Limpieza inicial de datasets

En este notebook se realiza la preparación básica de los datasets antes de su transformación al modelo documental.

El objetivo no es limpiar completamente los datos ni eliminar valores nulos, sino normalizar su formato para permitir su posterior integración en MongoDB.

Se aplican las siguientes transformaciones:

- Eliminación de espacios en blanco en columnas de texto
- Conversión de fechas a tipo datetime
- Normalización de tipos de datos
- Exportación a versión limpia (clean)

In [1]:
```
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)
```

Problemas detectados en los datasets originales

Durante la exploración inicial se observaron varios problemas comunes en datos procedentes de sistemas administrativos:

- Espacios en blanco al inicio o final de los textos
- Fechas almacenadas como cadenas de texto
- Columnas con tipos mezclados (número y texto)
- Valores vacíos representados de distintas formas

Estos problemas no impiden la lectura del CSV, pero sí pueden provocar:

- Errores al generar JSON

- Tipos inconsistentes en MongoDB
- Resultados incorrectos en agregaciones

Por ello se aplica una normalización ligera previa al modelado.

In [2]:
```python
def limpiar_strings(df):
    """Elimina espacios en blanco al inicio y final de todos los textos"""
    for col in df.select_dtypes(include=["object", "string"]).columns:
        df[col] = df[col].astype("string").str.strip()
        df[col] = df[col].replace("", pd.NA)
    return df


def convertir_fechas(df):
    """Convierte automáticamente columnas con fecha"""
    posibles = [c for c in df.columns if "fecha" in c.lower() or "fx_" in c.lower()]

    for col in posibles:
        try:
            df[col] = pd.to_datetime(df[col], errors="coerce", dayfirst=True)
        except:
            pass
    return df


def normalizar_tipos(df):
    """Evita tipos mezclados que rompen Mongo"""
    for col in df.columns:
        if df[col].dtype == "object":
            # intenta convertir a numérico si parece número
            convertido = pd.to_numeric(df[col], errors="ignore")
            df[col] = convertido
    return df

def limpiar_dataset(path):
    df = pd.read_csv(path, sep=';', low_memory=False)

    print("Cargando:", path)
    print("Shape original:", df.shape)
```

```
    df = limpiar_strings(df)
    df = convertir_fechas(df)
    df = normalizar_tipos(df)

    print("Limpieza completada\n")
    return df
```

Carga y limpieza de los datasets

Cada fichero se carga individualmente y se somete al mismo proceso de normalización para garantizar coherencia entre todos los conjuntos de datos.

In [3]:
```
df_terrazas = limpiar_dataset("../../data/raw/terrazas202312.csv")
df_locales = limpiar_dataset("../../data/raw/locales202312.csv")
df_licencias = limpiar_dataset("../../data/raw/licencias202312.csv")
df_actividad = limpiar_dataset("../../data/raw/actividadeconomica202312.csv")
```

```
Cargando: ../../data/raw/terrazas202312.csv
Shape original: (6788, 61)
Limpieza completada

Cargando: ../../data/raw/locales202312.csv
Shape original: (151162, 48)
Limpieza completada

Cargando: ../../data/raw/licencias202312.csv
Shape original: (150829, 49)
Limpieza completada

Cargando: ../../data/raw/actividadeconomica202312.csv
Shape original: (169559, 49)
Limpieza completada
```

Verificación de la limpieza

Se inspecciona la estructura resultante para comprobar:

- Tipos de datos correctos
- Ausencia de espacios residuales
- Conversión adecuada de fechas

In [4]:
```
df_terrazas.head()
```

Out[4]:

	id_terraza	id_local	id_distrito_local	desc_distrito_local	id_barrio_local	desc_barrio_local	id_ndp_edificio	id_clase_ndp_edificio	id_via
0	33	270403150	4	SALAMANCA	404	GUINDALERA	11018870	1	
1	1168	80000291	8	FUENCARRAL-EL PARDO	806	VALVERDE	20057165	1	
2	42	40002970	4	SALAMANCA	403	FUENTE DEL BERRO	11016907	1	
3	1176	40002488	4	SALAMANCA	406	CASTELLANA	11019901	1	
4	56	170000773	17	VILLAVERDE	1704	LOS ROSALES	11126803	1	

