

# Evidencias de la Tarea 5 y Practica 06

## Ejercicio 1:

The screenshot displays a Jupyter Notebook interface with the following content:

### 5.Carga de Datos (Mysql)

#### Paso 1: Conexion con mysql

```
[8]: !pip install sqlalchemy pymysql
```

Requirement already satisfied: sqlalchemy in c:\users\ponol\anaconda\lib\site-packages (2.0.34)  
collecting pymysql  
Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)  
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\ponol\anaconda\lib\site-packages (from sqlalchemy) (4.11.0)  
Requirement already satisfied: greenlet<=0.4.17 in c:\users\ponol\anaconda\lib\site-packages (from sqlalchemy) (3.0.1)  
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)  
Installing collected packages: pymysql  
Successfully installed pymysql-1.1.1

```
[11]: from sqlalchemy import create_engine
import pandas as pd

# crear conexión con SQLAlchemy
engine = create_engine("mysql+pymysql://root:13579@localhost/salaries")

# leer la tabla real
df_salaries = pd.read_sql("SELECT * FROM salaries_la", engine)
df_salaries.head()
```

```
[11]:
```

	id	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	1	2025.0	SE	FT	Data Product Owner	170000	USD	170000	US	0.0	US	
1	2	2025.0	SE	FT	Data Product Owner	110000	USD	110000	US	0.0	US	
2	3	2025.0	SE	FT	Data Product Owner	170000	USD	170000	US	0.0	US	
3	4	2025.0	SE	FT	Data Product Owner	110000	USD	110000	US	0.0	US	
4	5	2025.0	SE	FT	Engineer	143000	USD	143000	US	0.0	US	

#### Paso 2: Ver estructura de los datos

```
[13]: df_salaries.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73158 entries, 0 to 73157
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   id                  73158 non-null  int64
 1   work_year           73153 non-null  float64
 2   experience_level     73153 non-null  object
 3   employment_type     73153 non-null  object
 4   job_title           73153 non-null  object
 5   salary              73143 non-null  object
 6   salary_currency     73153 non-null  object
 7   salary_in_usd       73143 non-null  object
 8   employee_residence  73153 non-null  object
 9   remote_ratio        73153 non-null  float64
10   company_location    73153 non-null  object
11   company_size        73153 non-null  object
dtypes: float64(2), int64(1), object(9)
memory usage: 6.7+ MB
```

```
[14]: print(df_salaries.size)
877806
```

```
[15]: print(len(df_salaries))
73158
```

#### Paso 3: Limpieza de datos

```
[17]: # Eliminar registros donde falte 'experience_level'
df_salaries = df_salaries.dropna(subset=['experience_level'])
# Eliminar registros donde falte 'experience_level'
df_salaries = df_salaries.dropna(subset=['experience_level'])
print(len(df_salaries))
73153
```

#### Paso 4: Transformación de datos

```
[18]: # Selección de columnas clave
df_salaries = df_salaries[['experience_level', 'employment_type', 'salary_in_usd', 'remote_ratio']]

# Transformación de valores
df_salaries['experience_level'] = df_salaries['experience_level'].str.upper()
df_salaries['monthly_salary'] = pd.to_numeric(df_salaries['salary_in_usd'], errors='coerce') / 12
df_salaries = df_salaries.dropna()
```

Tarea

Project Overview | Cloud Mon. | +

localhost:8888/notebooks/Documents/Uni/ECBD/9A%20DGS%20ECBD/9A%20DGS%20ECBD/Practica06/Tarea.ipynb?

Jupyter Tarea Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help Trusted

Python [conda env:base] Anaconda ToolBox

### Paso 4: Transformación de datos

```
[18]: # Selección de columnas clave
df_salaries = df_salaries[['experience_level', 'employment_type', 'salary_in_usd', 'remote_ratio']]

# Transformación de valores
df_salaries['experience_level'] = df_salaries['experience_level'].str.upper()
df_salaries['monthly_salary'] = pd.to_numeric(df_salaries['salary_in_usd'], errors='coerce') / 12
df_salaries = df_salaries.dropna()

print(len(df_salaries))
df_salaries.head()
```

```
73128
```

	experience_level	employment_type	salary_in_usd	remote_ratio	monthly_salary
0	SE	FT	170000	0.0	14166.666667
1	SE	FT	110000	0.0	9166.666667
2	SE	FT	170000	0.0	14166.666667
3	SE	FT	110000	0.0	9166.666667
4	SE	FT	143000	0.0	11916.666667

