

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

Manipulación convierte estructuras crudas en útiles, resuelve inconsistencias (ej: locales) y optimiza (reduce duplicados via UIDs). Es vital para análisis preciso, usando bucles implícitos como ARRAYFORMULA para extender operaciones sin código explícito sobre rangos enteros.

- **Problemas comunes:**

- Separadores locales (coma vs punto) que confunden cálculos.
- JSON anidado o arrays desestructurados que necesitan tabularse.
- Duplicados que distorsionan joins sin UIDs únicos.
- **Importancia de UIDs:** Actúan como claves primarias (ej: ID cliente) para unir tablas o eliminar duplicados con precisión (ej: =UNIQUE elimina repeticiones).
- **Cómo funciona:** Piensa en un array {"a", "b"} – manipularlo implica convertirlo a columnas o unir con otra tabla via UID.

Diferencias entre herramientas:

- **Sheets/Excel:** Usan QUERY para consultas SQL-like y Power Query para transformaciones visuales.
- **DAX:** Ofrece RELATED para joins y SUMX para reducciones en modelos.
- **VBA/AppScript:** Permiten loops explícitos (ej: For Each) para transformaciones personalizadas.

Pausa: Imagina un array {"a","b"} en A1:A2. ¿Cómo lo convertirías a una tabla con encabezados? Sigue para aprender.

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

Problema: "1,234.56" (US, punto decimal, coma de miles) vs "1.234,56" (EU, coma decimal, punto de miles) genera errores (ej: suma ve texto en lugar de número).

- **Solución corregida y detallada:**

- **Fórmula ajustada:** Usa una secuencia de sustituciones para preservar integridad numérica.
 - Paso 1: `=SUSTITUIR(A1, ",", "")` elimina comas → "1234.56" (de "1,234.56").
 - Paso 2: `=SUSTITUIR(resultado, ".", ",")` intercambia punto por coma → "1234,56" (según locale).
 - Combinado: `=SUSTITUIR(SUSTITUIR(A1, ",", ""), ".", ",")` – Ajusta según necesidad (punto a coma o viceversa).
- **Configuración global:** Archivo > Configuración > Locale (ej: Español para coma decimal, US para punto) estandariza todo el documento.
- **Bucle implícito:** `=ARRAYFORMULA(SUSTITUIR(SUSTITUIR(A1:A10, ",", ""), ".", ","))` procesa un rango entero como un for implícito, ideal para columnas largas.
- **Nota importante:** La solución anterior fallaba al dejar residuos (dos puntos/comas); esta secuencia asegura un número válido antes de conversiones.

- **Ejemplo práctico:**

- A1: "1,234.56" → `=SUSTITUIR(SUSTITUIR(A1, ",", ""), ".", ",")` → "1234,56".
- A2:A5 con valores similares →
`=ARRAYFORMULA(VALUE(SUSTITUIR(SUSTITUIR(A2:A5, ",", ""), ".", ",")))` convierte a números para cálculos.

- **Diferencias entre herramientas:**

- **Sheets:** Locale se adapta automáticamente; usa `=TEXT(1234.56, "#.##0,00")` para formato EU.
- **Excel:** Opciones > Avanzado > Separador decimal permite ajuste manual por usuario.
- **DAX:** `FORMAT([Valor], "General Number")` ignora locale en salida, útil para consistencia.

- **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

Regex busca o reemplaza patrones complejos (ej: extraer "123" de "ID:123").

- **Funciones y ejemplos detallados:**

- **REGEXMATCH:** =REGEXMATCH(A1, "^\\d+\\$") verifica si A1 contiene solo dígitos (ej: "123" → VERDADERO, "12a" → FALSO).
- **REGEXREPLACE:** =REGEXREPLACE(A1, "\\s+", " ") unifica múltiples espacios en uno (ej: "hola mundo" → "hola mundo").
- **REGEXEXTRACT:** =REGEXEXTRACT(A1, "(\\d+)") extrae el primer número (ej: "ID:123" → "123").
- **Bucle implícito:** =ARRAYFORMULA(REGEXREPLACE(A1:A10, "\\s+", " ")) aplica a todo un rango, simulando un for por fila.

