

LOS MEJORES 70 TRUCOS

DE MICROSOFT EXCEL



fx



Los mejores 70 trucos de Microsoft Excel

que todo experto debe conocer

©Didier Atehortua Morales, 2021

©Julio César Rendón Orozco, 2021

©Yibson Atehortua Morales, 2021

©Cristhian Alberto Cabra Hernández, 2021

ISBN: 978-958-53268-1-1

Sello editorial: DATA ICE S.A.S (978-958-53268)
Reservados todos los derechos. No se permite la reproducción total o parcial de esta obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de los titulares del copyright. La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.

Los mejores 70 trucos de Microsoft Excel

2021

Contenido Resumido

Capítulo 1: Fundamentos.....	12
Capítulo 2: Fórmulas y Funciones.....	37
Capítulo 3: Informes y Gráficos	102
Capítulo 4: Excel VBA.....	137
Capítulo 5: Hacks Power Query	189

Contenido Especifico

Contenido Resumido	4
Contenido Especifico	5
Expresiones de Gratitud	8
Dedicatorias	9
Acerca de los autores.....	10
Introducción	11
Capítulo 1: Fundamentos.....	12
1.1 Analizar con pocos clics y subtotales.....	12
1.2 Controlar todo con validación de datos	17
1.3 Validación de datos dependiente.....	20
1.4 Convertir números como texto sin que se te cierre el Excel (No responde).....	22
1.5 Convertir valores a fechas sin formulas.....	24
1.6 Comentarios y notas en office 365	26
1.7 Poner Excel modo presentación manual	29
1.8 Navegar por Excel como un experto.....	32
1.9 Agregar complementos de Excel (Tienda)	33
1.10 Recuperar archivos no guardados	36
Capítulo 2: Fórmulas y Funciones.....	37
2.1 Dos o más criterios de búsqueda en BUSCARV.....	38
2.2 Campo dinámico en BDSUMA	43
2.3 El poder del SUMAR.SI.CONJUNTO	46
2.4 La incógnita función SIFECHA	48
2.5 Proteger fórmulas a otro nivel	50
2.6 Restringir fechas con DIASEM	55
2.7 Dominando el CONTAR.SI.CONJUNTO con operadores y comodines	59
2.8 Simplificando la concatenación con UNIRCADENAS	63
2.9 Revolucionando las funciones MAX y MIN	65
2.10 Realiza búsquedas de columnas de forma horizontal.....	67
2.11 La magia de concatenar varios tipos de formatos	69
2.12 Rangos dinámicos DESREF.....	72
2.13 Encontrar el último registro en una base de datos	78
2.14 El uso correcto de los comodines “*”	82
Caso 1: Comodín a la derecha	82
Caso 2: Comodín a la izquierda.....	84
Caso 3: Comodín ambos lados.....	85
2.15 Tabular formulas a nivel profesional.....	86
2.16 Nombrar rangos a un solo Clic	88
2.17 Sacando el potencial del INDIRECTO	90
2.18 Sacar valores UNICOS ordenados y automatizados	92
2.19 Búsquedas aproximadas	94
2.20 Sumar valores no continuos.....	97

Capítulo 3: Informes y Gráficos102

3.1 Gráfico de barras con lupa	102
3.2 Anexar papeles de trabajo a informes	109
3.3 Validar en segmentación de datos la selección de dos campos	112
3.4 Resaltar cumplimiento de presupuesto en gráfico de barras	116
3.5 Tablas dinámicas con minigráficos	121
3.6 Gráfico de barras para presentar variaciones de VENTAS y PRESUPUESTO	124
3.7 Resaltar registro con base a valor de una lista desplegable.....	128
3.8 Expresar cifras en los informes en miles y millones.....	130
3.9 Copiar Estilos de segmentación de datos de un informe a otro.....	132
3.10 Gráficos premium para ahorrar tiempo	135

Capítulo 4: Excel VBA.....137

4.1 Duplicar macros entre proyectos método 1	139
4.2 Duplicar macros entre proyectos método 2	141
4.3 Variables obligatorias en los proyectos.....	144
4.4 Atajos de escritura al declarar variables	146
4.5 Evitar parpadeo de pantalla	147
4.6 Evita colapsarte por cálculos automáticos	149
4.7 Eventos que obstaculizan las macros	151
4.8 Alertas para tus macros	153
4.9 Mensajes sin interrumpir la macro	157
4.10 Conoce los tiempos de tus macros	159
4.11 Aprende a buscar palabras con atajos.....	160
4.12 Realiza pruebas a los resultados	163
4.13 Inspecciona tus variables	166
4.14 Valida errores con depuración y compilar	168
4.15 El poder del filtro avanzado en búsquedas.....	169
4.16 Mejora tus búsquedas con Array	173
4.17 Cómo cargar PDFs en un formulario.....	177
4.18 Autocompletar palabras en VBA	182
4.19 Buscar definiciones y última posición.....	183
4.20 Application.InputBox	185
4.21 Bonus 3 palabras mágicas para buscar	187

Capítulo 5: Hacks Power Query189

5.1 Consolidar todas las tablas de un libro de Excel en el mismo archivo.....	190
Ir a Power Query	191
Editar consultas	197
Aprovechar el nombre de las Tablas.....	199
5.2 Ruta dinámica	203
5.3 Segmentar o filtrar información mediante listas en Excel	211
Columnas personalizadas	214
5.4 Crear tabla calendario automática en Power Query.....	218
5.5 Identificar campos faltantes entre dos tablas a un clic.....	225
5.6 Desactiva el encabezado y tipo cambiado automático y te evitas dolores de cabeza	232
5.7 Como trabajar como millones de registros en Excel con solo conexión	237
Usar datos de solo conexión.....	239

5.8 Eliminar errores al inicio, te dará un buen final.....	241
5.9 Ahorra pasos de transformación con este gran truco.....	248
5.10 Refrescar consultas automáticas con Excel y VBA	254
Actualización Automática	254
Actualización con VBA	256

Expresiones de Gratitud

- Primero que todo agradecemos a Dios por darnos los dones y talentos de poder transmitir información tan valiosa y contribuir a la educación continua de miles de personas
- A las decenas de empresas que han confiado en nosotros en desarrollos y capacitaciones, y gracias a ellas aprendemos e implementamos todo lo plasmado en este libro
- A nuestros estudiantes que cada día nos siguen y se apuntan a vivir esta aventura de los datos en las diferentes herramientas que enseñamos, gracias mil gracias
- A nuestras familias que son las que viven el intenso horario y ausencia en que pasamos escribiendo este material, aunque no las vean, ellos están ahí detrás de bambalina y con su mayor esfuerzo cuidando de nuestros pequeños hijos para sacar este plan adelante
- A todos nuestros amigos que siempre celebran cada victoria lograda, especialmente al equipo de ExcelAprende y Dataice, saben que lo que hacemos es con propósito

Dedicatorias

Este libro va dedicado especialmente a mi esposa Vanessa Quintero, siempre que inicio un proyecto como este y como cualquier otro, cree en mis locuras; a mis hijos Sofía, Miguel y Simón, que pueda ser una inspiración para ellos en todo lo que hago, a mi madre Olga Lucía Morales, que nunca dejó de creer en mí aun en las situaciones más difíciles.

--- **Didier Atehortúa Morales**

Dedico este libro a mi familia. A mi esposa quien ha estado a mi lado siempre, apoyándome e impulsándome a ir por más, a mis hijos quienes un día me motivaron preguntándome, “papi cuando vas a lanzar tu libro”, aquí lo tienen mis gemelos. A mis hermanos y mi madre quien siempre está en mi corazón y mente, y a todos aquellos que se alegran con los triunfos del otro, Dios los bendiga.

--**Yibson Atehortúa Morales**

Quiero dedicarle este libro a 4 personas que me han apoyado y hecho crecer en este último año, principalmente a mi novia Deny Milena Guerrero Saavedra quien confió en mí y me apoyo a seguir este camino al ver mi pasión en Excel, a mi equipo de trabajo Julio Rendón, Didier Atehortúa y Yibson Atehortúa quienes me dieron la oportunidad de crecer y aprender más cada día junto a ellos.

--**Cristhian Cabra Hernández**

Primeramente a DIOS que todo lo hace posible, a mi esposa que permite que mi vida sea genial, a mi madre Days que sin duda no sé qué sería de mi sin ella, mi padre que es mi ejemplo a seguir, mi familia que me apoya en todo y que sería de nuestra comunidad sin cada uno de sus miembros ;)

--**Julio C. Rendón Orozco**

Acerca de los autores



Didier Atehortúa Morales

Ingeniero financiero y de negocios, labora como consultor empresarial y docente en el manejo de las herramientas de datos, tales como Excel y Power BI.

Coautor de los libros Inteligencia de Negocios con Excel y Power BI, Estados Financieros en Excel y Power BI.



Yibson Atehortúa Morales

Administrador comercial y de mercadeo, amplia experiencia como analista en inteligencia de negocios, docente en el manejo de Excel y Power BI, consultor en procesos de automatización empresarial y de datos.



Julio César Rendón Orozco

Enamorado de DIOS, de mi familia y apasionado por compartir mi pasión por Excel. Administrador de empresas y negocios internacionales, premiado como MVP Microsoft y fundador de una de las más grandes comunidades de toda Latinoamérica en análisis de datos, Excelaprende.com con estudiantes en más de 20 países.



Cristhian Alberto Cabra

Administrador de negocios internacionales, apasionado por el desarrollo de nuevos negocios, creación de nuevos proyectos y la analítica de datos, consultor en desarrollo de soluciones basadas en Excel y VBA para la automatización de procesos en empresas mejorando los tiempos de trabajo.

Introducción

Dominar herramientas ofimáticas, de datos o de cualquier tipo de desarrollo es tener mayor ventaja sobre otro profesional, pero cada herramienta tiene secretos o trucos en su manejo; Excel es una de las herramientas más usadas a nivel mundial, con un gran portafolio de opciones, funciones, gráficos, códigos, procesos etc. En este libro vamos a tener los mejores 70 trucos que cualquier usuario de Excel debe aprender, ya que no se trata de cómo manejar la herramienta, sino de que como potencializar lo que ya se de este programa.

Cada truco aprendido en este libro lo llamamos un AS bajo la manga, ya que no solo sorprendes a la audiencia cuando los estas usando, sino que ahorras tiempo y el tiempo es vida.

Con muchos años de experiencia en enseñar, aprender y desarrollar en Excel, 4 profesionales nos hemos unido y seleccionado con mucho cuidado cada truco y su funcionalidad, y se han segmentado cada uno de ellos en Hacks de:

1. Fundamentos
2. Fórmulas y funciones
3. Informes y gráficos
4. Visual Basic for Applications
5. Power Query

Cada truco trae su material de trabajo y con una explicación tan detallada para que cualquier usuario ya sea básico o avanzado lo entienda, repita y aplique.

Este libro es para ti, solo si eres de esas personas amante de los datos, de las automatizaciones, de los atajos para llegar a un resultado, y de los que busca siempre enriquecer su conocimiento con pequeños trucos que te lleven una escala arriba. Todo analista, profesional o persona del común que piense que el tiempo es valioso y debe ser aprovechado buscando siempre su optimización en las tareas del día a día, debe adquirir este libro.

Capítulo 1: Fundamentos

Queremos darte la bienvenida a una aventura en la que hemos plasmado los Hacks “Trucos” que sin duda alguna para nosotros se han convertido en herramientas fundamentales para explotar al máximo Microsoft Excel, así que prepárate para una lectura que cambiara el rumbo de tus días.

En este primer capítulo revisaremos los fundamentos de las características más relevantes de Excel, en las que aprovecharemos los recursos con base en una de nuestras filosofías **EL PODER DE LO SIMPLE**.

Te recomendamos descargar los archivos complementarios para ir realizando las practicas e ir profundizando en lo aprendido. Iniciemos entonces esta aventura y aprendamos juntos.

1.1 Analizar con pocos clics y subtotales

Uno de los aspectos más relevantes de Excel no es conocer cómo funciona la herramienta, en cambio de esto es analizar con la herramienta, por lo que en este caso veremos la función subtotales.

Iniciemos por analizar el contexto de este caso en que contamos con una tabla y deseamos hacer un resumen de las ventas, y en caso de aplicar algún filtro debemos contar con las ventas de los datos visibles.

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446
Subtotal -->			\$ 34.768	
Total -->			\$ 34.768	

Figura 1.0 Contexto Hacks subtotales

En este caso lo aplicamos en un campo independiente de la tabla y al usar la función SUBTOTALES podremos definir en su argumento **núm_función** que operación queremos que esta realice, al seleccionar 109 indicaremos que se use la función SUMA, a diferencia de lo que vemos en la opción #2, es que al filtrar con subtotales solo veremos el resultado de los datos visibles.

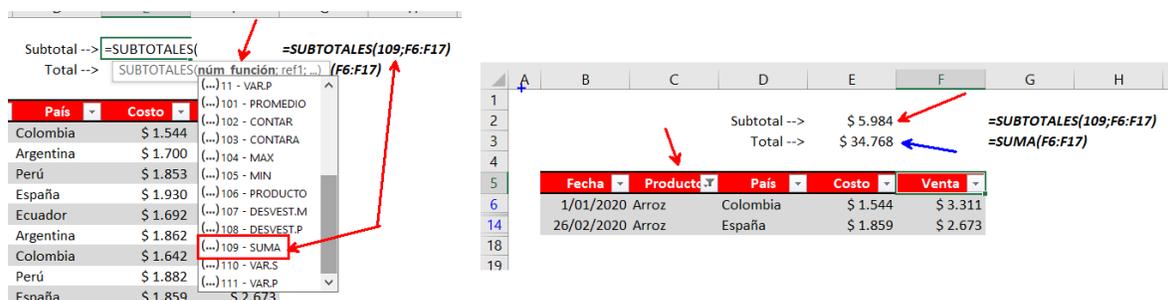


Figura 1.1 Insertar función subtotales

Lo anterior se puede aplicar si lo deseas usar desde un rango fijo y en una celda independiente, pero en la actualidad es mucho mejor trabajar con tablas y la funcionalidad de subtotales que esta trae por defecto.

Primero debemos seleccionar todos nuestros datos y convertirlos en una tabla desde la pestaña insertar y el comando Tabla, o presionando el atajo **Control + T**. Seguido de esto validamos que esté marcado el check La "tabla contiene encabezados".

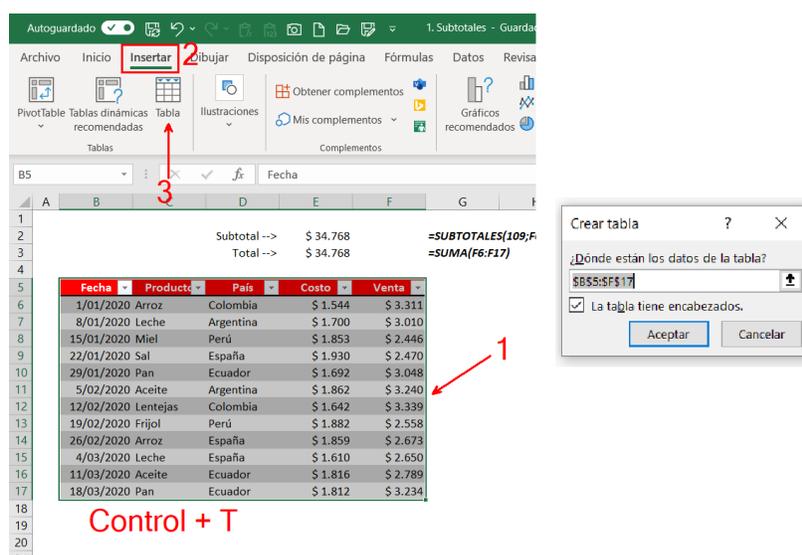


Figura 1.2 Insertar tabla

A continuación, veremos cómo solo basta con activar la opción de filas de totales y de forma automática Excel aplicará la función subtotaes, esto facilitará que puedas navegar por las diferentes funciones sin tener que insertar como en el caso anterior.

Otra de las ventajas es que esta se aplica a todo el rango de la tabla sin que tengas que definir un rango en específico.

The screenshot shows the Excel interface with the 'Diseño de tabla' ribbon active. The 'Fila de totales' checkbox is checked. The spreadsheet displays a table with the following data:

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446
22/01/2020	Sal	España	\$ 1.930	\$ 2.470
29/01/2020	Pan	Ecuador	\$ 1.692	\$ 3.048
5/02/2020	Aceite	Argentina	\$ 1.862	\$ 3.240
12/02/2020	Lentejas	Colombia	\$ 1.642	\$ 3.339
19/02/2020	Frijol	Perú	\$ 1.882	\$ 2.558
26/02/2020	Arroz	España	\$ 1.859	\$ 2.673
4/03/2020	Leche	España	\$ 1.610	\$ 2.650
11/03/2020	Aceite	Ecuador	\$ 1.816	\$ 2.789
18/03/2020	Pan	Ecuador	\$ 1.812	\$ 3.234
Total			\$ 34.768	

The callout box shows the formula bar with the formula: `=SUBTOTALS(109;[Venta])`

Figura 1.3 Subtotales desde tablas

Ahora podemos usar para los campos de texto la función recuento y nos dirá la cantidad de registros que tiene nuestra base de datos.



ADVERTENCIA: En los campos de tipo texto no se pueden usar funciones como promedio o suma, ya que estos generan un error por lo que se recomienda usar recuento para los campos de tipo texto.

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446
22/01/2020	Sal	España	\$ 1.930	\$ 2.470
29/01/2020	Pan	Ecuador	\$ 1.692	\$ 3.048
5/02/2020	Aceite	Argentina	\$ 1.862	\$ 3.240
12/02/2020	Lentejas	Colombia	\$ 1.642	\$ 3.339
19/02/2020	Frijol	Perú	\$ 1.882	\$ 2.558
26/02/2020	Arroz	España	\$ 1.859	\$ 2.673
4/03/2020	Leche	España	\$ 1.610	\$ 2.650
11/03/2020	Aceite	Ecuador	\$ 1.816	\$ 2.789
18/03/2020	Pan	Ecuador	\$ 1.812	\$ 3.234
Total		12	\$ 34.768	\$ 34.768

Figura 1.4 Subtotales para campos tipo texto



HACKS: Al usar las Tablas podemos insertar segmentación de datos y esto hace más poderoso los subtotales y la presentación de los datos.

Miremos a continuación como lo podemos lograr.

Seleccionamos la tabla → Clic en la pestaña Diseño de tabla → Insertar segmentación de datos → En la ventana de campos disponibles, podemos seleccionar los que deseemos e incluirlos como filtros. En este caso Producto → clic en Aceptar.

