False
YearsOfExperience         False
SeniorityLevel            False
TechnologyStack           False
CodeChallengeScore        False
TechnicalInterviewScore   False
IsHired                   False
dtype: bool
```

We make sure that there are no null/empty values in the dataframe and there certainly are not and this is what indicates FALSE results in all columns.



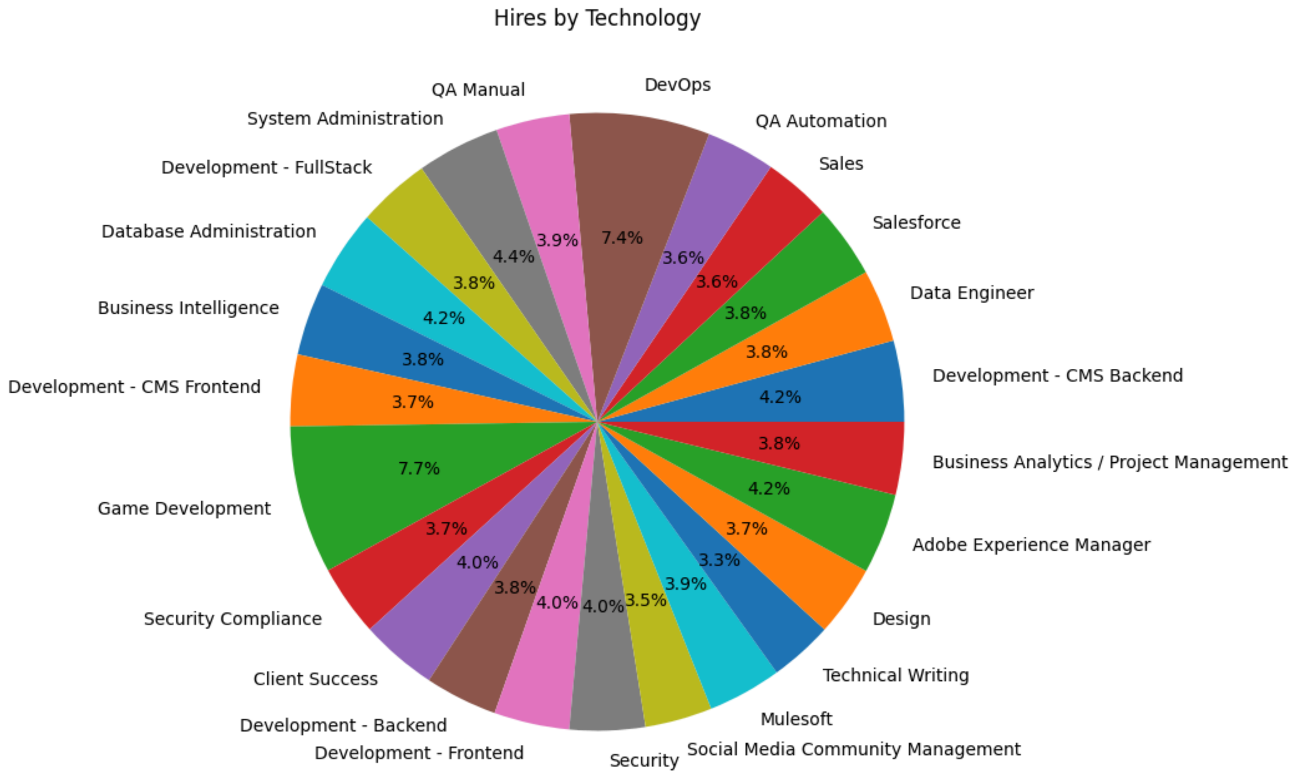
```
df[['YOE','Code Challenge Score','Technical Interview Score']].describe()
```

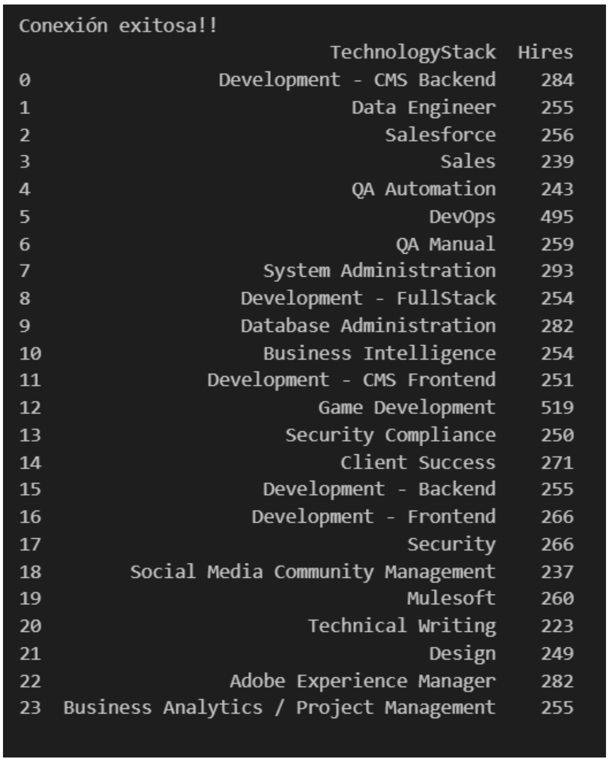
	YOE	Code Challenge Score	Technical Interview Score
count	50000.000000	50000.000000	50000.000000
mean	15.286980	4.996400	5.003880
std	8.830652	3.166896	3.165082
min	0.000000	0.000000	0.000000
25%	8.000000	2.000000	2.000000
50%	15.000000	5.000000	5.000000
75%	23.000000	8.000000	8.000000
max	30.000000	10.000000	10.000000

We provide a statistical summary of three important variables which are: Years Of Experience, Code Challenge Score and Technical Interview Score where we can conclude that the average experience of the candidates is approximately 15 years, indicating a moderately high level of experience overall, the standard deviation is 8.83, indicating a significant variability in the experience of the candidates, with