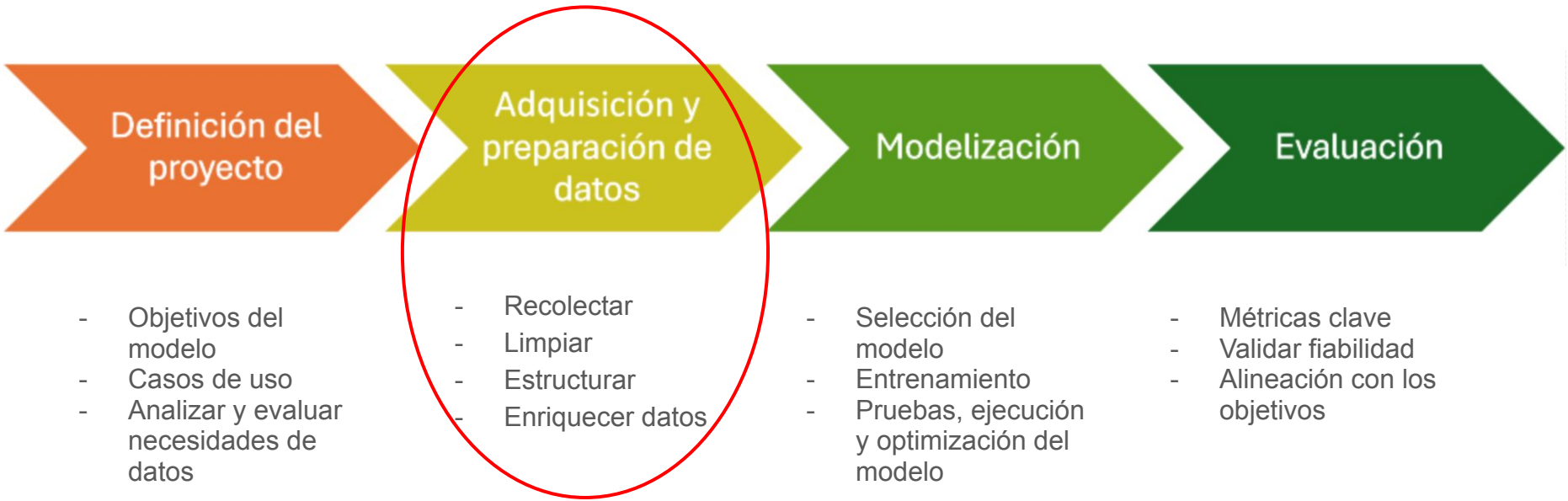
Procesos ETL

Tema 6: Limpieza, transformación y normalización de datos. Parte III

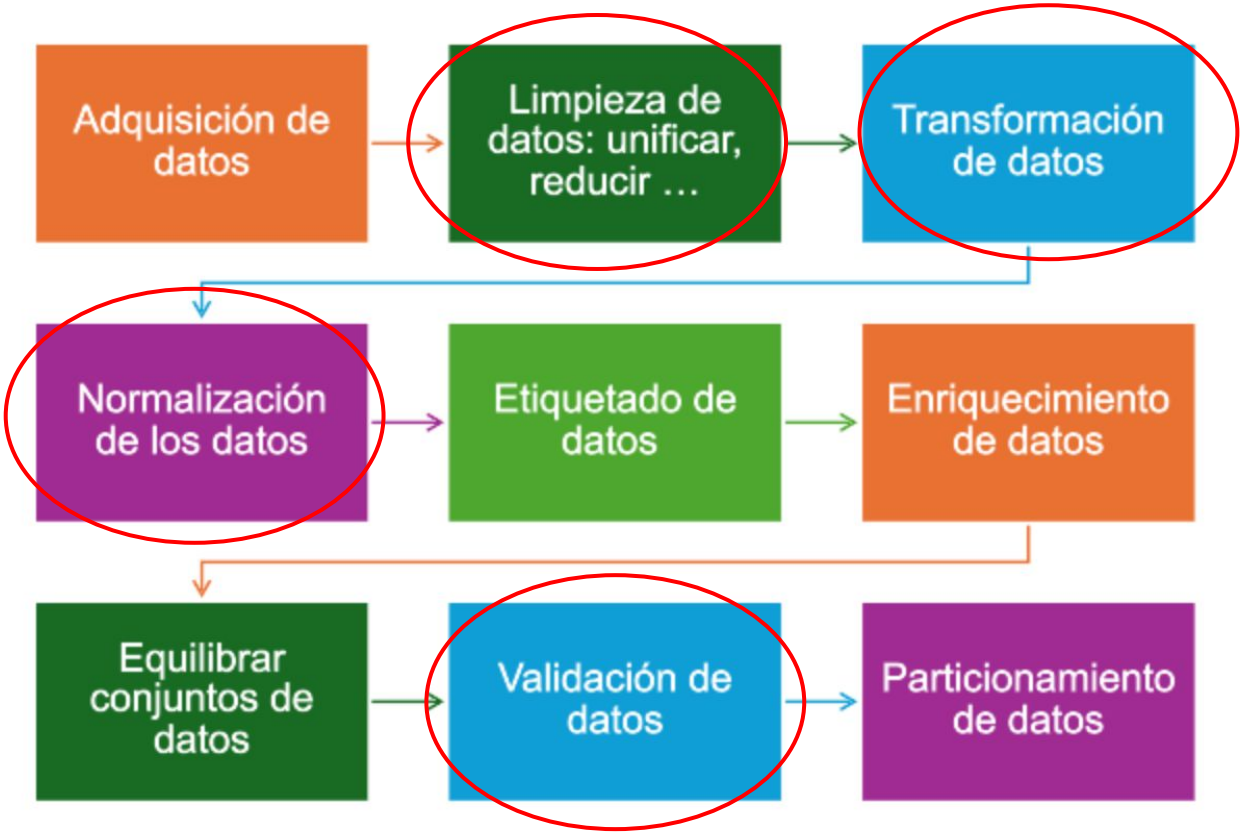
Adquisición y preparación de datos

Universitat d'Alacant
Universidad de Alicante

Ciclo de vida del proyecto IA



Preparación de datos



Transformación y normalización

El objetivo de la **normalización** es escalar las características de un conjunto de datos a un rango común, asegurando que ninguna variable domine a otras debido a su magnitud.

Esto es crucial para mejorar la **equidad** y el **rendimiento** de los modelos.

Métodos de normalización:

- ***Min-Max Scaling***: escala para ajustarlos a un rango normalmente [0,1]
- ***Z-score*** *Standard Scaler*: ajusta los datos para que tengan una media de cero y una desviación estándar de uno

Gestión e identificación de valores atípicos (**Outliers**), son valores que se desvían significativamente de la mayoría de los datos.

Normalización de datos

Normalización de datos *Min-Max Scaling*

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

```
import numpy as np

edad = np.array([5, 16, 18, 15, 25, 35, 50])

print('Min-Max scaling: ',np.round((edad - edad.min())/(edad.max() - edad.min()),2))


Min-Max scaling:  [0.   0.24 0.29 0.22 0.44 0.67 1.  ]
```

Normalización de datos

Z-score (Standard Scaler)

$$x' = \frac{x - \bar{x}}{\sigma}$$

```
import numpy as np
edad = np.array([5, 16, 18, 15, 25, 35, 50])
print('Z-score: ',(edad - edad.mean())/edad.std())
print('Z-score: ',np.round(((edad - edad.mean())/edad.std()), 3))


Z-score:  [-1.33308858 -0.53736904 -0.39269276 -0.60970718  0.11367422  0.83705562
1.92212772]

Z-score:  [-1.333 -0.537 -0.393 -0.61   0.114  0.837  1.922]
```

Outliers

Un valor atípico ***outlier***, es una **observación que se desvía** significativamente de la mayoría de los datos en un conjunto.

Pueden ser el resultado de **errores** en la recolección de datos o representar **eventos inusuales** pero reales.

