

The screenshot shows a Jupyter Notebook titled 'Practica_6.ipynb' in a dark-themed IDE. The Explorer panel on the left shows a folder named '9A_IDGS' containing files: 'comentarios_filtrados.csv', 'Practica_6.ipynb', 'salaries.csv', and 'users_placeholder.csv'. The main editor area displays the following Python code:

```
# 5.1 - Importación de datos
import pandas as pd
from sqlalchemy import create_engine, text

# Conexión usando el usuario data_user
engine = create_engine(
    "mysql+mysqlconnector://data_user:Contrasenasegura123%40@localhost:3306/employees"
)

# Prueba de conexión
with engine.connect() as conn:
    print("Conexión OK:", conn.execute(text("SELECT 1")).scalar())

# Leemos la tabla 'employees'
df_emp = pd.read_sql_table('employees', con=engine)
print("Dimensiones (filas, columnas):", df_emp.shape)
df_emp.head(5)
```

The status bar at the bottom indicates 'Python' and 'Cell 37 of 60'.

The screenshot shows the same Jupyter Notebook interface, but now displaying the output of the previous code cell. The output includes the connection status, the dimensions of the DataFrame, and a preview of the first five rows of the 'employees' table.

```
Conexión OK: 1
Dimensiones (filas, columnas): (63, 6)
```

	emp_no	birth_date	first_name	last_name	gender	hire_date
0	10001	1970-01-01	Juan	García	M	1990-01-01
1	10002	1970-01-02	María	Martínez	F	1990-01-02
2	10003	1970-01-03	José	Rodríguez	M	1990-01-03
3	10004	1970-01-04	Luis	López	F	1990-01-04
4	10005	1970-01-05	Ana	Hernández	M	1990-01-05

```
#muestra metadatos actuales
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73163 entries, 0 to 73162
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   work_year              73162 non-null  float64
1   experience_level        73158 non-null  object
2   employment_type         73158 non-null  object
3   job_title              73158 non-null  object
4   salary                 73156 non-null  float64
5   salary_currency         73159 non-null  object
6   salary_in_usd          73159 non-null  float64
7   employee_residence     73159 non-null  object
8   remote_ratio            73159 non-null  float64
9   country_location       73159 non-null  object
```

The status bar at the bottom indicates 'Python' and 'Cell 37 of 60'.

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

Practica 6 - Preparación de datos para Análisis (ETL) > Análisis Básico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

base (Python 3.12.7)

Limpieza de datos

```
# 5.2 - Limpieza de datos

# 5.2.1 - Conteo de nulos antes
print("Nulos antes de limpieza:\n", df_emp.isnull().sum())

# 5.2.2 - Eliminamos filas sin datos críticos
df_clean = df_emp.dropna(subset=[
    'first_name',
    'last_name',
    'birth_date',
    'gender',
    'hire_date'
]).copy()

# 5.2.3 - Revisión post-limpieza
print(f"Filas antes: {len(df_emp)} - Filas después: {len(df_clean)}")
print("Nulos después de limpieza:\n", df_clean.isnull().sum())
```

[52]

```
...
Nulos antes de limpieza:
emp_no      0
birth_date  0
first_name  0
last_name   0
gender      0
hire_date   0
dtype: int64
Filas antes: 63 - Filas después: 63
Nulos después de limpieza:
emp_no      0
birth_date  0
first_name  0
last_name   0
gender      0
hire_date   0
dtype: int64
```

Python

Spaces: 4 Cell 37 of 60 Go Live

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

Practica 6 - Preparación de datos para Análisis (ETL) > Análisis Básico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

base (Python 3.12.7)

Transformar datos

```
# 5.3 - Transformar datos

# Convertimos strings a datetime
df_clean['birth_date'] = pd.to_datetime(df_clean['birth_date'])
df_clean['hire_date'] = pd.to_datetime(df_clean['hire_date'])

# Calculamos años de servicio
today = pd.Timestamp.today().normalize()
df_clean['years_service'] = ((today - df_clean['hire_date']).dt.days / 365).astype(int)

# Asignamos nivel según antigüedad
df_clean['service_level'] = df_clean['years_service'].apply(
    lambda y: 'Senior' if y >= 10 else 'Junior'
)

# Mostramos un ejemplo
df_clean[['emp_no', 'first_name', 'last_name', 'hire_date', 'years_service', 'service_level']].head(5)
```