some having very little experience and others having up to 30 years of experience. Fifty percent of the candidates have between 8 and 23 years of experience, indicating a relatively even distribution with the mean. The standard deviation is 3.17, showing moderate variability in the scores.

With all this analysis we can observe that there will be no imputations of any kind since the data that we have are perfect to use in the analysis, then proceed to make the graphs.

Hires by technology (pie chart)

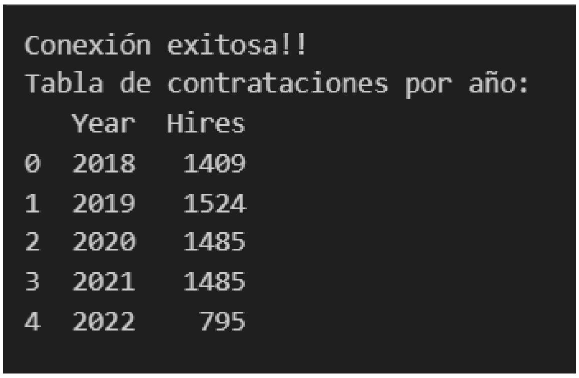




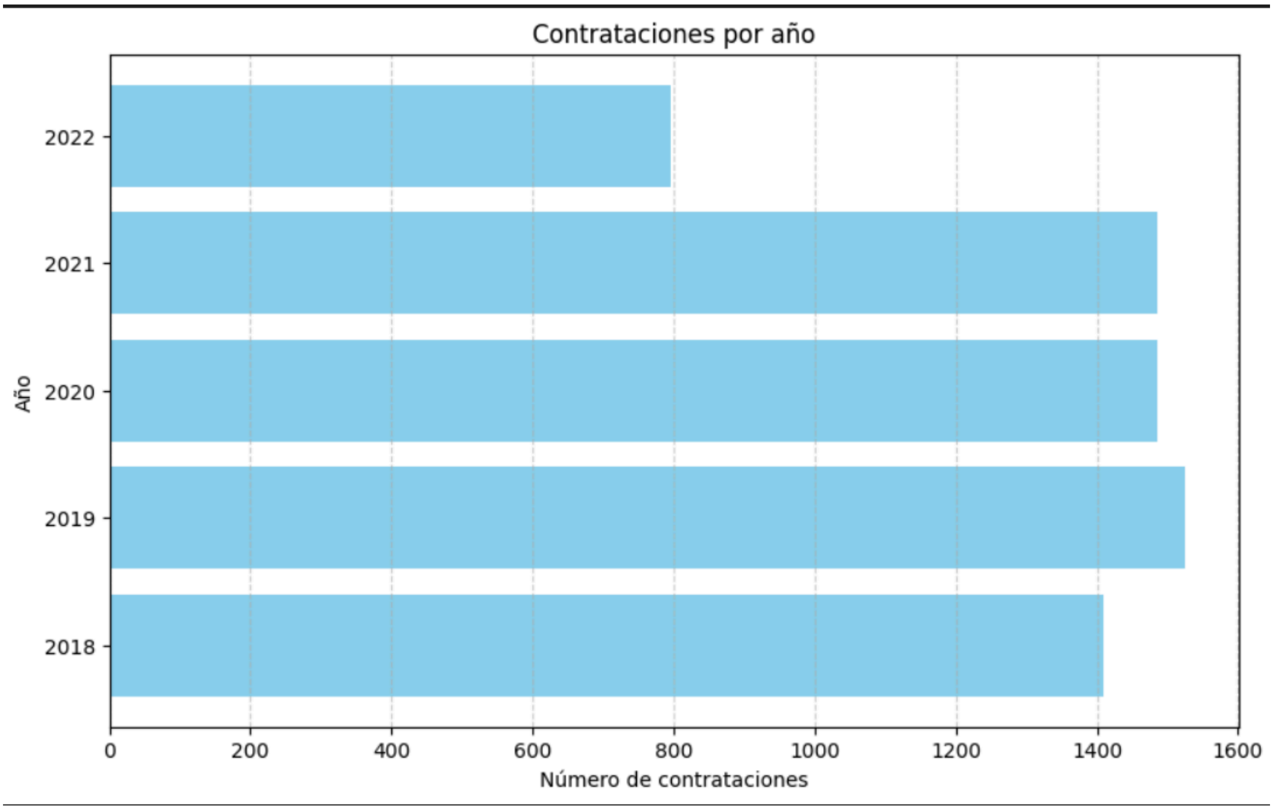
```
Conexión exitosa!!
                              TechnologyStack  Hires
0               Development - CMS Backend    284
1                           Data Engineer    255
2                              Salesforce    256
3                                   Sales    239
4                           QA Automation    243
5                                  DevOps    495
6                               QA Manual    259
7                   System Administration    293
8                   Development - FullStack    254
9                 Database Administration    282
10                  Business Intelligence    254
11              Development - CMS Frontend    251