Figura 1.5 Insertar segmentación de datos

Con la segmentación podremos analizar información de forma rápida y al complementarlo con la activación de los Subtotales se resumirán los datos visibles, esto te permitirá tener mayor alcance.

The screenshot shows an Excel spreadsheet with a data table and a filter pane. The formula bar at the top displays the formula `=SUBTOTALES(109;[Venta])`. The data table has columns for Fecha, Producto, País, Costo, and Venta. The filter pane on the right shows a list of products, with 'Leche' selected. Red arrows indicate the relationship between the formula, the data column, and the filter selection.

Fecha	Producto	País	Costo	Venta
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
4/03/2020	Leche	España	\$ 1.610	\$ 2.650
Total	2			\$ 5.660

Subtotal --> \$ 5.660
Total --> \$ 34.768

`=SUBTOTALES(109;F6:F17)`
`=SUMA(F6:F17)`

Producto: Aceite, Arroz, Frijol, **Leche**, Lentejas, Miel, Pan, Sal

Figura 1.6 Aplicar filtros con segmentación de datos

1.2 Controlar todo con validación de datos

Una de las herramientas más poderosas de Excel es la validación, es como hacer un ahorro programado. Te explico con mayor detalle que significa esto de un ahorro 😊

Si dedicas el tiempo a controlar lo que posiblemente pueden ser errores de digitación o uso de las plantillas que desarrollamos, esto es un ahorro de tiempo que posteriormente en la etapa de consolidación o transformación de datos no tendremos que usar, ya que validaremos que se puede usar, como en este caso los nombres de productos.

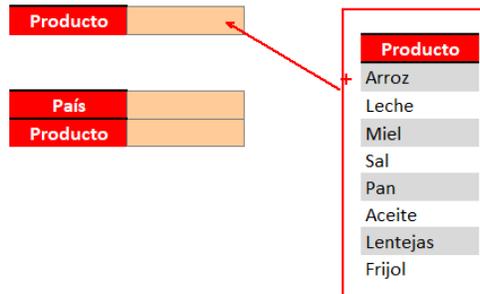


Figura 1.7 Listado productos para validación

Para poder lograrlo, primero debemos hacer una selección de la celda o rango de celdas donde deseamos hacer la validación → pestaña datos → Validación de datos.

En el cuadro de dialogo que se nos muestra, seleccionamos permitir Lista → En el campo origen, seleccionamos el rango de los nombres de producto que permitiremos → clic en Aceptar.

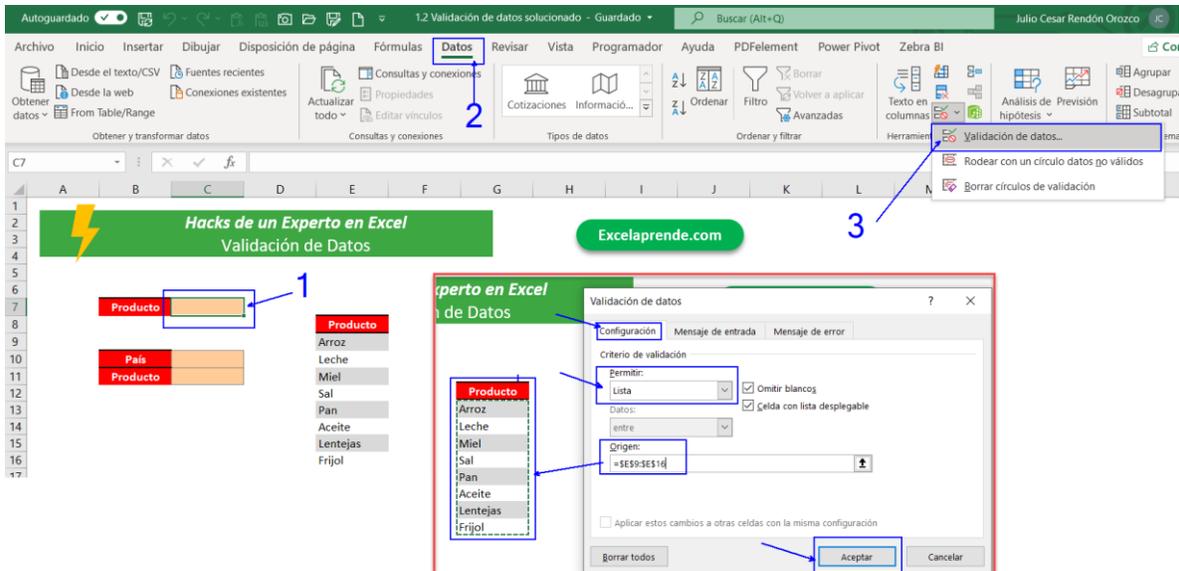


Figura 1.8 Validación de datos

Ahora contamos con la lista para seleccionar los productos y en este campo solo se permitirán estos y ningún otro.



Figura 1.9 Campos validados



HACKS: Podemos también personalizar el mensaje y alertar a los usuarios cuando intenten ingresar un valor no permitido, buscando siempre que el manejo de la plantilla sea el más óptimo posible.

En este caso y por defecto se muestra un mensaje predeterminado de error como se muestra a continuación.

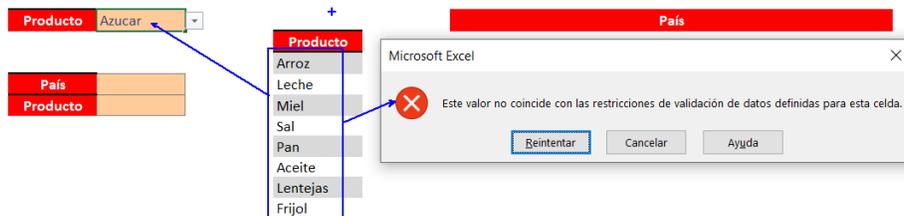


Figura 1.10 Mensaje de error en validación de datos

En este caso, Azúcar no se encuentra en la lista y nos muestra este error en pantalla, pero el usuario no tiene un mensaje claro de lo que debería hacer para poder ingresar el producto o por qué le está generando un error.

Vamos a personalizar este mensaje de tal forma que ofrezcamos una plantilla profesional para los usuarios.

Seleccionamos la celda o rango de celdas donde tenemos la validación → Datos → Validación de datos → Pestaña mensaje de error → Ingresamos el título personalizado → Damos la descripción o detalles que puede generar el error → Clic en Aceptar.

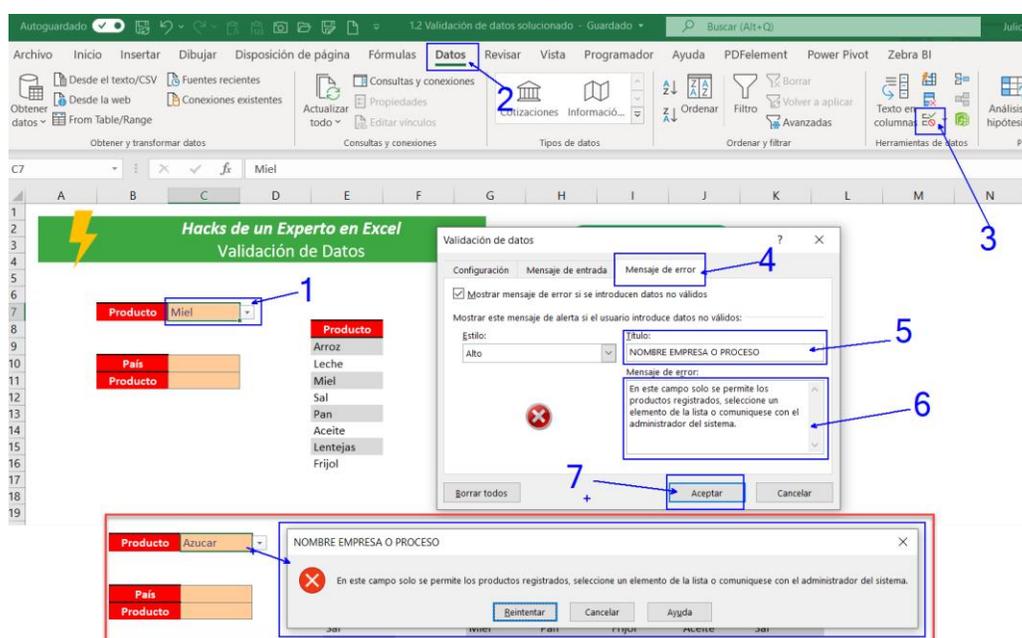


Figura 1.11 Personalizar mensaje de error en validación de datos



HACKS: En caso de que la lista que has creado deba crecer y tenga registros nuevos, te recomendamos que el rango que sirve de origen esté en formato de tabla, de esta forma al tener datos nuevos en automático aparecerán en la lista.

1.3 Validación de datos dependiente

Si de controlar lo que se ingresa en un campo se trata, contamos con la validación de datos, ahora te explicaremos como puedes potenciarla usando las listas dependiendo de una lista anterior.

Con esto lograremos que una segunda lista muestre los datos con base a lo seleccionado en una primera lista, veremos como con este truco controlaremos el ingreso de información en las plantillas y formatos.

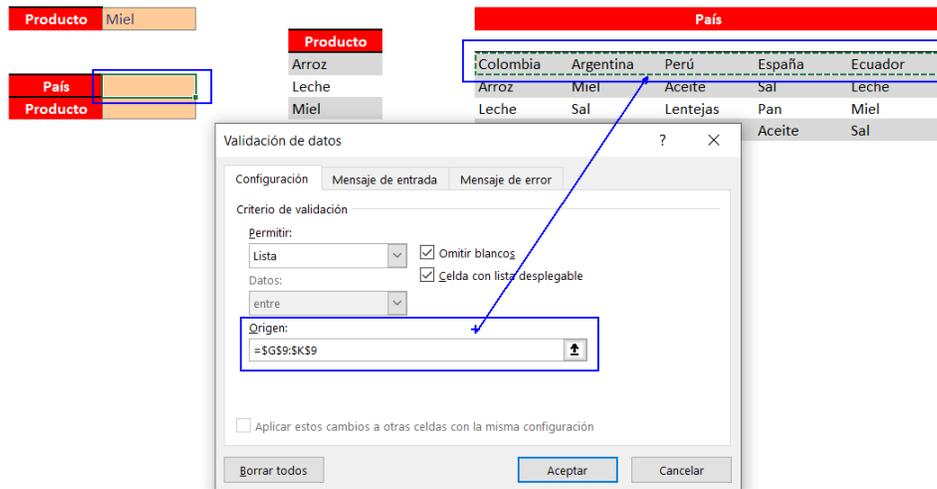


Figura 1.12 Validación de datos en lista

Vemos que también podemos tener un origen para nuestra validación de forma horizontal, y en este caso tendremos los nombres de los países y al seleccionar uno de ellos lo que buscamos en el campo producto es que solo se muestren los que corresponden a este país.

Lo primero que debemos hacer es nombrar los rangos, de esta manera Excel sabrá que debe buscar para mostrar cuando se le indique el país.

Seleccionamos el rango con los datos para crear los nombres → Pestaña fórmulas → Comando crear desde la selección → Dejar marcado solo Fila superior → Clic en Aceptar.

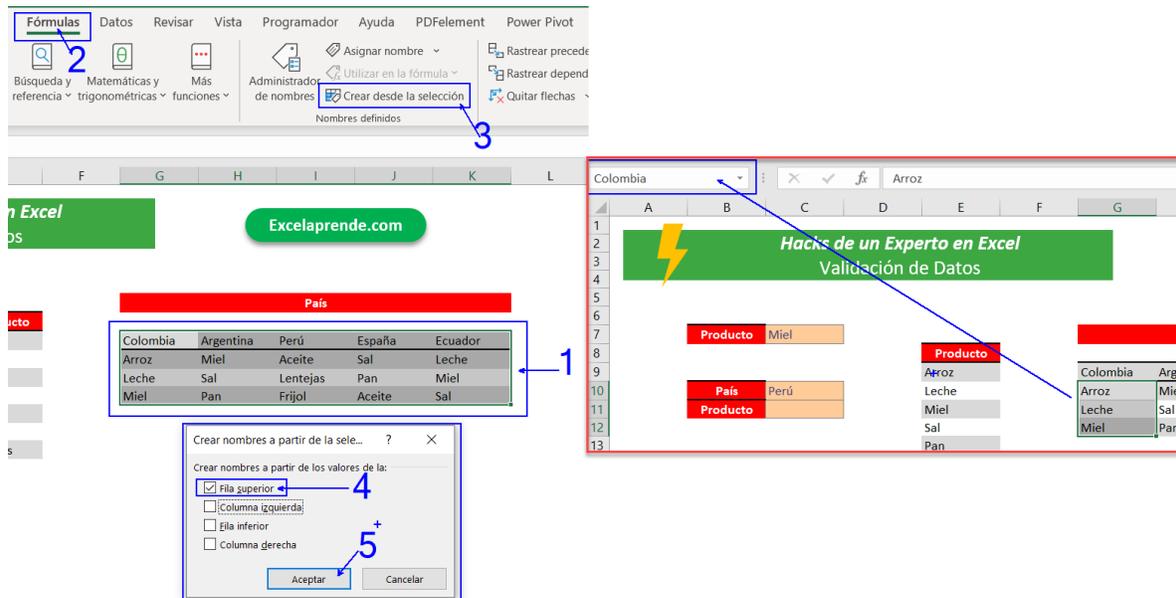


Figura 1.13 Nombrar rangos de celdas

Con este proceso hemos puesto los nombres a cada rango con base en los encabezados, como puede ver ahora, si seleccionamos desde el cuadro de nombres Colombia se han seleccionado los productos de este país.

El siguiente paso es realizar la validación para el campo producto.

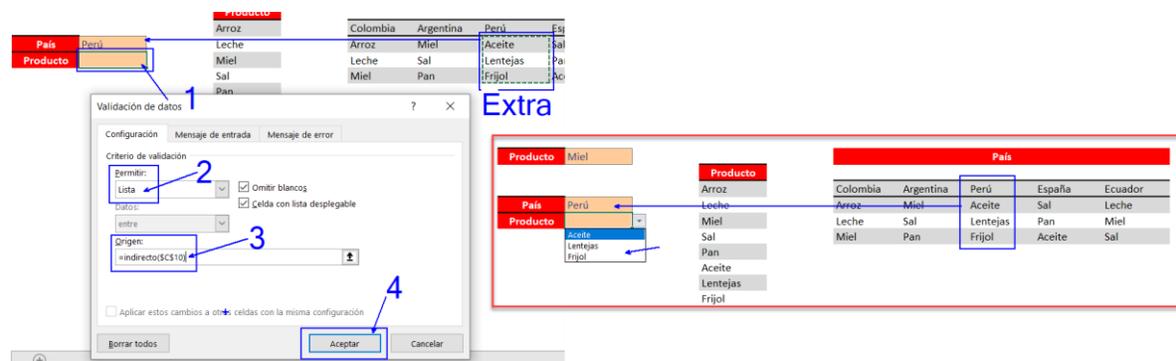


Figura 1.14 Función indirecto para adoptar nombre



FUNCIÓN: =INDIRECTO(\$C\$10)

Al usar la función indirecto lograremos que tome el nombre del país y busque el rango con el mismo nombre y de esta forma muestre su contenido.

1.4 Convertir números como texto sin que se te cierre el Excel (No responde)

Existen trucos que pueden salvar tu día y este es uno de ellos, como lograr convertir una serie de números almacenados como texto en números, esto es un proceso que puedes realizar manualmente y que en la mayoría de los casos bloquea Excel.

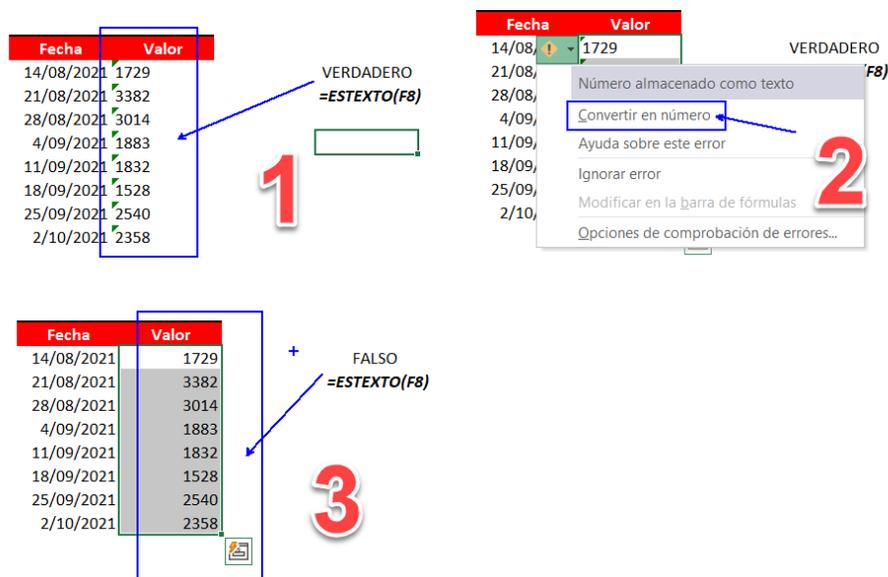


Figura 1.15 Convertir números desde la selección

Cuando tenemos números almacenados como texto podemos observar que están alineados de forma general a la izquierda y si lo validamos con la función ESTEXTO podemos ver que dará como resultado VERDADERO, de esta manera si son pocos registros podemos seleccionarlos y usar desde el menú contextual la opción de convertir a números, esto en cambio cuando son muchos registros puede generar que se cierre el Excel o deje de funcionar, ocasionando probablemente pérdida de los datos o formulas aplicadas en lo que estés trabajando.

Por lo que vamos a enseñarte el truco para que independientemente de la cantidad de registros puedas solucionar esto sin tener el riesgo de un **“Excel no responde...”**

Ingresamos en una celda que tenga formato general el número 1 → Seleccionamos el rango de datos que deseamos convertir a números → Vamos a seleccionar la opción Pegar en la pestaña Inicio, seguido seleccionamos Pegado especial (Podemos usar el atajo Control + Alternativo + V) → En el cuadro de dialogo Pegado especial seleccionamos Valores en Pegar y Multiplicar en Operación → Clic en Aceptar.

