Workshop 01
Sebastian Diaz Noguera
2216299

1. Migración del dataset a la base de datos

Para la importación del dataset **candidates.csv** se utilizó la librería de postgres psycopg2 para realizar la conexión a la base de datos y cargar el csv. Se realizó de la siguiente manera:

```
def connection():
    with open("C:/Users/SEBASTIAN/ETL/dbconfig.json") as f:
        db_file=json.load(f)
    try:
        conn = psycopg2.connect(
            host = 'localhost',
            user = db_file["user"],
            password = db_file["password"],
            database = db_file["database"]
        )
        print('¡Conexion exitosa!!')
        return conn

    except Exception as ex:
        print(ex)
```

Se utilizó un archivo json para guardar las credenciales y posteriormente se importó el archivo json para extraer cada una de las credenciales necesarias para la conexión.

Posteriormente se creó la tabla vacía a su vez indicando que tipo de dato es cada variable:

```
def createtable():
    conn = connection()
    cursor = conn.cursor()
    sql = '''
                CREATE TABLE candidates
               (first_name varchar,
                last_name varchar,
                email varchar,
                application_date date,
                country varchar,
                YOE INT,
                seniority varchar,
                technology varchar,
                code_challenge_score INT,
                technical_interview_score INT
               );
                '''
    cursor.execute(sql)
    conn.commit()
    print('¡Tabla creada!')
createtable()
```

Luego de estar conectado a la base de datos y haber creado la tabla, para cargar la información del csv inicial se creó la consulta para copiar la información de todas las columnas.

```
def copydata():
    conn = connection()
    cursor = conn.cursor()

    sql = '''
            COPY candidates(first_name, last_name, email, application_date, country, YOE, seniority, technology, code_challenge_
            FROM 'C:/data/candidates.csv' DELIMITER ';' CSV HEADER;
          '''
    cursor.execute(sql)
    conn.commit()
    print('¡Registros ingresados!')
copydata()
```

```
¡Conexion exitosa!!
¡Registros ingresados!
```

2. Análisis exploratorio de los datos

Para empezar con el reto lo primero que se realizó fue un análisis para conocer un poco más a detalle el dataset. A partir de este análisis se obtuvieron diferentes conclusiones que ayudaron a tener una visión más precisa del dataset. Algunas de las observaciones más importantes fueron:

- Al utilizar el comando .**shape** se pudo observar que en cuanto a las características básicas del dataset este cuenta con 50.000 filas y 10 columnas

```
candidates_df.shape
```

```
(50000, 10)
```

- Luego para ver qué tipo de dato era cada variable se usó el comando **.dtype**, se puede observar que 7 columnas son object y las demás de tipo entero.

```
candidates_df.dtypes
```

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

- Posteriormente se ejecutó el comando **.describe** para las columnas de tipo object.

```
candidates_df.describe(include="object")
```

	First Name	Last Name	Email	Application Date	Country	Seniority	Technology
count	50000	50000	50000	50000	50000	50000	50000
unique	3007	474	49833	1646	244	7	24
top	Sarai	Murazik	fern70@gmail.com	2020-07-07	Malawi	Intern	Game Development
freq	33	138	3	50	242	7255	3818

De esta consulta se obtuvo el recuento de cada variable y la cantidad de valores únicos que poseen. Esta información es de gran ayuda ya que también proporciona información sobre la frecuencia con la que se repiten las variables, lo que nos permite tener ideas sobre cómo puede ser el análisis y cuáles son las columnas que pueden tener un impacto significativo en los resultados, como por ejemplo : se puede observar que la columna 'technology', Game Development tiene la mayor freq y lo que probablemente se reflejará de alguna manera en nuestros gráficos.

- Ya por último se utilizó el comando **.isnull** para verificar si contábamos con valores null en las variables y como se puede observar todas las variables no contienen valores null

```
candidates_df.isnull().any()
```

```
First Name                  False
Last Name                   False
Email                       False
Application Date            False
Country                     False
YOE                         False
Seniority                   False
Technology                  False
Code Challenge Score        False
Technical Interview Score   False
dtype: bool
```

3. Operaciones y Gráficos

En este reto se requieren las siguientes visualizaciones :

a. Contrataciones por tecnología (gráfico circular)
b. Contrataciones por año (gráfico de barras horizontales)
c. Contrataciones por antigüedad (gráfico de barras)
d. Contrataciones por país a lo largo de los años (sólo EE.UU., Brasil, Colombia y Ecuador) (gráfico multilínea)

Para cumplir con los requisitos del desafío,era esencial determinar si un candidato estaba contratado, lo cual implicaba que ambas puntuaciones fueran iguales o mayores a 7.

