Universidad Internacional de La Rioja

Máster Universitario en Análisis y Visualización de Datos Masivos

Actividad 1: Limpieza de datos

Trabajo presentado por:	Henry Alejandro Gerena Ricardo
Asignatura:	Bases de Datos para el Big Data
Docente:	Marlon Cárdenas Bonett
Fecha:	01 Diciembre 2025

Contenido

1. Introducción

Viajar genera una enorme cantidad de información: registros de llegadas, gasto, tipos de alojamiento, transporte, actividades, entre otros datos. En este contexto, España aparece como uno de los destinos más atractivos. No solo influyen su clima generalmente templado, la variedad de paisajes o la facilidad para desplazarse dentro del territorio, sino también el idioma compartido con numerosos países de habla hispana y la fama de cercanía y hospitalidad de su población. Todo ello hace que, cuando se habla de turismo internacional, España suela considerarse un país que hay que conocer al menos una vez.

Según las estadísticas oficiales de Frontur y Egatur, En los primeros nueve meses de 2025 han llegado al país cerca de 76,4 millones de visitantes internacionales, y con ello dejando un ingreso aproximado de 13,000 M€.[1] Una cifra importante que obliga a las empresas del sector a reaccionar con rapidez, entender qué buscan las personas y apoyar sus decisiones en la información disponible. Por tal motivo surge el presente trabajo: adoptar la perspectiva de un analista de datos y aprovechar distintas fuentes públicas para estudiar el comportamiento del sector turístico.

El turismo, como cualquier otro sector económico, tiene características propias que requieren interpretar los datos con cuidado y en contexto. Además, al trabajar con fuentes abiertas, es frecuente encontrarse con problemas de calidad: datos faltantes, errores en digitación, falta de estandarización. Por ello, más allá de describir y explorar los datos, para la presente actividad nos centraremos en la calidad del dato y en su limpieza, para que las conclusiones que se extraigan de esta información aporte conocimiento para poder tomar decisiones fundamentadas.

[1] https://www.mintur.gob.es/es-es/GabinetePrensa/NotasPrensa/2025/Paginas/gasto-turistas-internacionales-espana-supera-los-105000-millones-hasta-septiembre.aspx

2. Selección de los datos

Luego de dar una introducción y ponernos en contexto para desarrollar la presente actividad, pasamos a la selección del conjunto de datos (de ahora en adelante dataset) tres conjuntos de datos representativos del sector turístico, estas fuentes permiten analizar distintos aspectos: comportamiento del turista, capacidad y oferta de alojamientos, y movilidad a través del transporte aéreo.

Tabla 1 Catalogo de datos seleccionados

Dataset	Origen	Justificación	Enlace	Fichero
Actividades turísticas Madrid	Catálogo de ayuntamientos	Para analizar comporta-miento y preferencias de turistas	datos.madrid.es	EstudioAtencion Visitante2024.xlsx
Alojamientos turísticos	Comunidad de Madrid	Para entender ocupación y tipos de alojamiento	datos.comunidad.madrid	alojamientos_turisticos.csv
Información de vuelos	Transporte aéreo AENA	Para análisis de demanda y movilidad turística	AENA Info Vuelos	infovuelos_sample.csv

2.1. Dataset 1. Actividades turísticas Madrid

Contiene 998 registros y 43 columnas, con información sobre la valoración de los usuarios de los Servicios de Atención al Visitante de Madrid (SAV). Los datos contienen información relacionada a la percepción en la calidad del servicio, instalaciones, personal, organización y recursos disponibles.

2.2. Dataset 2. Alojamientos turísticos

Contiene 10,974 registros y 14 columnas, este dataset almacena información de distintos tipos de alojamientos en la Comunidad de Madrid, con información sobre nivel de categoría, ubicación. Este dataset nos permite analizar sobre las opciones de alojamiento turístico y posibles tendencias de ocupación.

2.3. Dataset 3. Información de vuelos

Contiene 39,102 registros y 25 columnas, esta información fue tomada de un repositorio web de info "Infovuelos" de AENA, almacena registros de vuelos con origen o destino en aeropuertos españoles durante un periodo. Permite analizar la movilidad turística, la demanda de vuelos y posible tendencia de destinos.