Es importante detectar y gestionar adecuadamente los valores atípicos:

- Pueden **distorsionar los resultados** estadísticos. Por ejemplo ciertas medidas como la media son sensibles a los valores atípicos
- **Afectar el rendimiento** de los modelos. Un valor atípico puede sesgar el modelo, haciendo que su rendimiento de predicción sea menos preciso.

Se pueden eliminar, transformar o reemplazar.

Outliers

Tipos y posibles acciones:

- **Error** en la recolección de datos, errores de entrada e incluso de codificación
  - →Se tratan en el filtrado de datos, se eliminan o se marcan como datos faltantes
- Datos en un **acontecimiento extraordinario**
  - →Se puede considerar no representativo y se podría eliminar
- Datos **extraordinarios** para los que **no se tiene explicación**
  - →Hacer el experimento con y sin estas observaciones y analizar la influencia sobre los resultados.
- **Conjunto** de observaciones cuyos **valores se desvía de la norma** formando un patrón distinto aunque cada valor podría parecer “normal”
  - →Se podría tener en cuenta pero es importante analizar cómo influyen.

Outliers

Problemática

1. Son elementos que pueden no ser representativos de la población pudiendo distorsionar seriamente el comportamiento de los contrastes estadísticos.

2. Pueden ser indicativos de las características de un segmento de la población no tenido en cuenta, por lo que puede representar falta de representatividad de la muestra.

Outliers

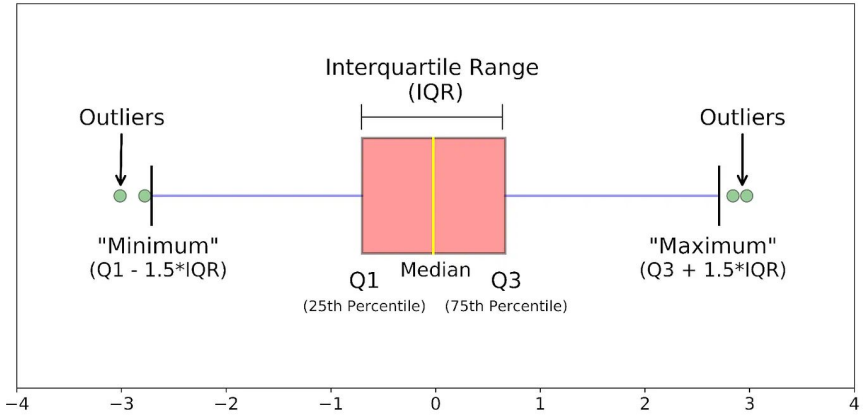
Gestionar los valores atípicos

```
edad = [5, 16, 9, 18, 15, 25, 35, 50, 52, 45, 48];

edadMedia = 28.9


edad = [5, 16, 9, 18, 15, 25, 35, 50, 52, 45, 48, 300];

edadMedia = 42.25

//La media quizá ya no es un valor representativo del conjunto
```

Outliers

Gestionar los valores atípicos

Diagrama de cajas y rango intercuartílico (IQR)



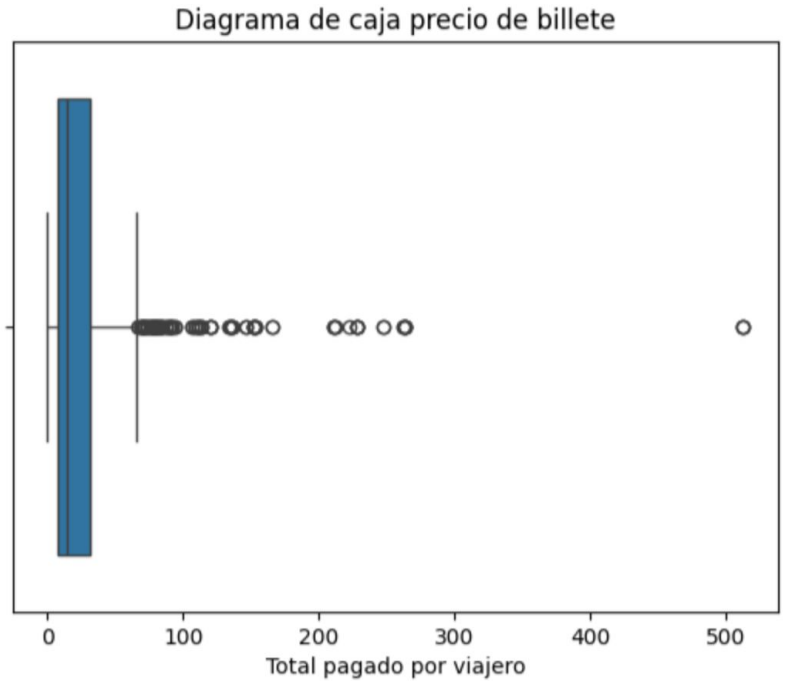
Q1 primer cuartil 25% Q3 tercer cuartil 75% IQR = Q3-Q1 Límite inferior = Q1-1,5*IQR Límite superior = Q3+1,5*IQR	Outliers están fuera de los límites inferior o superior

Outliers

Gestionar los valores atípicos *Outliers*

Diagrama de cajas

```
# Crear un diagrama de caja
g = sns.boxplot(data = titanic, x = 'fare')

# Título y eje X
g.set_title('Diagrama de caja precio de billete')
g.set_xlabel('Total pagado por viajero')
```



Outliers

Gestionar los valores atípicos *Outliers* - Rango intercuartílico (IQR)

```
# Rango intercuartílico (IQR)
# Calcular Q1 Q3 percentiles
q3 = titanic['fare'].quantile(0.75)
q1 = titanic['fare'].quantile(0.25)

# Obtener IQR
iqr = q3 - q1

# Limites superior e inferior
limitSup = q3 + (1.5 * iqr)
limitInf = q1 - (1.5 * iqr)

# Obtener el subconjunto de este dataset
outliers = titanic[(titanic['fare'] < limitInf) | (titanic['fare'] > limitSup)]
outliers.head()

#Obtener número de outliers
print(f'Outliers: {len(outliers)}')


Outliers: 116
```

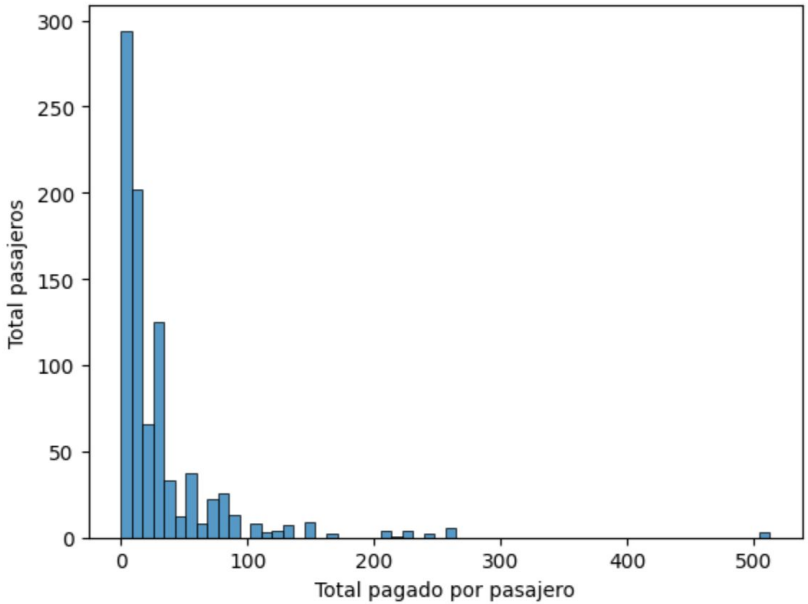
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
27	0	1	male	19.0	3	2	263.0000	S	First	man	True	C	Southampton	no	False
31	1	1	female	NaN	1	0	146.5208	C	First	woman	False	B	Cherbourg	yes	False
34	0	1	male	28.0	1	0	82.1708	C	First	man	True	NaN	Cherbourg	no	False
52	1	1	female	49.0	1	0	76.7292	C	First	woman	False	D	Cherbourg	yes	False

Outliers

Gestionar los valores atípicos *Outliers*

Histograma

```
# Crear histograma
g = sns.histplot(data = titanic, x = 'fare')

# Añadir etiquetas
g.set_xlabel('Total pagado por pasajero')
```