Para lograr esto, mi primer paso fue modificar la tabla que había creado y agregar la columna '**hired**' mediante la función **update_table()**, la cual contendría datos de tipo booleano

```
def update_table():
    conn = connection()
    cursor = conn.cursor()
    update = """ALTER TABLE candidates ADD COLUMN hired boolean"""
    print("tabla actualizada")
    cursor.execute(update)
    conn.commit()
update_table()
```

```
¡Conexion exitosa!!
tabla actualizada
```

Ya teniendo la tabla actualizada con la nueva columna que agregue, Luego por medio de la función **update_hired()** establecí la condición para que se considerara como verdadero (true).

```
def update_hired():
    conn = connection()
    cursor = conn.cursor()
    update = """UPDATE candidates
            SET hired = (code_challenge_score >= 7 AND technical_interview_score >= 7)"""
    cursor.execute(update)
    print("Columna hired actualizada")
    conn.commit()
update_hired()
```

```
¡Conexion exitosa!!
Columna hired actualizada
```

1. Respuesta Primer punto

En esta consulta se pedía sacar el numero de contrataciones por tecnologia que hay en el dataset, para ello se llamó a la conexión con la base de datos y el cursor, para posteriormente hacer el Query, en donde le indicamos que nos seleccione '**Technology**', que haga un conteo de todos los datos disponibles en donde la columna '**Hired**' sea true, es decir, que si fueron contratados.

```
def query1():
    conn = connection()
    cursor = conn.cursor()
    technology=[]
    Q1 = """
    SELECT technology,
    COUNT(*) AS conteo FROM candidates WHERE hired = true GROUP BY technology
    """
    cursor.execute(Q1)
    results=cursor.fetchall()
    for row in results:
        technology.append(row)
    df = pd.DataFrame(technology)
    df.columns=['Technology','Count']
    print(df)
    conn.commit()
query1()
```

Luego sacamos todos los resultados con la función **'fetchall()**' lo metemos a un arreglo vacío y lo convertimos en un DataFrame, de esta forma:

```
¡Conexion exitosa!!
                                Technology  Count
0                Development - CMS Backend    284
1                                Salesforce    256
2                             Data Engineer    255
3                             QA Automation    243
4                                     Sales    239
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
```

2. Respuesta segundo punto.

En este punto, el objetivo era determinar cuántas contrataciones se realizan cada año, para ello seleccionamos la columna llamada '**application_date**' que es la fecha de aplicación y con la cláusula **EXTRACT** sacamos únicamente el año, hacemos un conteo de los empleados donde la columna '**hired**' es verdadero

```
def query2():
    conn = connection()
    cursor = conn.cursor()
    year=[]
    Q2 = """
    SELECT EXTRACT(year FROM application_date) as year,
    COUNT(*) AS conteo FROM candidates WHERE hired = true GROUP BY year;
    """

    cursor.execute(Q2)
    results=cursor.fetchall()
    for row in results:
        year.append(row)
    df = pd.DataFrame(year)
    df.columns=['Year','Count']
    print(df)
    conn.commit()
query2()
```

De igual manera que en la consulta anterior podemos ver los resultados insertados en un DataFrame:

```
¡Conexion exitosa!!
   Year  Count
0  2021   1485
1  2020   1485
2  2022    795
3  2018   1409
4  2019   1524
```

3. Respuesta tercer punto.

En el siguiente objetivo se pedía contratación por seniority, para esto se selecciono la columna '**seniority**' que se puede tomar como nivel profesional, después se realiza el conteo en donde '**Hired**' sea igual a True y luego lo agrupó por las categorías de '**seniority**'

```python
def query3():
    conn = connection()
    cursor = conn.cursor()
    seniority = []
    Q3 = """
    SELECT seniority,
    COUNT(*) AS conteo FROM candidates WHERE hired = true GROUP BY seniority
    """

    cursor.execute(Q3)
    results=cursor.fetchall()
    for row in results:
        seniority.append(row)
    df = pd.DataFrame(seniority)
    df.columns=['Seniority','Count']
    print(df)
    conn.commit()
query3()
```

Realizada la consulta se puede ver el resultado obtenido en un Dataframe

```
¡Conexion exitosa!!
   Seniority  Count
0    Trainee    973
1     Senior    939
2  Architect    971
3     Intern    985
4       Lead    929
5  Mid-Level    924
6     Junior    977
```