12                       Game Development    519
13                    Security Compliance    250
14                         Client Success    271
15                  Development - Backend    255
16                 Development - Frontend    266
17                               Security    266
18        Social Media Community Management    237
19                               Mulesoft    260
20                      Technical Writing    223
21                                 Design    249
22                Adobe Experience Manager    282
23  Business Analytics / Project Management    255
```

There is a high demand in specific areas notably as Development Operations and Game Development indicating that they are particularly valued or necessary in the industry compared to Technical Writing and Social Media Community Management that have a lower number of contracts indicating that they are less demanded or have fewer employment opportunities. We can say what analysis can provide for those looking to orient their career or training towards areas with high demand or for companies looking to better understand the labor market in the technology sector.

Hires by year (horizontal bar chart)

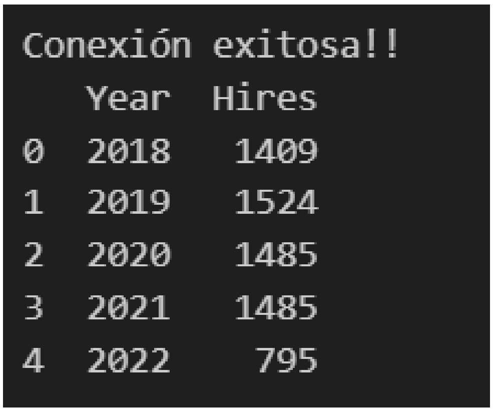


```
Conexión exitosa!!
