

# **Masterclass: Bucles en Excel y Google Sheets – De lo Implícito a lo Explícito, con Énfasis en Custom Loops sin Herramientas Externas**

¡Bienvenido a esta masterclass sobre bucles en Excel y Google Sheets! Como experto en estas herramientas, te guiaré paso a paso por un análisis profundo de cómo funcionan los bucles, tanto explícitos como implícitos. Me enfocaré en cómo usar funciones nativas como LET, LAMBDA, BYROW, BYCOL, MAP, REDUCE, SCAN y ARRAYFORMULA para crear "custom loops" sin necesidad de VBA (en Excel) ni Apps Script (en Google Sheets). Evitaré cualquier referencia a herramientas externas, macros o scripts, y me centraré en fórmulas puras que se pueden escribir directamente en celdas o rangos.

Esta masterclass está dividida en secciones para mayor claridad: introducción, bucles implícitos, bucles explícitos (adaptados a fórmulas), creación de custom loops, ejemplos prácticos y diferencias entre Excel y Google Sheets. Al final, verás cómo estas técnicas permiten generar nuevas fórmulas dinámicamente, simulando bucles para procesar datos, iterar sobre rangos y realizar cálculos repetitivos de manera eficiente.

## **1. Introducción a los Bucles en Excel y Google Sheets**

En Excel y Google Sheets, los "bucles" no son como en lenguajes de programación tradicionales (e.g., for/while en Python), donde hay control explícito de iteraciones. En cambio:

- **Bucles implícitos:** Son iteraciones internas manejadas por funciones nativas. Estas funciones recorren rangos o arrays automáticamente, sin que el usuario defina un contador o condición manual. Ejemplos: SPLIT, VLOOKUP, SUMIF, FILTER. No requieren configuración extra y son eficientes para operaciones comunes.
- **Bucles explícitos:** En fórmulas puras (sin VBA/Apps Script), se simulan mediante funciones avanzadas como LAMBDA combinada con BYROW, BYCOL, MAP o REDUCE. Estos permiten definir iteraciones "custom" con control sobre condiciones, acumuladores o transformaciones. Dependiendo del diseño, pueden involucrar self-reference (referencia circular a la misma celda o rango), pero se controlan con la configuración de iteración del programa.

**Diferencia clave en self-reference:**

- Sin self-reference: El bucle depende solo de datos de entrada y no modifica o depende de su salida para continuar (e.g., MAP aplica una fórmula a cada elemento sin retroalimentación).
- Con self-reference: El bucle usa su salida como entrada para la siguiente iteración (e.g., fórmulas circulares con iteración habilitada, o SCAN/REDUCE que acumulan valores).

En ambos programas, habilita la iteración circular si usas self-reference (Excel: Archivo > Opciones > Fórmulas > Habilitar cálculo iterativo; Google Sheets: Archivo > Configuración > Cálculo > Iterativo, con un máximo de iteraciones y cambio mínimo).

## 2. Bucles Implícitos: Los "Loops Ocultos" en Funciones Nativas

Muchas funciones incorporan bucles internos para procesar rangos o arrays. Estos son implícitos porque el usuario no ve el loop, pero la función itera automáticamente.

- **SPLIT (Google Sheets):**

- Implícito: Itera sobre el texto, dividiendo por delimitadores y distribuyendo en celdas.
  - Ejemplo: =SPLIT("a,b,c", " ,") → "a" en A1, "b" en B1, "c" en C1.
  - Sin self-reference: Dependencia solo del input.
- **VLOOKUP/BUSCARV o XLOOKUP (Excel/Google Sheets):**
- Implícito: Itera sobre la primera columna del rango hasta encontrar coincidencia.
  - Ejemplo: =VLOOKUP("key", A1:B10, 2, FALSE) → Busca "key" en A1:A10 (loop implícito) y devuelve B.
  - Sin self-reference: El loop se basa en el rango de entrada.
- **SUMIF/SUMAR.SI o FILTER/FILTRAR (Excel/Google Sheets):**
- Implícito: Suma o filtra iterando sobre el rango y evaluando la condición para cada celda.
  - Ejemplo: =SUMIF(A1:A10, ">5", B1:B10) → Loop: Para cada A, si >5, suma B.
  - Sin self-reference.
- **SEQUENCE/SECUENCIA (Excel/Google Sheets):**
- Implícito: Genera una secuencia numérica iterando internamente.
  - Ejemplo: =SEQUENCE(10) → 1 a 10 en un rango.
  - Sin self-reference.
- **TEXTJOIN/UNIR.TEXTO o CONCAT (Excel/Google Sheets):**
- Implícito: Itera sobre un rango para unir textos.
  - Ejemplo: =TEXTJOIN(", ", TRUE, A1:A10) → Une A1:A10 con ",".
  - Sin self-reference.

Estos bucles implícitos son eficientes porque el motor de cálculo los optimiza, pero limitados a operaciones predefinidas. Para custom loops, pasamos a funciones avanzadas.