Estrategias y técnicas de limpieza, transformación y normalización

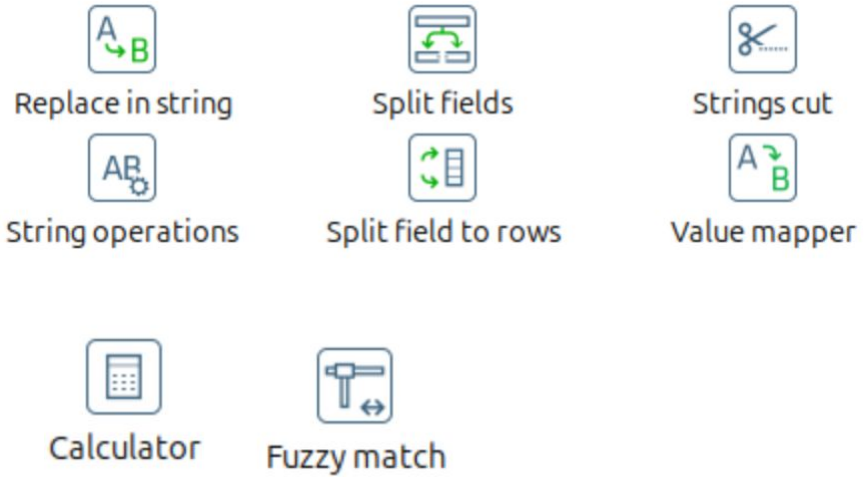
Estrategias y técnicas para la limpieza de datos

- Uso de expresiones regulares para encontrar y reemplazar formatos específicos.
- Limpieza para eliminar espacios en blanco, caracteres especiales, puntuación no deseada o caracteres extraños.
- Algoritmos para la comparación de cadenas (*string matching*), se identifican errores tipográficos, duplicados.
- Validación con datos del dominio →Mapeo de datos a vocabularios controlados.
- Estandarización aplicando un formato único o valor de referencia.

PDI Limpieza, transformación y normalización

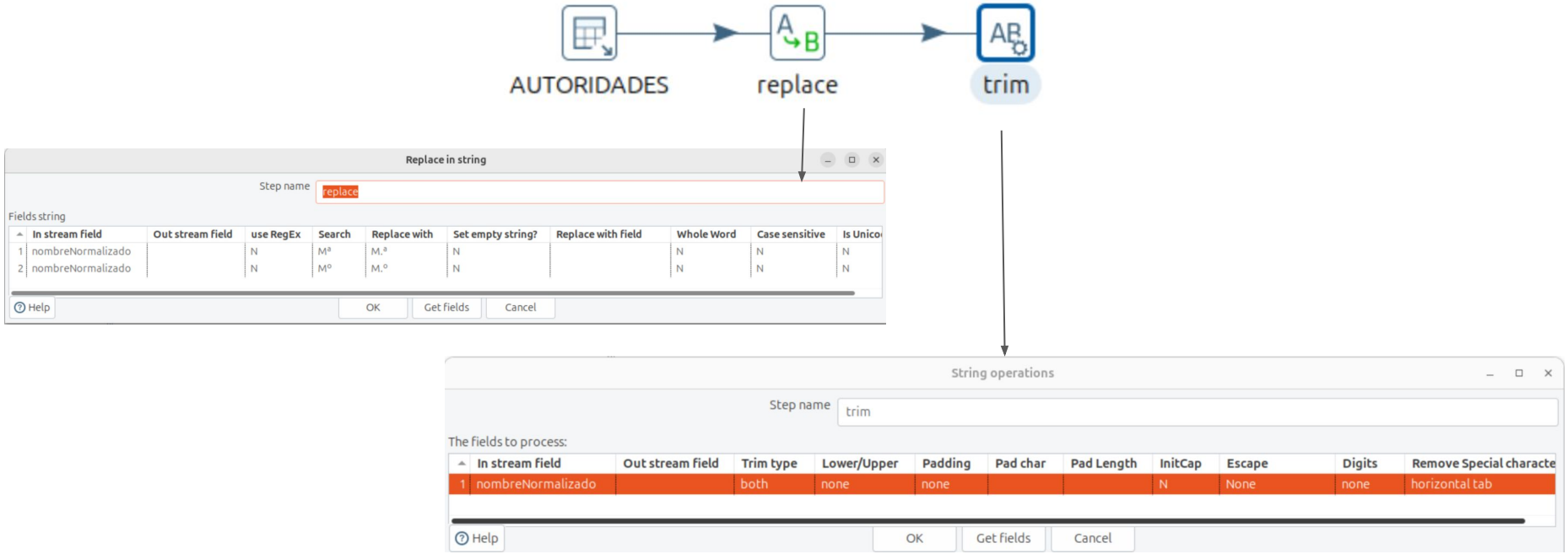
Algunos de los pasos más frecuentes para la limpieza de datos:

- *Replace in string*
- *Split fields*
- *Split field to rows*
- *Strings cut*
- *String operations*
- *Value mapper*
- *Calculator*
- *Fuzzy match*



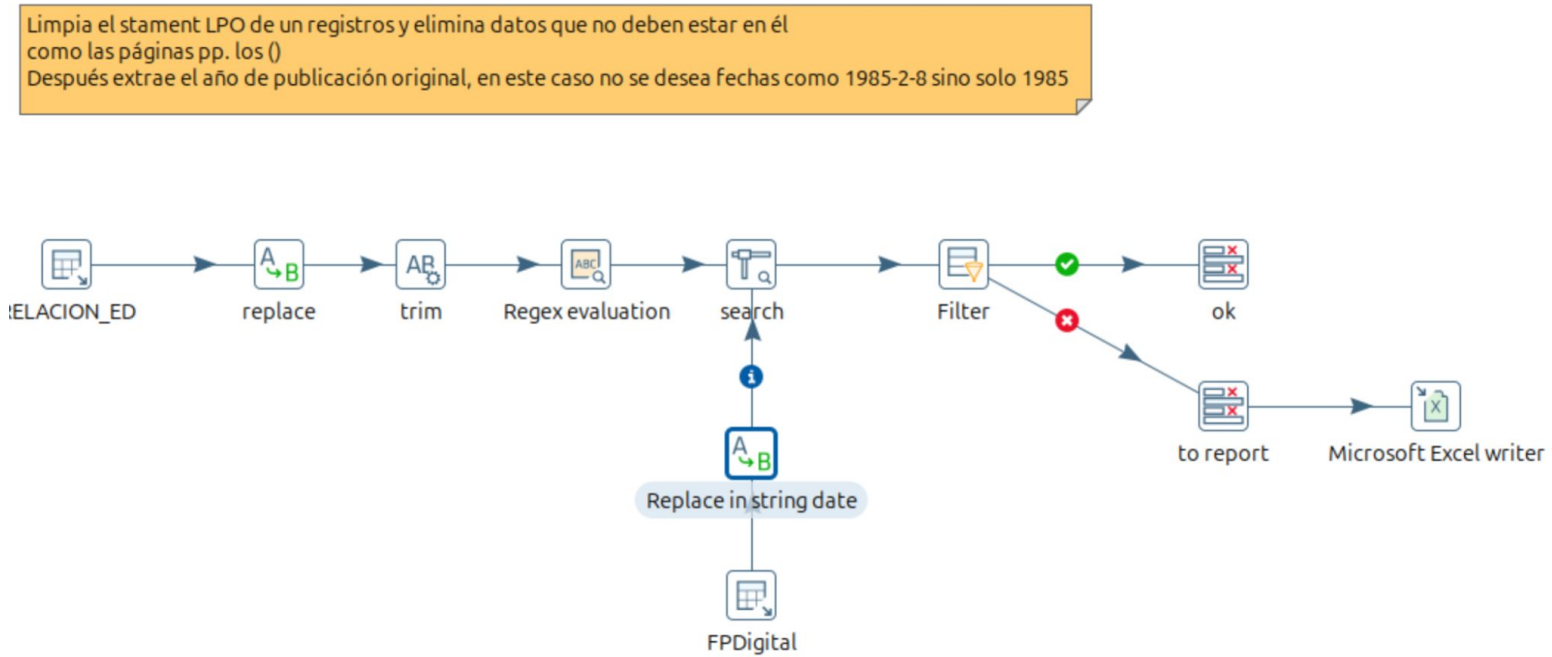
PDI Limpieza, transformación y normalización

Ejemplo limpieza, transformación y normalización de datos



PDI Limpieza, transformación y normalización

Ejemplo limpieza de datos



String Matching Algorithms

String Matching Algorithms

1. **Levenshtein y Damerau-Levenshtein**: Calcula la distancia entre dos strings en función del número de cambios necesarios para pasar de una cadena a otra.