[53]

```
...
emp_no first_name last_name hire_date years_service service_level
0      10001     Juan   García 1990-01-01           35         Senior
1      10002     María  Martínez 1990-01-02           35         Senior
2      10003    José    Rodríguez 1990-01-03           35         Senior
3      10004    Luis    López    1990-01-04           35         Senior
4      10005     Ana    Hernández 1990-01-05           35         Senior
```

Exportar datos a una tabla

```
# 5.3.1 - Exportar datos a una tabla
```

Python

Spaces: 4 Cell 37 of 60 Go Live

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

6 > Practica_6.ipynb > Practica 6 - Preparacion de datos para Analisis (ETL) > Analisis Basico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline ...

base (Python 3.12.7)

2	10003	José	Rodríguez	1990-01-03	35	Senior
3	10004	Luis	López	1990-01-04	35	Senior
4	10005	Ana	Hernández	1990-01-05	35	Senior

Exportar datos a una tabla

Generate + Code + Markdown

```
# 5.4 - Exportar datos a MySQL

df_clean.to_sql(
    name='employees_service',
    con=engine,
    if_exists='replace',
    index=False
)

# Verificamos el conteo en la nueva tabla
cnt = pd.read_sql("SELECT COUNT(*) AS total FROM employees_service", con=engine).iloc[0,0]
print(f"Tabla 'employees_service' creada con {cnt} registros.")
```

[54] Python

... Tabla 'employees_service' creada con 63 registros.

Ejercicio 2

Importacion de datos

```
# Librerias necesarias
import requests
import pandas as pd
```

[55] Python

Windows taskbar: WhatsApp, Firefox, VS Code, File Explorer, U盘, Spotify, etc.

Spaces: 4 Cell 37 of 60 Go Live

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

6 > Practica_6.ipynb > Practica 6 - Preparacion de datos para Analisis (ETL) > Analisis Basico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline ...

base (Python 3.12.7)

Tabla 'employees_service' creada con 63 registros.

Ejercicio 2

Generate + Code + Markdown

Importacion de datos

```
# Librerias necesarias
import requests
import pandas as pd
```

[55] Python

```
# Consumimos la API de usuarios de JSONPlaceholder
url = "https://jsonplaceholder.typicode.com/users"
resp = requests.get(url)
resp.raise_for_status()
```

[56] Python

```
# Normalizamos el JSON en un DataFrame
data = resp.json()
df_users = pd.json_normalize(data)

print("Dimensiones:", df_users.shape)
df_users.head(3)
```

[57] Python

... Dimensiones: (10, 15)

	id	name	username	email	phone	website	address	city	state	country	latitude	longitude	company	name	code
0	1	name	name	name	phone	website	address	city	state	country	latitude	longitude	company	name	code

Windows taskbar: WhatsApp, Firefox, VS Code, File Explorer, U盘, Spotify, etc.

Spaces: 4 Cell 37 of 60 Go Live

File Edit Selection View Go ... 9A_IDGS

EXPLORER ... Practica_6.ipynb x

6 > Practica_6.ipynb > Practica 6 - Preparacion de datos para Analisis (ETL) > Analisis Basico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

base (Python 3.12.7)

```
print("Dimensiones:", df_users.shape)
df_users.head(3)
```

Python

Dimensiones: (10, 15)

	id	name	username	email	phone	website	address.street	address.suite	address.city	address.zipcode	address.geo.lat	address.geo.lng	company.name	company.phrase
0	1	Leanne Graham	Bret	Sincere@april.biz	1-770-736-8031 x56442	hildegard.org	Kulas Light	Apt. 556	Gwenborough	92998-3874	-37.3159	81.1496	Romaguera-Crona	Multi-layered client-server neural-net
1	2	Ervin Howell	Antonette	Shanna@melissa.tv	010-692-6593 x09125	anastasia.net	Victor Plains	Suite 879	Wisokyburgh	90566-7771	-43.9509	-34.4618	Deckow-Crist	Proactive didactic contingency
2	3	Clementine Bauch	Samantha	Nathan@yesenia.net	1-463-123-4447	ramiro.info	Douglas Extension	Suite 847	McKenziehaven	59590-4157	-68.6102	-47.0653	Romaguera-Jacobson	Face to face bifurcated interface

Limpieza de datos

```
# 2.1 - Eliminamos duplicados por 'id'
df_clean = df_users.drop_duplicates(subset='id').copy()

# 2.2 - Quitamos filas sin correo electrónico
df_clean = df_clean.dropna(subset=['email'])

print("Tras limpieza:", df_clean.shape)
```

Python

Tras limpieza: (10, 15)

