Limpieza y carga del dataset de alojamientos turísticos (Airbnb)

En esta fase se realiza el proceso ETL sobre el dataset de alojamientos turísticos antes de su integración en el modelo documental existente.

El objetivo no es únicamente limpiar los datos, sino adaptarlos semánticamente para que puedan relacionarse con la colección `locales` mediante información geográfica común (distrito).

Carga del dataset

Se carga el archivo JSON descargado desde InsideAirbnb y se transforma a un DataFrame de pandas para facilitar su manipulación y análisis.

Este paso permite inspeccionar la estructura original del dataset y preparar la fase de selección de atributos.

In [1]:
```
import json
import pandas as pd
from pymongo import MongoClient
from pathlib import Path
```

In [2]:
```
ruta = "../../../data/raw/airbnb_listings.json"

with open(ruta, encoding="utf-8") as f:
    data = json.load(f)

df = pd.DataFrame(data)

print("Registros originales:", len(df))
df.head(3)
```

```
Registros originales: 16313
```

Out[2]:

	accommodates	amenities	bedrooms	beds	country	host_id	id	location	name	neighbourhood_group_clear
0	2	[TV, Wireless Internet, Air conditioning, Kitc...	0.0	1.0	Spain	71597	18628	{'coordinates': [40.424715317537384, -3.698638...	Greta Studio Wifi Chueca en Madrid	Ce
1	2	[TV, Cable TV, Wireless Internet, Air conditio...	0.0	1.0	Spain	74966	19864	{'coordinates': [40.413418179848634, -3.706838...	PLAZA MAYOR (wifi, air conditioning)	Ce
2	2	[TV, Cable TV, Internet, Wireless Internet, Ai...	0.0	1.0	Spain	82175	21512	{'coordinates': [40.42492023544748, -3.7134461...	Studio in Plaza de España	Moncloa - Ara

Selección de atributos relevantes

El dataset original contiene numerosos campos orientados a la plataforma Airbnb (host, políticas, descripciones, etc.) que no son relevantes para el análisis urbano-comercial del proyecto.

Por ello se seleccionan únicamente los atributos necesarios:

- Identificador del alojamiento
- Nombre
- Precio
- Capacidad
- Habitaciones y camas
- Tipo de alojamiento
- Número de reseñas
- Servicios (amenities)
- Distrito
- Localización geográfica

Esto reduce ruido en el modelo y mejora la eficiencia de almacenamiento y consulta.

In [3]:
```
df = df[[
    "id",
    "name",
    "price",
    "accommodates",
    "bedrooms",
    "beds",
    "room_type",
    "number_of_reviews",
    "amenities",
    "neighbourhood_group_cleansed",
    "location"
]]

df.head(3)
```

Out[3]:

	id	name	price	accommodates	bedrooms	beds	room_type	number_of_reviews	amenities	neighbourhood_group_cleans
0	18628	Greta Studio Wifi Chueca en Madrid	54.0	2	0.0	1.0	Entire home/apt	37	[TV, Wireless Internet, Air conditioning, Kitc...	Cen
1	19864	PLAZA MAYOR (wifi, air conditioning)	65.0	2	0.0	1.0	Entire home/apt	56	[TV, Cable TV, Wireless Internet, Air conditio...	Cen
2	21512	Studio in Plaza de España	40.0	2	0.0	1.0	Entire home/apt	36	[TV, Cable TV, Internet, Wireless Internet, Ai...	Moncloa - Arav

Homogeneización del esquema

Se renombran los campos para mantener coherencia con la colección `locales`.

El identificador `id` pasa a `_id` para cumplir el estándar de MongoDB y evitar duplicidad de claves.

Además, el campo de barrio/distrito se unifica bajo el nombre `distrito`, que será el punto de relación entre colecciones.

In [ ]:
```
df = df.rename(columns={
    "id": "_id",
    "name": "nombre",
    "price": "precio",
    "accommodates": "capacidad",
    "bedrooms": "habitaciones",
    "beds": "camas",
    "room_type": "tipo",
    "number_of_reviews": "reviews",
    "neighbourhood_group_cleansed": "distrito"
})
```

Limpieza y transformación de datos

Se aplican transformaciones para garantizar consistencia:

- Conversión de valores numéricos
- Eliminación de espacios en blanco
- Normalización del distrito en mayúsculas
- Preparación para futuras agregaciones entre colecciones

Esto permite realizar comparaciones directas con los datos de actividad comercial.

In [5]:
```
df["distrito"] = df["distrito"].str.upper().str.strip()
df["nombre"] = df["nombre"].str.strip()

# convertir a numéricos
df["precio"] = pd.to_numeric(df["precio"], errors="coerce")
df["habitaciones"] = pd.to_numeric(df["habitaciones"], errors="coerce")
df["camas"] = pd.to_numeric(df["camas"], errors="coerce")
df["capacidad"] = pd.to_numeric(df["capacidad"], errors="coerce")
```

Adaptación de coordenadas

Las coordenadas se transforman al estándar GeoJSON utilizado por MongoDB.

