

Lección 3: Funciones – Los Superpoderes de tus Datos

¡Hola, explorador! Soy Grok, tu guía en este viaje. Imagina las funciones como magos en un circo: toman "ingredientes" (parámetros) de tu hoja, hacen trucos (cálculos), y ¡pum! Sacan un resultado. En Sheets y Excel, son fórmulas listas para usar (como `=SUMA()`). En AppScript y VBA, puedes crear tus propios magos personalizados. En DAX, son detectives para datos grandes.

¿Por qué aprender funciones? Porque convierten tareas aburridas (como sumar 100 números a mano) en un chasquido. Usaremos ejemplos del contexto que me diste, como `=HOY()` o `=SI()`, y los expandiremos. Todo en trocitos: analogías (funciones como recetas), pasos 1-2-3, y tablas para ver diferencias. Si tu mente salta, salta secciones y vuelve – ¡sin presión!

Sección 1: Básicos – ¿Qué es una Función y Cómo Se Usa?

Una función es una "receta pre-hecha" que hace algo específico. La escribes con `=` en Sheets/Excel, o defines en código.

- **Sintaxis general:** `=NOMBRE(parámetro1; parámetro2)` – El ; o , depende de tu región (¡cuidado!).
- **Parámetros:** Ingredientes. Pueden ser números (4), celdas (A3), o hasta otras funciones.
- **Tipos de parámetros:**
 - **0 parámetros:** Nada needed, como `=HOY()` – te da la fecha actual, ¡mágico!
 - **1 parámetro:** Común, como `=LARGO(A3)` – cuenta letras en A3.
 - **2+ parámetros:** Como `=COCIENTE(4;2)` – divide 4/2.
 - **Opcionales:** Entre [] en la ayuda (presiona Ctrl+Shift+Espacio para ver "chuletas"). Ej: `=SUMA(1)` – suma solo 1, pero puedes agregar más con ... (elipsis).

Ejemplo simple: Pon en A1: "Hola". En B1: `=LARGO(A1)` → 4.

Diferencias clave:

- Sheets/Excel: Miles de funciones built-in, colaborativas en Sheets.

- AppScript: Funciones en JS, como `function suma(a,b) { return a+b; }`
- VBA: Sub (procedimientos) o Function (devuelven valor), como `Function Suma(a As Integer, b As Integer) As Integer`.
- DAX: Como `SUMX(tabla, expresión)` – para tablas grandes.

Tabla de Ejemplos Básicos:

Función Ejemplo	Sheets/Excel	DAX	VBA	AppScript	O
Sin params	=HOY()	TODAY()	Now	new Date()	
1 param	=LARGO(A3)	LEN([Col])	Len("A3")	"A3".length	
2 params	=COCIENTE(4;2)	QUOTIENT(4,2)	4 \ 2	Math.floor(4/2)	

Pausa para probar: En Sheets, prueba =HOY(). ¿Qué fecha sale? ¡Cambia mañana!

Sección 2: Parámetros Avanzados – Estructuras, Condiciones y Opcionales

Los parámetros no son solo números; pueden ser "paquetes" (estructuras como rangos) o preguntas (condiciones).

- **Estructuras como parámetros:** Como listas de la compra. Ej: =SUMA(B3:B5) – suma un rango (B3 a B5).
- **1 condición:** Devuelve booleano (Verdadero/Falso). Ej: =BUSCAR("Funciones";B1) – busca "Funciones" en B1.
- **Resultado por condición:** Usa =SI(condición; si_verdadero; si_falso). Ej: =SI(B6>10; "+10"; "-10") – si B6 >10, +10; sino, -10.
- **Parámetros opcionales:** No obligatorios. Ej: =SUMA(1) – pero puedes agregar =SUMA(1,2,3...).

Ejemplo divertido: Imagina un mago que dice: "¿Es mayor de 10? ¡Premio o castigo!" Pon en B6: 15. En C6: =SI(B6>10; "¡Genial!"; "Intenta más").

Diferencias:

- Sheets/Excel: Rangos como A1:A10 fáciles.