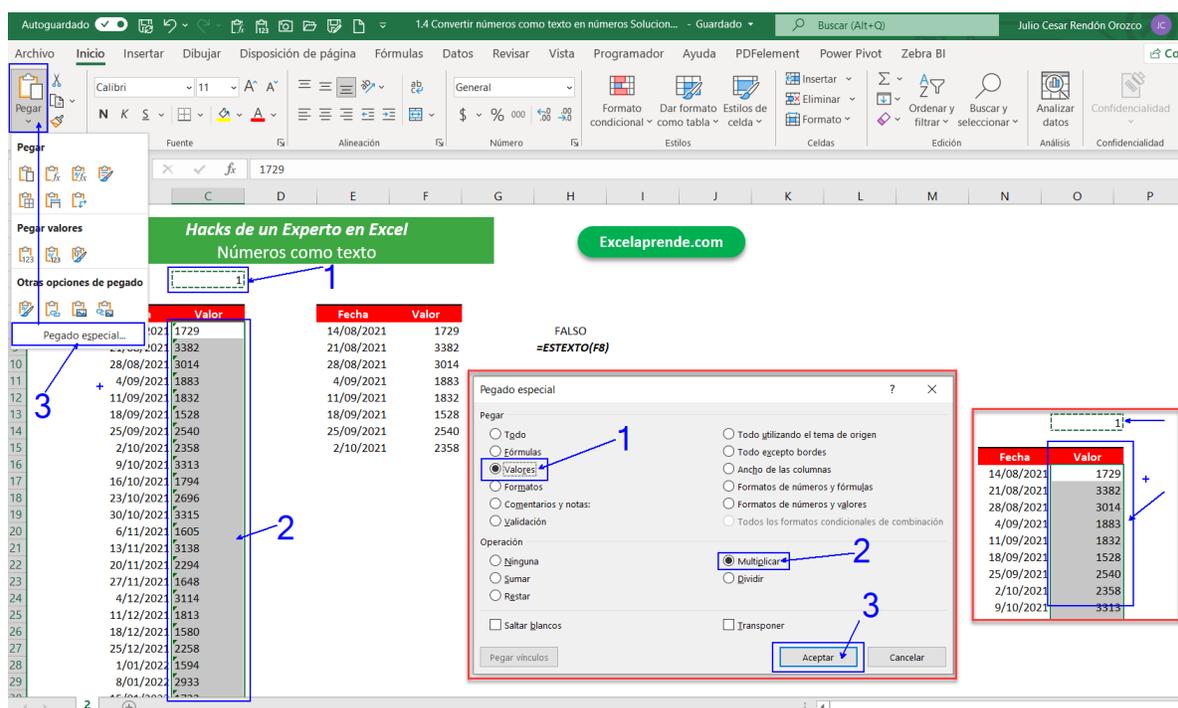


Figura 1.16 Convertir a números usando pegado especial

Vemos como ahora podemos tener todos nuestros números que estaban almacenados como texto en números como valor, lo que permitirá que usted pueda tener un resultado rápido y efectivo sin que su Excel tenga que cerrarse y dejar de funcionar.

1.5 Convertir valores a fechas sin formulas

Al momento de descargar información de un software frecuentemente nos encontramos con información almacenada con formatos especiales, por lo que debemos convertirlas en el formato adecuado. En este caso encontramos una fecha almacenada en el formato DDMMAAAA y se requiere este en el formato DD/MM/AAAA esto lo podríamos tratar con fórmulas y funciones, pero en esta oportunidad te mostraremos el gran poder de la opción texto en columnas, si tal como leíste con la opción de texto en columnas, por eso este libro te está ayudando a descubrir aquellos hacks que deberías conocer de Excel.

Fecha	Valor	Fecha 2
14/08/2021	1729	03022021
21/08/2021	3382	10022021
28/08/2021	3014	17022021
4/09/2021	1883	24022021
11/09/2021	1832	03032021
18/09/2021	1528	10032021

Figura 1.17 Comparar tipos de datos

Vamos a seleccionar los datos o el campo donde se encuentran los valores a convertir → Seguido de esto seleccionamos la pestaña Datos → Comando Texto en columnas → Se nos mostrará un nuevo cuadro de dialogo, en este será donde realizaremos la aplicación de nuevo formato.

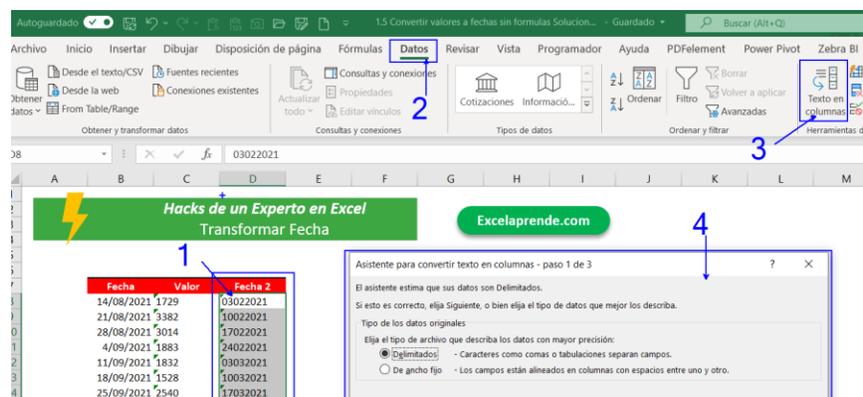


Figura 1.18 Abrir asistente Texto en columnas

Vamos a aplicar los 3 pasos de este cuadro de dialogo, estos de forma sencilla y con un resultado extraordinario.

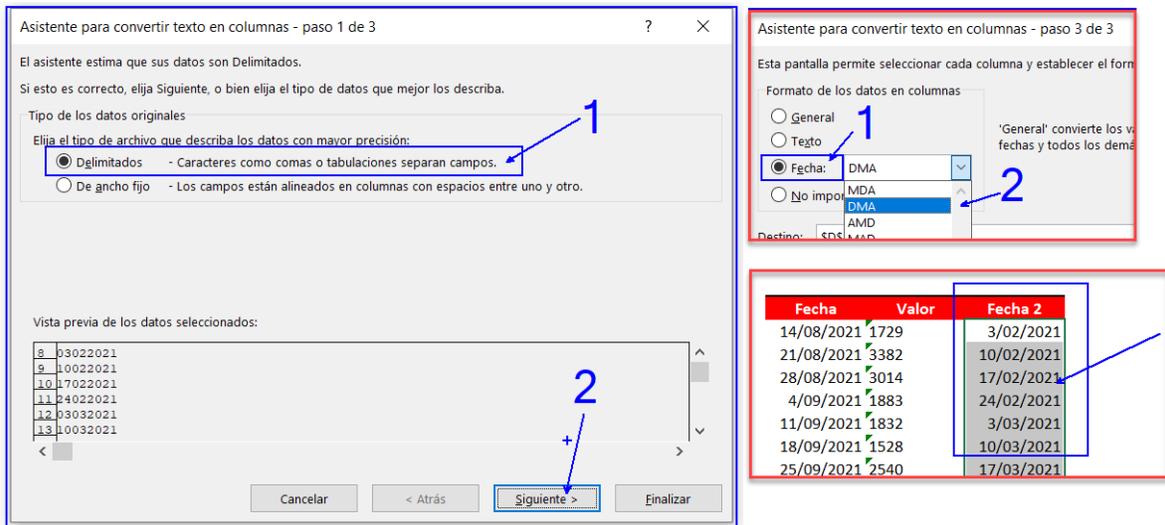


Figura 1.19 Aplicar formato de fecha a valores

Paso 1 de 3, seleccionamos la opción delimitados y presionamos clic en Siguiente.

Paso 2 de 3, nos aseguramos de que está marcada la opción solo de tabular y presionamos clic en siguiente.

Paso 3 de 3, Seleccionamos la opción Fecha y dentro de las opciones de la lista buscamos la que se adapte al resultado que queremos tener, en este caso DMA.

Presionamos clic en Aceptar y el resultado que vemos es precisamente lo que estábamos buscando, las fechas ya las podemos ver de forma práctica y rápida.

1.6 Comentarios y notas en office 365

La suite office ha estado evolucionando de tal forma que los usuarios tengamos mejores herramientas para realizar trabajo colaborativo, por eso el truco en esta oportunidad tiene que ver con los nuevos comentarios que nos ofrece Excel y la interacción con nuestros compañeros de trabajo desde la hoja de cálculo y que ellos puedan recibir notificaciones directamente a su bandeja de correo con los detalles para revisar las novedades o simplemente responder lo requerido.

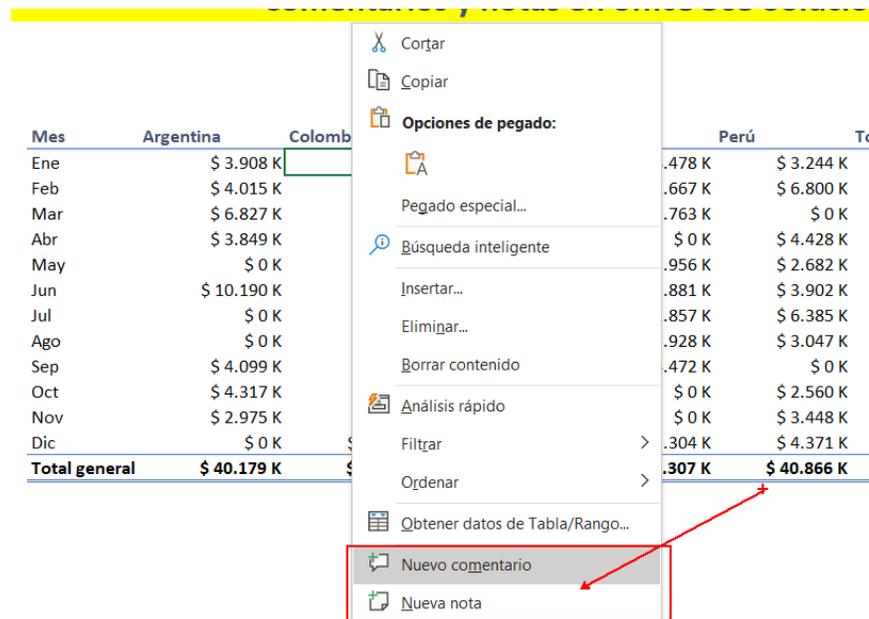


Figura 1.20 Aplicar formato de fecha a valores

Al presionar clic derecho sobre una de las celdas, vemos como ya no solo se nos ofrece una opción para dejar un comentario, contrario a esto tenemos la opción de Nueva Nota.

The image shows a screenshot of an Excel spreadsheet with a new note added to a cell. The spreadsheet has columns for 'Colombia', 'Ecuador', and 'España'. The rows represent values in thousands of dollars. A yellow note box is overlaid on the cell containing '\$ 0 K' under the 'Colombia' column. The note text reads 'Julio C. Rendón' and 'CLASICA ;)'. A red arrow points to the note box.

	Colombia	Ecuador	España
K	\$ 0 K		\$ 8.
K	\$ 3.582 K		\$ 2.
K	\$ 3.595 K		\$ 5.
K	\$ 7.242 K		
K	\$ 2.515 K	\$ 12.234 K	\$ 3.

Figura 1.21 Nueva Nota

El truco es usar la opción de Nuevo comentario en el cual podremos explotar la gran utilidad de conectividad que nos brinda la suite office 365.

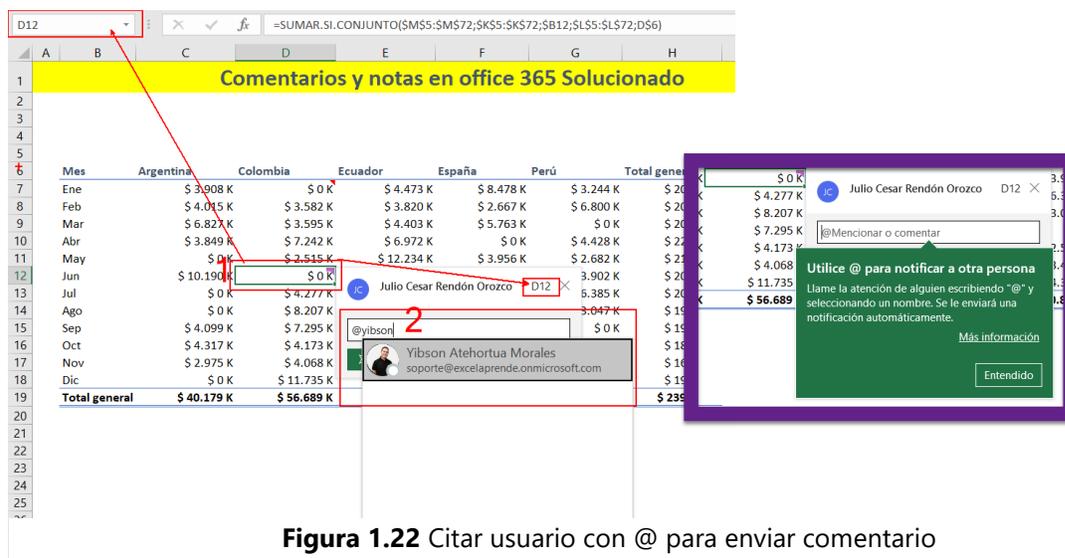


Figura 1.22 Citar usuario con @ para enviar comentario

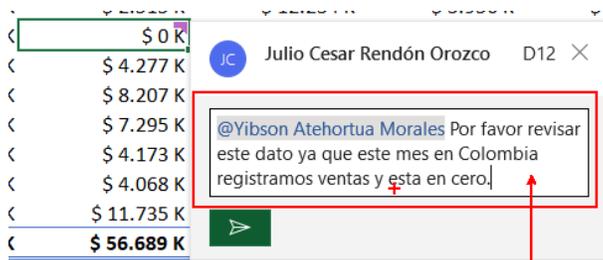


Figura 1.23 Comentario citando usuario

Debemos tener presente que, si el comentario se está realizando desde un archivo que no está compartido con el usuario citado deberás realizarlo, lo bueno es que esta notificación te saldrá de inmediato si ese es el caso.

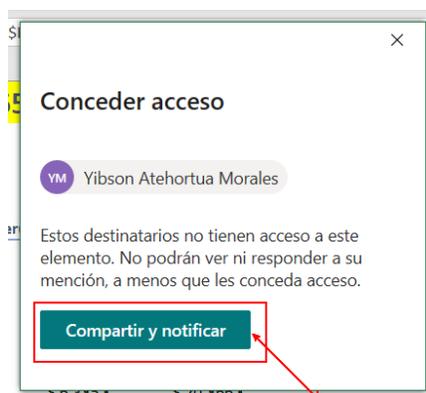


Figura 1.24 Compartir y notificar archivo

El usuario es notificado vía correo y podrás solo presionar clic en Ir al comentario y responderá.

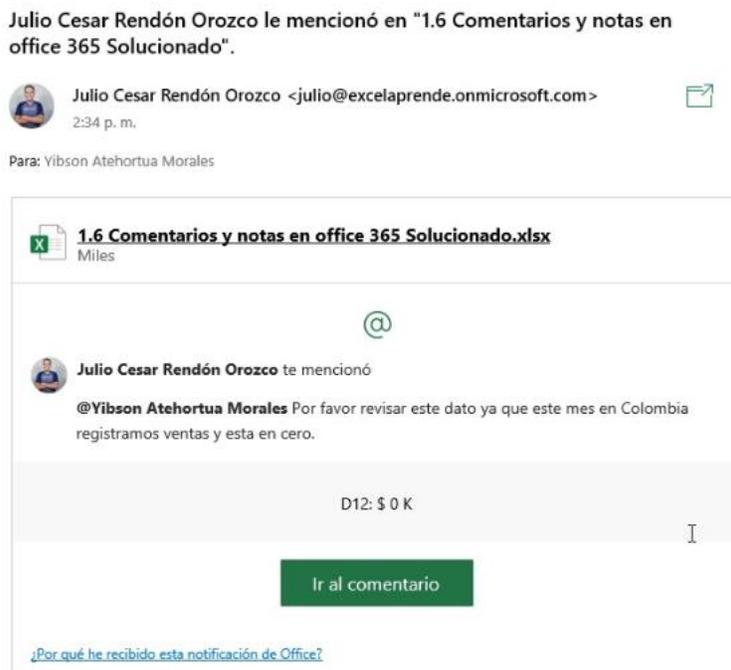


Figura 1.25 Notificación correo comentario en Excel

Podemos ver la respuesta sin tener que cerrar el archivo y se abrirá un hilo de conversación, el cual podrá ser eliminado o resolver.

Mes	Argentina	Colombia	Ecuador	España	Perú	Total general
Ene	\$ 3.908 K	\$ 0 K	\$ 4.473 K	\$ 8.478 K	\$ 3.244 K	\$ 20.104 K
Feb	\$ 4.015 K	\$ 3.582 K	\$ 3.820 K	\$ 2.667 K	\$ 6.800 K	\$ 20.884 K
Mar	\$ 6.827 K	\$ 3.595 K	\$ 4.403 K	\$ 5.763 K	\$ 0 K	\$ 20.587 K
Abr	\$ 3.849 K	\$ 7.242 K	\$ 6.972 K	\$ 0 K	\$ 4.428 K	\$ 22.490 K
May	\$ 0 K	\$ 2.515 K	\$ 12.234 K	\$ 3.956 K	\$ 2.682 K	\$ 21.388 K
Jun	\$ 10.190 K	\$ 0 K				\$ 20.633 K
Jul	\$ 0 K	\$ 4.277 K				\$ 20.866 K
Ago	\$ 0 K	\$ 8.207 K				\$ 19.417 K
Sep	\$ 4.099 K	\$ 7.295 K				\$ 19.117 K
Oct	\$ 4.317 K	\$ 4.173 K				\$ 18.019 K
Nov	\$ 2.975 K	\$ 4.068 K				\$ 16.625 K
Dic	\$ 0 K	\$ 11.735 K				\$ 19.409 K
Total general	\$ 40.179 K	\$ 56.689 K				\$ 239.539 K

Julio Cesar Rendón Orozco D12 ... 3.902 K

@Yibson Atehortua Morales Por favor revisar este dato ya que este mes en Colombia registramos ventas y esta en cero. 6.385 K

15/08/2021 2:33 p. m. 3.047 K

2.560 K

3.448 K

Editar 4.371 K

Yibson Atehortua Morales 19.409 K

ok entendido 19.417 K

15/08/2021 2:36 p. m. 19.117 K

@mención o respuesta

Figura 1.26 Respuesta comentario en Excel

1.7 Poner Excel modo presentación manual

Cuando desarrollas un informe en Excel y quieres presentarlo desde el mismo y no pasar todo tu trabajo a presentaciones en PowerPoint, puedes usar el siguiente truco para convertir a Excel en modo presentación, sin que apenas su audiencia note que está usando Excel.

Tomemos cualquiera de los archivos que hemos trabajado hasta el momento como referencia.