Tabla de contrataciones por año:
   Year  Hires
0  2018   1409
1  2019   1524
2  2020   1485
3  2021   1485
4  2022    795
```

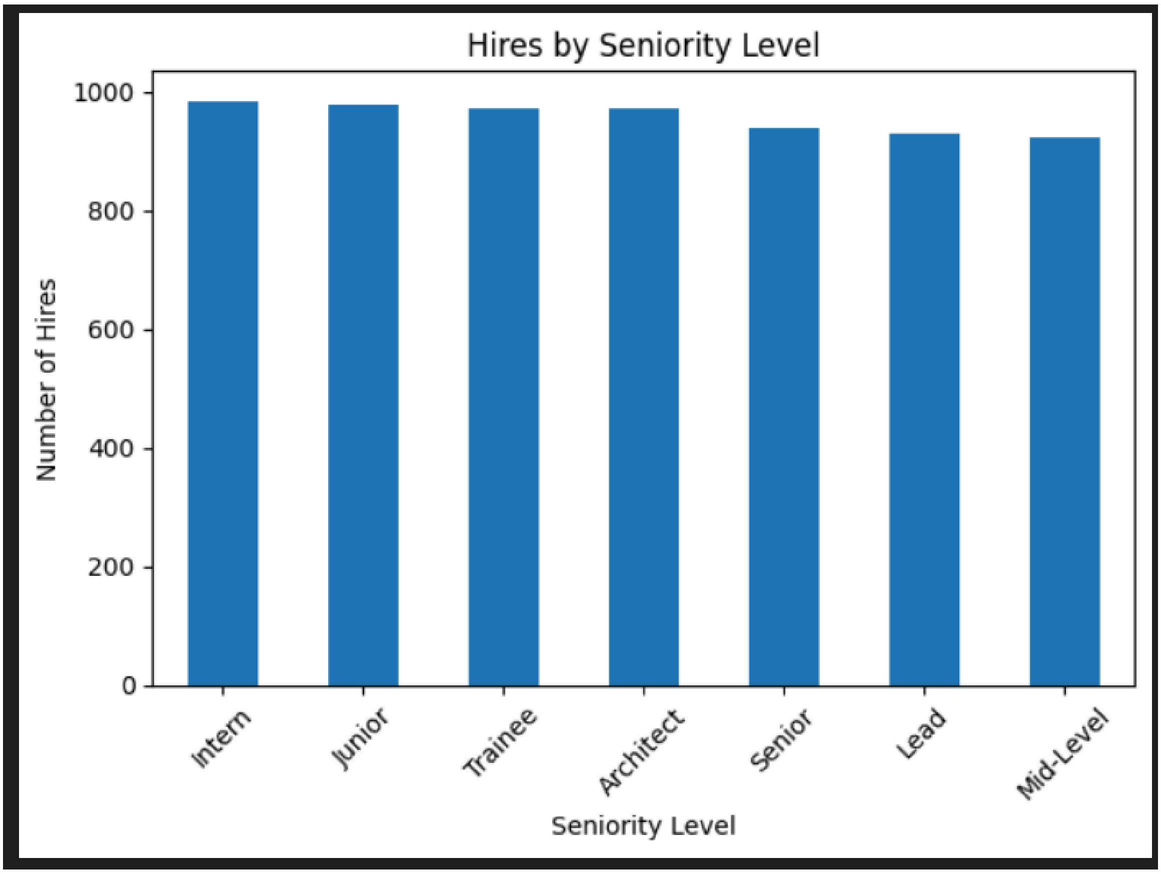


There was growth from 2018 to 2019, indicating a positive trend while the number of hires remained stable during 2020 and 2021 suggesting a break-even.

