

Lección 6: Loops – Repetición para Automatizar Tareas

Bienvenido de nuevo. En esta lección, abordamos los loops (bucles): mecanismos para repetir acciones hasta cumplir una condición o un número fijo de veces. Basado en el enfoque de lógica de programación anterior, adaptamos conceptos como iteración fija (for), condicional (while) y post-condicional (do-while) a nuestras herramientas. Usamos el contenido de los documentos adjuntos para enriquecer con ejemplos implícitos en fórmulas (como ARRAYFORMULA con IFs) y explícitos en scripting.

Los loops son esenciales para procesar rangos, arrays o tablas repetidamente, evitando trabajo manual. Mantendremos secciones breves, ejemplos paso a paso, tablas comparativas y pausas para probar. El tono es accesible para todos, con énfasis en diferencias prácticas entre Google Sheets, Excel, AppScript, VBA y DAX.

Sección 1: Introducción a los Loops – ¿Qué Son y Por Qué Usarlos?

Un loop es como un ciclo repetitivo: ejecuta código o fórmulas varias veces. En programación tradicional, hay tipos explícitos (for, while); en ofimática, a menudo implícitos (iteran sobre rangos sin escribir "loop").

- **Tipos principales:**

- **For (iteración fija):** Repite un número conocido de veces (ej: por cada celda en un rango).
- **While (iteración condicional):** Repite mientras una condición sea verdadera (hasta que falle).
- **Do-While:** Similar a while, pero ejecuta al menos una vez antes de comprobar.

Por qué importan: Automatizan tareas repetitivas, como sumar filas o filtrar datos. En Sheets/Excel, funciones como MAP o REDUCE simulan loops implícitos sin código imperativo.

Del contexto: En Excel/Sheets, "bucles" no son estructuras explícitas como en JavaScript/Python, sino funciones nativas que iteran sobre rangos (ej: ARRAYFORMULA con IF para condicionales).

Diferencias generales:

- Sheets/Excel: Loops implícitos via fórmulas (eficientes, pero limitados a rangos finitos).
- DAX: Iteración en medidas (ej: SUMX itera sobre tablas).
- VBA/AppScript: Loops explícitos (flexibles, pero requieren código).

Tabla de Tipos Básicos:

	Sheets/Excel			
Tipo de Loop	(Implícito)	DAX	VBA	AppScript
For (Fijo)	MAP/BYROW (por fila/columna)	ITERATE/Generate	For...Next	for (let i=0; i<n; i++)
While (Condicional)	REDUCE/SCAN (hasta condición)	WHILE en bucles custom	While...Wend	while (condición)
Do-While	No nativo; simula con IF	No directo	Do...Loop While	do { } while (condición)

Pausa: Reflexiona: ¿Qué tarea repetitiva en tu hoja podría automatizarse? Sigue para ejemplos.

Sección 2: Loop For (Iteración Fija) – Repite un Número Conocido de Veces

El for itera un conteo fijo, como "por cada elemento en un rango". Ideal para procesar arrays o rangos.

• Sintaxis básica:

- En código: Repite de 1 a N.
- En fórmulas: Funciones aplican operación a cada elemento.

Ejemplos del contexto:

- En Excel (con MAP/SEQUENCE): =MAP(SEQUENCE(10), LAMBDA(i, i*2)) – Itera 10

veces, multiplica i por 2. Resultado: Array [2,4,...,20].

- Simula: for i in range(1,11): print(i*2).
- En Sheets (con BYROW): =BYROW(A1:A10, LAMBDA(row, row*2)) – Multiplica cada valor en A1:A10 por 2, fila por fila.
- Simula: for row in range: row *= 2.
- Utilidad: Generar secuencias numéricas, aplicar operaciones repetitivas (ej: multiplicar cada celda por 2), procesar tablas (totales por fila).

Diferencias:

- Sheets: BYROW directo para rangos; ARRAYFORMULA envuelve para extender.
- Excel: SEQUENCE genera contador; no detiene dinámicamente.
- DAX: GENERATESERIES para secuencias fijas.
- VBA: For i = 1 To 10: ... Next i.
- AppScript: for...of para arrays.

Ejemplo práctico:

1. En Sheets, en A1:A5 pon números 1-5.
2. En B1: =BYROW(A1:A5, LAMBDA(row, row*2)) – Resultado spills a B1:B5 con 2,4,6,8,10.

Pausa: Prueba en Excel: =MAP(SEQUENCE(5), LAMBDA(x, x+1)). ¿Qué array obtienes?

Sección 3: Loop While (Iteración Condicional) – Repite Hasta que Falle una Condición

El while continua mientras la condición sea verdadera. Útil para acumular hasta un límite o hasta fallo.

- Sintaxis básica:
 - En código: Mientras (condición), haz algo.
 - En fórmulas: REDUCE/SCAN acumulan hasta condición.

Ejemplos del contexto:

- En Excel (con REDUCE): =REDUCE(0, SEQUENCE(100), LAMBDA(acc, iter, IF(acc < 50,

`acc + 5, acc)))` – Acumula +5 mientras acc <50; detiene en 50.

