

```
In [2]: #import de librerias a utilizar  
!pip install pyodbc pandas matplotlib
```

```

Collecting pyodbc
  Using cached pyodbc-5.2.0-cp313-cp313-win_amd64.whl.metadata (2.8 kB)
Collecting pandas
  Using cached pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting matplotlib
  Downloading matplotlib-3.9.3-cp313-cp313-win_amd64.whl.metadata (11 kB)
Collecting numpy>=1.26.0 (from pandas)
  Using cached numpy-2.2.0-cp313-cp313-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\diego\appdata\roaming\python\python313\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Using cached pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Using cached tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.1-cp313-cp313-win_amd64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.55.3-cp313-cp313-win_amd64.whl.metadata (168 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.7-cp313-cp313-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: packaging>=20.0 in c:\users\diego\appdata\roaming\python\python313\site-packages (from matplotlib) (24.2)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.0.0-cp313-cp313-win_amd64.whl.metadata (9.3 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: six>=1.5 in c:\users\diego\appdata\roaming\python\python313\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Using cached pyodbc-5.2.0-cp313-cp313-win_amd64.whl (69 kB)
Using cached pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
Downloading matplotlib-3.9.3-cp313-cp313-win_amd64.whl (7.8 MB)
----- 0.0/7.8 MB ? eta -:--:--
----- 1.0/7.8 MB 6.5 MB/s eta 0:00:02
----- 2.6/7.8 MB 7.7 MB/s eta 0:00:01
----- 4.5/7.8 MB 8.0 MB/s eta 0:00:01
----- 6.0/7.8 MB 7.6 MB/s eta 0:00:01
----- 7.3/7.8 MB 7.4 MB/s eta 0:00:01
----- 7.8/7.8 MB 7.2 MB/s eta 0:00:00
Downloading contourpy-1.3.1-cp313-cp313-win_amd64.whl (220 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.55.3-cp313-cp313-win_amd64.whl (2.2 MB)
----- 0.0/2.2 MB ? eta -:--:--
----- 1.3/2.2 MB 6.7 MB/s eta 0:00:01
----- 2.2/2.2 MB 7.1 MB/s eta 0:00:00
Downloading kiwisolver-1.4.7-cp313-cp313-win_amd64.whl (55 kB)
Using cached numpy-2.2.0-cp313-cp313-win_amd64.whl (12.6 MB)
Downloading pillow-11.0.0-cp313-cp313-win_amd64.whl (2.6 MB)
----- 0.0/2.6 MB ? eta -:--:--
----- 1.6/2.6 MB 9.7 MB/s eta 0:00:01
----- 2.6/2.6 MB 8.2 MB/s eta 0:00:00
Downloading pyparsing-3.2.0-py3-none-any.whl (106 kB)
Using cached pytz-2024.2-py2.py3-none-any.whl (508 kB)
Using cached tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Installing collected packages: pytz, tzdata, pyparsing, pyodbc, pillow, numpy, kiwis

```

olver, fonttools, cycler, pandas, contourpy, matplotlib
Successfully installed contourpy-1.3.1 cycler-0.12.1 fonttools-4.55.3 kiwisolver-1.4.7 matplotlib-3.9.3 numpy-2.2.0 pandas-2.2.3 pillow-11.0.0 pyodbc-5.2.0 pyparsing-3.2.0 pytz-2024.2 tzdata-2024.2

```
In [3]: #Recuperación de datos desde SQL Server
import pyodbc
import pandas as pd

# Configuración global de la conexión
server = 'localhost\\SQLEXPRESS' # Cambia si usas otra instancia
database = 'AnálisisFallecidos' # Reemplaza con tu base de datos
username = 'sa' # Usuario de SQL Server
password = 'diegoarmira123' # Contraseña del usuario

# Conexión
conn = pyodbc.connect(
    f"DRIVER={{ODBC Driver 17 for SQL Server}};"
    f"SERVER={server};"
    f"DATABASE={database};"
    f"UID={username};"
    f"PWD={password}"
)

# Consulta SQL para obtener los datos
query = """
SELECT departamento, municipio
FROM [AnálisisFallecidos].[dbo].[CentralTable];
"""

# Leer los datos
data = pd.read_sql(query, conn)

# Mostrar los primeros registros para verificar
print(data.head())
```

