Validación del modelo mediante consultas en MongoDB

En este notebook se desarrollan una serie de consultas sobre la colección `alojamientos` con el objetivo de validar el modelo documental implementado.

Las consultas permiten:

- Verificar la consistencia de los datos tras el proceso de limpieza y carga.
- Evaluar la capacidad del modelo para responder a consultas analíticas reales.

Se utilizarán principalmente pipelines de agregación junto con operadores como `$unwind`, `$group`, `$match`, `$project` y `$lookup`, demostrando la flexibilidad del modelo desnormalizado adoptado.

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

pd.set_option("display.max_columns", None)
pd.set_option("display.max_colwidth", None)
pd.set_option('display.max_rows', None)

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

col_locales = db["locales"]
col_alojamientos = db["alojamientos"]

print("Locales:", col_locales.count_documents({}))
print("Alojamientos:", col_alojamientos.count_documents({}))
```

```
Locales: 151162
Alojamientos: 16313
```

In [2]:
```
def pretty(df):
    """Convierte _id de Mongo en columnas normales sin romper si es string"""
    if "_id" in df.columns:
```

```
        if isinstance(df["_id"].iloc[0], dict):
            id_df = pd.json_normalize(df["_id"])
            df = pd.concat([id_df, df.drop(columns=["_id"])], axis=1)
        else:
            df = df.rename(columns={"_id": "distrito"}) #en este caso usaremos distrito
    return df
```

Objetivo

Tras ampliar el modelo de datos con la colección `alojamientos`, se realizan consultas analíticas para comprobar que la estructura híbrida permite combinar:

- Datos embebidos → terrazas y licencias
- Datos referenciados → alojamientos turísticos

Las consultas demostrarán que el modelo permite análisis territoriales y temporales sin duplicar información.

Consulta A

Total de alojamientos, locales y terrazas por distrito y barrio.

In [ ]:
```
pipeline = [
    {
        "$group": {
            "_id": {
                "distrito": "$desc_distrito_local",
                "barrio": "$desc_barrio_local"
            },
            "total_locales": {"$sum": 1},
            "total_terrazas": {"$sum": {"$size": {"$ifNull": ["$terrazas", []]}}}
        }
    },
    {
        "$lookup": {
            "from": "alojamientos",
            "localField": "_id.distrito",
```

```
            "foreignField": "distrito",
            "as": "alojamientos"
        }
    },
    {
        "$addFields": {
            "total_alojamientos": {"$size": "$alojamientos"}
        }
    },
    {
        "$project": {
            "_id": 1,
            "total_locales": 1,
            "total_terrazas": 1,
            "total_alojamientos": 1
        }
    },
    {"$sort": {"_id.distrito": 1, "_id.barrio": 1}}
]

result = list(col_locales.aggregate(pipeline))
display(pretty(pd.DataFrame(result)).head(10).style.hide(axis='index'))
```

distrito	barrio	total_locales	total_terrazas	total_alojamientos
ARGANZUELA	ACACIAS	1236	97	944
ARGANZUELA	ATOCHA	246	6	944
ARGANZUELA	CHOPERA	878	50	944
ARGANZUELA	DELICIAS	862	71	944
ARGANZUELA	IMPERIAL	826	61	944
ARGANZUELA	LEGAZPI	508	49	944
ARGANZUELA	PALOS DE LA FRONTERA	1377	99	944
BARAJAS	AEROPUERTO	508	4	115
BARAJAS	ALAMEDA DE OSUNA	459	24	115
BARAJAS	CASCO H.BARAJAS	491	28	115

Se agrupan los locales por distrito y barrio y posteriormente se incorporan los alojamientos mediante un `$lookup` por distrito.

Dado que los alojamientos no pertenecen a un barrio concreto, el número será común para todos los barrios del mismo distrito.

Consulta B

Barrios con mayor número de alojamientos y terrazas con licencias concedidas en los últimos dos años

In [5]:
```
from datetime import datetime, timedelta

limite_anos = 5 # deberiamos poner 2 pero para obtener resultados interesantes en el dataset vamos a ampliar el rango a 5 años
fecha_limite = datetime.now() - timedelta(days=365*limite_anos)
fecha_ahora = datetime.now()

pipeline = [

    {"$unwind": "$terrazas"},
    {
```

```
        "$match": {
            "terrazas.Fecha_confir_ult_decreto_resol": {"$lte": fecha_ahora, "$gte": fecha_limite}
        }
    },
    {
        "$group": {
            "_id": "$desc_distrito_local",
            "terrazas_recientes": {"$sum": 1}
        }
    },
    {
        "$lookup": {
            "from": "alojamientos",
            "localField": "_id",
            "foreignField": "distrito",
            "as": "alojamientos_distrito"
        }
    },
{
    "$addFields": {
        "total_alojamientos": {"$size": "$alojamientos_distrito"}
    }
},
{
    "$project": {
        "alojamientos_distrito": 0
    }
},
{
    "$addFields": {
        "total_conjunto": {
            "$add": ["$terrazas_recientes", "$total_alojamientos"]
        }
    }
},
{"$sort": {"total_conjunto": -1}}

]
```

```
result = list(col_locales.aggregate(pipeline))
display(pretty(pd.DataFrame(result)).style.hide(axis='index'))
```

distrito	terrazas_recientes	total_alojamientos	total_conjunto
CENTRO	166	8474	8640
SALAMANCA	161	1096	1257
ARGANZUELA	70	944	1014
RETIRO	59	536	595
CARABANCHEL	68	457	525
LATINA	64	421	485
CIUDAD LINEAL	64	405	469
PUENTE DE VALLECAS	28	307	335
HORTALEZA	48	230	278
USERA	38	181	219
VILLAVERDE	37	93	130
CHAMBERI	129	0	129
BARAJAS	9	115	124
CHAMARTIN	105	0	105
MORATALAZ	23	77	100
VILLA DE VALLECAS	36	48	84
TETUAN	63	0	63
FUENCARRAL-EL PARDO	55	0	55
SAN BLAS-CANILLEJAS	50	0	50
MONCLOA-ARAVACA	45	0	45
VICALVARO	10	0	10

Se analizaron las zonas con mayor intensidad turística reciente combinando:

- Alojamientos turísticos (Airbnb)
- Terrazas con decreto o resolución confirmado en los últimos dos años

El dataset de alojamientos únicamente dispone de información a nivel de distrito, sin correspondencia fiable con el barrio municipal. Por este motivo la agregación se realiza a nivel de distrito en lugar de barrio.

Proceso:

1. Se expanden las terrazas almacenadas como array embebido
2. Se filtran aquellas cuya fecha de confirmación de decreto/resolución es reciente
3. Se agrupan por distrito
4. Se incorporan los alojamientos mediante `$lookup`
5. Se calcula un indicador combinado de presión turística

Este indicador permite identificar las áreas con mayor crecimiento reciente de actividad turística y hostelera.

Optimización del $lookup en agregaciones

Durante la implementación de la consulta combinada entre locales y alojamientos se detectó que el operador `$lookup` devolvía el array completo de documentos relacionados.

Esto generaba dos problemas:

- Resultados extremadamente grandes
- Visualización incorrecta en herramientas analíticas
- Consumo innecesario de memoria

Dado que el objetivo era únicamente contar alojamientos por distrito, no era necesario mantener los documentos embebidos tras la unión.

Por ello, tras calcular el tamaño del array mediante `$size`, se elimina el campo utilizando `$project`.

De esta forma se obtiene un resultado compacto y eficiente, equivalente a un `JOIN + COUNT` en bases de datos relacionales, manteniendo el rendimiento del pipeline de agregación.