

# Lección 9: Manipulación Avanzada de Estructuras – Transformando y Uniendo Datos (Revisada)

Bienvenido de nuevo. En esta lección revisada, ajustamos la manipulación avanzada de estructuras, corrigiendo soluciones y ampliando detalles. Transformamos datos (arrays, JSON a tablas), manejamos separadores locales ("1,234.56" vs "1.234,56"), usamos regex para patrones, aplicamos operaciones de base de datos (joins, append, transpose), y destacamos UIDs para deduplicar/unir. Incorporamos bucles implícitos (SPLIT, VLOOKUP) y expandimos Sección 6 con Power Query (append, joins por UID) y QUERY en Sheets, incluyendo tablas dinámicas y slicers brevemente.

---

## Sección 1: Introducción – Manipulación para Datos Eficientes

**Explicación Detallada:** La manipulación avanzada convierte estructuras crudas (como listas o JSON) en formatos estructurados (tablas) para análisis. Resuelve problemas como separadores locales (ej: coma vs punto en números) y optimiza datos eliminando duplicados con UIDs (identificadores únicos). Usa bucles implícitos (ej: ARRAYFORMULA aplica TRIM a un rango entero, simulando un "for each" sin código explícito) para eficiencia.

- **Problemas Comunes:**
  - **Separadores Locales:** "1,234.56" (US) vs "1.234,56" (EU) confunde cálculos.
  - **JSON Anidado:** Datos jerárquicos difíciles de tabular.
  - **Duplicados:** Sin UIDs, joins fallan o datos se repiten.
- **Importancia de UIDs:** Un UID (ej: "Cliente\_001") es una clave única que asegura precisión en joins (unir tablas) y deduplicación (eliminar repeticiones con =UNIQUE).
- **Ejemplo Práctico:** Tienes un array {"a","b";1,2}. ¿Cómo lo conviertes en una tabla? Usa =FLATTEN(TRANSPOSE(array)) para una columna.

## Diferencias entre Herramientas:

- **Sheets/Excel:** QUERY (SQL-like) y Power Query para transformaciones visuales.
- **DAX:** Usa RELATED (joins) y SUMX (reduce) en modelos de Power BI.
- **VBA/AppScript:** Loops personalizados (ej: For Each para procesar rangos).

**Pausa:** Piensa en un array {"a","b"}. ¿Lo transformarías con FLATTEN o TRANSPOSE?

Prueba en Sheets.

---

## Sección 2: Formatos Locales – Separadores de Miles y Decimales (Revisada)

**Explicación Detallada:** Los separadores varían por región (ej: "1,234.56" en US usa coma para miles y punto para decimales; "1.234,56" en EU invierte). Esto causa errores si el software asume un locale diferente. La solución corregida usa una secuencia de sustituciones para estandarizar.

- **Solución Corregida Paso a Paso:**

1. **Eliminar Comas:** =SUSTITUIR(A1, ",", "") convierte "1,234.56" a "1234.56".
  2. **Intercambiar Punto por Coma:** =SUSTITUIR(resultado, ".", ",") da "1234,56".
    - Fórmula completa: =SUSTITUIR(SUSTITUIR(A1, ",", ""), ".", ",").
    - Nota: La versión anterior fallaba dejando dos puntos/comas; esta preserva el valor numérico.
- **Configuración Global:** Archivo > Configuración > Locale (ej: Español para coma decimal) ajusta el comportamiento predeterminado.
  - **Bucle Implícito:** =ARRAYFORMULA(SUSTITUIR(SUSTITUIR(A1:A10, ",", ""), ".", ",")) aplica a todo un rango, simulando un loop "for each".

- **Ejemplo Práctico:**

- A1: "1,234.56" (US format).
- =SUSTITUIR(A1, ",", "") → "1234.56".
- =SUSTITUIR("1234.56", ".", ",") → "1234,56" (EU format).
- Resultado usable en cálculos con =VALUE.

### Diferencias entre Herramientas:

- **Sheets:** Locale auto-detecta; usa =TEXT(1234.56, "#.##0,00") para formato EU.
- **Excel:** Opciones > Avanzado > Separador decimal (punto o coma).
- **DAX:** FORMAT([Valor], "General Number") ignora locale en salida.

## Tabla de Separadores:

Problema	Sheets/Excel	DAX	AppScript/VBA	
Swap coma/punto	=SUSTITUIR(SUSTITUIR(A1, ",", ""), ".", ",")	REPLACE	replace(str, /[.,]/, locale)	
Convertir a Número	=VALUE(A1)	VALUE([Col])	Val(str)	

**Pausa:** Ingresa "1,234.56" en A1 de Sheets. Aplica la fórmula y cambia locale a Español.  
¿Qué obtienes?

---

## Sección 3: Regex – Patrones para Búsqueda y Reemplazo

**Explicación Detallada:** Regex (expresiones regulares) busca o reemplaza patrones en texto (ej: extraer "123" de "ID:123"). Es ideal para limpiar datos sucios o extraer datos específicos como emails o códigos.