2. **Needleman-Wunsch:** Calcula la similitud de dos secuencias, calculando una penalización por espacio.

3. **Jaroa y Jaro-Winkler**: Calcula un índice de similitud entre dos strings. El resultado es una fracción entre 0 (sin similitud) y 1 (coincidencia idéntica).

4. **Pair letters similarity**: Este algoritmo corta ambas cadenas en pares y comparan los conjuntos de pares.

5. **Metaphone, Double Metaphone, Soundex y RefinedSoundEx**: Estos algoritmos todos intentan hacer coincidir los strings en función de cómo "sonarían" y también se denominan fonéticas algoritmos.

String Matching Algorithms

String Matching Algorithms

Levenshtein y Damerau-Levenshtein: Calcula la distancia entre dos strings en función del número de cambios necesarios para pasar de una cadena a otra.

- CASTERS y CASTRO es de 2
- Paso 1: eliminar la E; Paso 2: reemplazar S por O

Needleman-Wunsch: Calcula la similitud de dos secuencias, calculando una penalización por espacio.

- Puntuación de -2 para CASTERS a CASTRO

String Matching Algorithms

String Matching Algorithms

Jaroa y JaroWinkler: Calcula un índice de similitud entre dos strings. El resultado es una fracción entre 0 (sin similitud) y 1 (coincidencia idéntica).

- Levenshtein entre CASTERS y POOH es de 7
- Jaro y de Jaro-Winkler es 0 porque no hay similitud entre las dos cadenas

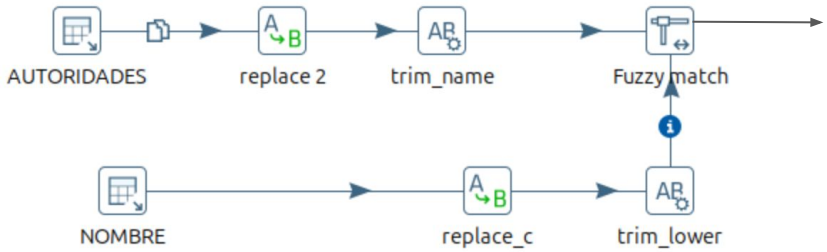
Pair letters similarity: Este algoritmo corta ambas cadenas en pares y comparan los conjuntos.

- CASTERS y CASTRO se transformará en:
  - {CA, AS, ST, TE, ER, RS} y {CA, AS, ST, TR, RO}
  - Calcula la similitud dividiendo el número de pares comunes (multiplicado por dos) por la suma de los pares de ambas cadenas → hay tres pares comunes (CA, AS, ST) y once pares en total →puntuación de similitud de (2*3)/11 = 0,545 (está bien)

String Matching Algorithms

Objetivo: determinar si dos nombres de autor, que parecen ligeramente diferentes, en realidad se refieren a la misma persona

Si la distancia es baja →asumimos que son duplicados



Examine preview data

Rows of step: Fuzzy match (1000 rows)

idAutoridad	nombreNormalizado	nombreNormalizadoOriginal	match	measure valu
17436	rojasfranciscode	Rojas, Francisco de, , (O.F.M.)	rojasfranciscode	1,0
18387	arcosfranciscode	Arcos, Francisco de , (O.SS.T.)	rojasfranciscode	0,7333333333
24996	castillafranciscode	Castilla, Francisco de (S.I.)	rojaszorrillafranciscode	0,7317073171
22570	marianojosédesevilla	Mariano José de Sevilla , (O.F.M.Cap.)	larramarianojoséde	0,7222222222
18892	torresfranciscode	Torres, Francisco de , (O.F.M.)	rojasfranciscode	0,7096774194
22515	torresfranciscode	Torres, Francisco de , (S.I.)	rojasfranciscode	0,7096774194
24756	torresfranciscode	Torres, Francisco de (O.P.)	rojasfranciscode	0,7096774194
24227	sosafranciscode	Sosa, Francisco de (O.F.M.)	rojasfranciscode	0,6896551724
1685	ruizdealarcónhernando	Ruiz de Alarcón, Hernando (S. 17º)	ruizdealarcónymendozajuan	0,6818181818
17645	araujofranciscode	Araujo, Francisco de , (O.P.), (1580-1664)	quevedofranciscode	0,6666666667
23779	franciscodeosuna	Francisco de Osuna (O.F.M.)	rojasfranciscode	0,6666666667

Close | Stop | Get more rows

String Matching Algorithms

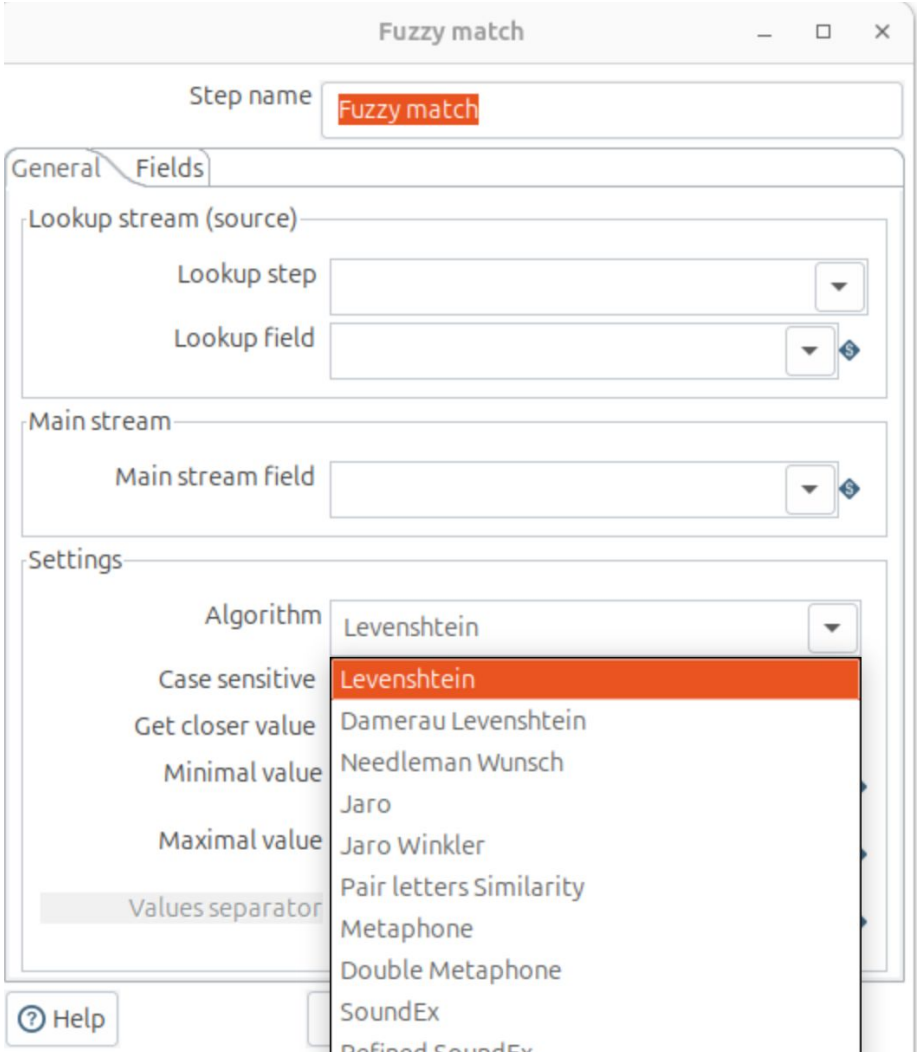
String Matching Algorithms

Metaphone, Double Metaphone, Soundex y RefinedSoundEx: Estos algoritmos intentan hacer coincidir los strings en función de cómo "sonarían", también se denominan algoritmos fonéticos.