Transformar datos

File Edit Selection View Go ... 9A_IDGS

EXPLORER ... Practica_6.ipynb x

6 > Practica_6.ipynb > Practica 6 - Preparacion de datos para Analisis (ETL) > Analisis Basico de DataFrame (Datos Estadísticos Generales) > Valores no nulos > counts = df["work_year"].value_counts().sort_index()

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

base (Python 3.12.7)

Transformar datos

```
# 3.1 - Convertimos latitud/longitud a tipo numérico
df_clean['lat'] = df_clean['address.geo.lat'].astype(float)
df_clean['lng'] = df_clean['address.geo.lng'].astype(float)

# 3.2 - Renombramos columnas para mayor claridad
df_clean = df_clean.rename(columns={
    'address.street': 'street',
    'address.suite': 'suite',
    'address.city': 'city',
    'address.zipcode': 'zipcode',
    'company.name': 'company_name',
    'company.catchPhrase': 'company_phrase'
})

# 3.3 - Añadimos usuario en minúsculas y dominio de email
df_clean['username_lower'] = df_clean['username'].str.lower()
df_clean['email_domain'] = df_clean['email'].str.split('@').str[1]

df_clean[
    'id', 'name', 'username_lower', 'email', 'email_domain',
    'street', 'suite', 'city', 'zipcode', 'lat', 'lng',
    'company_name', 'company_phrase'
].head(3)
```

Python

	id	name	username_lower	email	email_domain	street	suite	city	zipcode	lat	lng	company_name	company_phrase
0	1	Leanne Graham	bret	Sincere@april.biz	april.biz	Kulas Light	Apt. 556	Gwenborough	92998-3874	-37.3159	81.1496	Romaguera-Crona	Multi-layered client-server neural-net
1	2	Ervin Howell	antonette	Shanna@melissa.tv	melissa.tv	Victor Plains	Suite 879	Wisokyburgh	90566-7771	-43.9509	-34.4618	Deckow-Crist	Proactive didactic contingency
2	3	Clementine Bauch	samantha	Nathan@yesenia.net	yesenia.net	Douglas Extension	Suite 847	McKenziehaven	59590-4157	-68.6102	-47.0653	Romaguera-Jacobson	Face to face bifurcated interface

Exportar datos

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

6 > Practica_6 - Preparacion de datos para Analisis (ETL) > Analisis Basico de DataFrame (Datos Estadisticos Generales) > Valores no nulos > counts = df[["work_year"]].value_counts().sort_index()

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

base (Python 3.12.7)

2	3	Clementine Bauch	samantha	Nathan@yesenia.net	yesenia.net	Douglas Extension	Suite 847	McKenziehaven	59590-4157	-68.6102	-47.0653	Romaguera-Jacobson	Face to face bifurcated interface
---	---	------------------	----------	--------------------	-------------	-------------------	-----------	---------------	------------	----------	----------	--------------------	-----------------------------------

Exportar datos a un csv

Generate + Code + Markdown

```
# 4.1 - Guardar a CSV
df_clean.to_csv("users_placeholder.csv", index=False)
print(" users_placeholder.csv generado")

# 4.2 - Guardar en MySQL como tabla 'users_api'
from sqlalchemy import create_engine

df_clean.to_sql(
    name='users_api',
    con=engine,
    if_exists='replace',
    index=False
)

print(" Tabla 'users_api' creada en MySQL")
```

(60) Python

... users_placeholder.csv generado
Tabla 'users_api' creada en MySQL

[] Python

OUTLINE
TIMELINE

Spaces: 4 Cell 37 of 60 Go Live

File Edit Selection View Go ... 9A_IDGS

EXPLORER

- 9A_IDGS
 - PRACTICA_6
 - comentarios_filtrados.csv
 - Practica_6.ipynb
 - salaries.csv
 - users_placeholder.csv