- **Funciones Clave:**

- **REGEXMATCH:** =REGEXMATCH(A1, "^\\d+\$/) verifica si A1 contiene solo dígitos (verdadero/falso).
- **REGEXREPLACE:** =REGEXREPLACE(A1, "\\s+", " ") unifica múltiples espacios en uno (ej: "a b" → "a b").
- **REGEXEXTRACT:** =REGEXEXTRACT(A1, "(\\d+)") extrae el primer número (ej: "ID:123" → "123").
- **Bucle Implícito:** =ARRAYFORMULA(REGEXREPLACE(A1:A10, "\\s+", " ")) aplica a un rango entero, simulando un loop.

- **Ejemplo Práctico:**

- A1: "Hola Mundo 2023".
- =REGEXREPLACE(A1, "\\s+", " ") → "Hola Mundo 2023".
- =REGEXEXTRACT(A1, "(\\d+)") → "2023".

**Problema Común:** Extraer datos de texto mixto (ej: "Email: test@ej.com" → "test@ej.com"); solución: =REGEXEXTRACT(A1, "(\\w+@\\w+.\\w+)").

### Diferencias entre Herramientas:

- **Sheets/Excel:** Regex nativo (limitado en versiones antiguas de Excel).
- **DAX:** Usa SEARCH o SUBSTITUTE (sin regex nativo).
- **VBA/AppScript:** Objeto RegExp para patrones avanzados (ej: RegExp.Test para match).

### Tabla de Regex:

Función	Sheets/Excel	DAX	AppScript/VBA	○
Match	=REGEXMATCH(A1, pattern)	No directo	test(pattern)	
Replace	=REGEXREPLACE(A1, pattern, new)	SUBSTITUTE	replace / RegExp.Replace	

**Pausa:** Ingresa "ID:123" en A1. Usa =REGEXEXTRACT(A1, "(\\d+)") para extraer "123".

---

### Sección 4: Transformar Estructuras a Tablas – Arrays, JSON y Más

**Explicación Detallada:** Convierte estructuras no tabulares (arrays, JSON) en tablas para análisis. Incluye transformación de fechas (ej: "Oct, 23 2025" a "23/10/2025").

- **Arrays a Tabla:**

- **FLATTEN:** =FLATTEN(A1:C3) aplana una matriz (ej: {"a","b";1,2} → columna "a,1,b,2").
- **TRANSPOSE:** =TRANSPOSE(A1:A3) gira filas a columnas (ej: A1:A3 → fila).
- **SLICE:** =SLICE(A1:A10, 2, 5) toma un subrango (del 2º al 6º elemento).

- **JSON a Tabla:**

- **Sheets:** Usa =IMPORTJSON(URL) (add-on) para importar y expandir.
- **Excel:** Power Query > De JSON > Expandir columnas anidadas.
- **Bucle Implícito:** =MAP(JSON array, LAMBDA(item, SPLIT(item, ","))) procesa cada item.

- **Transformación de Fechas:**

- Ejemplo: A1: "Oct, 23 2025".
- =TEXT(DATEVALUE(SUBSTITUTE(A1, "Oct", "10") & " 2025"), "DD/MM/YYYY") → "23/10/2025".
- Nota: Ajusta mes a número (Oct=10) y usa DATEVALUE.

- **Ejemplo Práctico:**

- Array A1:B2: {"a","b";1,2}.
- =FLATTEN(TRANSPOSE(A1:B2)) → columna "a,1,b,2".
- Fecha A3: "Oct, 23 2025" → "23/10/2025".

### Diferencias entre Herramientas:

- **Sheets:** FLATTEN nativo; ARRAYFORMULA para rangos.
- **Excel:** TOCOL/TOROW en 365 para aplastar.
- **DAX:** UNNEST para JSON en Power BI.

### Tabla de Transformación:

Estructura	Sheets/Excel	DAX	AppScript/VBA	O
Array a Tabla	=FLATTEN(A1:B2)	GENERATE	flatMap / Loop	
JSON a Tabla	Power Query / IMPORTJSON	PARSEJSON	JSON.parse + loop	
Fecha Transformada	=TEXT(DATEVALUE(...), "DD/MM/YYYY")	FORMAT([Date], "DD/MM/YYYY")	new Date().toLocaleDateString	

**Pausa:** Ingresa "Oct, 23 2025" en A1. Aplica la fórmula de fecha.

---

## Sección 5: Transformación de Bases de Datos – Joins, Append, Condicionales

**Explicación Detallada:** Opera datos como base de datos. Incluye todos los tipos de joins y distingue append vs update rows con UIDs.