La debilidad de todos estos algoritmos fonéticos es que se basan en el inglés y no sería de mucha utilidad en un entorno francés, español u holandés.

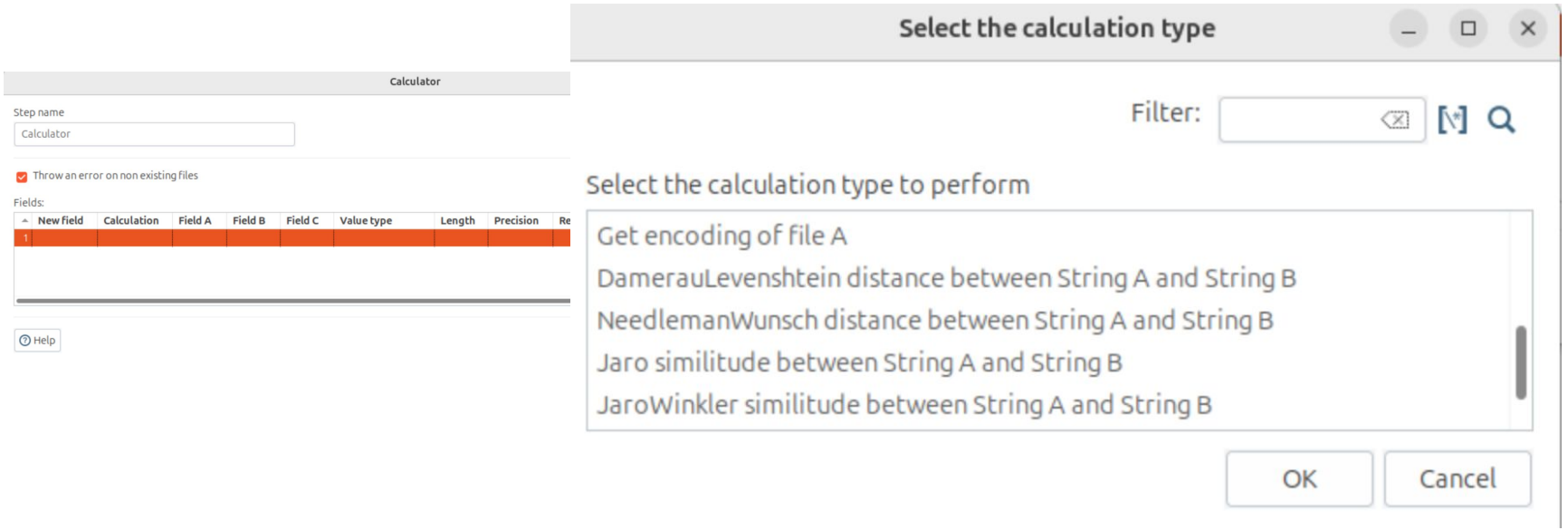
String Matching Algorithms

Fuzzy match



String Matching Algorithms

Calculator step



Estrategias y técnicas para la limpieza de datos

- Uso de expresiones regulares para encontrar y reemplazar formatos específicos.
- Limpieza para eliminar espacios en blanco, caracteres especiales, puntuación no deseada o caracteres extraños.
- Algoritmos para la comparación de cadenas (*string matching*), se identifican errores tipográficos, duplicados.
- Validación con datos del dominio →Mapeo de datos a vocabularios controlados.
- Estandarización aplicando un formato único o valor de referencia.

Validación de datos

Objetivo: verificar y validar los datos recopilados, garantizando que son **confiables** y **adecuados**.

Deben cumplir las **reglas** de negocio predefinidas y marcar o rechazar cualquier registro que no cumpla estos requisitos.

Cualquier tarea en la gestión de datos (adquisición, transformación o carga), debe incluir la validación de datos para garantizar resultados precisos:

- verificar que los datos se extraigan correctamente de las fuentes de origen
- la transformación se realice con precisión de acuerdo con las reglas definidas
- y la carga de datos en destino se produzca sin pérdida ni corrupción

Una forma sencilla de validación de datos son las tablas de referencia, donde la regla es: “el valor debe estar presente en tabla de referencia”.

La mayoría de las reglas de validación son un poco más complejas y tienen dependencia del dominio.

Validación de datos

- Verificar si un dato debe estar dentro de un **rango de valores**
- Verificar **consistencia** en el conjunto de datos y entre diferentes conjuntos de datos
- Verificar **Integridad referencial** garantizando que se mantengan las relaciones entre los datos de diferentes tablas o bases de datos
- Verificar las **restricciones de identidad**, garantizando que los datos como identificadores, DNI, email o teléfono, son únicos
- Comprobar que los **datos esenciales** están **presentes**
    - gestión de valores faltantes vista anteriormente

Validación de datos

- Comprobar **tipo** de datos y **tamaño**
  - por ejemplo una contraseña de 12 caracteres alfanuméricos y con ciertas restricciones, un código postal, email
- Comprobar el **formato**
  - por ejemplo la fecha formato AAAA-MM-DD o números de teléfono
- Comprobar valor dentro de un **vocabulario controlado**
  - por ejemplo los idiomas tiene un conjunto de datos aceptables, los formatos de una obra tiene una lista o tabla de referencia que determina los valores válidos, los meses del año, el género, latitud, longitud, etc.
- **Restricciones** según **dominio**
  - Los valores de entrada deben estar en mayúsculas/minúsculas
  - Los importes no pueden superar el valor X
  - Los suscriptores deben tener al menos 18 años

Validez de los datos

Tablas de referencia

Using Reference Tables

Comprobar información consultando fuentes externas validadas (datos maestros o de referencia).

No siempre lo tenemos disponible.

Por ejemplo países, ciudades, autores, libros, idiomas, etc.