### Paso 5: Exportar datos a tabla temporal

```
[22]: from sqlalchemy import text

# crear tabla definitiva (o de respaldo)
with engine.connect() as con:
    con.execute(text("""
        CREATE TABLE IF NOT EXISTS salaries_transformado (
            experience_level VARCHAR(10),
            employment_type VARCHAR(5),
            salary_in_usd FLOAT,
            remote_ratio FLOAT,
            monthly_salary FLOAT
        )
    ""))

[23]: df_salaries.to_sql(
    name='salaries_transformado',
    con=engine,
    if_exists='replace', # 'append' si no quieres reemplazar la tabla
    index=False
)
```

```
73128
```

Tarea

Project Overview | Cloud Mon. | +

localhost:8888/notebooks/Documents/Uni/ECBD/9A%20DGS%20ECBD/9A%20DGS%20ECBD/Practica06/Tarea.ipynb?

Jupyter Tarea Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help Trusted

Python [conda env:base] Anaconda ToolBox

### Paso 5: Exportar datos a tabla temporal

```
[22]: from sqlalchemy import text

# crear tabla definitiva (o de respaldo)
with engine.connect() as con:
    con.execute(text("""
        CREATE TABLE IF NOT EXISTS salaries_transformado (
            experience_level VARCHAR(10),
            employment_type VARCHAR(5),
            salary_in_usd FLOAT,
            remote_ratio FLOAT,
            monthly_salary FLOAT
        )
    ""))

[23]: df_salaries.to_sql(
    name='salaries_transformado',
    con=engine,
    if_exists='replace', # 'append' si no quieres reemplazar la tabla
    index=False
)
```

```
73128
```

### Paso 6: Verificación de exportación

```
[24]: verification = pd.read_sql("SELECT * FROM salaries_transformado LIMIT 5", engine)
verification
```

	experience_level	employment_type	salary_in_usd	remote_ratio	monthly_salary
0	SE	FT	170000	0.0	14166.666667
1	SE	FT	110000	0.0	9166.666667
2	SE	FT	170000	0.0	14166.666667
3	SE	FT	110000	0.0	9166.666667
4	SE	FT	143000	0.0	11916.666667

```
[ ]:
[ ]:
[ ]:
```

## Ejercicio 2:

Jupyter Tarea Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help Trusted

Python [conda env:base] Anaconda Toolbox

### Con Api

#### Importar datos de la Api

```
[25]: import requests
import pandas as pd

# obtener los primeros 20 pokémon
url = "https://pokeapi.co/api/v2/pokemon?limit=20"
response = requests.get(url)
data = response.json()

# Extraer los enlaces a los detalles de cada Pokémon
pokemon_urls = [pokemon["url"] for pokemon in data["results"]]

# Obtener datos de cada Pokémon
pokemon_data = []

for url in pokemon_urls:
    res = requests.get(url).json()
    pokemon_data.append({
        "name": res["name"],
        "height": res["height"],
        "weight": res["weight"],
        "base_experience": res["base_experience"],
        "types": ", ".join([t["type"]["name"] for t in res["types"]])
    })

# Crear DataFrame
df_pokemon = pd.DataFrame(pokemon_data)
df_pokemon.head()
```

	name	height	weight	base_experience	types
0	bulbasaur	7	69	64	grass, poison
1	ivysaur	10	130	142	grass, poison
2	venusaur	20	1000	236	grass, poison
3	charmander	6	85	62	fire
4	charmeleon	11	190	142	fire

#### Limpiar datos

```
[26]: # verificar datos
```

Jupyter Tarea Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help Trusted

Python [conda env:base] Anaconda Toolbox

#### Limpiar datos

```
[26]: # verificar datos
print(df_pokemon.isnull().sum())

# Eliminar filas con nulos si hubiera
df_pokemon = df_pokemon.dropna()
```

	name	height	weight	base_experience	types
0	bulbasaur	7	69	64	grass, poison
1	ivysaur	10	130	142	grass, poison
2	venusaur	20	1000	236	grass, poison
3	charmander	6	85	62	fire
4	charmeleon	11	190	142	fire