3. Apartado 1 - Problemas detectados y limpieza de datos

A continuación, se describen los principales problemas encontrados sobre la calidad de cada conjunto de datos, así como una forma de detectarlos, medir su impacto y propuesta de solución. Una vez estos dataset pasen por el proceso de limpieza permitirán segmentar turistas, analizar alojamientos por geolocalización o análisis de correlación en retrasos de vuelos.

3.1. Dataset 1. Actividades turísticas Madrid

Problema 1: Nombres de columnas largos y poco manejables

Qué problema es: los nombres de varias columnas son frases que contienen espacios, caracteres especiales como tildes y signos de interrogación. Ejemplos:

```
1. [129 caracteres] ¿Qué instalaciones del Servicio de Atención e Información Turística ha visitado? (Seleccione todas las opciones que correspondan)
```

Cómo se ha detectado: luego de cargar el fichero en un `DataFrame` de la librería pandas y revisar `df.columns`, se observa que un número significativo de columnas contienen tildes, espacio y signos ¿ ,?.

Por qué es un problema: va en contra de las recomendaciones de la guía de calidad para datos abiertos, que aconseja nombres cortos pero que describan su información, sin caracteres especiales, sin espacios.

Forma de solucionarlo: definir una función con un patrón de renombrado sistemático (minúsculas, caracteres ASCII, sin caracteres especiales, reemplazando los espacios por guiones bajo *(snake_case))*.

Justificación: se alinean los datos con buenas prácticas para la recolección de datos, y poder manipularlo de una manera que se alinea a un marco de trabajo ya establecido por gran parte la industrial que trabaja con datos, por otro lado, mejora la legibilidad, reusabilidad y su futura conversión a formatos JSON.

Problema 2: Mezcla de tipos de datos en dominio numérico

Qué problema es: en las variables de satisfacción se mezclan valores numéricos (1-10) con palabras abreviadas *NS/NC* y en algunos otros casos palabras que no aplican a la descripción de la variable, debido que solicita información de satisfacción en una escala, esta se interpreta como una variable cuantitativa ordinal y no podrá llevar texto. Ejemplos:

```
Satisfacción con la información facilitada sobre     Satisfacción con la información facilitada
Visitas Guiadas
0                                                 8                                                NS/NC
```

Cómo se ha detectado: analizando los tipos de datos (`dtype object`) y posteriormente utilizando una función para revisar los valores únicos (`df[col].unique()`).

Por qué es un problema: impide realizar análisis a estas variables numéricas de forma directa impidiendo realizar análisis estadístico por falta de homogeneidad.

Forma de solucionarlo: identificar los valores no numéricos y recodificarlos con valor nulo (null), posteriormente trasformar la columna a numérica.

Justificación: se habilita la variable para realizar análisis estadístico como medias, desviaciones, comparaciones, de forma consistente proporcionando un sentido de coherencia y consistencia de manera confiable sin que este genere error a quien desee analizar los datos.

3.2. Dataset 2: Alojamientos turísticos

Problema 3: Inconsistencia y falta de normalización en campos de dirección

Qué problema es: los campos de dirección (via_tipo, via_nombre, numero, planta, puerta), presentan mezcla de formatos no normalizados, adicionalmente aparecen mayúscula/minúsculas en la variable puerta, codificaciones variadas de planta/puerta. Ejemplos:

```
# Puerta, array([nan, 'D', '2', 'EXT_IZ', 'A', 'Dcha.', 'Izda.', 'PTA. I', 'IZDA'])
# Planta, array([nan, '3ª', '8ª', '4º', '2', '3', '2ª', '4', '5º', '3º 4º'])
```

Cómo se ha detectado: por medio de análisis exploratorio con pandas, implementando la función `df[col].value_counts()` y revisando muestras.

Por qué es un problema: dificulta la geocodificación, el análisis espacial y la integración con datos oficiales.

Forma de solucionarlo: definir los valores oficiales y homologar los datos nuevos con esa fuente oficial, separar los valores numéricos de planta y puerta.

Justificación: mejorar el análisis geoespacial por ende la calidad de la fuente, además que aporta valor para quien lo requiera pudiendo sectorizar si así lo requiere un análisis zonal, evita posibles errores en duplicación de datos y facilita la integración futura con otras fuentes.