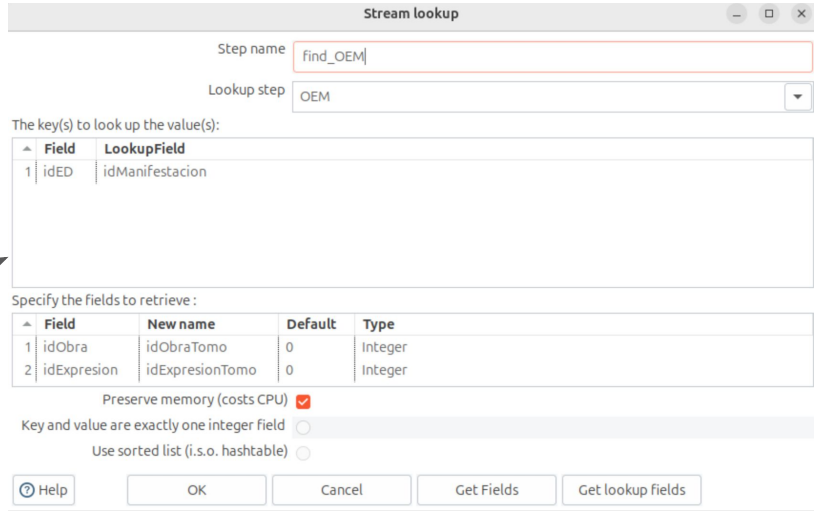
Wikidata

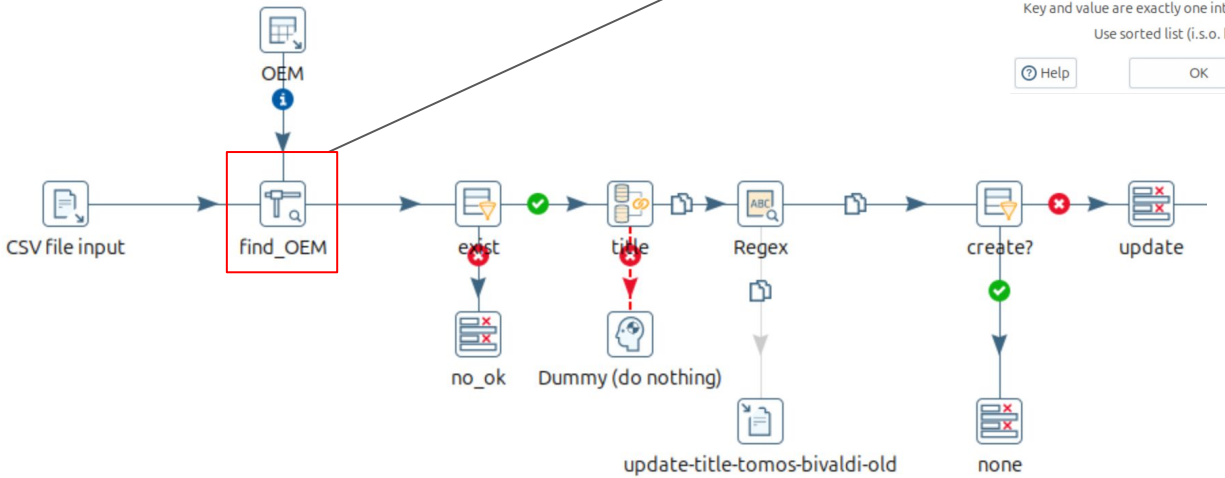
Geonames

Tablas de referencia

Conforming Data Using Lookup Tables

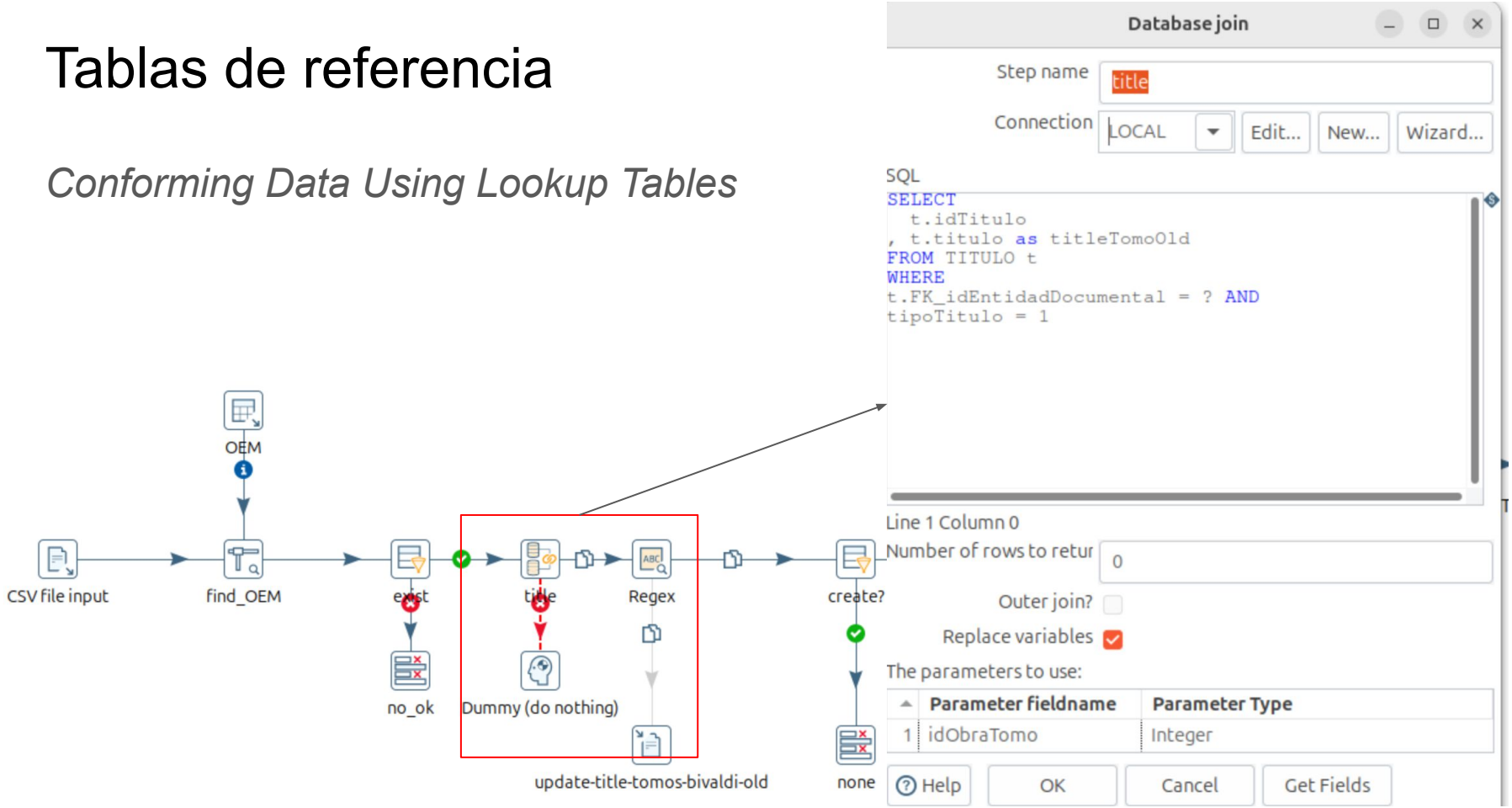
Realizar una búsqueda basada en coincidencia de campos





Tablas de referencia

Conforming Data Using Lookup Tables



Tablas de referencia

Se crea tabla de referencia como tabla maestra de comprobación de datos.

Se deben cumplir dos requisitos básicos:

- Cada posible valor del sistema fuente necesita una correspondencia
- La correspondencia debe conducir a un conjunto único de valores

Ejemplo: se crea tabla de referencia para los idiomas con las normas ISO sobre identificación del idioma

- ISO 639-2 estándar internacional que define los código de identificación única de los idiomas
- ISO 639-3 define código para para un idioma individual incluyendo idiomas vivos, extintos y antiguos
- Norma ISO 639-1 es más restrictiva que las otras normas que cubren una gama más amplia de idiomas y variaciones

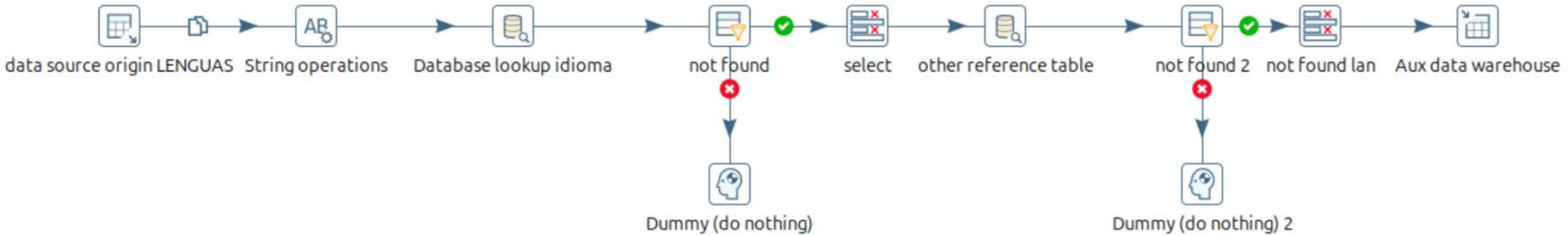
idIdioma ▲ 1	iso6392Code	iso6393Code	iso6391Code	nombre
26	spa	spa	es	español
27	swe	swe	sw	sueco
28	syr	syr		sirio
30	lat	lat	la	llatí
32	gre* / ell	ell	el	grec
33	gre* / ell	ell	el	griego
37	syc	syc		siriaco
38	spa	spa	es	castellà
39	cat	cat	ca	català
45	pol	pol	pl	polaco
46	cat	cat	ca	valenciano
47	baq* / eus	eus	eu	euskera
48	dan	dan	da	danés