Hires by seniority (bar chart)

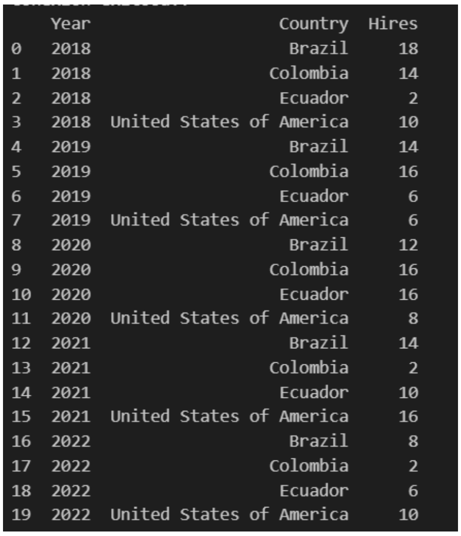


```
Conexión exitosa!!
   Year  Hires
0  2018   1409
1  2019   1524
2  2020   1485
3  2021   1485
4  2022    795
```

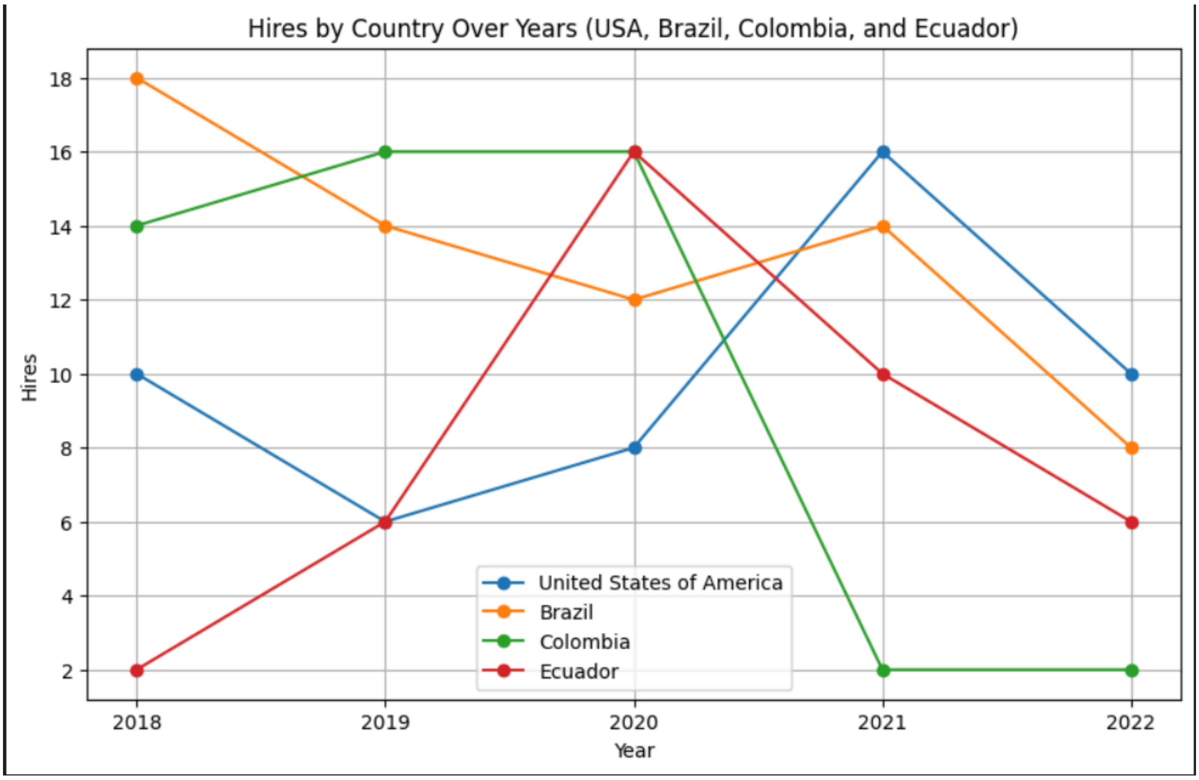


There is a decrease when the seniority level increases, the demand remains consistent at all levels indicating a balanced approach, with this, levels such as Intern, Junior and Trainee have the highest number suggesting high demand in early stages These data suggest that there are significant opportunities when entering the workforce.