- **Casos de uso:**

- Extraer emails: =REGEXEXTRACT(A1, "([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,})").
- Limpiar texto: Reemplazar caracteres especiales con =REGEXREPLACE(A1, "[^a-zA-Z0-9]", "").
- **Problema común:** Texto mixto (ej: "Precio: \$45.67") necesita extracción; solución: REGEXEXTRACT con patrón adecuado.

- **Diferencias entre herramientas:**

- **Sheets/Excel:** Regex nativo (limitado en versiones antiguas de Excel; usa Power Query en lugar).
- **DAX:** No soporta regex; usa SEARCH o SUBSTITUTE para aproximaciones.
- **VBA/AppScript:** RegExp objeto permite patrones avanzados (ej: Set reg = New RegExp; reg.Pattern = "\d+").

- **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+)") y verifica el resultado.

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

Convierte estructuras crudas (arrays, JSON) en tablas organizadas.

- **Transformaciones detalladas:**

- **Arrays a Tabla:**

- `=FLATTEN(A1:C3)` aplana una matriz (ej: {"a","b";1,2} → columna "a,1,b,2").
 - `=TRANSPOSE(A1:A3)` gira filas a columnas (ej: A1:A3 = "a","b","c" → fila "a,b,c").
 - `=SLICE(A1:A10, 2, 5)` extrae subrango (ej: A1:A10 = 1-10 → A2:A6).

- **Transformación de Fechas:** Convierte texto a fecha estándar.

- Ej: "Oct, 23 2025" → `=DATEVALUE(SUBSTITUTE(SUBSTITUTE(A1, "Oct", "10"), ",", "")) & TEXT(..., "dd/mm/yyyy")` → "23/10/2025".
 - Bucle implícito: `=ARRAYFORMULA(DATEVALUE(SUBSTITUTE(SUBSTITUTE(A1:A10, "Oct", "10"), ",", "")))`.

- **JSON a Tabla:**

- `=IMPORTJSON(URL)` (add-on) convierte JSON a columnas (ej: {"name":"Ana","age":25} → tabla).
 - Power Query > De JSON > Expandir anidaciones (ej: arrays dentro de JSON).
 - Bucle implícito: `=MAP(JSON array, LAMBDA(item, SPLIT(item, ",")))` procesa elementos (ej: ["a,b","c,d"] → filas).

- **Ejemplo práctico:**

- A1: {"a","b";1,2} → `=FLATTEN(TRANSPOSE(A1:B2))` → columna.
 - A1: "Oct, 23 2025" → `=TEXT(DATEVALUE(SUBSTITUTE(A1, "Oct", "10")), "dd/mm/yyyy")` → "23/10/2025".

- **Diferencias entre herramientas:**

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

- **Tabla de Transformación:**

Estructura	Sheets/Excel	DAX	AppScript/VBA	Q
Array a Tabla	=FLATTEN(A1:B2)	GENERATE	flatMap / Loop flatten	
JSON a Tabla	Power Query / IMPORTJSON	PARSEJSON	JSON.parse + loop	
Fecha (Texto a Fecha)	=TEXT(DATEVALUE(...), "dd/mm/yyyy")	FORMAT([Date], "...")	Date.parse / CDate	

Pausa: Convierte "Oct, 23 2025" en A1 a "23/10/2025" con la fórmula.

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

Opera datos como una base de datos relacional.

- **Operaciones detalladas:**

- **Inner Join:** =QUERY(A1:B10 & C1:D10, "select * where Col1=Col3") une por coincidencia (ej: UID en ambas tablas).
 - Ejemplo: A1:B5 (UID, Nombre) y C1:D5 (UID, Edad) → QUERY filtra donde UID coincide.