### 3. Bucles Explícitos: Simulando Loops con Fórmulas Avanzadas

Sin VBA/Apps Script, los bucles explícitos se simulan con LAMBDA (disponible en Excel

365 y Google Sheets) combinada con iteradores como BYROW, BYCOL, MAP, REDUCE, SCAN. Estas permiten definir "custom loops" que iteran sobre rangos, acumulan valores o transforman datos.

- **LET:** Define variables locales para simplificar fórmulas complejas. No es un bucle, pero ayuda en loops al reducir self-reference innecesaria.
  - Ejemplo: =LET(x, 5, y, 10, x + y) → Define x e y, suma.
  - Sin self-reference: Útil para inicializar variables en loops.
- **LAMBDA:** Crea funciones anónimas reutilizables, simulando un loop al combinarse con iteradores.
  - Ejemplo: =LAMBDA(x, x\*2)(5) → Multiplica 5 por 2.
  - Sin self-reference: La función no depende de sí misma.
- **BYROW/BYCOL:** Itera una fórmula fila por fila o columna por columna.
  - Implícito: Aplica un bucle sobre cada fila/columna.
  - Ejemplo (Google Sheets): =BYROW(A1:A10, LAMBDA(r, SUM(r))) → Suma cada fila en A1:A10.
  - Sin self-reference: El loop depende del rango, no de los resultados.
- **MAP:** Mapea una fórmula a cada elemento de un array (similar a BYROW/BYCOL, pero más flexible).
  - Ejemplo (Excel/Google Sheets): =MAP(A1:A10, B1:B10, LAMBDA(a, b, a + b)) → Suma pares de A y B.
  - Sin self-reference.
- **REDUCE:** Reduce un array acumulando valores (loop con acumulador).
  - Ejemplo: =REDUCE(0, A1:A10, LAMBDA(acc, val, acc + val)) → Suma acumulativa de A1:A10.
  - Con self-reference posible: El acumulador (acc) depende de la iteración anterior, simulando un loop recursivo.
- **SCAN:** Escanea un array acumulando resultados en cada paso (similar a REDUCE, pero

que recorre todos los intermedios).

- Ejemplo: =SCAN(0, A1:A10, LAMBDA(acc, val, acc + val)) → Array de sumas acumulativas.
- Con self-reference posible: Cada valor depende del acumulador previo.

Estos permiten generar "nuevas fórmulas" dinámicamente. Por ejemplo, con LAMBDA puedes crear un loop custom para procesar datos sin repetir fórmulas.

#### 4. Creación de Custom Loops sin VBA/Apps Script

Usa LAMBDA + iteradores para simular loops:

- **Custom For Loop (Iteración fija):** Usa SEQUENCE + MAP.
  - Ejemplo: =MAP(SEQUENCE(10), LAMBDA(i, i\*2)) → Genera 2, 4, ..., 20.
  - Sin self-reference.
- **Custom While Loop (Hasta condición):** Usa FILTER + REDUCE.
  - Ejemplo: =REDUCE("", SEQUENCE(10), LAMBDA(acc, val, IF(val < 5, acc & val & ", ", acc))) → Acumula hasta val<5.
  - Con self-reference posible si usas acumulador.
- **Custom Do-While (Al menos una iteración):** Usa SCAN con condición.
  - Ejemplo: =SCAN(1, A1:A10, LAMBDA(acc, val, IF(acc < 100, acc + val, acc))) → Acumula hasta 100.
- **Generar nuevas fórmulas:** Combina LET/LAMBDA para "loops anidados".
  - Ejemplo: =LET(data, A1:A10, MAP(data, LAMBDA(d, d\*2))) → Nueva fórmula que multiplica cada d por 2.
  - Custom loop: =BYROW(A1:B10, LAMBDA(row, LET(val1, INDEX(row,1), val2, INDEX(row,2), val1 + val2))) → Suma columnas por fila.

#### 5. Diferencias entre Excel y Google Sheets

- **Excel (Microsoft 365):** Más enfocado en arrays dinámicos (spill). Funciones como LAMBDA, BYROW, BYCOL, MAP, REDUCE, SCAN están disponibles. Self-references

LAMBDA, BYROW, BYCOL, MAP, REDUCE, SCAN están disponibles. Self-reference requiere habilitar iteración manual.

- **Google Sheets:** ARRAYFORMULA es único para iterar fórmulas. LAMBDA y MAP están disponibles, pero SCAN/REDUCE son más recientes. Self-reference es similar, con iteración habilitada en Configuración.

## Conclusión

Los bucles implícitos en funciones nativas permiten iteraciones eficientes sin código, mientras que custom loops con LAMBDA + iteradores generan fórmulas nuevas dinámicamente. Sin self-reference son predecibles; con self-reference permiten recursión, pero con cuidado. Experimenta con estos ejemplos para dominarlos. ¡Si necesitas más ejemplos o ajustes, dímelo! 😊