Hires by country over years (USA, Brazil, Colombia, and Ecuador only)(multiline chart)



```
    Year                   Country  Hires
0   2018                    Brazil     18
1   2018                  Colombia     14
2   2018                   Ecuador      2
3   2018  United States of America     10
4   2019                    Brazil     14
5   2019                  Colombia     16
6   2019                   Ecuador      6
7   2019  United States of America      6
8   2020                    Brazil     12
9   2020                  Colombia     16
10  2020                   Ecuador     16
11  2020  United States of America      8
12  2021                    Brazil     14
13  2021                  Colombia      2
14  2021                   Ecuador     10
15  2021  United States of America     16
16  2022                    Brazil      8
17  2022                  Colombia      2
18  2022                   Ecuador      6
19  2022  United States of America     10
```



There is notable variability in the number of recruitments between countries over the years. Ecuador shows a significant increase in 2020, while Colombia has a marked decrease in 2021, Brazil shows a decrease from 2018 to 2020, and after a decrease in 2019 and 2020, the United States recovers in 2021, exceeding the levels of previous years.

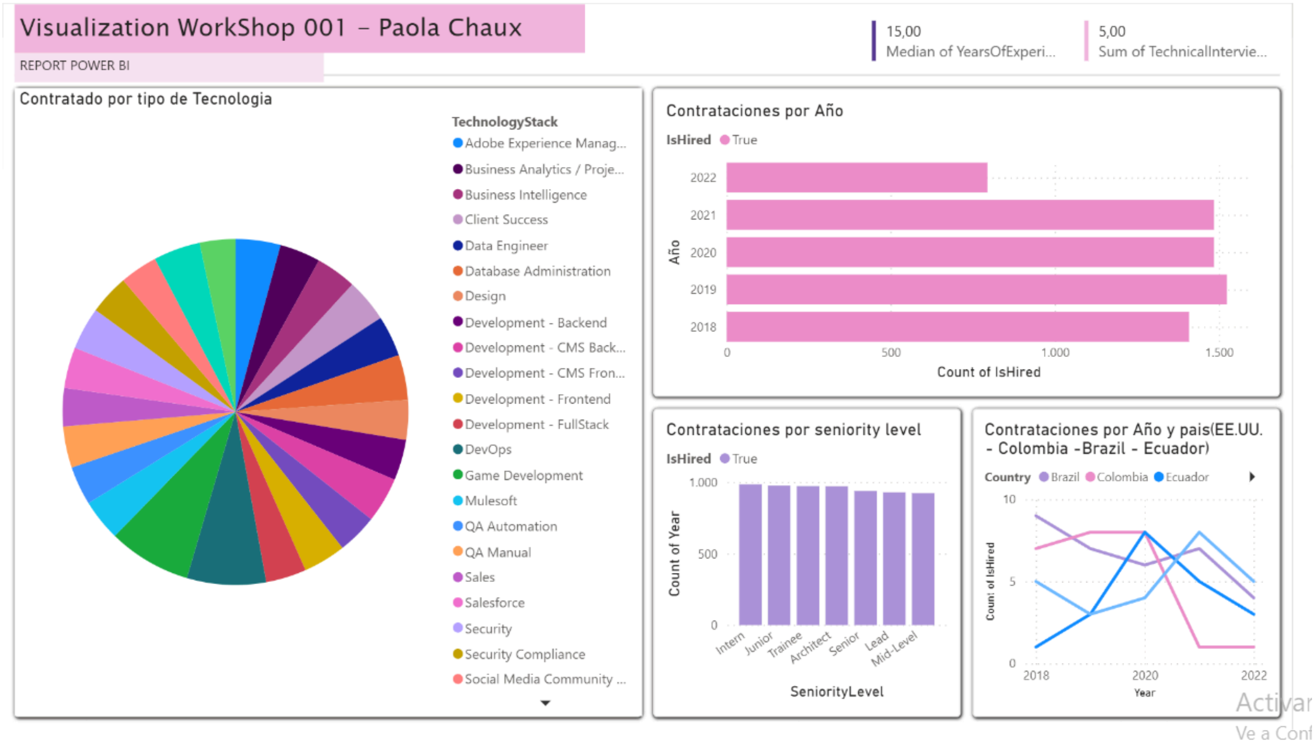
VISUALIZATION IN POWER BI

First of all, import modules: power bi client and pandas

I use Power BI Client to generate a general report which I will edit in the same jupyter notebook.