C:\Users\diego\AppData\Local\Temp\ipykernel_4760\3800792054.py:27: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.

```
data = pd.read_sql(query, conn)
```

```
departamento municipio
```

0	N/A	N/A
1	N/A	N/A
2	N/A	N/A
3	N/A	N/A
4	N/A	N/A

```
In [10]: # Filtrar valores N/A en los municipios
filtered_data = data[data['municipio'].notna()] # Excluir registros con NaN
filtered_data = filtered_data[filtered_data['municipio'] != 'N/A'] # Excluir regis

# Conteo de registros por municipio después del filtrado
municipios_count = filtered_data['municipio'].value_counts()

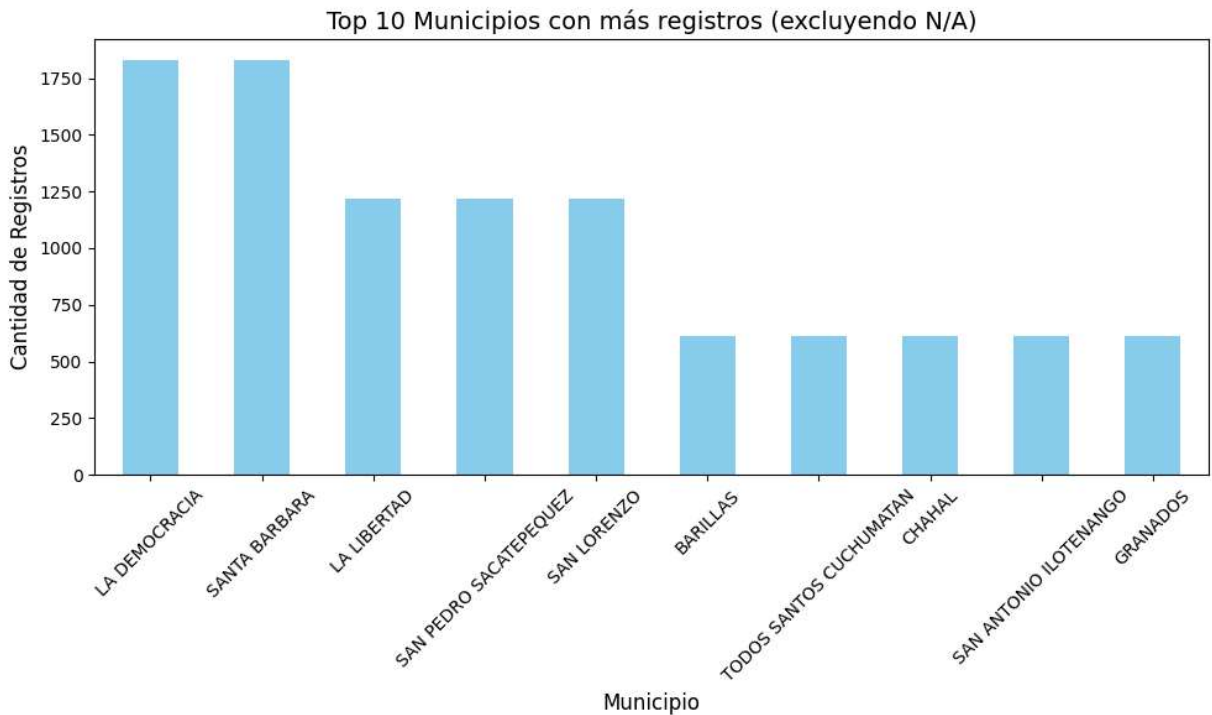
# Obtener los 10 municipios con más registros
```

```

top_municipios = municipios_count.head(10)

# Histograma para Los 10 municipios principales
plt.figure(figsize=(10, 6)) # Ajustar el tamaño del gráfico
top_municipios.plot(kind='bar', color='skyblue')
plt.title('Top 10 Municipios con más registros (excluyendo N/A)', fontsize=14)
plt.xlabel('Municipio', fontsize=12)
plt.ylabel('Cantidad de Registros', fontsize=12)
plt.xticks(rotation=45, fontsize=10) # Rotar las etiquetas
plt.tight_layout() # Ajustar el diseño
plt.show()