Problema 4: Codificación poco clara y redundante de tipo y categoría de alojamiento

Qué problema es: las columnas `alojamiento_tipo` y `categoria` mezclan información, por un lado, en `alojamiento_tipo` está el valor `PENSION`, y en `categoria` esta `2-PENSION`. Ejemplos:

```
   alojamiento_tipo       categoria               denominacion
0             HOTEL         3-HOTEL               GRAN LEGAZPI
```

Cómo se ha detectado: revisión de valores único con values_count() de alojamiento y categoría, además analizando agrupaciones y observación de patrones con la función groupby(alojamiento_tipo, categoría)

Por qué es un problema: genera redundancia al momento de analizar los datos y posible confusión para quien desee utilizar el conjunto de datos, no esta claro el nivel responsabilidades y dificulta filtros y análisis.

Forma de solucionarlo: mantener `alojamiento_tipo` como variable de tipo, normalizado (`HOTEL, HOSTAL PENSION, CASA_HUESPEDES`), dividir categoría en 2 campos uno con valor numérico (`1 ,2 ,3`) y `categoria_tipo` (`HOTEL, HOSTAL, PENSION`).

Justificación: estamos aplicando el principio de atomicidad sin redundancia al dividir responsabilidades y con la `categoria`, mejorando la estructura del dataset y facilitando la agregación y comparación de estas variables.

3.3. Dataset 3: Información de vuelos

Problema 5: Fechas y horas separadas y con formatos inconsistentes

Qué problema es: las fechas y horas de llegadas y salidas se almacenan en campos separados y con formatos distintos, se observa que las fechas de operación usan `dd/mm/aa`. Ejemplos:

```
dep_date   dep_time   arr_date   arr_time              timestamp
13/04/18      10:00   13/04/18      10:10    2018-04-13 10:33:45
```

Cómo se ha detectado: por medio de observación directa se ejecutó la función `df.info()` las variables que son de tiempo no se cargaron con su tipo de dato, eso es indicio de posible error en la fuente y posteriormente se observaron los datos con la función `df.sample(10).`

Por qué es un problema: el objetivo de este dataset va en función del tiempo, si tenemos errores o formatos diferentes no podremos encontrar valor, se dificulta el análisis y seria propenso a aumentar el riesgo entorpeciendo su utilidad en caso de querer relacionarlo con otra fuente de datos.

Forma de solucionarlo: primero convertir las fechas que no tengan el formato YYYY-MM-DD, posteriormente unificar y darle el formato debido a las fechas y horas.

Justificación: al corregir las variables temporales con un formato datetime estándar (ISO 8601) facilitara su análisis, precisión y mantenimiento para que se puedan validar momentos concretos en el tiempo y poder tomar decisiones informadas e integrar con otro recurso.

Problema 6: Campo de estados de vuelo como descripciones, no estructurado

Qué problema es: los estados del vuelo salida/llegada (`dep_status`, `arr_status`) se encuentran como texto libre, no estructurado combinando información importante con descripciones y dificultando su posterior análisis. Ejemplo:

```
dep_date   dep_time                          dep_status      arr_date   arr_time
13/04/18      10:00   El vuelo ha despegado a las 10:03      13/04/18      10:10
```

Cómo se ha detectado: mediante inspección directa de los datos `dep_status` y `arr_status` con la función `values_count()` se observan valores con frases descriptivas en texto libre.

Por qué es un problema: impide clasificar los vuelos por estado, no se puede medir la cantidad, ni los diferentes estados a lo largo del tiempo para detectar patrones.
Forma de solucionarlo: descomponer la variable y dejar en una nueva variable los valores que si describen el estado, por ejemplo (PROGRAMADO, DESPEGADO), y en otra el texto.
Justificación: pasamos de tener un texto libre que no nos aporta información, ha tener una variable consistente y analizable con claridad semántica.