Tabla de referencia del código de idioma

Tablas de referencia

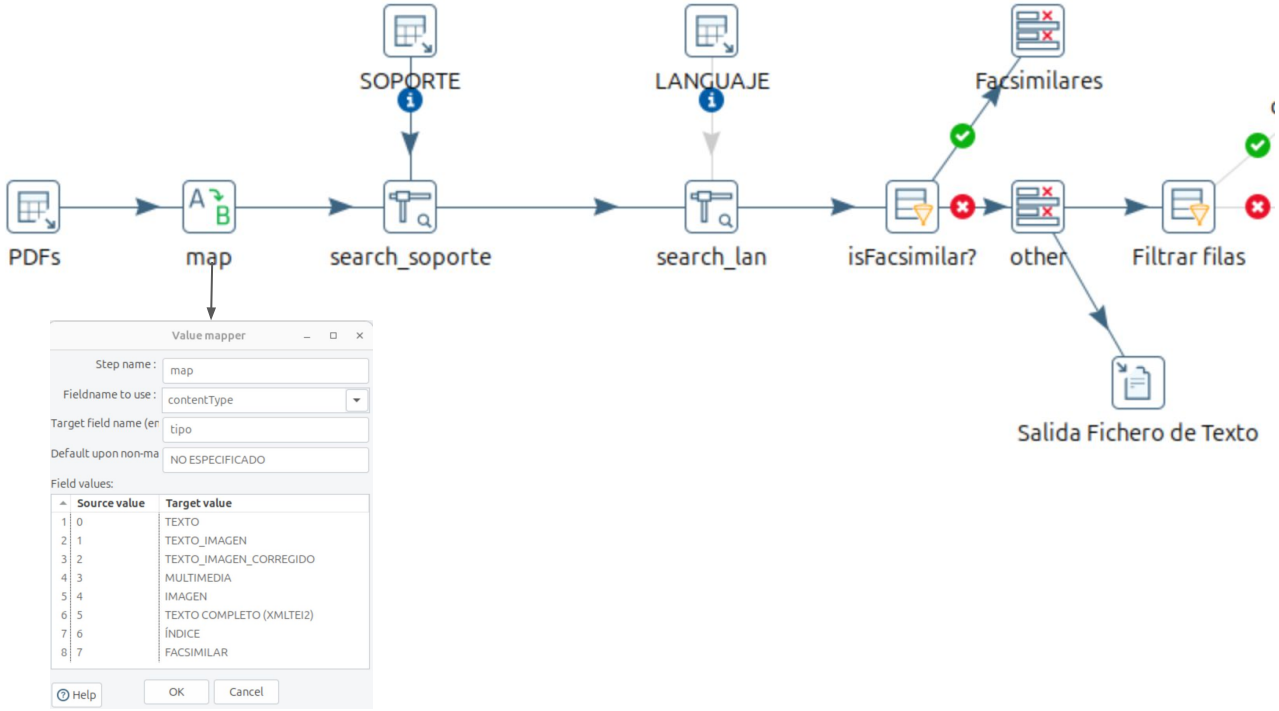
Conforming Data Using Reference Tables

Ejemplo



Validación de datos

Mapeo de datos a vocabularios controlados



Validación de datos

Applying Validation Rules

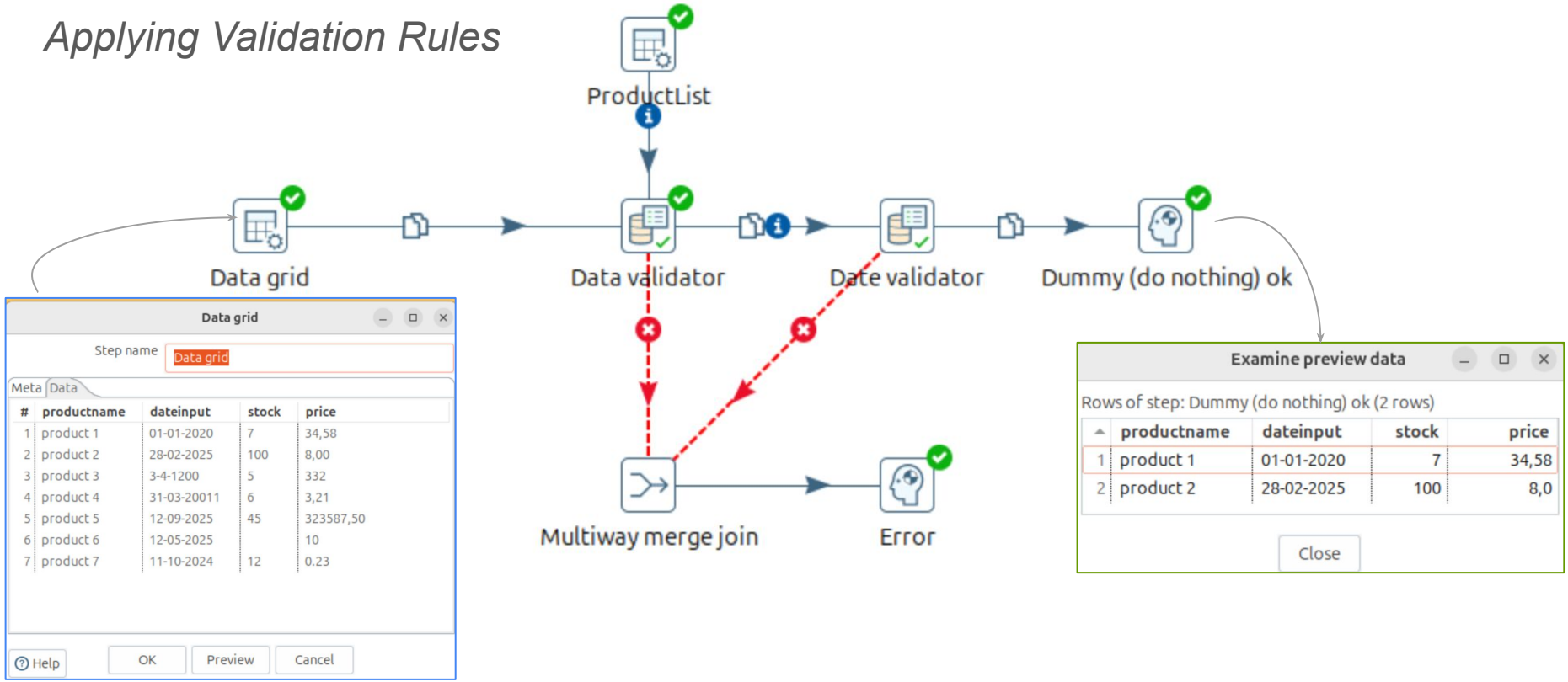
Ejemplo:

Los datos deben cumplir las siguientes reglas:

- Ninguno de los campos puede contener un valor NULL.
- Las fechas no pueden ser anteriores al 1 de enero de 2001.
- No se permiten nombres de productos fuera del catálogo que define la empresa.
- El número de artículos en stock debe estar entre 1 y 100. No puede estar vacío.
- El precio por artículo no puede exceder de 1000€.

