

Lección 7: Importar – Traer Datos y Recursos de Fuentes Externas

Bienvenido de nuevo. En esta lección, exploramos cómo importar datos y recursos en tus herramientas de ofimática y programación. Importar significa traer información de archivos externos (como CSV o PDF), módulos de código (como scripts .gs), referencias a otras hojas o libros, e integraciones con sistemas como Power BI o Python. Esto ahorra tiempo al reutilizar datos sin recrearlos manualmente.

Basado en prácticas estándar y métodos actualizados (como Power Query para PDF en Excel o gspread para Python en Sheets), cubriremos pasos prácticos. Todo en secciones concisas, con ejemplos 1-2-3 y tablas para diferencias entre Google Sheets, Excel, AppScript, VBA y DAX.

Sección 1: Introducción – ¿Qué Significa Importar y Por Qué?

Importar es como abrir una puerta a datos externos: carga archivos, conecta hojas o integra herramientas. Evita copiar-pegar manual, mantiene datos actualizados y soporta automatización.

- **Tipos clave:**

- Archivos: CSV (datos separados por comas), XLS/XLSX (hojas Excel), PDF (documentos escaneados o tabulares).
- Módulos: Archivos de código reutilizables (.gs en AppScript, .bas en VBA, .html para web).
- Referencias: A celdas/hojas en el mismo archivo o externo.
- Integraciones: Power BI para visualización, Python para procesamiento avanzado.

Por qué importa: Maneja grandes volúmenes (ej: importar ventas de CSV diario), colabora (referencias a sheets compartidos) y extiende funcionalidades (Python para análisis complejos).

Diferencias generales:

- Sheets: Nube-focalizado, fácil para web/CSV.

- Excel: Potente con Power Query para formatos complejos.
- AppScript/VBA: Scripts para imports custom.
- DAX: Importa en Power BI para modelos.

Pausa: Piensa en un archivo CSV que uses. Sigue para importarlo.

Sección 2: Importar Archivos – CSV, XLS/XLSX, PDF y Más

Importa datos de archivos externos para poblar hojas o tablas.

- **CSV:** Fácil, estructurado en comas.
 - Ejemplo en Sheets: Archivo > Importar > Subir > Selecciona CSV > Importar datos.
 - Ejemplo en Excel: Datos > Obtener Datos > De Archivo > De Texto/CSV > Selecciona > Cargar.
- **XLS/XLSX:** Hojas Excel completas.
 - Ejemplo en Sheets: Archivo > Importar > Subir XLSX > Importar como nueva hoja.
 - Ejemplo en Excel: Archivo > Abrir > Selecciona XLSX (o Power Query para combinar múltiples).
- **PDF:** Más complejo; extrae tablas o texto.
 - Ejemplo en Excel: Datos > Obtener Datos > De Archivo > De PDF > Selecciona páginas/tablas > Cargar (usa Power Query).
 - Ejemplo en Sheets: No nativo; usa add-ons como "PDF to Sheets" o AppScript con API externas (ej: UrlFetch a servicio de parsing).

Pasos generales para PDF:

1. En Excel: Datos > De PDF > Navega al archivo > Elige tablas detectadas.
2. En Sheets: Usa script:

```
function importPDF() { var pdf = DriveApp.getFilesByName('file.pdf').next(); /* parse con libs */ }
```

 – Requiere add-ons para parsing completo.

Diferencias:

- Otros formatos: JSON/XML via IMPORTXML en Sheets; TXT via Power Query en Excel.

Tabla de Importación de Archivos:

Formato	Sheets	Excel	AppScript/VBA/DAX	
CSV	Archivo > Importar o =IMPORTDATA(URL)	Power Query > De Texto/CSV	AppScript: Utilities.parseCsv; VBA: Workbooks.OpenText	
XLS/XLSX	Archivo > Importar	Archivo > Abrir o Power Query > De Excel	AppScript: SpreadsheetApp.openById; VBA: Workbooks.Open	
PDF	Add-ons o script (no nativo)	Power Query > De PDF	AppScript: UrlFetch + parsing; VBA: Custom con libs; DAX: En Power BI via connector	

Pausa: Importa un CSV simple en Sheets. Prueba =IMPORTDATA("url-to-csv").

Sección 3: Importar Módulos – Reutilizar Código Externo

Módulos son archivos de código que importas para funciones reutilizables (.gs para AppScript, .bas/.cls para VBA, .html para templates web).