- **Tipos de Joins:**

- **Inner Join:** Solo filas con coincidencias en ambas tablas (ej: =QUERY(A1:B10 & C1:D10, "select \* where Col1=Col3")).
- **Left Join:** Todas filas de la izquierda, coincidencias de la derecha (ej: =VLOOKUP con IFERROR para llenar nulos).
- **Right Join:** Todas de la derecha, coincidencias de la izquierda (simula con transpuesta y left join).
- **Outer Join:** Todas filas de ambas, nulos donde no coinciden (Power Query o QUERY avanzado).

- **Append vs Update Rows:**

- **Append:** Agrega filas nuevas (ej: = {A1:B10; C1:D10}). Si UID existe, duplica; usa UNIQUE si no.
- **Update Rows:** Actualiza filas existentes basadas en UID. Requiere relación previa o merge (Power Query).
- **Contexto UID:** Si tabla1 (UID, Nombre) y tabla2 (UID, Ventas) no estaban relacionadas, usa UID para:
  - **Append:** = {tabla1; tabla2} (duplica si UID repite).
  - **Update:** Power Query > Combinar > Usa UID > Actualizar valores (reemplaza Ventas si UID coincide).
- **Columnas Condicionales:** =ARRAYFORMULA(IF(A1:A10>10, "Alto", "Bajo")) crea dinámicamente.
- **Transpose/Flatten/Reduce:**
  - =TRANSPOSE gira; =FLATTEN aplana; =REDUCE(0, rango, LAMBDA(acc, val, acc+val)) suma acumulativa.
- **Ejemplo Práctico:**
  - Tabla1: UID, Nombre (1,"Ana"; 2,"Bob").
  - Tabla2: UID, Ventas (1,100; 3,300).
  - **Inner Join:** QUERY para UID=1 (Ana,100).
  - **Append:** {Tabla1; Tabla2} → 4 filas (duplica UID=1).
  - **Update:** Power Query merge por UID reemplaza Ventas de Ana a 100.

### Diferencias entre Herramientas:

- **Sheets:** QUERY simula SQL; no update nativo.
- **Excel:** Power Query maneja joins/update visualmente.
- **DAX:** RELATED para joins; no append directo.

### Tabla de Operaciones DB:

Operación	Sheets/Excel	DAX	AppScript/VBA	O
Inner Join	=QUERY o VLOOKUP	RELATED	Loop match	
Left/Right/Outer	QUERY avanzado / Power Query	CROSSJOIN/FILTER	Custom logic	
Append	= {rango1; rango2}	UNION	concat arrays	
Update Rows	Power Query merge	CALCULATE	Loop + match	
Reduce	=REDUCE(acc, rango, LAMBDA)	SUMX	reduce func	

**Pausa:** Prueba un inner join con QUERY en dos rangos.

---

## Sección 6: Power Query, QUERY Formula y SQL – Consultas Estructuradas (Revisada)

**Explicación Detallada:** Herramientas para consultas estructuradas y visuales.

- **Power Query (Excel):**
  - **Append:** Datos > Combinar Consultas > Anexar > Selecciona tablas (ej: Tabla1 & Tabla2) > Cargar. Une filas; duplica si UID repite.
  - **Combine usando UID:** Datos > Combinar Consultas > Combinar > Elige UID como clave > Tipo Inner/Left > Cargar. Actualiza o une según configuración.
  - **Otros:** Transformar > Transponer gira; Expandir columnas anidadas aplana JSON.
- **=QUERY en Sheets:** =QUERY(A:C, "select \* where A = 'x' order by B group by C") filtra (A=x), ordena (B), agrupa (C). Usa SQL sintaxis.
- **SQL:** En Power BI (ej: SELECT \* FROM tabla INNER JOIN otra ON UID) o scripts externos.
- **Tablas Dinámicas y Slicers:**
  - **Excel/Sheets:** Insertar > Tabla Dinámica > Selecciona rango > Añade campos (ej: UID, Ventas) > Insertar Slicer > Filtra por UID o fecha interactivamente.
- **Ejemplo Práctico:**
  - Power Query: Une Tabla1 (UID, Nombre) y Tabla2 (UID, Ventas) por UID con Inner Join.
  - QUERY: =QUERY(A1:B10, "select A, sum(B) where A is not null group by A") suma por grupo.

## Diferencias entre Herramientas:

- **Power Query:** Visual, ideal para batch (Excel solo).
- **QUERY:** Rápida, nativa en Sheets.

**Pausa:** Crea una tabla dinámica con slicer en Excel.

---

## Sección 7: Consejos Finales

### Explicación Detallada:

- Usa LAMBDA (ej: =LAMBDA(x, x\*2)(A1)) para funciones reutilizables en rangos.
- UIDs aseguran integridad (ej: =UNIQUE(A1:A10) elimina duplicados).
- Verifica locale en imports (Archivo > Configuración > Idioma) para evitar errores.

**Conclusión:** Manipulación optimiza datos. Integra con formateo previo para flujos limpios.

**Pausa Final:** Haz un join en Power Query con UID. ¿Tema 10? Ej: "Automatización". ¡Dime!