#### Transformar datos

```
[27]: # crear columna que calcule el "peso por experiencia"
df_pokemon["weight_per_exp"] = df_pokemon["weight"] / df_pokemon["base_experience"]
df_pokemon.head()
```

	name	height	weight	base_experience	types	weight_per_exp
0	bulbasaur	7	69	64	grass, poison	1.078125
1	ivysaur	10	130	142	grass, poison	0.915493
2	venusaur	20	1000	236	grass, poison	4.237288
3	charmander	6	85	62	fire	1.370968
4	charmeleon	11	190	142	fire	1.338028

#### Exportar datos

```
[28]: df_pokemon.to_csv("datos_pokemon.csv", index=False)
print("Exportado como datos_pokemon.csv")
Exportado como datos_pokemon.csv

[29]: from sqlalchemy import create_engine

engine = create_engine("mysql+pymysql://root:13579@localhost/salaries")

df_pokemon.to_sql(
    name="pokemon_api_data",
    con=engine,
    if_exists="replace",
    index=False
)
```

Tarea

Project Overview | Cloud Mon... | +

localhost:8888/notebooks/Documents/Uni/ECBD/9A%20IDGS%20ECBD/9A%20IDGS%20ECBD/Practica06/Tarea.ipynb?

JupyterTarea Last checkpoint: 3 hours ago

FileEditViewRunKernelSettingsHelp

Trusted

JupyterLabPython [conda envbase]\*Anaconda Toolbo...

```
df_pokemon.head()
```

	name	height	weight	base_experience	types	weight_per_exp
0	bulbasaur	7	69	64	grass, poison	1.078125
1	ivysaur	10	130	142	grass, poison	0.915493
2	venusaur	20	1000	236	grass, poison	4.237288
3	charmander	6	85	62	fire	1.370968
4	charmeleon	11	190	142	fire	1.338028

Exportar datos

```
[28]: df_pokemon.to_csv("datos_pokemon.csv", index=False)
      print("Exportado como datos_pokemon.csv")
      Exportado como datos_pokemon.csv

[29]: from sqlalchemy import create_engine
      engine = create_engine("mysql+pymysql://root:13579@localhost/salaries")
      df_pokemon.to_sql(
          name="pokemon_api_data",
          con=engine,
          if_exists="replace",
          index=False
      )

[29]: 20
[ ]:
[ ]:
[ ]:
[ ]:
```

Conexion con mongo

```
[2]: !pip install pymongo
      Collecting pymongo
      Downloading pymongo-4.13.2-cp312-win_...
      Collecting dnspython<2.0.0,=>1.24.0 (from pymongo)
      Downloading dnspython-2.7.0-py3-none-any...
```

02:03 p.m.  
18/06/2025

## Ejercicio extra:

**Conexion con mongo**

```
[7]: !pip install pymongo
Collecting pymongo
  Downloading pymongo-4.13.2-cp312-win_amd64.whl.metadata (22 kB)
Collecting dnspython<0.0,=1.16.0 (from pymongo)
  Downloading dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)
Collecting pymongo-4.13.2-cp312-win_amd64.whl (903 kB)
-----0.0/903.3 kB 7 eta :--:
-----0.0/903.3 kB 7 eta :--:
-----526.3/903.3 kB 3.4 MB/s eta 0:00:01
-----903.3/903.3 kB 2.3 MB/s eta 0:00:00
Downloaded dnspython-2.7.0-py3-none-any.whl (113 kB)
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.7.0 pymongo-4.13.2

[8]: from pymongo import MongoClient
import pandas as pd

# URI de MongoDB Atlas
uri = "mongodb+srv://lospozoles13579@cluster0.hrotage.mongodb.net/"

# Conectar al cliente
cliente = MongoClient(uri)

# Seleccionar base de datos y colección
db = cliente["salaries"] # Nombre de tu base de datos en Atlas
coleccion = db["salaries_ia"] # Nombre de la colección

# Leer datos
datos = list(coleccion.find())
df_mongo = pd.DataFrame(datos)
df_mongo.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
1	2025.0	SE	FT	Data Product Owner	170000.0	USD	170000.0	US	0.0	US	M
2	2025.0	SE	FT	Data Product Owner	110000.0	USD	110000.0	US	0.0	US	M
3	2025.0	SE	FT	Data Product Owner	170000.0	USD	170000.0	US	0.0	US	M
4	2025.0	SE	FT	Data Product Owner	110000.0	USD	110000.0	US	0.0	US	M
5	2025.0	MX	FT	Engineer	143000.0	USD	143000.0	US	0.0	US	M

**Insertion de los datos**