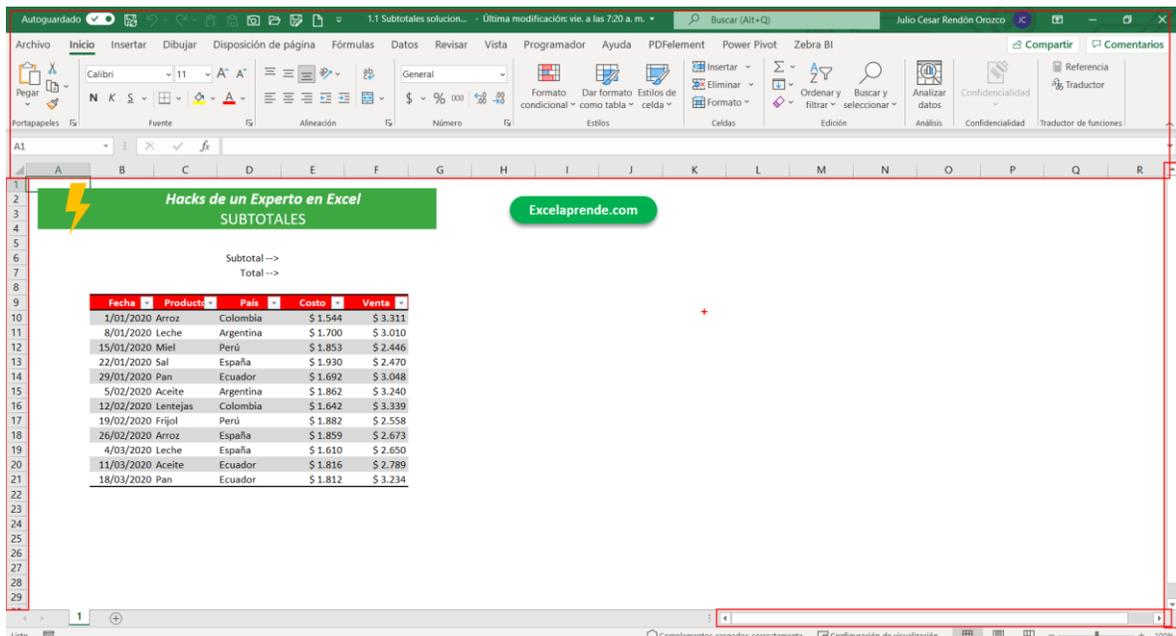


Figura 1.27 Referencia de informe

Vamos a ocultar todo lo que está en los recuadros rojos, para ello vamos a la cinta de opciones Archivo y presionamos clic en opciones, lo que permitirá que abramos el cuadro de dialogo Opciones de Excel.

Vamos a buscar el grupo Avanzadas → Desmarcamos las opciones de (Mostrar barra de desplazamiento horizontal, Mostrar barra de desplazamiento vertical, Mostrar pestañas de hojas) → Clic en Aceptar.

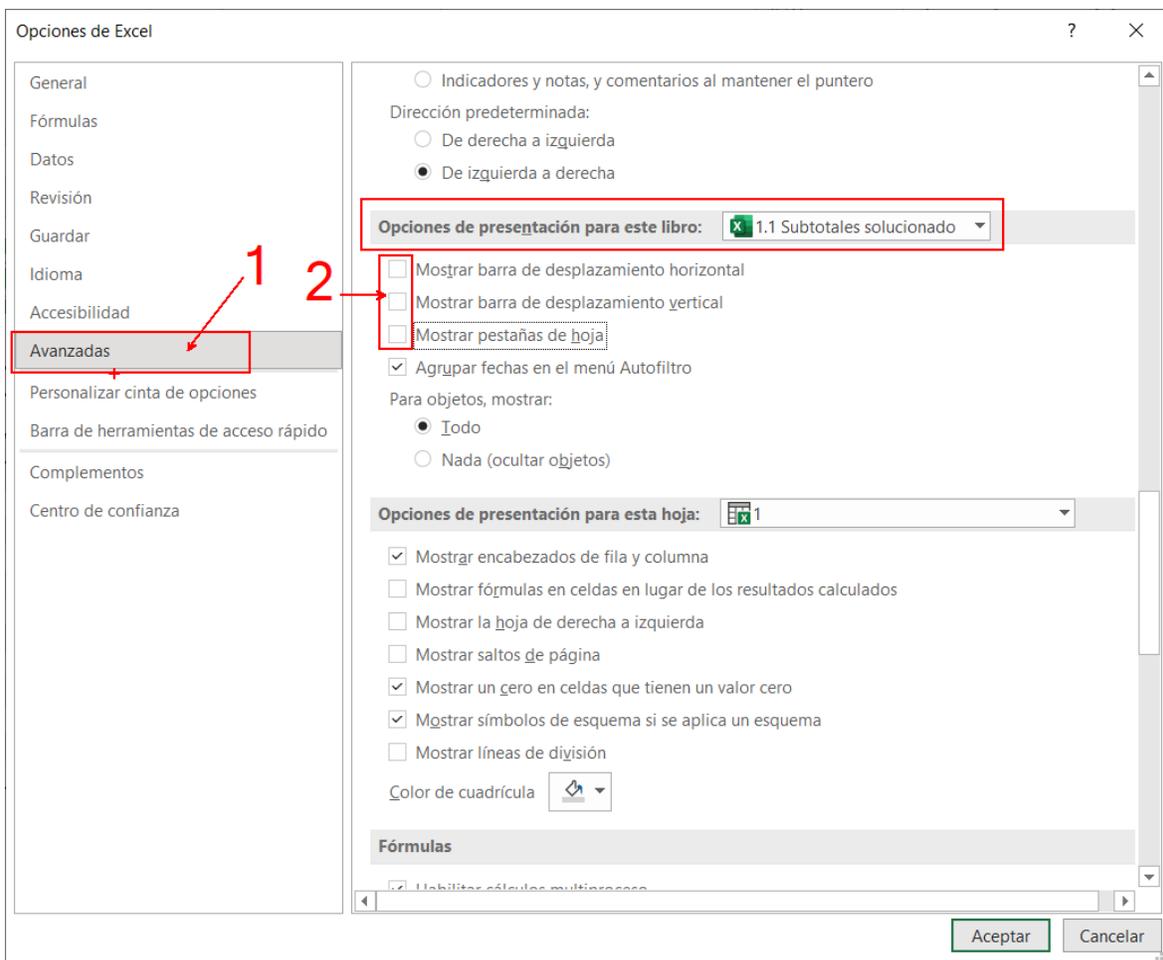


Figura 1.28 Opciones de Excel

Vamos a la pestaña Vista → Desmarcamos Líneas de cuadrícula, Barra de fórmulas y Encabezados.



Figura 1.29 Desactivar opciones en pestaña vista

Vamos a las opciones de presentación de la cinta de opciones → seleccionamos Ocultar automáticamente la cinta de opciones → a continuación nuestro Excel estará en modo presentación.

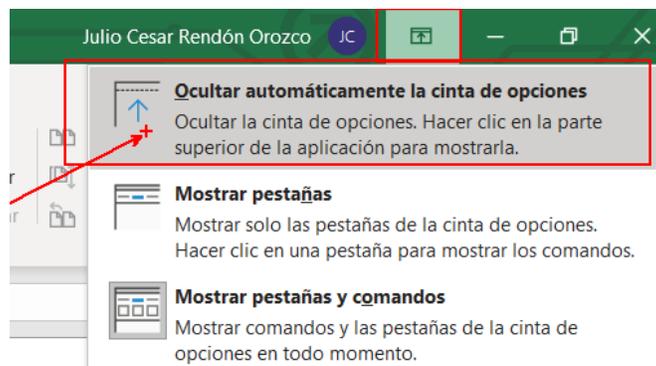


Figura 1.30 Ocultar cinta de opciones

Ahora tendremos una vista de nuestro informe limpia y que permitirá que los clientes del informe puedan concentrarse en los datos.

The image shows a screenshot of an Excel spreadsheet in presentation mode. The spreadsheet has a green header bar with a lightning bolt icon and the text 'Hacks de un Experto en Excel' and 'SUBTOTALES'. A button labeled 'Excelaprende.com' is visible in the top right. Below the header, there are two rows of summary data: 'Subtotal -->' and 'Total -->'. The main data is a table with the following columns: Fecha, Producto, País, Costo, and Venta. The table contains 14 rows of data.

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446
22/01/2020	Sal	España	\$ 1.930	\$ 2.470
29/01/2020	Pan	Ecuador	\$ 1.692	\$ 3.048
5/02/2020	Aceite	Argentina	\$ 1.862	\$ 3.240
12/02/2020	Lentejas	Colombia	\$ 1.642	\$ 3.339
19/02/2020	Frijol	Perú	\$ 1.882	\$ 2.558
26/02/2020	Arroz	España	\$ 1.859	\$ 2.673
4/03/2020	Leche	España	\$ 1.610	\$ 2.650
11/03/2020	Aceite	Ecuador	\$ 1.816	\$ 2.789
18/03/2020	Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 1.31 Excel modo presentación



NOTA: Para ver nuestro Excel normal, nuevamente vamos a deshacer todos los pasos aplicados 😊

1.8 Navegar por Excel como un experto

Al momento de navegar por archivos de Excel con muchas hojas suele ser tedioso el hecho de moverse e intentar ver la hoja en la que deseamos ubicarnos, por eso una de las mejores prácticas para hacerlo es aprovechar el recurso oculto de Excel que nos muestra un listado de las hojas activas de nuestro archivo.



Figura 1.32 Hojas en Excel

Solo basta con presionar clic derecho sobre el navegador de hojas y se nos mostrara un listado de las hojas activas, clic en la hoja que quieras ir y en Aceptar, de esta forma rápida podrás ir sin tener que presionar mil clics 😊

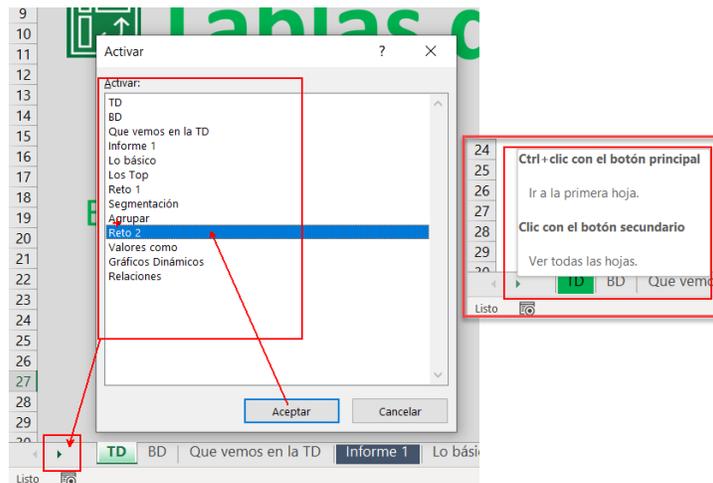


Figura 1.33 Hojas activas

1.9 Agregar complementos de Excel (Tienda)

Vamos a usar los complementos de Excel, si es un término que muchos ni siquiera conocen y menos que pueden agregar utilidades increíbles con pocos clics a su amada hoja de cálculo Excel.

En esta oportunidad buscamos agregar un complemento que facilite crear un organigrama y que sea dinámico.

Para agregar un complemento vamos a la pestaña Insertar → Grupo complementos → Mis complementos → pestaña Tienda.

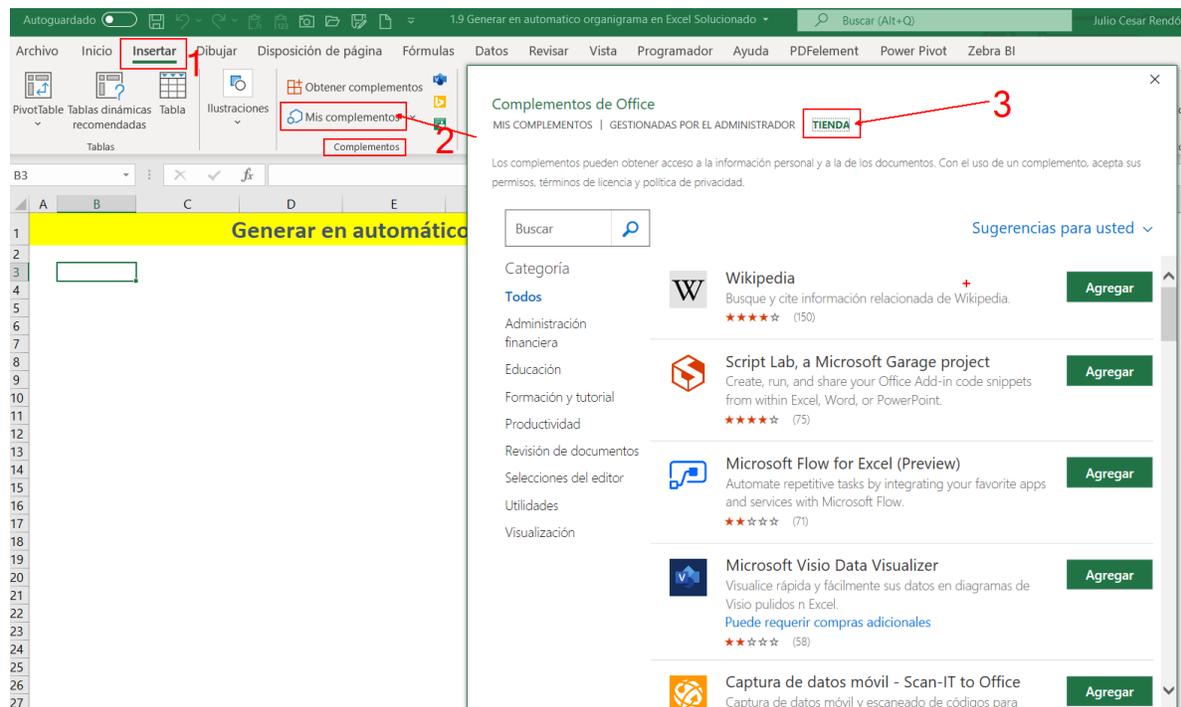


Figura 1.34 Agregar complemento

Aquí podremos buscar tanto complementos gratis como pagados, en esta oportunidad usaremos uno gratis y de gran utilidad.

Ingresamos la palabra ORGANIGRAMA → clic en buscar → clic en Agregar.

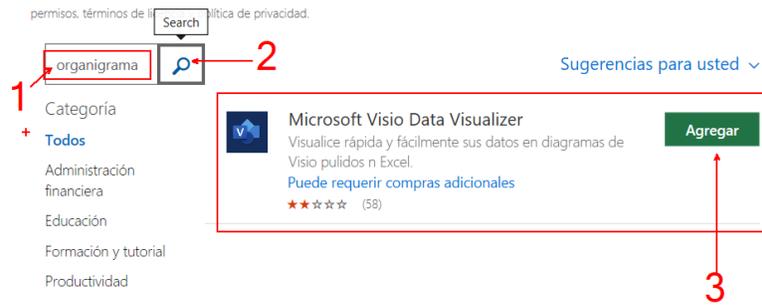


Figura 1.35 Buscar complementos de Excel

Se nos muestra por primera vez al momento de agregar este complemento un cuadro de dialogo con los términos y condiciones, solo presionamos clic en Continuar.

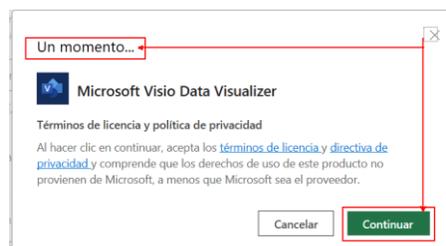


Figura 1.36 Aceptar términos y condiciones de complemento

Se nos muestra el cuadro de dialogo Data Visualizer, en este vamos a seleccionar Organigrama y luego escoger el de nuestra preferencia, para este caso seleccionamos vertical.

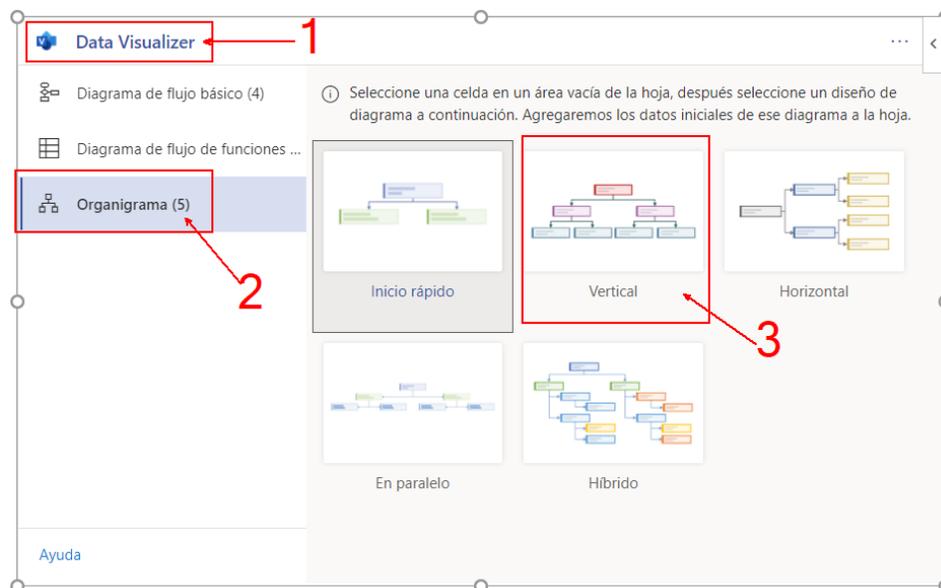


Figura 1.37 Seleccionar tipo de organigrama

En automático y sin que tengas que realizar ningún paso adicional se ha insertado un organigrama, el cual puedes iniciar a modificar con los datos de tu empresa o la de tu cliente.