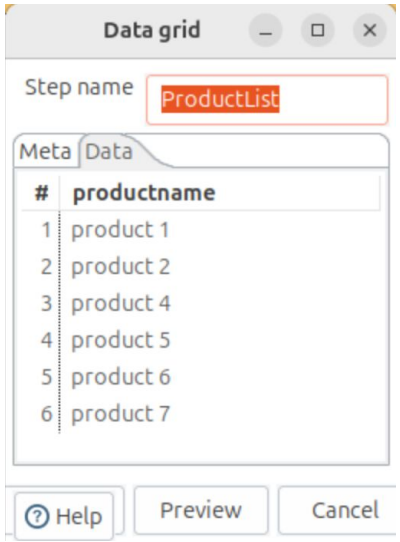
Validación de datos

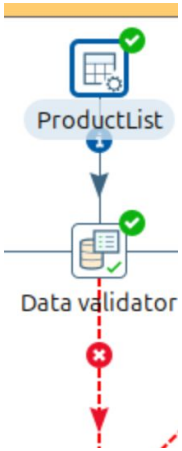
Applying Validation Rules

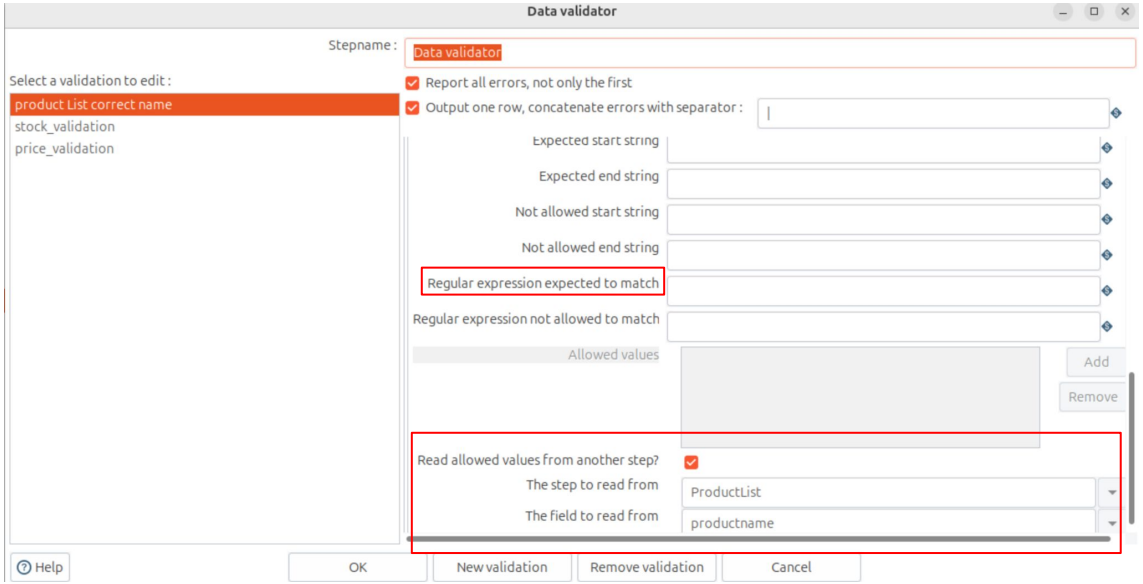


Validación de datos

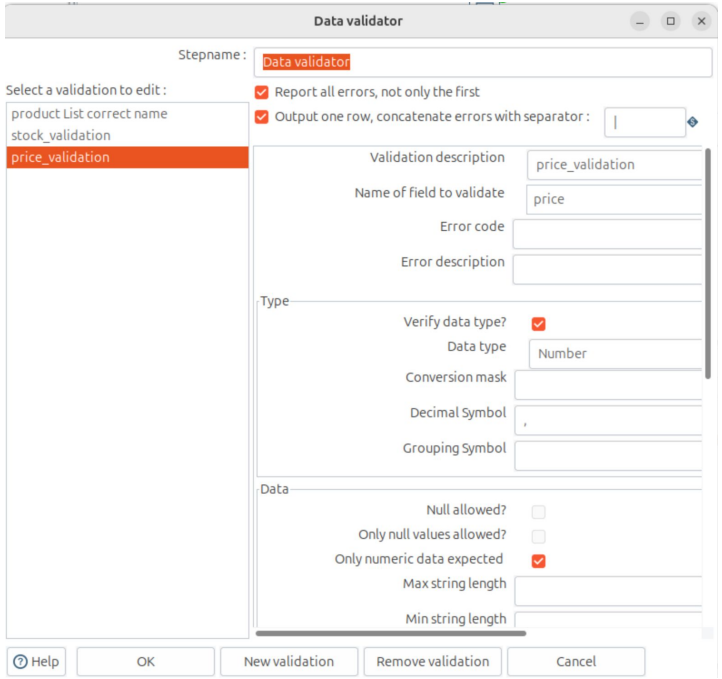
Applying Validation Rules

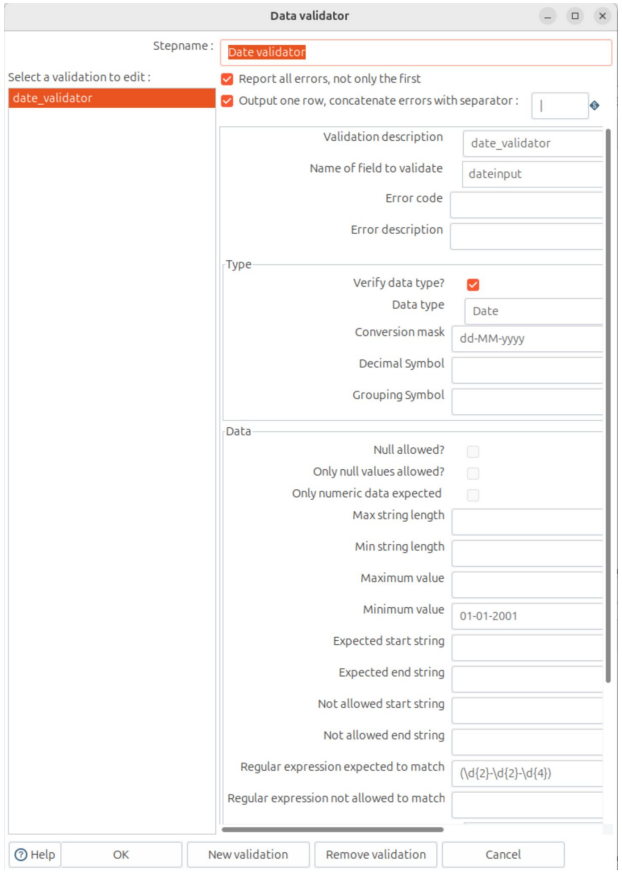






Validación de datos





Gestión de datos duplicados

El tratamiento de duplicados se basa en **identificar y eliminar esa redundancia** en base a los **atributos clave** que definen qué conforma un duplicado y las **reglas** de negocio propias del dominio

- Diversas fuentes de datos: Obtener la misma información de diferentes sistemas que no están sincronizados.
- Errores en la entrada de datos: Errores tipográficos que crean nuevas entradas para la misma entidad.
- Fusión de bases de datos: Combinar conjuntos de datos sin un proceso adecuado de deduplicación.

Problemas derivados:

- Baja calidad de los datos
- Inconsistencia ya que se pueden contener información contradictoria
- Espacio de almacenamiento no necesario
- Impacto en el rendimiento de los modelos

Gestión de datos duplicados

Resolver duplicados en el conjunto de datos se realiza durante la fase de transformación del proceso ETL.

El objetivo es **identificar y eliminar** los registros **duplicados** para mantener solo la versión más precisa y completa.

Pasos clave:

1. **Identificar**: encontrar los registros que son potencialmente duplicados.
    - Duplicados exactos
    - Duplicados no exactos
2. **Estandarizar**: **normalizar** los datos para facilitar la comparación
3. **Fusionar** (*merge*) o **eliminar** (*delete*)

Gestión de datos duplicados

Identificar

Encontrar los registros que son potencialmente duplicados.

- Duplicados exactos, usando identificadores únicos
- Duplicados no exactos, con técnicas *fuzzy matching* para encontrar coincidencias aproximadas. Ejemplos de uso de Jaro-Winkler o Levenshtein para calcular la similitud entre cadenas de texto

Herramientas:

- Herramientas ETL comerciales
- Librerías de código abierto: Python (Pandas, Dask, Dedupe)
- SQL usando por ejemplo GROUP BY

Gestión de datos duplicados

Estandarizar

Normalizar los datos para facilitar la comparación

Mayúsculas/minúsculas

Limpieza de datos

ej. "calle Mayor", "C/ Mayor", "C. Mayor" se convierten en "Calle Mayor"

Gestión de datos duplicados

Fusionar o eliminar

Una vez identificados, decidir qué hacer con los duplicados:

- **Fusionar** (Merge): Combinar la información de los registros duplicados en un único registro maestro, conservando los datos más completos y actualizados.
- **Eliminar** (Delete): Si no se puede fusionar, se elimina el duplicado menos relevante.