PRACTICA_6 > users_placeholder.csv > data

```
1 id,name,username,email,phone,website,street,suite,city,zipcode,address.geo.lat,address.geo.lng,company.name,company.phrase,company.bs.(af,lng,u
2 1,Leanne Graham,Bret Sincere,gopril.biz,1-770-730-0831,x56442,hildegard.org,Koles Light,Apt. 556,Gwenborough,92988-3874,-37.3159,61.1496,Romague
3 2,Ervin Howell,Antonette Shannam,ameliss-tv,010-692-6593,x89125,anastasia.net,Victor Plains,Suite 879,Wisokyburgh,98566-7771,-42.9589,-34.4618,0
4 3,Clementine Bauch,Samantha,Nathangyesenia.net,1-463-123-4447,ramiro.info,Douglas Extension,Suite 847,McKenziehaven,59590-4157,-68.6102,-47.065
5 4,Patricia Lebsack,Karlanne,Julianne,O'Conner,gory.org,493-178-9623,x156,kale.biz,Hoeger Mall,Apt. 692,South Elvis,53919-4257,29.4572,-164.2998,
6 5,Chelsey Dietrich,Kamren,Lucio,Hettingergannie.ca,(254)954-1289,demarco.info,Skiles Walks,Suite 351,Roscoeview,33263,-31.8129,62.5342,Keebler
7 6,Mrs. Dennis Schulist,Leopoldo Corkery,Karley,Dach,jasper.info,1-477-935-8478,x6438,ola.org,Norberto Crossing,Apt. 950,South Christy,23585-133
8 7,Kurtis Weissnat,Elwyn,Skiles,Telly,Hoeger,billy.biz,210.067.6132,elvis.io,Rex Trail,Suite 280,Hovemouth,58884-1899,24.8918,21.8984,Johns Grou
9 8,Nicholas Runolfsdottir V,Maxime Nienow,Sherwood,grosamond.me,586.493.6943,x148,jacynthe.com,Ellsworth Summit,Suite 729,Aliyaview,45169,-14.399
10 9,Glenna Reichert,Delphine,Chaim,McDermott,dana.io,(775)976-6794,x41286,conrad.com,Dayna Park,Suite 449,Bartholomewbury,76495-3109,24.6463,-168.
11 10,Clementina DuBuque,Moriah,Stanton,Rey,Padberg@karina.biz,824-648-3884,ambrose.net,Kattie Turnpike,Suite 198,Lebsackbury,31428-2261,-38.2386,
12
```

OUTLINE
TIMELINE

CSVLine Query Align

UTF-8 LF CSV Go Live Prettier

Mysql Tabla creada en Ejercicio 1, Carga de Datos

The screenshot shows the MySQL Workbench interface with the 'employees_service' table selected in the Schemas pane. The table contains 5 rows of data. The SQL query editor shows the query: `SELECT * FROM employees.employees_service;`. The Action Output pane shows the execution results.

#	emp_no	birth_date	first_name	last_name	gender	hire_date	years_service	service_level
1	10001	1970-01-01 00:00:00	Juan	García	M	1990-01-01 00:00:00	35	Senior
2	10002	1970-01-02 00:00:00	Maria	Martínez	F	1990-01-02 00:00:00	35	Senior
3	10003	1970-01-03 00:00:00	José	Rodríguez	M	1990-01-03 00:00:00	35	Senior
4	10004	1970-01-04 00:00:00	Luis	López	F	1990-01-04 00:00:00	35	Senior
5	10005	1970-01-05 00:00:00	Ana	Hernández	M	1990-01-05 00:00:00	35	Senior

The Action Output pane shows the following results:

#	Time	Action	Message	Duration / Fetc
1	23:28:57	SELECT * FROM employees.employees_service LIMIT 0, 1000	63 row(s) returned	0.0061 sec / 0.0
2	23:29:00	SELECT * FROM employees.users_api LIMIT 0, 1000	10 row(s) returned	0.0038 sec / 0.0

Tabla en Mysql del Api

The screenshot shows the MySQL Workbench interface with the 'users_api' table selected in the Schemas pane. The table contains 5 rows of data. The SQL query editor shows the query: `SELECT * FROM employees.users_api;`. The Action Output pane shows the execution results.

#	id	name	username	email	phone	website	street
1	Leanne Graham	Bret	Sincere@april.biz	1-770-736-8031 x56442	hildegard.org	Kulas Light	
2	Ervin Howell	Antonette	Shanna@melissa.tv	010-692-6593 x09125	anastasia.net	Victor Plains	
3	Clementine Bauch	Samantha	Nathan@yesenia.net	1-463-123-4447	ramiro.info	Douglas Extension	
4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org	493-170-9623 x156	kale.biz	Hoeger Mall	
5	Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	(254)954-1289	demarco.info	Skiles Walks	

The Action Output pane shows the following results:

#	Time	Action	Message	Duration / Fetc
1	23:28:57	SELECT * FROM employees.employees_service LIMIT 0, 1000	63 row(s) returned	0.0061 sec / 0.0
2	23:29:00	SELECT * FROM employees.users_api LIMIT 0, 1000	10 row(s) returned	0.0038 sec / 0.0
3	23:29:51	SELECT * FROM employees.users_api LIMIT 0, 1000	10 row(s) returned	0.00053 sec / 0.0