Then we will proceed to log in to Power BI to use its tools, we call the connection to the database in postgres and we make a query calling all the data in the table, we carry out a verification that if the data is being brought, I change to Datetime type to be sure that the year is with this type, I make an extra column to work separately year and it does not cause errors later and I can use the tool to visualize better,then I proceed to make the general report that Power BI Client creates, I print it and work on it, after having it done I save it and upload it to my group workspace.

Finally I look for the group identifier, the report identifier and code, having it ready, I view my report using the report id and group id.



CONCLUSIONS

The average experience of the candidates is approximately 15 years, indicating a moderately high level of experience overall. Fifty percent of the candidates have between 8 and 23 years of experience, indicating a relatively even distribution with the mean.

High demand in specific areas: Operations Development and Game Development, while Technical Writing and Social Media Community Management have fewer hires, indicating lower demand.

The demand for hire is consistent at all levels of seniority, although it decreases slightly as seniority increases. Entry levels, such as Intern, Junior and Apprentice, have the most recruitments.

There is considerable variability in hiring between countries over the years. Ecuador shows an increase in 2020, Colombia decreases in 2021, Brazil shows a decreasing trend, and the United States recovers in 2021.

REFERENCES

1. https://www.youtube.com/watch?v=jlvMxTn_fOU
2. https://www.youtube.com/watch?v=ag5vK3R_h7M
3. https://www.youtube.com/watch?v=pPhQfeSgO6o
4. https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbIJHTEN4VzV6WVI0cE9Nb2liR2tkcGptWUNUQXxBQ3Jtc0tuNmV3MEF3Z3NWNDczM0Iwb3FJc3NrTE9zUGZ6ak9SUU1ueUZLZURwWWs3Q0xVc1JacEdvemZDNGhxVWhKUlN5UnhWUzdqZEliSWJ2bGtlTkNGRmNTX3JSUEFraEdhRl9oaVZ4NEVlMWRyRnFVeVYzSQ&q=https%3A%2F%2Fgithub.com%2Flearn2excel%2FPowerBI&v=pPhQfeSgO6o
5. https://powerbi.microsoft.com/es-mx/blog/create-power-bi-reports-in-jupyter-notebooks/
6. https://bertia.es/incrustar-informes-de-power-bi-en-jupyter-notebook/
7. https://pypi.org/project/powerbiclient/
8. https://www.neoguias.com/como-conectarse-postgresql-python/#Como_conectarte_a_una_base_de_datos
9. https://www.studocu.com/bo/document/universidad-mayor-de-san-andres/programacion-i/tarea-4python-ejercicios/33129056
10. https://es.stackoverflow.com/questions/185298/importar-una-funci%C3%B3n-de-otro-archivo-ipynb-en-jupyter-notebook
11. https://github.com/dventep/workshop001_etl_education/blob/main/notebooks/eda_report.ipynb
12. https://learn.microsoft.com/es-es/power-bi/consumer/end-user-change-sort
13. https://pypi.org/project/powerbiclient/
14. https://learn.microsoft.com/es-es/power-bi/create-reports/jupyter-quick-report
15. https://learn.microsoft.com/es-es/javascript/api/overview/powerbi/powerbi-jupyter
16. https://learn.microsoft.com/es-es/power-bi/connect-data/service-tutorial-connect-to-github