- **En AppScript:** Añade bibliotecas (scripts externos).
 - Ejemplo: Editor > Bibliotecas > Añadir > Pega ID de script > Versión > Guardar. Usa como MiLibreria.funcion().
- **En VBA:** Importa módulos desde archivos.
 - Ejemplo: VBE > Archivo > Importar Archivo > Selecciona .bas > OK. Arrastra-drop en VBE para copiar.
- **HTML:** Para web apps en AppScript.
 - Ejemplo: Crea .html > Usa HtmlService.createHtmlOutputFromFile('miHtml').

Pasos para AppScript:

1. Obtén ID de script externo.

2. Añade como biblioteca.
3. Llama funciones importadas.

Diferencias:

- Sheets/Excel: No módulos nativos; usa add-ons.
- DAX: Importa via Power BI datasets, no código directo.

Tabla de Importación de Módulos:

Tipo	Sheets/AppScript	Excel/VBA	DAX	
.gs/.bas	AppScript: Añade biblioteca por ID	VBA: Importar en VBE	N/A (usa funciones custom en Power BI)	
.html	AppScript: createHtmlOutputFromFile	VBA: No directo; usa web queries	N/A	

Pausa: En AppScript, añade una biblioteca pública (ej: OAuth2) y prueba.

Sección 4: Referencias a Otras Hojas o Libros – Conectar Interno/Externo

Referencia datos de otras hojas (mismo archivo) o spreadsheets (externos).

- **Mismo archivo:** =Hoja2!A1.
- **Externo en Sheets:** =IMPORTRANGE("url-spreadsheet", "Hoja1!A1:B10").
- **Externo en Excel:** =[libro.xlsx]Hoja1!A1 (abre ambos para actualizar).

Pasos para IMPORTRANGE:

1. Copia URL del spreadsheet fuente.
2. En celda: =IMPORTRANGE("url", "rango").
3. Autoriza acceso primera vez.

Diferencias:

- Sheets: IMPORTRANGE para externo; actualiza en vivo.

- Excel: Referencias externas requieren abrir libros; usa Power Query para dinámico.

Tabla de Referencias:

Tipo	Sheets	Excel	AppScript/VBA
Misma hoja	=A1	=A1	getRange('A1')
Otra hoja mismo file	=Hoja2!A1	=Sheet2!A1	getSheetByName('Sheet2').getRange('A1')
Externo	=IMPORTRANGE(url, rango)	= [file.xlsx]Sheet1!A1	AppScript: SpreadsheetApp.openByUrl; VBA: Workbooks.Open

Pausa: Prueba IMPORTRANGE con un sheet tuyo.

Sección 5: Integraciones con Power BI y Python – Importar Avanzado

Extiende con herramientas externas.

- **Power BI:**

- Con Excel: Archivo > Exportar > Analizar en Power BI; o Power BI > Obtener Datos > De Excel.
- Con Sheets: Power BI > Obtener Datos > Google Sheets > URL > Conectar (usa connector).

- **Python:**

- En Excel: Fórmulas > Insertar Python > Escribe código (ej: import pandas as pd; df = pd.read_csv('file.csv')).
- En Sheets: Usa gspread (instala via pip fuera, pero en AppScript integra con Python via API); o Colab: from gspread import Client; sheet = gc.open('name').

Pasos para Python en Excel:

1. Activa Python en Excel (si disponible).

2. Celda: =PY("código python").
3. Para import: df = pd.read_excel('file.xlsx').

Diferencias:

- Power BI: Visualiza imports; integra con DAX para modelos.
- Python: Procesamiento avanzado; en Sheets via scripts o Colab.

Tabla de Integraciones:

Herramienta	Sheets	Excel	AppScript/VBA/DAX	?
Power BI	Conektor directo via URL	Importar workbook	DAX: En Power BI datasets	
Python	gspread en scripts/Colab	Nativo via Fórmulas > Python	VBA: No directo; AppScript: Llama APIs Python	

Pausa: Prueba Power BI con un sheet simple.

Sección 6: Consejos Finales y Diferencias

- **Seguridad:** Autoriza accesos; usa archivos confiables.
- **Limitaciones:** PDF no siempre perfecto (OCR para escaneados).
- **Sheets vs Excel:** Sheets nube-orientado; Excel robusto offline con Power Query.

Conclusión: Importar conecta tu trabajo. Comienza con CSV simples, avanza a integraciones.

Pausa final: Importa un PDF en Excel. ¿Tema 8? Ej: "Exportar". ¡Dime! 😊

↳ Ejemplos avanzados de IMPORTRANGE

↳ Exportar datos desde Sheets

↳ Más ejemplos prácticos en tablas