In [5]:
```
df_terrazas.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 6788 entries, 0 to 6787
Data columns (total 61 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id_terraza                      6788 non-null   int64
 1   id_local                        6788 non-null   int64
 2   id_distrito_local               6788 non-null   int64
 3   desc_distrito_local             6788 non-null   string
 4   id_barrio_local                 6788 non-null   int64
 5   desc_barrio_local               6788 non-null   string
 6   id_ndp_edificio                 6788 non-null   int64
 7   id_clase_ndp_edificio           6788 non-null   int64
 8   id_vial_edificio                6788 non-null   int64
 9   clase_vial_edificio             6788 non-null   string
 10  desc_vial_edificio              6788 non-null   string
 11  nom_edificio                    6788 non-null   string
 12  num_edificio                    6788 non-null   int64
 13  Cod_Postal                      6788 non-null   int64
 14  coordenada_x_local              6788 non-null   float64
 15  coordenada_y_local              6788 non-null   float64
 16  id_tipo_acceso_local            6788 non-null   int64
 17  desc_tipo_acceso_local          6788 non-null   string
 18  id_situacion_local              6788 non-null   int64
 19  desc_situacion_local            6788 non-null   string
 20  secuencial_local_PC             6788 non-null   int64
 21  Escalera                        123 non-null    string
 22  id_planta_agrupado              6784 non-null   string
 23  id_local_agrupado               5524 non-null   string
 24  coordenada_x_agrupacion         105 non-null    float64
 25  coordenada_y_agrupacion         105 non-null    float64
 26  rotulo                          6788 non-null   string
 27  id_periodo_terraza              6788 non-null   int64
 28  desc_periodo_terraza            6788 non-null   string
 29  id_situacion_terraza            6788 non-null   int64
 30  desc_situacion_terraza          6788 non-null   string
 31  Superficie_ES                   6788 non-null   float64
 32  Superficie_RA                   5610 non-null   float64
 33  Fecha_confir_ult_decreto_resol  6788 non-null   datetime64[us]
 34  id_ndp_terraza                  6788 non-null   int64
 35  id_clase_ndp_terraza            6788 non-null   int64
```

```
 36  id_vial                      6788 non-null   int64
 37  desc_clase                   6788 non-null   string
 38  desc_nombre                  6788 non-null   string
 39  nom_terraza                  6788 non-null   string
 40  num_terraza                  6788 non-null   int64
 41  cal_terraza                  581 non-null    string
 42  desc_ubicacion_terraza       6788 non-null   string
 43  hora_ini_LJ_es               6788 non-null   string
 44  hora_fin_LJ_es               6788 non-null   string
 45  hora_ini_LJ_ra               5610 non-null   string
 46  hora_fin_LJ_ra               5610 non-null   string
 47  hora_ini_VS_es               6788 non-null   string
 48  hora_fin_VS_es               6788 non-null   string
 49  hora_ini_VS_ra               5610 non-null   string
 50  hora_fin_VS_ra               5610 non-null   string
 51  mesas_aux_es                 6788 non-null   int64
 52  mesas_aux_ra                 6549 non-null   float64
 53  mesas_es                     6788 non-null   int64
 54  mesas_ra                     6549 non-null   float64
 55  sillas_es                    6788 non-null   int64
 56  sillas_ra                    6549 non-null   float64
 57  cal_edificio                 523 non-null    string
 58  fx_carga                     6788 non-null   datetime64[us]
 59  fx_datos_ini                 6788 non-null   datetime64[us]
 60  fx_datos_fin                 6788 non-null   datetime64[us]
dtypes: datetime64[us](4), float64(9), int64(21), string(27)
memory usage: 3.2 MB
```

Exportación de datasets normalizados

Los datasets resultantes se guardan en una versión *clean*.

Estos archivos no contienen cambios estructurales ni pérdida de información; únicamente se ha normalizado su formato para permitir su posterior transformación al modelo documental y su inserción en MongoDB.

In [6]:
```
df_terrazas.to_csv("../../data/cleaned/terrazas_clean.csv", index=False)
df_locales.to_csv("../../data/cleaned/locales_clean.csv", index=False)
df_licencias.to_csv("../../data/cleaned/licencias_clean.csv", index=False)
```

```
df_actividad.to_csv("../../data/cleaned/actividad_clean.csv", index=False)

print("Datasets limpios guardados en /data/clean")
```

```
Datasets limpios guardados en /data/clean
```