Este formato evita inconsistencias cuando faltan valores y permite la creación de índices geoespaciales (2dsphere) para consultas de proximidad.

Se garantiza así la integridad semántica de la información geográfica y la compatibilidad con futuras consultas espaciales.

In [ ]:
```python
def extraer_coords(loc):
    try:
        coords = loc.get("coordinates", None)

        if (
            isinstance(coords, list)
            and len(coords) == 2
            and coords[0] is not None
            and coords[1] is not None
        ):
            return {
                "type": "Point",
                "coordinates": [coords[1], coords[0]]
            }

        return None
    except:
        return None


df["coordenadas"] = df["location"].apply(extraer_coords)
df = df.drop(columns=["location"])
```

Limpieza de servicios (amenities)

El campo amenities es un array de características del alojamiento.

Se eliminan valores inconsistentes generados por la plataforma (por ejemplo, "translation missing") manteniendo únicamente información útil.

El campo se conserva como array, ya que representa una relación natural multivaluada propia de un modelo documental.

In [7]:
```
def limpiar_amenities(lista):
    if not isinstance(lista, list):
        return []
    return [
        a for a in lista
        if "translation missing" not in a.lower()
    ]

df["amenities"] = df["amenities"].apply(limpiar_amenities)
```

Conversión a documentos MongoDB

Tras la limpieza, los registros se transforman en documentos JSON listos para su inserción en la base de datos documental.

In [8]:
```
documentos = df.to_dict(orient="records")

print("Documentos preparados:", len(documentos))
documentos[0]
```

```
Documentos preparados: 16313
```

Out[8]:

```
{'_id': 18628,
 'nombre': 'Greta Studio Wifi Chueca en Madrid',
 'precio': 54.0,
 'capacidad': 2,
 'habitaciones': 0.0,
 'camas': 1.0,
 'tipo': 'Entire home/apt',
 'reviews': 37,
 'amenities': ['TV',
  'Wireless Internet',
  'Air conditioning',
  'Kitchen',
  'Smoking allowed',
  'Elevator',
  'Heating',
  'Family/kid friendly',
  'Hangers',
  'Hot water',
  'Microwave',
  'Coffee maker',
  'Refrigerator',
  'Dishes and silverware',
  'Cooking basics',
  'Long term stays allowed',
  'Well-lit path to entrance'],
 'distrito': 'CENTRO',
 'coordenadas': {'type': 'Point',
  'coordinates': [-3.6986381877058387, 40.424715317537384]}}
```


Exportación a formato JSON Lines

In [9]:
```
#definimos la ruta de salida
OUTPUT_DIR = Path("../../../data/output")
OUTPUT_DIR.mkdir(parents=True, exist_ok=True)

output_file = OUTPUT_DIR / "alojamientos.jsonl"

# Guardar los documentos en formato JSONL
with open(output_file, "w", encoding="utf-8") as f:
```

```
    for doc in documentos:
        f.write(json.dumps(doc, ensure_ascii=False) + "\n")

print("Archivo generado en:", output_file.resolve())
```

```
Archivo generado en: C:\Users\spara\Desktop\ENTREGA_BASE_DE_DATOS_NO_SQL\entrega_codigo\data\output\alojamientos.jsonl
```

Inserción en la base de datos

Los documentos se insertan en una nueva colección denominada `alojamientos`.

Se mantiene separada de `locales` para evitar duplicidad de información y permitir consultas conjuntas mediante agregaciones.

In [10]:
```
client = MongoClient("mongodb://localhost:27017")
db = client["actividad_comercial_madrid"]

col_airbnb = db["alojamientos"]

col_airbnb.drop()

col_airbnb.insert_many(documentos)

print("Insertados:", col_airbnb.count_documents({}))
```

```
Insertados: 16313
```

Verificación de la carga

Se realiza una consulta de comprobación para confirmar que los documentos se han insertado correctamente y que el esquema es consistente.

In [11]:
```
import pandas as pd

pd.DataFrame(list(col_airbnb.find({}, {"_id":1,"nombre":1,"distrito":1,"precio":1}).limit(5)))
```

Out[11]:

	_id	nombre	precio	distrito
0	18628	Greta Studio Wifi Chueca en Madrid	54.0	CENTRO
1	19864	PLAZA MAYOR (wifi, air conditioning)	65.0	CENTRO
2	21512	Studio in Plaza de España	40.0	MONCLOA - ARAVACA
3	21853	Bright and airy room	17.0	LATINA
4	23021	Elegant Apartment in Spain Square	90.0	MONCLOA - ARAVACA