```
[12]: import pandas as pd
from pymongo import MongoClient

# Leer tu archivo CSV local
ruta_csv = r"C:\Users\pooal\Documents\Un\ECBD\9A ID05 ECBD\Practica06\salaries.csv"
df = pd.read_csv(ruta_csv)

# Conectar a MongoDB Atlas
uri = "mongodb+srv://lospozoles13579@cluster0.hrotage.mongodb.net/"
cliente = MongoClient(uri)

# Base de datos y colección
db = cliente["salaries"]
coleccion = db["salaries_ia"]

# Insertar los datos en la colección
coleccion.insert_many(df.to_dict(orient="records"))

print("✅ Datos insertados correctamente en salaries.salaries_ia")
# Datos insertados correctamente en salaries.salaries_ia

[13]: # Ver cuántos documentos hay
print(coleccion.count_documents({}))

146320
```

Jupyter Tarea Last Checkpoint: 3 hours ago

### Limpieza de datos

```
[15]: # Ver columnas y nulos
df_mongo.columns
df_mongo.info()
df_mongo.isnull().sum()

# Limpieza básica
df_mongo = df_mongo.dropna(subsets=["experience_level", "salary_in_usd"])

<class 'pandas.core.frame.DataFrame'>
Index: 73146 entries, 0 to 73147
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   _id                 73146 non-null  object
 1   work_year           73146 non-null  float64
 2   experience_level     73146 non-null  object
 3   employment_type     73146 non-null  object
 4   job_title           73146 non-null  object
 5   salary              73146 non-null  float64
 6   salary_currency     73146 non-null  object
 7   salary_in_usd       73146 non-null  float64
 8   employee_residence  73146 non-null  object
 9   remote_ratio        73146 non-null  float64
10   company_location    73146 non-null  object
11   company_size        73146 non-null  object
dtypes: float64(4), object(8)
memory usage: 7.3+ MB
```

### Transformación de datos

```
[16]: # Asignar tipos correctos
df_mongo["salary_in_usd"] = pd.to_numeric(df_mongo["salary_in_usd"], errors="coerce")

# Crear columna salario mensual
df_mongo["monthly_salary"] = df_mongo["salary_in_usd"] / 12

# Pasar experiencia a mayúsculas
df_mongo["experience_level"] = df_mongo["experience_level"].str.upper()

# Ver resultado
df_mongo[["experience_level", "salary_in_usd", "monthly_salary"]].head()
```

Jupyter Tarea Last Checkpoint: 3 hours ago

### Transformación de datos

```
[16]: # Asignar tipos correctos
df_mongo["salary_in_usd"] = pd.to_numeric(df_mongo["salary_in_usd"], errors="coerce")

# Crear columna salario mensual
df_mongo["monthly_salary"] = df_mongo["salary_in_usd"] / 12

# Pasar experiencia a mayúsculas
df_mongo["experience_level"] = df_mongo["experience_level"].str.upper()

# Ver resultado
df_mongo[["experience_level", "salary_in_usd", "monthly_salary"]].head()
```

	experience_level	salary_in_usd	monthly_salary
0	SE	170000.0	14166.666667
1	SE	110000.0	9166.666667
2	SE	170000.0	14166.666667
3	SE	110000.0	9166.666667
4	MX	143000.0	11916.666667

### Exportar a MySQL (opcional)

```
[17]: from sqlalchemy import create_engine

engine = create_engine("mysql+pymysql://root:115799@localhost/salaries")

df_mongo.to_sql(
    name="salaries_mongo_exportado",
    con=engine,
    if_exists="replace",
    index=False
)
```