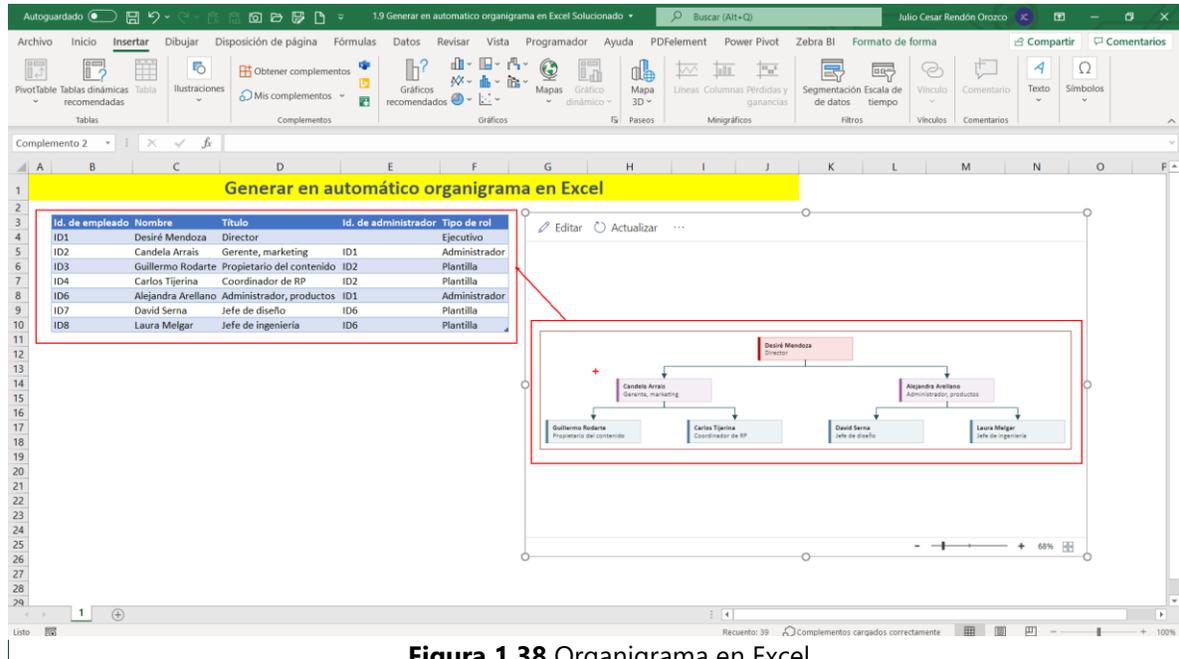


Figura 1.38 Organigrama en Excel



ADVERTENCIA: Los nombres de los encabezados de la tabla no se pueden modificar.

Al realizar cambios solo debes seleccionar el organigrama y presionar clic sobre Actualizar.



Figura 1.39 Organigrama en Excel

1.10 Recuperar archivos no guardados

Hace un par de días mi esposa me llamo asustada que había dedicado toda la mañana hacer un trabajo y que no había guardado el archivo de Excel, la primera pregunta mía fue si ya había apagado el PC, resulta fácil recuperar un archivo si no lo guardaste por primera vez o si solo abriste un archivo y realizaste un cálculo de una comisión o parte de un informe lo cierras y luego necesitas algo de este sin haberlo guardado.

Bueno ahora aprenderemos este truco para recuperar este tipo de archivos.

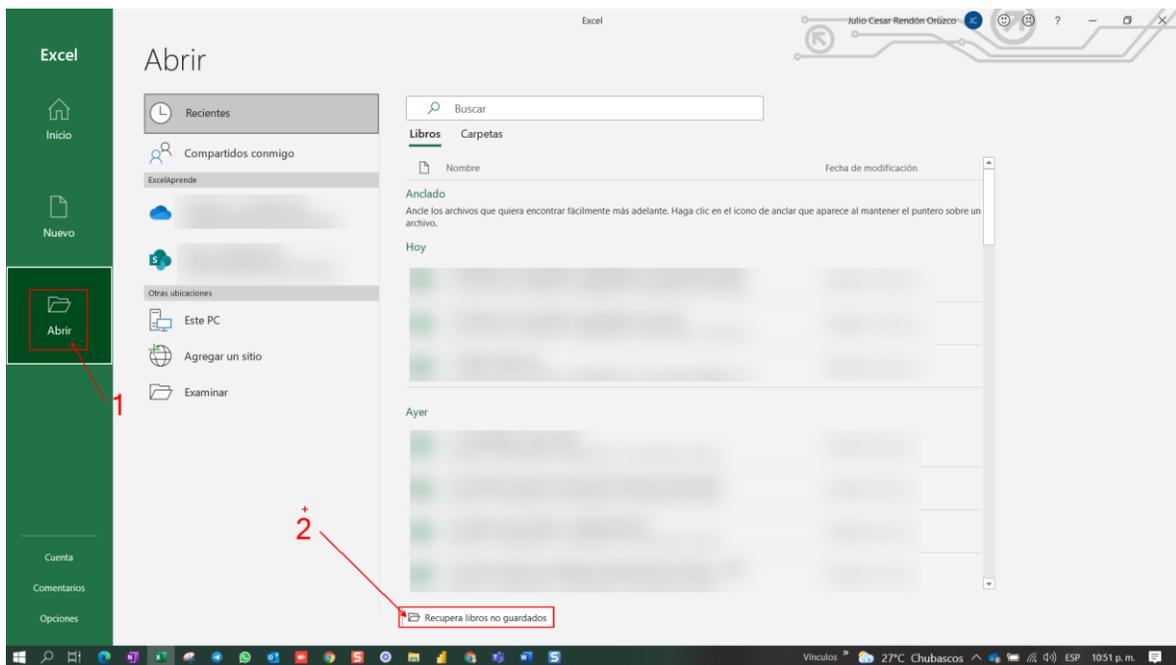


Figura 1.40 Recuperar archivos no guardados

Solo basta con abrir nuevamente Excel → vamos y presionamos clic en Abrir → en la parte inferior vamos a notar una opción que pocos detallan Recuperar libros no guardados → clic en esta opción.

Capítulo 2: Fórmulas y Funciones

Cuando se usa Excel es habitual confundirse con la diferencia entre una fórmula y una función, ambos recursos son para realizar operaciones matemáticas y por este motivo es fácil pensar que son lo mismo, pero en realidad tienen una diferencia. Las fórmulas son cálculos matemáticos que se realizan a partir de valores o datos introducidos en una celda, pueden crearse con valores directos o con referencias a otras celdas y son diseñadas, creadas y ejecutadas por el usuario.

Por su parte las funciones en Excel son fórmulas predefinidas por Microsoft en la herramienta, se ejecutan utilizando valores específicos que se conocen como argumentos. Se trata de valores que están estructurados y siguen una secuencia específica como si fuera una macro de Excel para obtener un resultado.



NOTA: Este es un ejemplo diferencial entre una fórmula y una función en Microsoft Excel.

Fórmula: =A1+B1

Función: =SUMA (A1:B1)

Tener conocimiento de fórmulas y funciones en Excel te permite dominar la mejor hoja de cálculo del mercado, Microsoft Excel. Pero quedarte en un nivel básico no tiene nada que ver con ser un experto en la herramienta y dominar los mejores y exclusivos trucos que podrás ver y desarrollar a continuación, trucos o hacks que te permitirán sacarle el máximo provecho a la herramienta ayudándote a realizar un trabajo óptimo y ahorrar mucho más tiempo.

2.1 Dos o más criterios de búsqueda en BUSCARV

La función **BUSCARV** es una función de búsqueda y referencia. Se usa para realizar la búsqueda vertical de un valor en la columna del extremo izquierdo de una tabla o selección y devolver el valor en la misma fila.

Su estructura o argumentos son los siguientes:



FUNCIÓN: BUSCARV (valor_buscado; matriz_tabla; indicador_columnas; [rango])

A continuación, se desarrollará un caso común del uso de la función BUSCARV.

Se cuenta con una base de datos con el producto, costo y venta. Y en un informe aparte se requiere que al desplegar los productos y seleccionar uno de ellos, por medio de la función BUSCARV encuentre y traiga el costo y venta de ese producto seleccionado.

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.859	\$ 2.673
Aceite	\$ 1.610	\$ 2.650
Lentejas	\$ 1.816	\$ 2.789
Frijol	\$ 1.812	\$ 3.234

Figura 2.1 Ejercicio BUSCARV.

Se selecciona el producto de la lista desplegable y se parametriza la función en la celda donde se solicita el costo (C7).

Producto	Costo	Venta
Leche	=BUSCARV(\$B7;\$B\$9:\$D\$17;2;0)	

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.859	\$ 2.673
Aceite	\$ 1.610	\$ 2.650
Lentejas	\$ 1.816	\$ 2.789
Frijol	\$ 1.812	\$ 3.234

Figura 2.2 Desarrollo función BUSCARV.

Esta misma función se arrastra para el campo Venta (D7) y se cambia el argumento indicador de columna.

Producto	Costo	Venta
Leche	1700	=BUSCARV(\$B7;\$B\$9:\$D\$17;3;0)

Figura 2.3 Cambio de indicador de columna.

Como resultado se obtiene el costo y venta del producto que se seleccione en la lista desplegable.

Producto	Costo	Venta
Leche	1700	3010

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.859	\$ 2.673
Aceite	\$ 1.610	\$ 2.650
Lentejas	\$ 1.816	\$ 2.789
Frijol	\$ 1.812	\$ 3.234

Figura 2.4 Resultado obtenido.

El ejercicio anterior muestra la funcionalidad del BUSCARV y los resultados que podemos obtener con la correcta utilización de sus argumentos, pero el reto viene a continuación cuando se nos pide que debemos buscar información para dos o más criterios de búsqueda (valor_buscado).

Producto	País	Costo	Venta
Arroz	Colombia	\$ 1.544	\$ 3.311
Leche	Argentina	\$ 1.700	\$ 3.010
Miel	Perú	\$ 1.853	\$ 2.446
Sal	España	\$ 1.930	\$ 2.470
Pan	Ecuador	\$ 1.692	\$ 3.048
Aceite	Argentina	\$ 1.862	\$ 3.240
Lentejas	Colombia	\$ 1.642	\$ 3.339
Frijol	Perú	\$ 1.882	\$ 2.558
Arroz	España	\$ 1.859	\$ 2.673
Leche	España	\$ 1.610	\$ 2.650
Aceite	Ecuador	\$ 1.816	\$ 2.789
Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 2.5 Planteamiento del problema.

Se puede evidenciar en la tabla anterior, como el producto Arroz se repite para el país Colombia y España, y lo mismo ocurre para los demás productos de la tabla, por lo que se hace necesario tener en cuenta un segundo criterio de búsqueda, teniendo en cuenta que al utilizar el BUSCARV de la forma tradicional solo nos retornaría el primer valor encontrado.

Es ahí donde viene el primer truco de este capítulo:



HACKS: Con la misma función **BUSCARV** y la concatenación o unión de los criterios de búsqueda, se puede dar solución a este problema.

Veamos el paso a paso de este truco:

1. Lo primero que debe hacerse es agregar una columna en la matriz o tabla donde se tiene la información a buscar y concatenar allí con el operador ampersand (&) los

criterios o valores buscados y arrastrar, esto con la intención de crear una columna de valores únicos.

	Producto	País	Costo	Venta
=G10&H10	Arroz	Colombia	\$ 1.544	\$ 3.311
	Leche	Argentina	\$ 1.700	\$ 3.010
	Miel	Perú	\$ 1.853	\$ 2.446
	Sal	España	\$ 1.930	\$ 2.470
	Pan	Ecuador	\$ 1.692	\$ 3.048
	Aceite	Argentina	\$ 1.862	\$ 3.240
	Lentejas	Colombia	\$ 1.642	\$ 3.339
	Frijol	Perú	\$ 1.882	\$ 2.558
	Arroz	España	\$ 1.859	\$ 2.673
	Leche	España	\$ 1.610	\$ 2.650
	Aceite	Ecuador	\$ 1.816	\$ 2.789
	Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 2.6 Crear columna con valores concatenados.

- El paso siguiente es programar la función BUSCARV en la celda de retorno, pero en esta ocasión en el argumento “valor_buscado” se debe unir con el operador AMPERSAND (&) los dos criterios a buscar:

Producto	País	Costo	Venta
Leche	Argentina	=BUSCARV(G7&H7;	

Figura 2.7 Unir criterios.



FUNCIÓN: =BUSCARV(G7&H7;\$F\$9:\$J\$21;4;0)



ADVERTENCIA: Se debe tener presente que la columna de la matriz o tabla donde se unieron los criterios debe seleccionarse como la primera columna del rango en el argumento “matriz_tabla”.

Por último, esta columna que se creó en la matriz o tabla se puede ocultar si se considera que estéticamente afecta la presentación de la información.

Producto	País	Costo	Venta
Leche	Argentina	1700	3010

Producto	País	Costo	Venta
Arroz	Colombia	\$ 1.544	\$ 3.311
Leche	Argentina	\$ 1.700	\$ 3.010
Miel	Perú	\$ 1.853	\$ 2.446
Sal	España	\$ 1.930	\$ 2.470
Pan	Ecuador	\$ 1.692	\$ 3.048
Aceite	Argentina	\$ 1.862	\$ 3.240
Lentejas	Colombia	\$ 1.642	\$ 3.339
Frijol	Perú	\$ 1.882	\$ 2.558
Arroz	España	\$ 1.859	\$ 2.673
Leche	España	\$ 1.610	\$ 2.650
Aceite	Ecuador	\$ 1.816	\$ 2.789
Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 2.8 Resultado final.

2.2 Campo dinámico en BDSUMA

La función BDSUMA se usa para obtener la sumatoria de un rango de celdas que cumplen con uno o varios criterios, esta función pertenece al grupo de funciones de base datos y en un principio puede tener similitud al SUMAR.SI.CONJUNTO, pero al utilizar el truco que veremos a continuación tendrá una ventaja muy marcada.

La estructura de BDSUMA es la siguiente:



FUNCIÓN: = BDSUMA (base_de_datos; nombre_de_campo; criterios)

Fx --> =+BDSUMA(B13:F25;E13;B9:D10)

Fecha	Producto	País
1/01/2020	Arroz	Colombia

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446
22/01/2020	Sal	España	\$ 1.930	\$ 2.470
29/01/2020	Pan	Ecuador	\$ 1.692	\$ 3.048
5/02/2020	Aceite	Argentina	\$ 1.862	\$ 3.240
12/02/2020	Lentejas	Colombia	\$ 1.642	\$ 3.339
19/02/2020	Frijol	Perú	\$ 1.882	\$ 2.558
26/02/2020	Arroz	España	\$ 1.859	\$ 2.673
4/03/2020	Leche	España	\$ 1.610	\$ 2.650
11/03/2020	Aceite	Ecuador	\$ 1.816	\$ 2.789
18/03/2020	Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 2.9 Ejercicio BDSUMA.

Como podemos evidenciar en el ejercicio anterior, el segundo argumento de la función BDSUMA es “nombre_campo” y este argumento solo permite ingresar uno solo, pero en nuestra tabla tenemos dos (costo y venta), y si quisiéramos evaluar Costo o Venta, tendríamos que crear la función dos veces. Pero con este truco veremos cómo se soluciona este problema y dejar el argumento “nombre_campo” de manera dinámica.



HACKS: Crear argumento “nombre_campo” dinámico para la función BDSUMA.

1. Inicialmente se debe crear una lista desplegable con los nombres de los campos que se quieren sumar:

Fecha	Producto	País	Costo	Venta
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446

Figura 2.10 Crear lista desplegable.

2. Modificar el segundo argumento de la función BDSUMA “nombre_campo” y dejarlo haciendo referencia a la celda donde se creó la lista desplegable.

Resultado --> `=BDSUMA(B13:F25;F9;B9:D10)`
 Fx --> `=BDSUMA(base_de_datos; nombre_de_campo; criterios)`

Fecha	Producto	País
1/01/2020	Arroz	Colombia

Figura 2.11 Cambiar el argumento “nombre_campo”.

- Al realizar este cambio el usuario podrá seleccionar la columna que se quiere sumar con la función BDSUMA.



NOTA: Este truco de asignar un campo dinámico para determinar la columna a sumar, no es posible programarlo en la función SUMAR.SI.CONJUNTO.

Al final solo con desplegar nuestra lista y seleccionar la columna a sumar, podremos obtener el resultado sin necesidad de crear dos funciones para la misma tabla.



Figura 2.12 Resultado final.

2.3 El poder del SUMAR.SI.CONJUNTO

La función **SUMAR.SI.CONJUNTO** nos permite sumar los valores de un rango de celdas que cumplan con varias condiciones que estipulemos. A diferencia de la función **SUMAR.SI**, que solo nos permite sumar los valores de las celdas que cumplen una condición concreta, con la función **SUMAR.SI.CONJUNTO** podremos definir hasta 127 criterios.

La estructura o argumentos de la función son:



FUNCIÓN: = **SUMAR.SI.CONJUNTO** (rango_suma; rango_criterios1; criterio1; rango_criterios2; criterios2...)

A continuación, veremos la diferencia de usar el **SUMAR.SI** y el **SUMAR.SI.CONJUNTO**.

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.859	\$ 2.673
Aceite	\$ 1.610	\$ 2.650
Lentejas	\$ 1.816	\$ 2.789
Frijol	\$ 1.812	\$ 3.234

Figura 2.13 SUMAR.SI

Como podemos ver en la función **SUMAR.SI** solo nos permite ingresar un rango y un criterio y luego el rango a sumar, si quisiéramos evaluar otro criterio debemos optar por la función **SUMAR.SI.CONJUNTO**.

=SUMAR.SI.CONJUNTO(J10:J21;\$G\$10:\$G\$21;\$G\$7;\$H\$10:\$H\$21;\$H\$7)

Producto	País	Costo	Venta
Arroz	Colombia	1544	3311

Producto	País	Costo	Venta
Arroz	Colombia	\$ 1.544	\$ 3.311
Leche	Argentina	\$ 1.700	\$ 3.010
Miel	Perú	\$ 1.853	\$ 2.446
Sal	España	\$ 1.930	\$ 2.470
Pan	Ecuador	\$ 1.692	\$ 3.048
Aceite	Argentina	\$ 1.862	\$ 3.240
Lentejas	Colombia	\$ 1.642	\$ 3.339
Frijol	Perú	\$ 1.882	\$ 2.558
Arroz	España	\$ 1.859	\$ 2.673
Leche	España	\$ 1.610	\$ 2.650
Aceite	Ecuador	\$ 1.816	\$ 2.789
Pan	Ecuador	\$ 1.812	\$ 3.234

Figura 2.14 SUMAR.SI.CONJUNTO

En esta función podemos observar cómo tiene en cuenta el criterio “Producto” y “País” y realiza la suma; y de esta misma manera podemos evaluar hasta 127 criterios, definitivamente un gran potencial en esta función.



ADVERTENCIA: Los rangos en los argumentos de la función SUMAR.SI.CONJUNTO deben ser iguales, si el rango de los criterios va de la fila A2:A15, este mismo número de filas debe tener el rango de la suma.



NOTA: En la función SUMAR.SI.CONJUNTO cambia el orden de los argumentos, respecto a la función SUMAR.SI, ya que la columna a sumar se debe ingresar como primer argumento.