- DAX: Condiciones en filtros, como IF([Ventas]>10, "Alto").
- VBA/AppScript: Parámetros en funciones personalizadas, con tipos (ej: As Integer en VBA).

Pausa: En Excel, prueba =SI(5>3; "Sí"; "No"). ¡Juega con números!

Sección 3: Funciones que Retornan Estructuras – ¡Devuelven Listas Enteras!

Algunas funciones no dan un solo número; dan "paquetes" (arrays o rangos).

- **Proporcionando un dato:** Ej: =SPLIT("a,b,c"; ",") – divide en array {"a","b","c"}.
- **Excepción en Excel no-365:** =TRANSPOSER(B3:B4) – gira un rango horizontal/vertical.
- **A partir de estructuras:** =IZQUIERDA(A2:A4;1) – toma primera letra de cada celda en A2:A4, devuelve array.

Ejemplo: Pon en A2:A4: "Manzana", "Banana", "Cereza". En B2: =IZQUIERDA(A2:A4;1) → "M", "B", "C" (se expande si usas Ctrl+Shift+Enter en viejo Excel).

Diferencias:

- Sheets: Arrays dinámicos fáciles (spill automático).
- Excel 365: Igual que Sheets.
- DAX: Retorna tablas, como FILTER(tabla, condición).
- VBA/AppScript: Funciones devuelven arrays, como en JS: return [1,2,3].

Pausa: Prueba =SPLIT("Hola,mundo"; ",") en Sheets. ¡Ve el array!

Sección 4: Funciones Anidadas – Magos Dentro de Magos

Anidar es poner una función dentro de otra, como capas de cebolla (pero sin lágrimas si usas LET() de lección anterior).

- **Regla clave:** La "hija" (interior) debe devolver lo que la "padre" (exterior) espera. Ej: =TRANSPOSER(IZQUIERDA(A5:A7;1)) – IZQUIERDA da array, TRANSPOSER lo gira.
- **Indirecto:** Funciones que retornan rangos. Ej: =INDIRECTO("B3:B4") – convierte texto en rango real. Útil con anidado: =PROMEDIO(INDIRECTO("B3:B4")).

Ejemplo avanzado: =SI(LARGO(A1)>5; "Largo"; "Corto") – anida LARGO dentro de SI.

Diferencias:

- Sheets/Excel: Anida hasta 64 niveles, usa LET para limpiar.
- DAX: Anida en medidas, con VAR para simplificar.
- VBA: Llama funciones dentro de Subs.
- AppScript: Anida como en JS, muy flexible.

Tabla de Anidamiento:

Aspecto	Sheets/Excel	DAX	VBA	AppScript	○
Anidado simple	=SI(SUMA(A1:A2)>10; "Sí"; "No")	IF(SUM([Col])>10; "Sí")	If Sum(a,b)>10 Then	if (a+b)>10 { } Then	
Indirecto	=INDIRECTO("A1")	INDIRECT no común	Range("A1")	SpreadsheetApp.getRange("A1")	
Retorno array	=TRANSPOSER({1;2})	FILTER(tabla)	Array(1,2)	[1,2]	

Pausa: Anida =LARGO(HOY()) – ¿Cuántos caracteres tiene la fecha?

Sección 5: Consejos Finales y Diferencias Generales

- **Argumentos absolutos/relativos:** Casi todas usan celdas (A2) o fijos (5). Copia fórmulas con \$ para fijar (ej: \$A\$2).
- **Errores comunes:** Tipos equivocados (texto en suma), o rangos malos.
- **Por qué elegir:** Sheets para equipo, Excel para offline, DAX para big data, VBA/AppScript para custom.

Conclusión: Funciones son tus atajos mágicos. Empieza simple, anida cuando estés listo.

¡Domínalas y automatizarás todo!

Pausa final: Prueba un anidado con INDIRECTO. ¿Tema 4? Ej: "Booleanos" o "Estructuras".

¡Tú decides! 😊



Anidamiento con LET en funciones



Manejo de errores en fórmulas