4. Respuesta cuarto punto.

Ya por último en este se nos solicitaba contrataciones por país a lo largo de los años (sólo EE.UU., Brasil, Colombia y Ecuador), seleccionamos las columnas de ‘country’ y ‘application_date’, a esta última le extraemos el año de acuerdo la condición de la columna ‘hired’, tambien dentro de esta condición especificamos únicamente los valores que queremos sacar de country, tales como Colombia, Brasil, Ecuador y Estados Unidos.

```python
def query4():
    conn = connection()
    cursor = conn.cursor()
    country = []
    Q4 = """
    SELECT EXTRACT(year FROM application_date) as year, country,
    COUNT(*) AS conteo FROM candidates WHERE hired = true AND country IN ('Brazil','Ecuador','United States of America','Colombia'
    """

    cursor.execute(Q4)
    results=cursor.fetchall()
    for row in results:
        country.append(row)
    df = pd.DataFrame(country)
    df.columns=['Year','Country','Count']
    print(df)
    conn.commit()
query4()
```

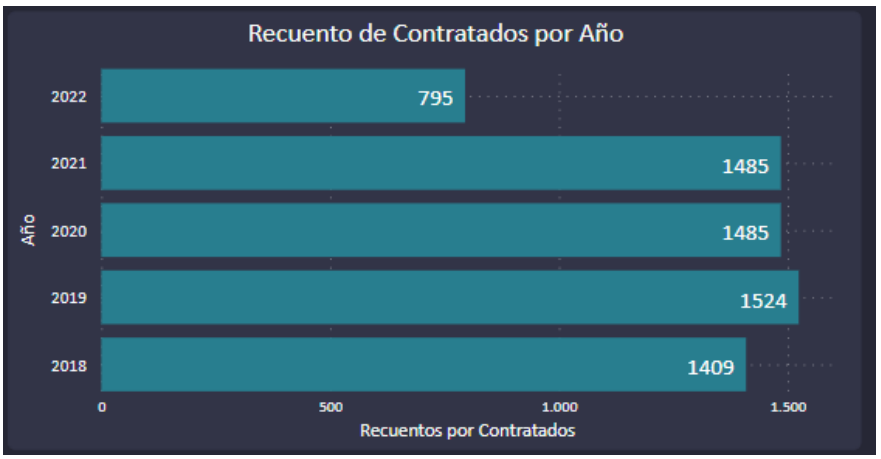
Sacamos el resultado del query y lo convertimos a un DataFrame para poder visualizarlo.

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

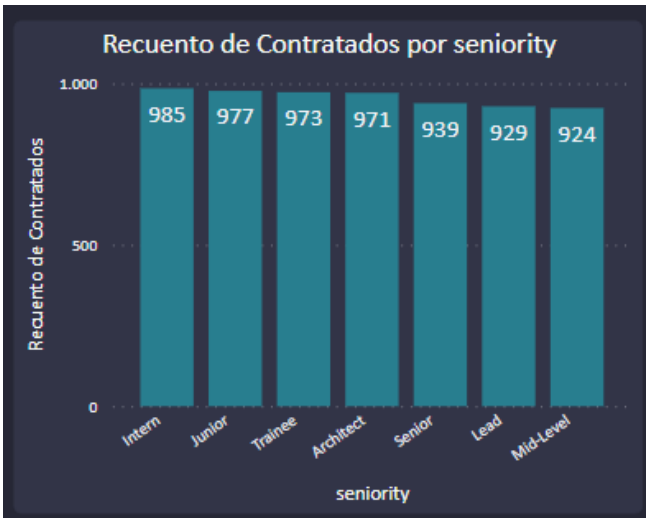
Gráficas

1. Una vez tenemos los resultados graficamos en la aplicación de PowerBi, colocando las variables usadas en este query y al analizar la grafica podemos concluir que, si tomamos que todos los empleados trabajan en una misma empresa habría más personal en '**Game Development**' con 7.7% y '**DevOps**' con 7.39% de todo el personal, mientras que el resto de tecnologías se distribuyen casi igual entre el 3.39% y 4.39%.

2. Después de realizada la gráfica en la herramienta Power Bi, se puede concluir que el año donde más candidatos fueron contratados fue en el año **2019** con **1524** incrementando respecto al año anterior un **115** de contratados , y que en los años **2020** y **2021** las contrataciones disminuyeron levemente manteniéndose en **1485**, respecto al año **2022** al tener este comportamiento en la grafica lo mas probable es que este no tenga los datos del año completo



3. En esta gráfica de contratados por seniority, se puede observar que en general todas están niveladas, es decir, se distribuyen normal, pero de si algo se puede resaltar es que de todos el nivel ‘**intern’** es el que tiene mayor número de contratados con 985 empleados mientras que ‘Mid-Level’ es el que menos tiene con 924 empleados.



4. Por último en esta gráfica se puede evidenciar que a medida de que pasan los años en los países han variado mucho en el número de contratados por cada uno de ellos, claro está el ejemplo de **colombia** que al principio de los años tuvo un gran número de contrataciones pero a medida del tiempo estos han bajado llegando a tener solo 1 contratación en el último año, cabe resaltar que esto puede deber por lo mencionado anteriormente que los datos en el 2022 no estén completo.