2.4 La incógnita función SIFECHA

La función SIFECHA es una función de fecha y hora. Se usa para devolver la diferencia entre DOS valores de fecha (fecha de inicio y fecha final), basada en el intervalo (unidad) especificado.

En pocas palabras esta función nos permite saber cuánto tiempo en año, mes o días hay entre dos fechas.



ADVERTENCIA: Esta función se encuentra en estado de reserva por lo que no muestra los argumentos de su estructura cuando se está desarrollando la función y tampoco la encontrarás en el cuadro de ayuda Fx.

Sintaxis o estructura de la función:



FUNCIÓN: = SIFECHA (fecha-inicio, fecha-final, unidad)

Fecha de inicio y fecha final: Son dos fechas entre las cuales usted quiere calcular la diferencia.

Unidad: Es el intervalo en tiempo especificado que puede ser uno de los siguientes:

Y = Año

M = Mes

D = Día

Punto de Partida	Punto final
1/08/2020	1/08/2021

Resultado -->
Fx -->

Figura 2.15 SIFECHA.

Si se quiere cambiar la unidad solo es modificarla en la función y ponerla entre comillas.



HACKS: Calcular rápidamente la edad de una persona con la función **SIFECHA**.

En Excel si se quisiera calcular la edad actual de una persona que nació el 15 de junio de 1989 sin la función SIFECHA, se tendría que hacer las siguientes operaciones:

1

Fecha Nacimiento	Resultado	Fx
15/06/1989	32	=+ENTERO(FRAC.AÑO(B17;HOY()))

Figura 2.16 Opción 1 para calcular edad.

2

Fecha Nacimiento	Resultado	Fx
15/06/1989	32	=(HOY()-B21)/365

Figura 2.17 Opción 2 para calcular edad.



NOTA: Para la segunda opción es necesario cambiar el formato celda a número.

Como se puede evidenciar las fórmulas tienen un grado de complejidad mayor al que puede tener la función SIFECHA ya que con solo 3 argumentos dará el mismo resultado.

Punto de Partida	Punto final
15/06/1989	13/08/2021

Resultado -->	32
Fx -->	=SIFECHA(B8;C8;"Y")

Figura 2.18 Calcular edad con SIFECHA.

2.5 Proteger fórmulas a otro nivel

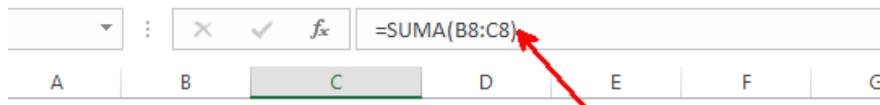
Cuando estamos desarrollando funciones o fórmulas en Microsoft Excel es fundamental pensar en el tema seguridad de la información, ya que en muchas ocasiones es necesario proteger las celdas donde se tienen las fórmulas para evitar que por error se borren o se altere la información, y aún más cuando se tiene la necesidad de ocultar la fórmula o función que se realizó.

Para proteger las fórmulas en Excel también existe una posibilidad con validación de datos, donde se puede restringir la sobre escritura sin necesidad de proteger toda la hoja, pero su debilidad es que si el usuario oprime la tecla DELETE o SUPRIMIR se eliminaría de igual forma la función o fórmula, por lo que solo se protegería la sobre escritura.

En este orden de ideas el siguiente truco es el más recomendado para proteger de manera total nuestras fórmulas y funciones.



HACKS: Protección de fórmulas y funciones nivel experto.



Valor 1	Valor 2
1500	2400

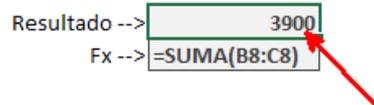


Figura 2.19 Visualización de la función en la barra de fórmulas.

En la imagen anterior se puede evidenciar como al posicionarme en la celda donde se tiene la función o fórmula se puede ver en la barra de fórmulas su composición y si por error escribo algo en esta celda se puede eliminar la función. Para evitar estos problemas vamos a realizar los siguientes pasos:

1. El primer paso que debe realizarse es seleccionar las celdas donde el usuario puede modificar la información, o que también llamamos rangos para el INPUT.

Valor 1	Valor 2
1500	2400

Figura 2.20 Seleccionar celdas modificables.

2. Luego debe darse clic derecho y seleccionar formato celda:

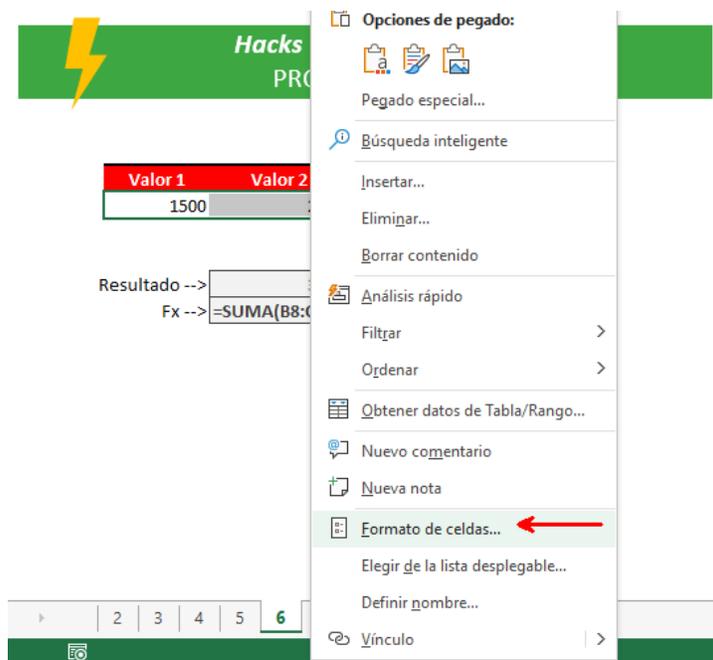


Figura 2.21 Clic derecho, formato celdas.

3. Se debe ir a la pestaña Proteger y posteriormente quitar el check box de la opción Bloqueada y darle aceptar.

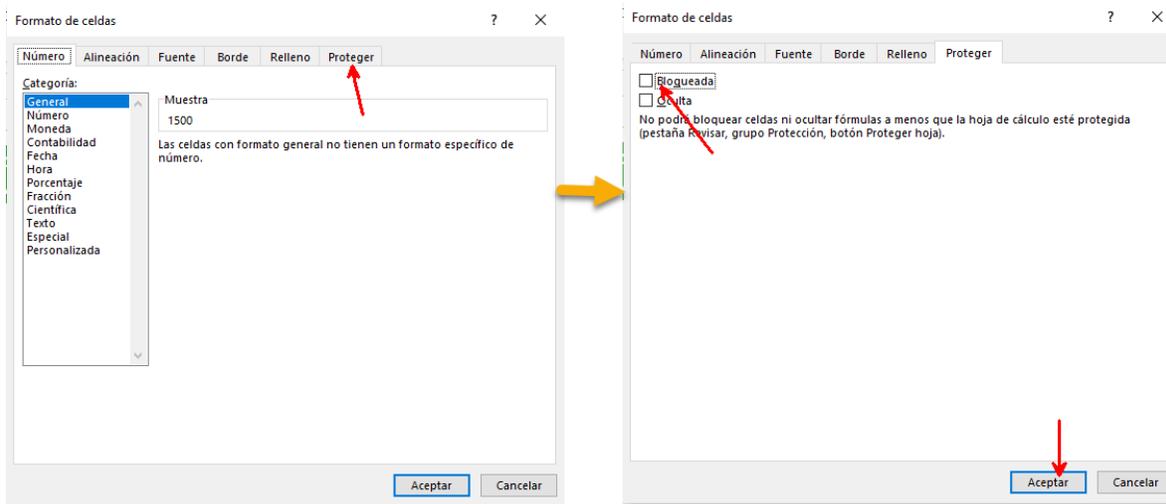


Figura 2.22 Opción Proteger, deshabilitar bloqueada.

4. Luego se debe posicionar en la celda donde se encuentre la fórmula o función y se da clic derecho y en la pestaña Proteger se debe habilitar la opción Oculta y darle aceptar:

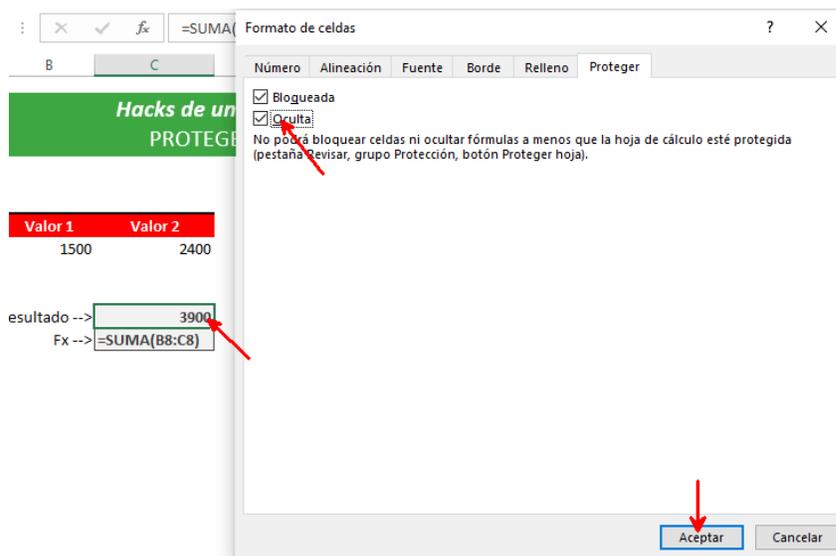


Figura 2.23 Opción Oculta en la celda de la fórmula.

5. Posteriormente se debe dar clic derecho sobre el nombre de la hoja, y se da clic a la opción Proteger hoja.

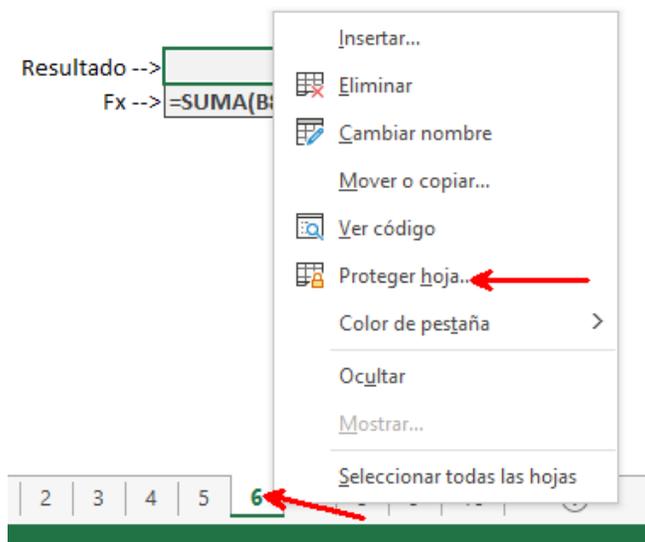


Figura 2.24 Clic derecho proteger hoja.

6. A continuación, se podrá ingresar una contraseña para desbloquear en una futura oportunidad y se da clic en aceptar.

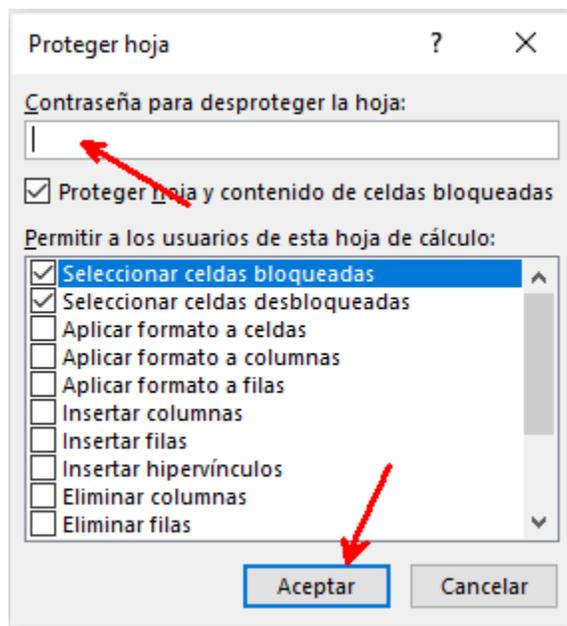


Figura 2.25 Asignar contraseña.

Si posteriormente a los pasos aplicados, nos posicionamos en la celda de la fórmula o función, no se podrá ver en la barra de fórmulas la función realizada y hasta que no se desproteja la hoja no lo permitirá hacer.

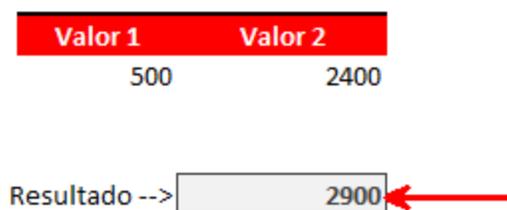
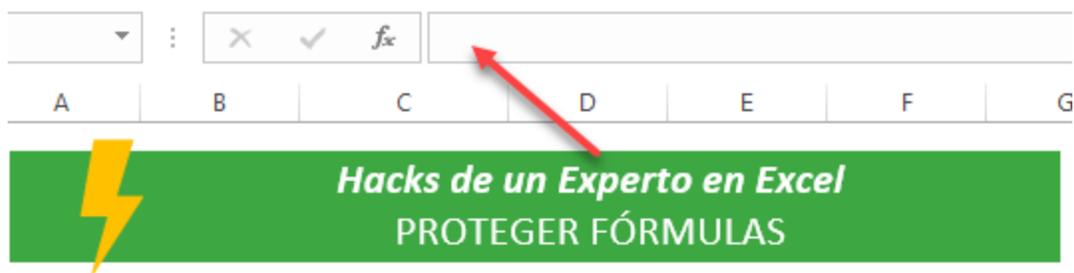


Figura 2.26 Fórmula oculta.

Si por cualquier descuido le das suprimir o intentas sobre escribir sobre la celda de la fórmula, Excel te enviará una advertencia de error:

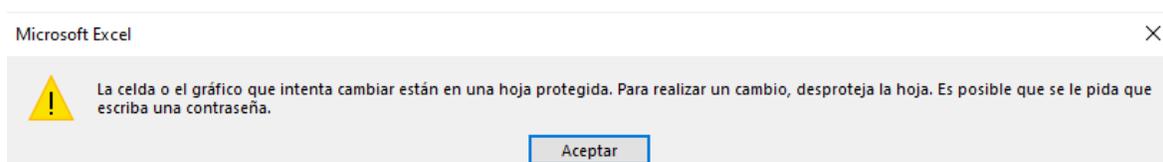


Figura 2.27 Advertencia de protección.

De esta manera se podrá proteger todas las celdas que se quieran en una hoja o libro de Microsoft Excel.

2.6 Restringir fechas con DIASEM

La función DIASEM en Excel nos devuelve el número que identifica un día de la semana, es decir, nos regresa un número entre 1 y 7 el cuál indicará qué día de la semana corresponde a una fecha determinada.

Esta función tiene un argumento que nos permite determinar en qué día empieza la semana y desde ahí asignara un número a cada día.

La sintaxis o argumentos de la función es:



FUNCIÓN: = DIASEM (núm_de_serie; [tipo])

Núm_de_serie: hace referencia a la fecha que se le va a identificar el día de la semana.

[tipo]: indicar el día en el que inicia la semana (1 = domingo, 2 = lunes ...)

Fecha	Día	Día Semana
2/08/2021	Lunes	=DIASEM(B7;2)
3/08/2021	Martes	2
4/08/2021	Miércoles	3
5/08/2021	Jueves	4
6/08/2021	Viernes	5
7/08/2021	Sábado	6
8/08/2021	Domingo	7

Figura 2.28 DIASEM.

A continuación, vamos a desarrollar paso a paso la forma de restringir una celda para que no se pueda ingresar fechas de días no hábiles como sábado y domingo.

1. Lo primero que debe hacerse es crear la celda donde el usuario debe ingresar la fecha y posteriormente darle clic en la pestaña datos:

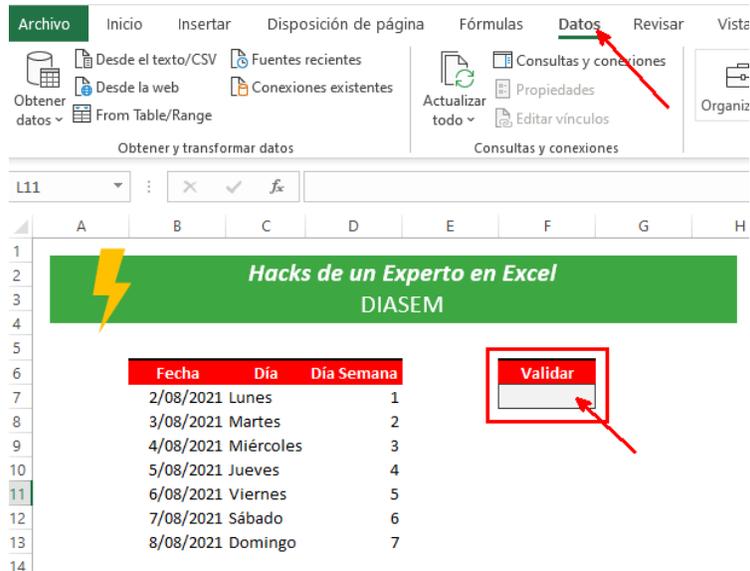


Figura 2.29 Pestaña datos.

2. Luego se debe buscar la opción de validación de datos, y desplegar el campo permitir donde se debe seleccionar personalizado:

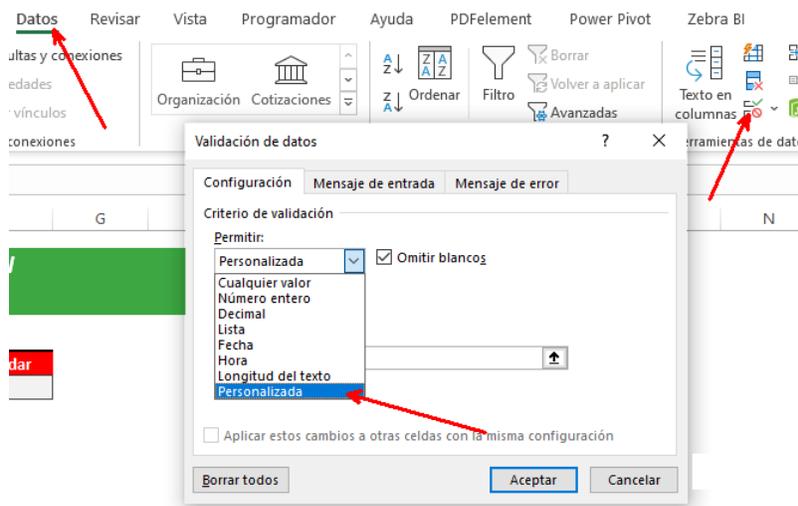


Figura 2.30 Validación de datos personalizada.

