

Formatos de Archivos TXT para Importar Datos a una Base de Datos

Aquí tienes 10 formatos en los que un archivo TXT puede estructurar datos para una base de datos (excluyendo CSV y JSON):

1. TSV (Tab-Separated Values)

Datos separados por tabulaciones.

```
ID  Nombre  Edad   Correo
1   Juan    25    juan@email.com
2   Ana 30   ana@email.com
```

Conversión a CSV (Python)

```
import pandas as pd

df = pd.read_csv('archivo.tsv', sep='\t')
df.to_csv('archivo.csv', index=False)
```

2. Pipe-Separated Values

Datos separados por |.

```
ID|Nombre|Edad|Correo
1|Juan|25|juan@email.com
2|Ana|30|ana@email.com
```

Conversión a CSV (Python)

```
import pandas as pd

df = pd.read_csv('archivo.txt', sep='|')
df.to_csv('archivo.csv', index=False)
```

3. XML (Extensible Markup Language)

Formato basado en etiquetas.

```
<personas>
  <persona>
    <ID>1</ID>
    <Nombre>Juan</Nombre>
    <Edad>25</Edad>
    <Correo>juan@email.com</Correo>
  </persona>
  <persona>
    <ID>2</ID>
    <Nombre>Ana</Nombre>
    <Edad>30</Edad>
    <Correo>ana@email.com</Correo>
  </persona>
</personas>
```

Conversión a JSON (Python)

```
import xml.etree.ElementTree as ET
import json

tree = ET.parse('archivo.xml')
root = tree.getroot()

personas = []
for persona in root.findall('persona'):
    personas.append({
        'ID': persona.find('ID').text,
        'Nombre': persona.find('Nombre').text,
        'Edad': persona.find('Edad').text,
        'Correo': persona.find('Correo').text
    })

with open('archivo.json', 'w') as f:
    json.dump(personas, f, indent=4)
```

4. Fixed-Width Format

Cada campo tiene una longitud fija.

ID	Nombre	Edad	Correo
1	Juan	25	juan@email.com
2	Ana	30	ana@email.com

Conversión a CSV (Python)

```
import pandas as pd

df = pd.read_fwf('archivo.txt')
df.to_csv('archivo.csv', index=False)
```

5. Key-Value Pairs

Cada línea representa una clave y un valor.

```
ID: 1
Nombre: Juan
Edad: 25
Correo: juan@email.com

ID: 2
Nombre: Ana
Edad: 30
Correo: ana@email.com
```

Conversión a JSON (Python)

```
import json

data = []
with open('archivo.txt', 'r') as f:
    persona = {}
    for line in f:
        if line.strip():
            key, value = line.split(': ')
            persona[key] = value.strip()
        else:
            data.append(persona)
            persona = {}
    if persona:
        data.append(persona)

with open('archivo.json', 'w') as f:
    json.dump(data, f, indent=4)
```

6. SQL Insert Statements

Archivo con sentencias SQL.

```
INSERT INTO personas (ID, Nombre, Edad, Correo) VALUES (1, 'Juan', 25,
'juan@email.com');
```

```
INSERT INTO personas (ID, Nombre, Edad, Correo) VALUES (2, 'Ana', 30,
'ana@email.com');
```

Conversión a CSV (Python)

```
import pandas as pd
import re

data = []
with open('archivo.sql', 'r') as f:
    for line in f:
        match = re.search(r'\((.*?)\)', line)
        if match:
            data.append(match.group(1).replace('"', '').split(', '))

df = pd.DataFrame(data, columns=['ID', 'Nombre', 'Edad', 'Correo'])
df.to_csv('archivo.csv', index=False)
```

7. YAML (Yet Another Markup Language)

```
personas:
- ID: 1
  Nombre: Juan
  Edad: 25
  Correo: juan@email.com
- ID: 2
  Nombre: Ana
  Edad: 30
  Correo: ana@email.com
```

Conversión a JSON (Python)

```
import yaml
import json

with open('archivo.yaml', 'r') as f:
    data = yaml.safe_load(f)

with open('archivo.json', 'w') as f:
    json.dump(data, f, indent=4)
```

8. Log Format

```
[2024-02-14 10:00:00] ID=1 Nombre=Juan Edad=25 Correo=juan@email.com  
[2024-02-14 10:05:00] ID=2 Nombre=Ana Edad=30 Correo=ana@email.com
```

Conversión a CSV (Python)

```
import pandas as pd  
import re  
  
data = []  
with open('archivo.log', 'r') as f:  
    for line in f:  
        match = re.findall(r'\[(\w+)=([^\ ]+)\]', line)  
        data.append(dict(match))  
  
df = pd.DataFrame(data)  
df.to_csv('archivo.csv', index=False)
```

9. INI File Format

```
[Persona1]  
ID=1  
Nombre=Juan  
Edad=25  
Correo=juan@email.com  
  
[Persona2]  
ID=2  
Nombre=Ana  
Edad=30  
Correo=ana@email.com
```

Conversión a JSON (Python)

```
import configparser  
import json  
  
config = configparser.ConfigParser()  
config.read('archivo.ini')  
  
data = {section: dict(config.items(section)) for section in config.sections()}  
  
with open('archivo.json', 'w') as f:  
    json.dump(data, f, indent=4)
```

10. Custom Delimiter Format

```
ID~Nombre~Edad~Correo  
1~Juan~25~juan@email.com  
2~Ana~30~ana@email.com
```

Conversión a CSV (Python)

```
import pandas as pd  
  
df = pd.read_csv('archivo.txt', sep='~')  
df.to_csv('archivo.csv', index=False)
```