

Fuentes y Preparación de Datos

Data Science (CC3084) - 2026

Fuentes y Preparación de Datos

Semestre 02, 2026

Recordatorio: Tipos de Datos

Antes de analizar o modelar, debemos clasificar las variables. El tipo de dato determina qué técnicas y qué transformaciones podemos aplicar.

Variables Cualitativas

- Nominales: categorías sin orden (ej. género, color, país).
- Ordinales: categorías con un orden (ej. nivel educativo, satisfacción).

Variables Cuantitativas

- Discretas: valores contables (ej. número de hijos).
- Continuas: valores medibles en un rango (ej. peso, edad, ingresos).

Niveles de medición

Nivel	Orden	Distancia	Cero real	Ejemplo	----- ----- ----- -----
-----	Nominal	No	No	No	color, país
	Ordinal	Sí	No	No	
	Intervalo	Sí	Sí	No	temperatura °C
	Razón	Sí	Sí	Sí	

edad, ingresos |

Estructura de los datos

- Estructurados: tablas con filas y columnas (bases de datos, CSV).
- Semiestructurados: tienen etiquetas pero no un esquema rígido (JSON, XML).
- No estructurados: texto libre, imágenes, audio, video.

Fuentes de Datos

Una fuente de datos es el origen de donde provienen los datos que vamos a analizar.

Según su origen

- Primarias: recolectadas por nosotros mismos para un objetivo específico (encuestas, experimentos, sensores).
- Secundarias: recolectadas por terceros y reutilizadas (open data, estudios previos, datasets públicos).

Según su ubicación

- Internas: generadas dentro de la organización (ventas, logs, CRM).
- Externas: provienen de fuera (redes sociales, portales públicos, proveedores).

Ejemplos comunes

- Archivos: CSV, Excel, JSON.
- Bases de datos: SQL (PostgreSQL, MySQL, SQLite).
- APIs web (REST).

- Web scraping.
- Sensores e IoT.
- Portales de datos abiertos (open data).
- Bitácoras (logs) y encuestas.

Obtención de Datos de Distintas Fuentes

Formatos comunes

| Formato | Uso típico | |-----|-----| | CSV | Tablas simples, intercambio | | JSON | APIs, datos semiestructurados | | XLSX | Hojas de cálculo | | XML | Documentos con etiquetas | | Parquet | Datos columnares, grandes volúmenes | | SQL | Bases de datos relacionales |

Leer archivos con pandas

```
import pandas as pd

df_csv = pd.read_csv("datos.csv")
df_excel = pd.read_excel("datos.xlsx")
df_json = pd.read_json("datos.json")
```

Leer desde una base de datos

```
import pandas as pd
import sqlite3

con = sqlite3.connect("base.db")
df = pd.read_sql("SELECT * FROM clientes", con)
```

Consumir una API

```
import requests

respuesta = requests.get("https://api.ejemplo.com/datos")
datos = respuesta.json()
```

Web scraping

```
import requests
from bs4 import BeautifulSoup

html = requests.get("https://sitio.com").text
soup = BeautifulSoup(html, "html.parser")

titulos = [h.text for h in soup.select("h2")]
```

Datos Crudos vs. Procesados

Datos crudos (raw)

- Tal como se obtienen de la fuente, sin ninguna modificación.
- Suelen tener errores, formatos inconsistentes y valores faltantes.
- Regla de oro: nunca se sobrescriben los datos crudos.

Datos procesados

- Ya fueron limpiados y transformados.
- Están listos para el análisis y el modelado.

El flujo de preparación

```
Datos crudos -> Limpieza / Transformación -> Datos procesados -> Análisis
```

Se suele decir que el 80% del trabajo de un análisis de datos se va en obtener y limpiar los datos, no en modelar.

Limpieza de Datos

Problemas comunes

- Valores faltantes (NaN).
- Registros duplicados.
- Tipos de dato incorrectos.
- Formatos inconsistentes (fechas, mayúsculas, unidades).
- Valores atípicos (outliers).
- Errores de captura y typos.

Técnicas frecuentes

- Faltantes: eliminar filas o imputar (media, mediana, moda).
- Duplicados: eliminarlos.
- Estandarizar formatos (fechas, texto, unidades).
- Convertir al tipo de dato correcto.
- Tratar outliers: revisarlos; no siempre se eliminan.

Ejemplo con pandas

```
df = df.drop_duplicates()
df["edad"] = df["edad"].fillna(df["edad"].median())
df["nombre"] = df["nombre"].str.strip().str.title()
df["fecha"] = pd.to_datetime(df["fecha"])
```

Datos ordenados (Tidy Data)

Hadley Wickham propone tres principios para estructurar los datos de forma que sean fáciles de analizar:

1. Cada variable es una columna.
2. Cada observación es una fila.
3. Cada tipo de unidad observacional es una tabla.

Code Book (Definición de Variables)

Definición

Un code book (o diccionario de datos) es un documento que describe cada variable del conjunto de datos, de modo que sea auto-explicativo para cualquier persona ajena al proyecto.

Qué incluye por cada variable

- Nombre de la variable.
- Descripción / significado.
- Tipo de dato.
- Nivel de medición.
- Unidades.
- Valores permitidos o codificación.
- Rango o ejemplo de valores.

Ejemplo

Variable	Descripción	Tipo	Medición	Valores	
				id cliente	Identificador único

string | Nominal | 10 caracteres | | edad | Edad del cliente en años | int | Razón | 18-99 | | genero | Género | string | Nominal | F, M, Otro | | satisfaccion | Nivel de satisfacción | int | Ordinal | 1-5 |

Documentación y Código Utilizado

Reproducibilidad

Cualquier persona (incluido tú en el futuro) debe poder repetir el análisis y obtener el mismo resultado.

Qué documentar

- De dónde vienen los datos: fuente, fecha y versión.
- Qué transformaciones se aplicaron y por qué.
- Los supuestos y las decisiones tomadas.

Herramientas

- README del proyecto.
- Notebooks (Jupyter) que combinan narrativa y código.
- Comentarios claros en el código.
- Control de versiones con git.
- Separar los datos crudos y procesados en carpetas distintas.

Estructura de proyecto sugerida

```
proyecto/  
├─ data/  
│   └─ raw/          # datos crudos, nunca se modifican
```

```
|   └─ processed/    # datos limpios listos para analizar
|─ notebooks/
|─ src/
|   └─ limpieza.py
|─ codebook.md
|─ README.md
```

Resumen

- Clasifica los tipos de datos antes de analizar.
- Las fuentes pueden ser primarias o secundarias y llegar en muchos formatos.
- Preserva siempre los datos crudos; trabaja sobre copias procesadas.
- La limpieza es el 80% del trabajo; busca datos ordenados (tidy data).
- Documenta cada variable en un code book.
- Documenta el proceso y el código para que sea reproducible.