3. Para finalizar se debe ingresar la siguiente función en la barra fórmula de la validación de datos, donde se hace referencia a la celda que nos servirá de INPUT o modificable por el usuario, se establece que si el resultado de DIASEM es menor o igual a 5 (lunes a viernes) me permita ingresar la información, de lo contrario nos arroje un error y no permita la escritura:

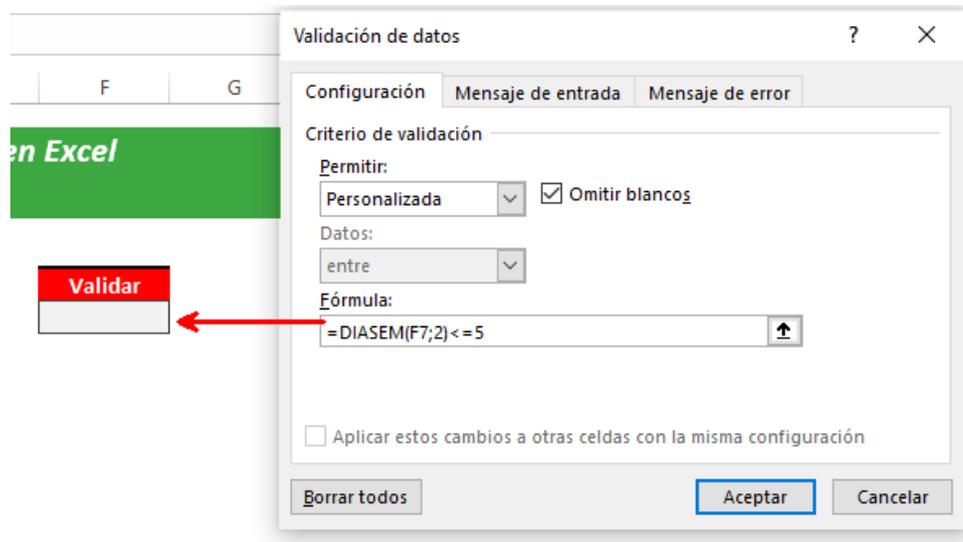


Figura 2.31 Fórmula personalizada.

4. Por último y como paso opcional podemos diseñar y definir el mensaje de error que nos saldrá cuando se ingrese una fecha que no cumple con la validación, para hacerlo se debe dar clic en la pestaña Mensaje de error y definir el mensaje:

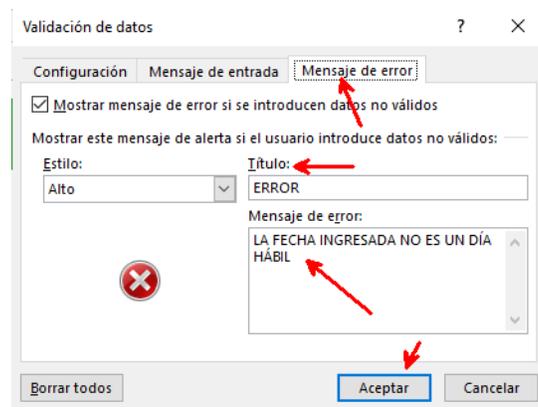


Figura 2.32 Mensaje de error.



HACKS: Restringir celdas para el registro de fechas con base a la función DIASEM y la validación de datos.

Podemos ver que al ingresar una fecha que no es un día hábil, nos arroja el error que definimos en el paso anterior.

Hacks de un Experto en Excel
DIASEM

Excelaprende.com

Fecha	Día	Día Semana
2/08/2021	Lunes	1
3/08/2021	Martes	2
4/08/2021	Miércoles	3
5/08/2021	Jueves	4
6/08/2021	Viernes	5
7/08/2021	Sábado	6
8/08/2021	Domingo	7

Validar
07/08/2021

ERROR

LA FECHA INGRESADA NO ES UN DÍA HÁBIL

Reintentar Cancelar Ayuda

Figura 2.33 Error al ingresar fecha no hábil.

Pero si se ingresa una fecha de lunes a viernes cumpliendo con la validación nos permite el ingreso de la información sin ningún tipo de restricción o error:

Hacks de un Experto en Excel
DIASEM

Excelaprende.com

Fecha	Día	Día Semana
2/08/2021	Lunes	1
3/08/2021	Martes	2
4/08/2021	Miércoles	3
5/08/2021	Jueves	4
6/08/2021	Viernes	5
7/08/2021	Sábado	6
8/08/2021	Domingo	7

Validar
2/08/2021

Figura 2.34 Resultado final.

2.7 Dominando el CONTAR.SI.CONJUNTO con operadores y comodines

La función CONTAR.SI.CONJUNTO es una función estadística. Se usa para contar el número de celdas seleccionadas basándose en varios criterios especificados.

Su sintaxis o estructura es la siguiente:



FUNCIÓN: =CONTAR.SI.CONJUNTO(rango_criterios1; criterios1; [rango_criterios2; criterios2];...)

- **criterios_rango1:** Obligatorio. El primer rango en el que se evalúan los criterios asociados.
- **criterios1:** Obligatorio. Los criterios en forma de número, expresión, referencia de celda o texto que determinan las celdas que se van a contar. Por ejemplo, los criterios se pueden expresar como 32, ">32", B4, "manzanas" o "32".



NOTA: Solo los dos primeros criterios son obligatorios, el resto son opcionales.

A continuación, veremos la aplicación de un CONTAR.SI.CONJUNTO para el siguiente caso:

Producto	País	Costo
Arroz	Colombia	\$ 1.544
Leche	Argentina	\$ 1.700
Miel	Perú	\$ 1.853
Sal	España	\$ 1.930
Pan	Ecuador	\$ 1.692
Aceite	Argentina	\$ 1.862
Lentejas	Colombia	\$ 1.642
Frijol	Perú	\$ 1.882
Arroz	España	\$ 1.859
Leche	España	\$ 1.610
Aceite	Ecuador	\$ 1.816
Pan	Ecuador	\$ 1.812

Figura 2.35 Caso CONTAR.SI.CONJUNTO

En la imagen anterior se tiene una tabla con la descripción de unos productos, país y costos, se nos pide contar los productos del país Ecuador que sus costos sean mayores a 1.700 para dar solución a esto se desarrolla la siguiente función:

Producto	Costo >1700
Ecuador	2

Fx --> =CONTAR.SI.CONJUNTO(C11:C22;B7;D11:D22;">1700")

Producto	País	Costo
Arroz	Colombia	\$ 1.544
Leche	Argentina	\$ 1.700
Miel	Perú	\$ 1.853
Sal	España	\$ 1.930
Pan	Ecuador	\$ 1.692
Aceite	Argentina	\$ 1.862
Lentejas	Colombia	\$ 1.642
Frijol	Perú	\$ 1.882
Arroz	España	\$ 1.859
Leche	España	\$ 1.610
Aceite	Ecuador	\$ 1.816
Pan	Ecuador	\$ 1.812

Figura 2.36 Solución Caso CONTAR.SI.CONJUNTO

Como se pudo evidenciar en la solución de este caso se usó el operador mayor que (>) y este operador hace parte del grupo de operadores que queremos enseñarte en este truco.



HACKS: CONTAR.SI.CONJUNTO con operadores y comodines.

Operador	Descripción
*	REEMPLAZA VARIOS CARACTERES
?	REEMPLAZA 1 CARÁCTER
<	MENOR QUE
>	MAYOR QUE
<>	DIFERENTE QUE
=	IGUAL A

Figura 2.37 Operadores.

Con el operador asterisco (*) se puede reemplazar una cadena de texto continua a una referencia, y de la cual no se tiene la certeza de la cantidad de caracteres siguientes y su composición, veamos un ejemplo:

País	Costo	#
Arg	>1500	2

`Fx --> =+CONTAR.SI.CONJUNTO(K11:K22; K7&" L11:L22; ">1500")`

País	Costo
COloMbla	\$ 1.544
Argent	\$ 1.700
Peru	\$ 1.853
Esp	\$ 1.930
Ecu	\$ 1.692
Argentina**:	\$ 1.862
Colombi	\$ 1.642
Perú	\$ 1.882
Espain	\$ 1.859
España	\$ 1.610
Ecu	\$ 1.816
Ecuador	\$ 1.812

Figura 2.38 Operador asterisco.

Se puede evidenciar como nos dan una tabla con los nombres de los países de manera incompleta o con caracteres especiales, esto nos dificulta el conteo por el criterio país, pero se puede crear una lista con los caracteres que se encuentren de manera correcta, en este caso los 3 primeros caracteres del país.

Con este operador podemos unirlo (&) a la celda donde se tiene los 3 primeros caracteres del país y reemplazará los caracteres siguientes, para poder evaluar el criterio país.

La concatenación puede hacerse haciendo referencia a la celda o escribiéndola de manera implícita, veamos un ejemplo:

Referencia a la celda =R3&"*"

Texto implícito ="Arg*"

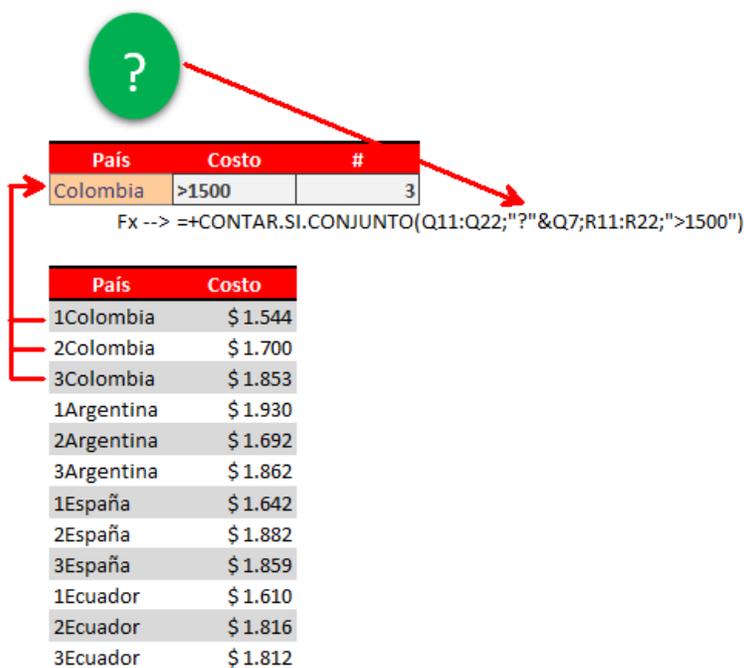


Figura 2.39 Operador signo interrogación.

Con este operador podemos reemplazar un solo carácter, indiferente la ubicación.

Con el resto de los operadores como el < (menor), > (mayor) e = (igual) ya se han podido evidenciar su potencial al utilizar dentro de las funciones y es importante resaltar que siempre deben usarse encerradas en comillas " " .

Menor: "<100" **Mayor:** ">100" **Diferente:** "<>100" **Igual:** "=100"

2.8 Simplificando la concatenación con UNIRCADENAS

Cuando en Excel estamos trabajando con cadenas de texto y se requiere unirlos en una misma celda, podíamos hacerlo de dos maneras, una de ellas por medio del operador AMPERSAND (&) o con la función CONCATENAR, la cual en Excel sirve para unir dos o más cadenas de texto en una sola.



FUNCIÓN: =CONCATENAR (texto 1; texto 2...)

Solo tiene como argumento obligatorio texto 1 y los demás son opcionales, y si se requiere unir texto como delimitadores, estos deben incluirse entre comillas como un argumento de texto más.



ADVERTENCIA: En Excel 2016, Excel Mobile y Excel para la Web, esta función se ha sustituido por la función CONCAT. Aunque la función CONCATENAR sigue estando disponible para garantizar la compatibilidad con versiones anteriores, le recomendamos que use la función CONCAT a partir de ahora. El motivo es que puede que CONCATENAR ya no esté disponible en versiones futuras de Excel.



HACKS: UNIRCADENAS la forma simple de concatenar texto en una sola celda.

Pero sin lugar a duda la nueva función para UNIRCADENAS es una gran evolución de las funciones anteriores, esta función combina el texto de varios rangos o cadenas e incluye el delimitador que se especifique entre cada valor de texto que se combinará. Si el delimitador es una cadena de texto vacío, esta función concatenará los rangos.



FUNCIÓN: =UNIRCADENAS (delimitador, ignorar_vacío, Texto1, [Texto2],...)

El poder de esta función se basa en que nos solicita el delimitador en un solo argumento y desde el inicio, por su parte tiene en cuenta si la cadena de texto a unir tiene espacios no tenerlos en cuenta.



NOTA: Esta función está disponible en Windows o Mac si tiene Office 2019 o si tiene una suscripción Microsoft 365 .

Nombre 1	Nombre 2	Apellido 1	Apellido 2
Julio	Cesar	Rendon	Orozco

Opción tradicional	
Julio Cesar Rendon Orozco	
Fx -->	=CONCAT(B8; " ";C8; " ";D8; " ";E8)

Nueva Opción	
Julio Cesar Rendon Orozco	
Fx -->	=UNIRCADENAS(" ";VERDADERO;B8:E8)

Figura 2.40 Ejemplo UNIRCADENAS.

En el ejemplo anterior podemos ver como en la función CONCAT se debe ingresar el delimitador como un argumento por cada ocasión que se requiera utilizar, mientras que en UNIRCADENAS nos permite hacerlo desde el inicio una única vez.

Por otro lado, nos permite parametrizar si se va a tener en cuenta o no las celdas vacías o con espacios.

Nombre 1	Nombre 2	Apellido 1	Apellido 2
Julio			Orozco

Cadena con espacios CONCAT	
Julio	Orozco

Cadena con espacios UNIRCADENAS	
Julio	Orozco

Figura 2.41 Manejo de espacios.

2.9 Revolucionando las funciones MAX y MIN

Cuando se quería extraer el valor máximo o mínimo de un rango de datos, se debía usar la función MAX y MIN, pero si se requería evaluar esta misma acción con base a una condición o criterio no era posible. En la actualidad Microsoft Excel nos trae una revolución completa en estas dos funciones y se trata de las funciones MAX.SI.CONJUNTO y MIN.SI.CONJUNTO, veamos su composición y funcionalidad.



FUNCIÓN: MAX.SI.CONJUNTO(rango_máximo; rango_criterios1; criterios1; [rango_criterios2; criterios2];...)

MAX		
Costo	Venta	
\$ 1.930	\$ 3.311	

MIN		
Costo	Venta	
\$ 1.544	\$ 2.446	

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.859	\$ 2.673
Aceite	\$ 1.610	\$ 2.650
Lentejas	\$ 1.816	\$ 2.789
Frijol	\$ 1.812	\$ 3.234

Figura 2.42 MAX y MIN.

En el ejemplo se evidencia cómo funciona el MAX y MIN sin ningún tipo de evaluación condicional, estas funciones solo evalúan el rango y arrojan el resultado sin criterios.

MAX.SI.CONJUNTO		
Producto	Costo	Venta
Leche	1700	3010

MIN.SI.CONJUNTO		
Costo	Venta	
1610	2650	

Producto	Costo	Venta
Arroz	\$ 1.544	\$ 3.311
Leche	\$ 1.700	\$ 3.010
Miel	\$ 1.853	\$ 2.446
Sal	\$ 1.930	\$ 2.470
Pan	\$ 1.692	\$ 3.048
Aceite	\$ 1.862	\$ 3.240
Lentejas	\$ 1.642	\$ 3.339
Frijol	\$ 1.882	\$ 2.558
Arroz	\$ 1.859	\$ 2.673
Leche	\$ 1.610	\$ 2.650
Aceite	\$ 1.816	\$ 2.789
Pan	\$ 1.812	\$ 3.234

Figura 2.43 MAX.SI.CONJUNTO y MIN.SI.CONJUNTO

En el cuadro anterior se realiza el cálculo del valor máximo y mínimo, pero con el condicional del producto que se tenga seleccionado en el campo producto.



ADVERTENCIA: Estas funciones están disponibles en Windows o Mac si tiene Office 2019 o si tiene una suscripción Microsoft 365 .



HACKS: MAX.SI.CONJUNTO una manera más completa de extraer el valor máximo y mínimo de un rango de datos que cumplen determinados criterios.

2.10 Realiza búsquedas de columnas de forma horizontal

Cuando se nos pide encontrar un valor en un rango compuesto por varias columnas donde algunas de ellas se encuentran vacías y que independiente de su ubicación nos debe consolidar la información en una sola columna, no existe una sola función que pueda solucionar este inconveniente y entra a jugar la importancia de las fórmulas parametrizadas por nosotros mismos, veamos:

Nombre	Distribuidor	PV #1	PV #2	PV #3	Código	Precio Venta
Martillo	Mundo Art	1515			ADD 17	
Tuercas	Mundo Art		1928		ADD 18	
Llantas	Col Basic		1596		ADD 18	
Lamparas	Plast PM	2751			ADD 18	
Bloques	Col Basic			2836	ADD 17	

Figura 2.44 Caso consolidación de información.

Como se evidencia en la imagen anterior existen unos precios de venta distribuidos en 3 columnas diferentes y se requiere traer la información en una sola.



HACKS: Utilización de la fórmula INDICE y COINCIDIR para traer información de varias columnas y consolidar en una sola.

Antes de solucionar el caso, veamos cómo actúa la evaluación de celdas por medio del operador <> (diferente):

PV #1	PV #2	PV #3	Evaluación
1515			=[{VERDADERO\FALSO\FALSO}]

Figura 2.45 Validación de rango con <>.