4. Apartado 2 - Propuesta de formato JSON

Dataset 1. Actividades turísticas Madrid

Diseño: se propone un modelo para representar cada respuesta de la encuesta como un documento independiente, con esto facilitamos su almacenamiento y consultas posteriores en una base de datos NOsql como MongoDB, adicionalmente es utilizado el patrón *embebido* para agrupar toda la información de la encuesta:

- Se ha incluido un identificador único y secuencial para cada documento con el nombre `respuesta_id`
- Se añadió un campo de tipo `timestamp` en formato ISO 8601 (YYYY-MM-DDTHH:MM:SS), que indica el momento que fue creado el documento JSON.
- El contenido de la encuesta se limpió y normalizo en todas las variables de encuesta, se anidan en un objeto con el nombre `datos`.
- Las variables numéricas se almacenan en rangos de 1 a 10, para los valores faltantes se envía nulo (null).

Justificación: cuando se analizan respuestas de encuestas, se necesita la información de las variables descriptivas (edad, país) como las valoraciones del servicio (trato, instalaciones), implementando el patrón embebido permite realizar consultas directas extrayendo el documento completo con la información disponible.

- Dataset 3: Información de vuelos

Diseño: se propone un modelo basado en un documento por vuelo, utilizando un patrón *embebido* para los datos fuertemente relacionados con el vuelo:

- La información de salida (`departure`), llegada (`arrival`) y del clima (`weather`), esta información fue embebida en el mismo documento de vuelo.

Justificación: cuando se consulta un vuelo, casi siempre todos estos datos son retornados en la misma consulta y evita tener que relacionar con "joins" la información, adicionalmente no se emplea el patrón referenciado por que el tamaño de cada documento no supera el tamaño recomendado que son 16MB.

5. Apartado 3 - Metodología de limpieza implementada

Se implemento la siguiente metodología, estructurada en las siguientes etapas, aplicada a los conjuntos de datos, en este análisis son 3:

1. Carga y exploración inicial

- Importación de los ficheros CSV/xlsx con pandas.
- Revisión:
  - Número de filas y columnas.
  - Tipos de datos detectados por la herramienta `df.info()`.
  - Consulta de las primeras filas y valores únicos de las variables.
- Aplicación de código exploratorio (value_counts, unique, isnull, patrones) para localizar:
  - Valores nulos.
  - Codificaciones irregulares texto en variables numéricas.
  - Valores duplicados
  - Problemas de formato en fechas, textos y variables categóricas.
  - Mezcla de tipos en una misma columna.

2. Definición de reglas de limpieza

- Definir reglas particulares para cada caso de uso:
  - Estandarización de valores y formatos.
  - Separación de atributos atómicos (texto/numérico, texto/fecha).

3. Aplicación de las transformaciones

- Implementación de reglas con código Python (ejemplo., creación de nuevas columnas derivadas, homologación de valores, normalización de formatos implementando estándares).

4. Validación posterior

- Verificación:
    - Que se apliquen adecuadamente las transformaciones (por ejemplo, NS/NC en campos numéricos).
    - Los formatos de fecha y hora son homogéneos y correctos.
- Comprobación de que los datos pueden representarse sin ambigüedades en los modelos JSON definidos.

6. Apartado 4 - Mejoras propuestas en los conjuntos de datos

1. Estandarización y documentación:
    - Implementar un diccionario de datos (nombres, tipos, dominios), en las fuentes donde fueron extraídos estos datasets, no contaban con metadatos dificultando la elección de los valores correctos para cada variable.
    - Unificar formatos (fechas ISO 8601, nulos NULL).
2. Atomicidad (ACID) y control de tipos de valores:
    - Usar códigos separados para categóricas, evitar texto libre.
    - Asegurar campos atómicos, separando significados.
    - Justificación: Facilita el análisis, la interoperabilidad y el modelado en JSON.
3. Propuestas por dataset

- Dataset 1 – Encuesta a visitantes
    - Diseñar formularios con normalización en nombres de preguntas.
    - Estandarizar selección múltiple (listas de códigos).
- Dataset 2 – Alojamientos turísticos
    - Validar y catalogar direcciones.
    - Separar "tipo de alojamiento" de "nivel de categoría" en campos distintos.
- Dataset 3 – Información de Vuelos
    - Almacenar horas programadas/reales en datetime estándar.
    - Codificar estados de vuelo con catálogo cerrado.
    - Justificación: Permite análisis temporal robusto y clasificación clara.

7. Tamaño de la muestra entregada

Los ficheros generados para esta actividad contienen una muestra de 100 documentos cada uno, por temas prácticos, con el objetivo de facilitar su manejo en el momento de depositarlo en el repositorio de la universidad.