**💡 Tema 1: Variables - La Caja de Herramientas Digital 📦**

**1. El Concepto Fundacional: ¿Qué es una Variable? (El Contenedor)**

Imagina que estás construyendo algo con datos. Una **Variable** es simplemente un **contenedor etiquetado** 📦 en la memoria de tu computadora que guarda un valor.

* **Etiqueta (Nombre):** Cómo llamas a la caja (ej: NombreCliente).
* **Contenido (Valor):** Lo que hay dentro (ej: "Ana López" o el número 42).
* **Tipo (Material de la Caja):** Qué tipo de datos puede guardar (texto, números, fechas).

| Plataforma | ¿Dónde se usa la variable? |
| --- | --- |
| **Sheets/Excel** | Dentro de una **fórmula** o como un **Rango Nombrado**. |
| **AppScript/VBA** | Dentro de un **programa** (macro) para hacer cálculos complejos o repetir tareas. |
| **DAX** | Dentro de una **medida** o **columna calculada** para guardar resultados intermedios. |

Exportar a Hojas de cálculo

**2. Variables Rápidas en Fórmulas: LET() y Rangos Nombrados**

En Excel y Google Sheets, las variables te ayudan a simplificar fórmulas largas, evitar calcular el mismo valor varias veces, y hacer que el código sea fácil de leer.

**A. La Función LET() (Excel / Google Sheets) 💨**

Piensa en LET() como ponerle un **apodo temporal** a un cálculo.

* **Uso:** Defines la variable *solo* dentro de esa fórmula.
* **Ventaja clave:** Una vez que el valor complejo está en la variable, si lo usas 10 veces, solo se calcula **una vez**. ¡Ahorras tiempo y recursos!
* **Ejemplo:** =LET(Impuesto; A2\*0.18; A2 + Impuesto)
  + **Impuesto** es la variable. Solo existe en esa celda.

**B. Rangos Nombrados (Excel / Google Sheets) 🏷️**

Esto es como poner un nombre **público y permanente** a una celda o un valor.

* **Diferencia clave con LET():** El nombre es visible en **todo el archivo**. Es útil para constantes (como la tasa de IVA) que usas en muchas fórmulas diferentes.

**3. Variables DAX: El Botón de Pausa del Calculador ⏸️**

DAX (Power BI, Power Pivot) es un lenguaje de fórmulas, no de programación. Aquí, la variable (VAR) tiene un objetivo muy claro: **optimizar el cálculo**.

* **Sintaxis:** Se usa la palabra clave **VAR** dentro de una Medida o Columna.
* **DAX VAR (El Almacén de Resultados):** Cuando DAX encuentra una variable VAR, **guarda el resultado** de esa expresión *inmediatamente* (se le llama **materialización**). Luego, puedes usar ese resultado guardado sin que DAX tenga que recalcularlo cada vez.
* **Alcance:** Es **local**. Solo existe dentro de la Medida o la función de consulta donde la defines.
* **Tipo de Contenido:** Puede guardar un solo número **(Scalar)** o una **Tabla completa** (una "Tabla Variable" temporal).

**4. Variables de Programación: VBA (Estricto) vs. AppScript (Flexible) 🚦**

En los lenguajes de *scripting* (AppScript, VBA), las variables son los bloques de construcción de tu código. La gran diferencia está en las reglas del juego:

**A. El Tipado (El Material de la Caja)**

| Tipo de Tipado | VBA (Visual Basic for Applications) 👨‍🏫 | AppScript (JavaScript) 🧑‍🤝‍🧑 |
| --- | --- | --- |
| **Modelo** | **Fuerte y Explícito** (Estricto). | **Débil y Dinámico** (Flexible). |
| **Declaración** | **Obligatoria.** Debes decirle *exactamente* qué tipo de datos va a guardar la caja (Dim i As Integer). | **Flexible.** Simplemente declaras (let i = 10;). El código **adivina** (infiere) el tipo basándose en el contenido. |
| **Riesgo** | Menos riesgo de errores, pero más lento de escribir. | Más rápido de escribir, pero más riesgo de errores inesperados (ej: sumar un texto con un número). |

Exportar a Hojas de cálculo

**B. Alcance y Ciclo de Vida (Dónde puedes usar la caja)**

El **Alcance (Scope)** define dónde en tu código puedes "ver" o usar la variable.

| Contexto | VBA (Dim, Public, Private) | AppScript (var, let, const) |
| --- | --- | --- |
| **Local (Dentro de una función)** | **Dim** o **Private**. La variable muere cuando la función termina. | **let** y **const** (El estándar moderno, solo visibles en el bloque de código {}). |
| **Global (En todo el proyecto)** | **Public**. La variable existe mientras el libro esté abierto. | **var** (Antiguo, se extiende a toda la función y puede causar confusión). **const** (Para valores que nunca cambian, ej: π). |

Exportar a Hojas de cálculo

**5. Por Valor vs. Por Referencia: ¿Compartir una Copia o la Llave Única? 🔑**

Este es el concepto **más importante** para entender cómo funcionan las variables en programación (VBA/AppScript).

**1. Paso por Valor (Una Copia de la Nota) 📝**

* **¿Qué pasa?** El programa hace una **copia** del valor y se la pasa a la función.
* **Consecuencia:** Si la función cambia el valor, solo cambia la copia; **el valor original queda intacto**.
* **Aplica a:** Tipos de datos simples (números, texto, booleanos) en VBA (por defecto) y AppScript.

**2. Paso por Referencia (La Llave a la Caja Central) 🗝️**

* **¿Qué pasa?** El programa no pasa el contenido, sino la **dirección de memoria** (la "llave" 🗝️) donde está almacenado.
* **Consecuencia:** Si la función usa esa llave para cambiar el contenido, el **valor original SÍ cambia**.
* **Aplica a:**
  + **VBA:** Debes usar la palabra clave **ByRef** explícitamente (o ByVal para forzar la copia).
  + **AppScript/JS:** Automáticamente para **Objetos y Arrays**. Si pasas un Array a una función, esa función puede modificar el Array original sin que te des cuenta.

**6. Contenedores Complejos: Arrays y Tablas (La Estantería) 🧱**

A veces, una sola caja no es suficiente; necesitas una **estantería** para guardar muchos valores relacionados. Esto es un Array (una lista ordenada) o una Tabla.

| Plataforma | ¿Cómo se define la "Estantería"? | Características clave |
| --- | --- | --- |
| **Sheets/Excel** | **Literal de Array** con llaves **{}**. | Se usa para crear bloques de datos **temporales** en una fórmula: ={"Nombre"\"Edad"; "Ana"\25}. |
| **VBA** | **Dim Nombre(5) As Integer**. | La estantería es de **tamaño fijo** (si no usas ReDim). Puedes elegir si empieza a contar desde 0 o 1 (Option Base). |
| **AppScript** | **let miLista = []**. | El Array es **dinámico** (puede crecer). Es la herramienta principal para leer/escribir datos de la hoja con getValues() y setValues(). |
| **DAX** | **Tablas Variables** (con VAR). | No es una estantería, sino una **Tabla de Datos completa** (con filas y columnas). Solo existe temporalmente durante el cálculo de la Medida. |