```



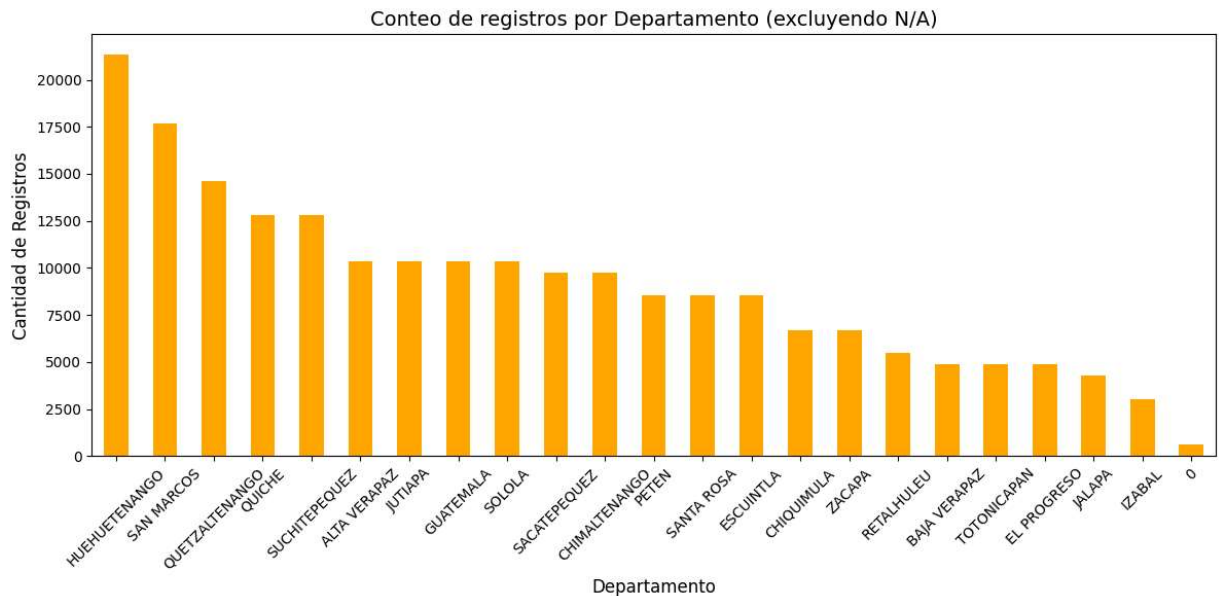
```

In [11]: # Filtrar valores N/A en Los departamentos
filtered_departments = data[data['departamento'].notna()] # Excluir registros con
filtered_departments = filtered_departments[filtered_departments['departamento'] !=

# Conteo de registros por departamento
departamentos_count = filtered_departments['departamento'].value_counts()

# Histograma
plt.figure(figsize=(12, 6))
departamentos_count.plot(kind='bar', color='orange')
plt.title('Conteo de registros por Departamento (excluyendo N/A)', fontsize=14)
plt.xlabel('Departamento', fontsize=12)
plt.ylabel('Cantidad de Registros', fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.tight_layout()
plt.show()

```



Comparativo: Análisis Exploratorio vs. Análisis Descriptivo

Análisis Exploratorio	Análisis Descriptivo
Departamentos	Medidas por Departamento
<ul style="list-style-type: none">- Identificamos que Huehuetenango y San Marcos tienen los registros más altos de fallecidos.- Se encontraron valores N/A en los datos, los cuales se eliminaron para obtener resultados más claros.- Usamos histogramas para observar la distribución de los datos por departamentos y municipios.	<ul style="list-style-type: none">- Huehuetenango lidera con más de 20,000 registros de fallecidos, seguido de San Marcos con aproximadamente 18,000.- Los departamentos con menos fallecidos incluyen Izabal y Jalapa, ambos con menos de 500 registros.- Existe una gran disparidad en los datos: los departamentos con mayor cantidad de fallecidos superan ampliamente a los demás.
Municipios	Medidas por Municipio
<ul style="list-style-type: none">- Municipios como La Democracia y Santa Bárbara destacan en el Top 10 de registros.- Identificamos municipios con conteos bajos y valores vacíos.	<ul style="list-style-type: none">- La Democracia y Santa Bárbara tienen más de 1,700 registros cada uno, liderando el Top 10.- Municipios como Granados tienen significativamente menos registros, mostrando una distribución desigual entre municipios.