- **Append vs Update Rows:**

- **Append:** = {A1:B10; C1:D10} apila tablas verticalmente (ej: añade nuevas filas de otra tabla con mismo esquema).
 - Sheets: Simple concatenación; no actualiza existentes.
 - Excel: Power Query > Anexar > Selecciona tablas > Carga (nueva tabla).
 - DAX: UNION apila sin relación previa.
 - AppScript/VBA: concat arrays o Range.Copy.
- **Update Rows:** Actualiza filas existentes basándose en UID.
 - Sheets: No nativo; usa VLOOKUP + IF para reemplazar (ej: =IFERROR(VLOOKUP(UID, nueva_tabla, 2, FALSE), valor_original)).
 - Excel: Power Query > Combinar > Usa UID > Reemplazar valores.
 - DAX: Relaciona tablas con UID y usa CALCULATE para actualizar.
 - AppScript: Loop con getRange/setValue; VBA: Loop con Find.
- **Contexto UID:** Si tablas no tenían relación previa pero comparten UID, Power Query/DAX crean relación dinámica; Sheets requiere QUERY manual.
- **Columnas Condicionales:** =ARRAYFORMULA(IF(A1:A10>10, "Alto", "Bajo")) genera columna dinámica.
- **Transpose/Flatten/Reduce:**
 - =TRANSPOSE(A1:A3) gira (ej: columna a fila).
 - =FLATTEN(A1:C3) aplana (ej: matriz a columna).
 - =REDUCE(0, A1:A10, LAMBDA(acc, val, acc+val)) suma acumulativa.

- Ejemplo práctico:
 - A1:B3 (UID, Nombre) y C1:D3 (UID, Edad) → =QUERY({A1:B3; C1:D3}, "select * where Col1 is not null").
- Diferencias entre herramientas:
 - Sheets: QUERY simula SQL; append manual.
 - Excel: Power Query maneja joins/append visualmente.
 - DAX: RELATED para relaciones; SUMMARIZE agrupa.
- Tabla de Operaciones DB:

Operación	Sheets/Excel	DAX	AppScript/VBA	🔗
Inner Join	=QUERY o VLOOKUP	RELATED	Loop match	
Append	= {rango1; rango2}	UNION	concat arrays	
Update Rows	VLOOKUP + IF	CALCULATE	Loop setValue	
Reduce	=REDUCE(acc, rango, LAMBDA)	SUMX	reduce func	

Pausa: Prueba un append con dos rangos en Sheets.

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

Herramientas para transformaciones avanzadas.

- **Power Query (Excel):**
 - **Append:** Datos > Combinar Consultas > Anexar > Selecciona tablas (ej: Tabla1 y Tabla2) > Carga como nueva tabla.
 - **Combine usando UID:** Datos > Combinar Consultas > Combinar > Elige columna UID > Selecciona Inner Join > Carga.
 - **Otros:** Transformar > Transponer gira filas/columnas; Expandir columnas anidadas aplana JSON.
 - Ejemplo: Une Tabla1 (UID, Nombre) y Tabla2 (UID, Edad) por UID.
- **=QUERY en Sheets:** =QUERY(A:C, "select * where A = 'x' order by B group by C") filtra (A=x), ordena (B), agrupa (C).
 - Ejemplo: A1:C10 con datos → QUERY agrupa por C con suma de B.
- **SQL:** En Power BI o scripts (ej: SELECT * FROM Tabla1 INNER JOIN Tabla2 ON Tabla1.UID = Tabla2.UID).
 - Usa para joins complejos o actualizaciones.

• **Tablas Dinámicas y Slicers:**

- **Excel/Sheets:** Insertar > Tabla Dinámica > Selecciona rango > Añade campos (ej: UID, Nombre) > Insertar Slicer > Filtra por UID o fecha.
 - Ejemplo: Slicer filtra ventas por mes en tabla dinámica.
- **Bucle implícito:** QUERY itera internamente; Power Query procesa batch sin loops visibles.
- **Diferencias:** Power Query es visual y potente; QUERY es rápida para Sheets.

Pausa: Crea una tabla dinámica con un rango y añade un slicer.

Sección 7: Consejos Finales

- **Usa LAMBDA:** Crea funciones reutilizables (ej: =LAMBDA(x, x*2)(A1)).
- **UIDs esenciales:** Aseguran integridad en joins y deduplicación.
- **Verifica locale:** Ajusta en imports 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! 😊

↳ Detalles sobre Transformación de Fechas

↳ Automatización de Datos

↳ Más Ejemplos Prácticos