- Simula: acc = 0; while acc <50: acc +=5.
- En Sheets (con SCAN): `=SCAN(0, SEQUENCE(100), LAMBDA(acc, iter, IF(acc <50, acc +5, acc)))` – Array de valores acumulados hasta 50.
- Simula while con rastreo de pasos.
- Utilidad: Acumular valores (ej: suma hasta >100), procesar hasta condición (ej: iterar hasta encontrar valor negativo). No infinito; limitado por rango.

Diferencias:

- Sheets: SCAN muestra intermedios; ARRAYFORMULA envuelve para rangos.
- Excel: REDUCE/SCAN potentes para acumuladores; self-reference posible pero controlado por SEQUENCE.
- DAX: No while directo; usa GENERATE con filtros.
- VBA: While condición: ... Wend (riesgo infinito si no actualizas).
- AppScript: while (condición) {} – Flexible, pero vigila bucles infinitos.

Ejemplo práctico:

1. En Excel: `=REDUCE(0, SEQUENCE(10), LAMBDA(acc, i, IF(acc <20, acc + i, acc)))` – Acumula suma hasta >=20.

Pausa: En Sheets, prueba `=SCAN(0, SEQUENCE(5), LAMBDA(acc, i, acc + i))`. ¿Qué array ves?

Sección 4: Loop Do-While – Ejecuta al Menos Una Vez y Luego Comprueba

Similar a while, pero garantiza una ejecución inicial. No nativo en fórmulas; simúlalo con IF/REDUCE.

- Sintaxis básica: Haz algo, luego mientras condición.

Ejemplos del contexto (simulados):

- En fórmulas: Usa ARRAYFORMULA con IF para condicional post-ejecución (ej: procesa

rango, luego titra).

- **En VBA:** Do: ... Loop While condición.
- **Utilidad:** Procesar datos donde necesitas al menos una iteración (ej: acumular hasta condición, pero inicia siempre).

Diferencias:

- Sheets/Excel: Simula con REDUCE (inicia acc, luego condiciona).
- DAX: Bucles custom en medidas.
- VBA/AppScript: Explícito; AppScript: do { } while.

Ejemplo: En AppScript: do { sum += i; i++; } while (sum <10).

Pausa: Piensa en un caso donde necesites ejecutar al menos una vez. Prueba en VBA si tienes Excel.

Sección 5: Loops Implícitos en Fórmulas – Bucles Ocultos en Funciones Nativas

Del contexto: Muchas funciones iteran implícitamente sin escribir loops.

• Ejemplos:

- **SPLIT:** Itera sobre texto dividiendo por delimitador (ej: =SPLIT("manzana,pera,banana", ",") – Array ["manzana","pera","banana"]).
- **BUSCARV/VLOOKUP:** Busca hasta coincidencia (loop por celdas en rango).
- **SUMIF/FILTER:** Filtra/itera condicionalmente.
- **ARRAYFORMULA:** Extiende fórmula a rango (ej: =ARRAYFORMULA(A1:A10*2) – Multiplica cada celda).

Diferencias:

- Sheets: ARRAYFORMULA envuelve; más directo con BYROW.
- Excel: En 365, spill automático; usa Ctrl+Shift+Enter en viejo.

Tabla de Implícitos:

Función	Sheets/Excel	DAX	VBA/AppScript
---------	--------------	-----	---------------



SPLIT	=SPLIT(texto, delim)	No directo	Split(str, delim) / str.split
BUSCARV	=BUSCARV(valor, rango, col)	LOOKUPVALUE	Loop hasta find
FILTER	=FILTER(rango, condición)	FILTER(tabla, cond)	Filter array

Pausa: En Sheets, prueba =SPLIT("a,b,c", ","). ¿Qué array sale?

Sección 6: Loops en Scripting – Explícitos y Potentes

En VBA/AppScript, loops son directos.

- **VBA For:** For Each celda In Rango: celda.Value = celda.Value * 2 Next.
- **AppScript While:** while (i <10) { sum += i; i++; }
- **Del contexto:** Usa LAMBDA para custom loops en fórmulas, pero scripting para complejos.

Pausa: En Apps Script, escribe un for simple.

Sección 7: Consejos, Limitaciones y Diferencias Finales

- **Limitaciones (del contexto):** No infinitos; predice con SEQUENCE. No self-reference directa sin control.
- **Consejos:** Evita infinitos actualizando variables. Usa LAMBDA para reusable.
- **Sheets vs Excel:** Sheets más directo con BYROW; Excel potente con REDUCE/SCAN.
- **DAX:** Iteración en tablas, no imperativa.
- **VBA/AppScript:** Flexibles, pero código-heavy.

Conclusión: Loops automatizan repetición. Implícitos en fórmulas para simple, explícitos en código para avanzado.

Pausa final: Prueba un REDUCE. ¿Tema 7? Ej: "Funciones Avanzadas". ¡Dime! 😊

↳ Ejemplos avanzados de SCAN

↳ Condicionales anidadas



Más ejemplos en DAX