En la imagen anterior se escribió la siguiente fórmula para evaluar cómo actúa el operador diferente, `=D22:F22<>"` y al oprimir la tecla **F9** nos muestra cuál sería el resultado a obtener, por lo que nos queda más claro como evaluará la información la siguiente fórmula como solución al caso inicialmente planteado:

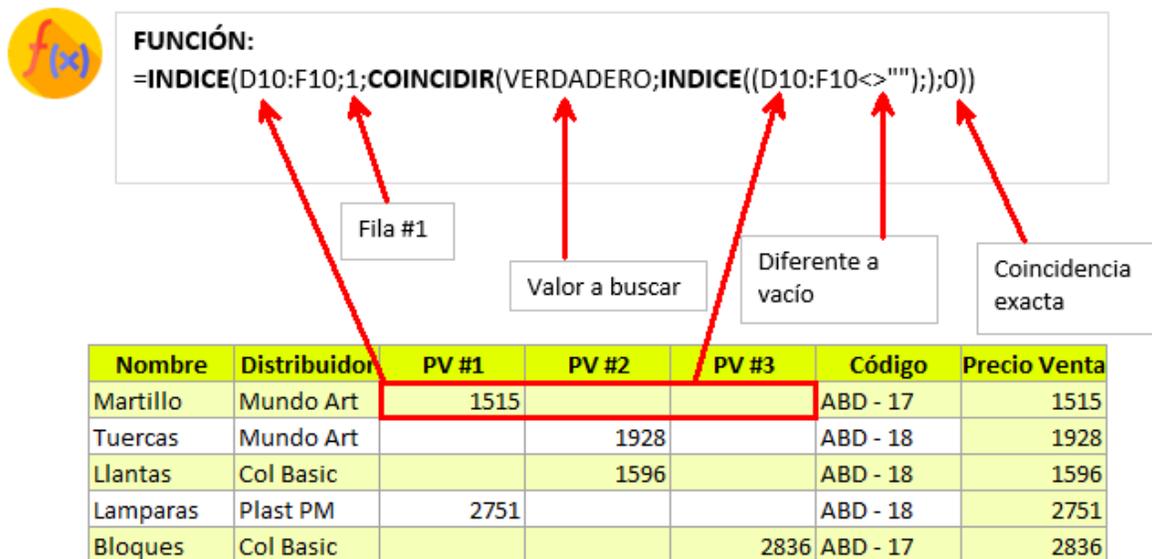


Figura 2.46 Explicación fórmula.

Como se nota en el ejemplo, al utilizar esta fórmula nos permite consolidar en una sola columna el precio de venta independiente en la columna que se encontraba, una forma eficaz de solucionar el caso.

Con solo arrastrar la fórmula de manera vertical se podrá obtener la información consolidada en una sola columna.

2.11 La magia de concatenar varios tipos de formatos

Con el operador AMPERSAND o las funciones de CONCATENAR, CONCAT o UNIRCADENAS podemos unir varias cadenas de texto o números en una misma celda, pero la incógnita viene cuando se nos pide unir una celda que contiene texto, con otra celda que tiene una fecha y otra que tiene un porcentaje, vamos a intentarlo y veamos el resultado:

Título	Informe comparativo
Fecha	14/08/2021
% Cumplimiento	53%

Informe comparativo 44422 %Cumplimiento 0,53395447123235

`=+CONCAT(C8;" ";C9;" ";"%Cumplimiento ";C10)`

Figura 2.47 Exposición caso unir formatos.

Podemos observar como la función no nos reconoce el formato tipo fecha ni el porcentaje, de esta manera no sería viable presentar nuestro informe sin el formato adecuado.



HACKS: Darle formato a la unión de diferentes cadenas de texto, fecha o número.



FUNCIÓN: =TEXTO (Valor al que quiere dar formato, "Código de formato que quiere aplicar")

La función TEXTO le permite cambiar la apariencia de un número aplicándole formato con códigos de formato. Es útil para mostrar números en un formato más legible o para combinarlos con texto o símbolos.

Veamos entonces como aplicar la función texto, dentro de la función CONCAT para unir las cadenas de texto con el formato adecuado.

1. Lo primero que debemos realizar es posicionarnos en la celda que tiene el formato que necesitamos → Clic derecho → Formato de celdas → Opción número → Personalizada → Copiar formato en la barra tipo:

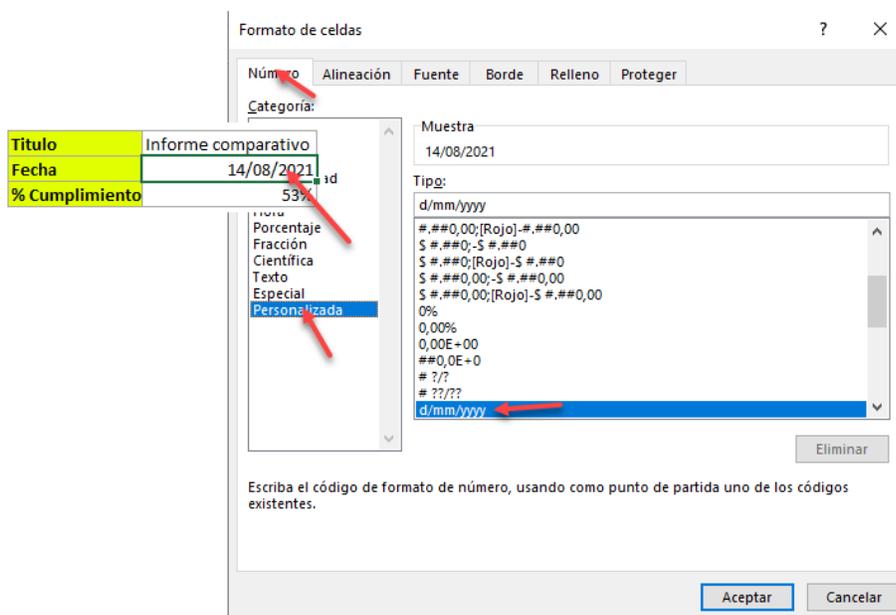


Figura 2.48 Copiar formato deseado

2. Posteriormente vamos a la formula concatenada y agregamos la función texto y en el segundo argumento de esta función pegar entre comillas el formato que se copió en la acción anterior:

```
=+CONCAT(C8;" ";TEXTO(C9;"d/mm/yyyy");" ";"%Cumplimiento ";C10)
```

Figura 2.49 pegar formato deseado en función TEXTO.



NOTA: Esta misma acción debe repetir con los formatos de porcentajes, horas y cualquiera que se quiera incluir en la función TEXTO.

Título	Informe comparativo
Fecha	14/08/2021
% Cumplimiento	53%
Informe comparativo 14/08/2021 %Cumplimiento 53%	

Figura 2.50 Resultado final.

Por último, tenemos el resultado esperado, y este tipo de títulos se pueden agregar a gráficos u objetos como formas en nuestros informes, veamos rápidamente como agregar este título a una forma:

1. En la pestaña insertar → Ilustraciones → Formas → se agrega una forma rectangular:

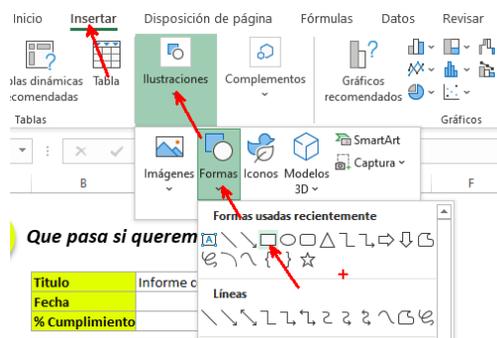


Figura 2.51 Insertar forma.

2. Cuando se agregue la forma, teniéndola seleccionada nos posicionamos en la barra de fórmulas y escribimos el signo = haciendo referencia a la celda del título concatenado:

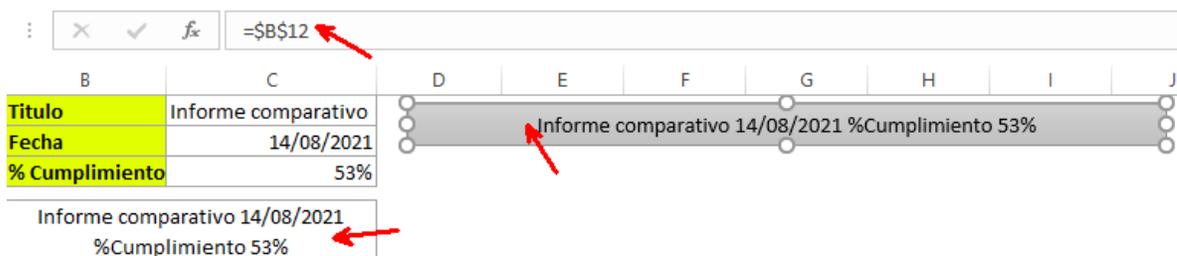


Figura 2.52 Llevar título a una forma.

2.12 Rangos dinámicos DESREF

La función DESREF es una función de búsqueda y referencia. Se usa para devolver una referencia a una celda desplazada de la celda especificada (o la celda o rango de celdas de la parte izquierda superior) a un número concreto de filas y columnas.



FUNCIÓN: DESREF(referencia, filas, columnas[, [altura] [, [ancho]])

Esta función puede anidarse con otras funciones como CONTARA para definir rangos dinámicos y de esta manera incluirlo en validación de datos como listas desplegables o también en tablas dinámicas para no tener que estar actualizando los rangos de la base de datos cuando esta se actualice, vamos a desarrollar un ejemplo de cómo opera la función DESREF.

Fecha	Producto	País	Costo	Venta	Utilidad
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311	\$ 1.767
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010	\$ 1.310
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446	\$ 593
22/01/2020	Sal	España	\$ 1.938	\$ 2.478	\$ 540

Figura 2.53 Ejemplo DESREF.

En la imagen anterior podemos ver que al estar posicionado en la celda 1 de la columna costo y se requiere hacer un desplazamiento de dos columnas hacia la derecha y luego bajar 3 filas para llegar al objetivo que sería la última celda de la columna utilidad. Para dar solución a este problema se debe realizar la función DESREF de la siguiente manera:



Figura 2.54 Argumentos DESREF.

Como podemos ver, la función DESREF nos permite llegar a un objetivo por medio del movimiento entre columnas y filas, pero también podemos hacerlo de manera negativa, es decir si quisiéramos devolvemos del valor (540) de la utilidad, hacia (1.544) del costo solo basta con hacerlo de la siguiente manera:



Figura 2.55 Argumentos negativos.

Con solo cambiar la celda de inicio y el número de filas y columnas que debe devolverse, se puede llegar al objetivo planteado.

Ahora vamos a definir como crear la fórmula para nuestro rango dinámico y al ingresar información sea a nivel de filas o columnas, el rango se actualice automáticamente sin necesita de ir a ingresar o arrastras el rango hasta el nuevo dato registrado.

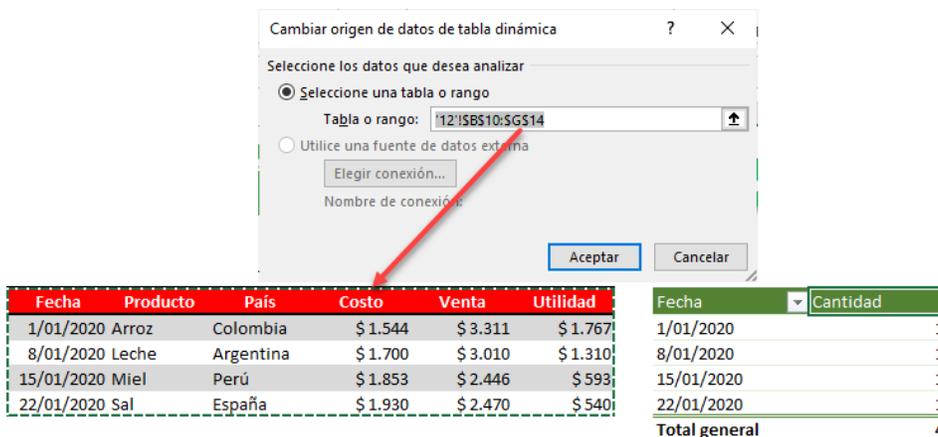


Figura 2.56 Origen de datos

Al ingresar una tabla dinámica y se dé clic en la opción de cambiar origen de datos, este nos resaltaré el rango que se tiene definido para la tabla dinámica y se puede evidenciar que lo fija con una referencia absoluta, el cual no permite que se mueva ni a nivel de columnas ni filas. '\$12!\$B\$10:\$G\$14'

El inconveniente surge que, si se ingresa un nuevo dato, por ejemplo, tenemos un nuevo registro con fecha 25/01/2020 y si no se cambia manualmente en el rango de datos, Excel no reconoce este nuevo registro y por ende no se visualiza en la tabla dinámica, veamos:

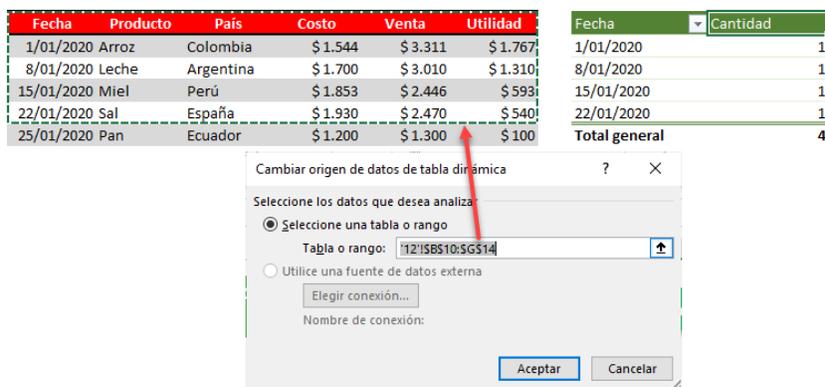


Figura 2.57 Nuevo dato en la BD.

Vamos a desarrollar paso a paso la función para solucionar este inconveniente:

1. Lo primero es ir a la pestaña Fórmulas → Asignar nombre → Definir nombre

Resultado --> \$ 1.544
Fx --> =DESREF(G14;-3;-2)

Fecha	Producto	País	Costo	Venta	Utilidad
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311	\$ 1.767
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010	\$ 1.310
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446	\$ 593
22/01/2020	Sal	España	\$ 1.930	\$ 2.470	\$ 540
25/01/2020	Pan	Ecuador	\$ 1.200	\$ 1.300	\$ 100

Fecha	Cantidad
1/01/2020	1
8/01/2020	1
15/01/2020	1
22/01/2020	1
Total general	4

Figura 2.58 Definir nombre.

2. Luego se debe asignar un nombre a nuestro rango dinámico, el cual se ingresa en la tabla dinámica o demás funciones, y posteriormente se digita la fórmula:

Figura 2.59 Parametrizar el rango.

Para esta ocasión se definirá el nombre del rango así: “RANGOBD”, y la fórmula es la siguiente:



HACKS: Fórmula para definir rango dinámico:

=DESREF('12'!\$B\$10;0;0;CONTARA('12'!\$B:\$B);CONTARA('12'!\$10:\$10))

Se refiere a:

`=DESREF('12'!B10;0;0;CONTARA('12'!$B:$B);CONTARA('12'!$10:$10))`

Cuenta la cantidad de filas

Celda donde inicia

No moverse ni filas ni columnas

Cuenta la cantidad de columnas

Figura 2.60 Explicación fórmula.

Teniendo claro los argumentos y toda la estructura de la fórmula planteada, procedemos a evaluar su funcionamiento, donde al ingresar información nueva a nuestro rango el la tomara sin necesidad de actualizarlo y con solo actualizar la tabla dinámica se incluirán sean las filas o columnas.



ADVERTENCIA: para que este rango queda establecido en la tabla dinámica o fórmula, se debe modificar una única vez en el campo “Cambiar origen de datos” y asignarle el nombre que se le dio al rango.

Cambiar origen de datos de tabla dinámica

Seleccione los datos que desea analizar

Seleccione una tabla o rango

Tabla o rango: 12'!\$B\$10:\$G\$14 ← Borrar

Utilice una fuente de datos externa

Elegir conexión...

Nombre de conexión:

Cambiar origen de datos de tabla dinámica

Seleccione los datos que desea analizar

Seleccione una tabla o rango:

Tabla o rango: RANGODB ← Escribir nombre

Utilice una fuente de datos externa

Elegir conexión...

Nombre de conexión:

Aceptar Cancelar

Figura 2.61 Reemplazar rango por nombre.

Luego de haber cambiado el nombre del rango de datos y asignar el nombre que definimos en un inicio (Ver figura 2.59). ya se podrá quitar o agregar filas o columnas y nuestro rango al estar completamente dinámico lo detectará.

Es importante precisar y como ya se ha hecho, de que este rango dinámico no solo puede usarse en tablas dinámicas, sino también en las funciones o fórmulas donde se hace necesario ingresar un rango y que este puede estar cambiando, como por ejemplo en el argumento de matriz o tabla en la función BUSCARV.



NOTA: Esta fórmula también puede modificarse el argumento del conteo de columnas, si solo se va a evaluar a nivel de filas, se puede dejar fijo el número 1, para indicar que solo tenga en cuenta una sola columna, pero que sea dinámico a nivel de fila.

The screenshot shows an Excel interface with a dynamic table, a dialog box, and a pivot table. The dynamic table has columns: Fecha, Producto, País, Costo, Venta, Utilidad. The dialog box is titled 'Cambiar origen de datos de tabla dinámica' and shows 'RANGOBD' as the selected range. The pivot table shows a summary of the data by date.

Fecha	Producto	País	Costo	Venta	Utilidad
1/01/2020	Arroz	Colombia	\$ 1.544	\$ 3.311	\$ 1.767
8/01/2020	Leche	Argentina	\$ 1.700	\$ 3.010	\$ 1.310
15/01/2020	Miel	Perú	\$ 1.853	\$ 2.446	\$ 593
22/01/2020	Sal	España	\$ 1.930	\$ 2.470	\$ 540
25/01/2020	Pan	Ecuador	\$ 1.200	\$ 1.300	\$ 100
Total general					5

Figura 2.62 Resultado final.

2.13 Encontrar el último registro en una base de datos

Tenemos una base de datos de pedidos por cliente, donde al ser una tabla o base de datos transaccional se encontrará el mismo cliente N veces que se le ha facturado, y la orden es traer la última factura de cada cliente, para llevar el control de periodicidad.

CLIENTE	Pedido	Fecha/Pedido	Num Factura	Fecha/Factura	Vir Total Factura
AA 1	1075656	02/01/2015	249249	16/01/2015	\$467.376
AA 1	1075656	02/01/2015	249705	21/01/2015	\$6.735.792
AA 1	1075656	02/01/2015	250101	23/01/2015	\$5.550.108
AA 1	1075656	02/01/2015	250230	24/01/2015	\$2.678.856
AA 1	1075656	02/01/2015	250470	26/01/2015	\$17.848.734
AA 1	1075656	02/01/2015	250754	28/01/2015	\$12.036.102
AA 1	1075656	02/01/2015	250938	29/01/2015	\$6.517.782
AA 1	1075656	02/01/2015	251194	31/01/2015	\$2.228.676
AA 1	1075656	02/01/2015	251585	04/02/2015	\$1.796.040
AA 1	1075656	02/01/2015	252173	10/02/2015	\$315.684
AA 1	1075656	02/01/2015	252443	12/02/2015	\$1.121.574
AA 1	1075656	03/01/2015	252698	13/02/2015	\$295.812
AA 1	1075656	02/04/2016	253709	23/02/2015	\$433.368
AA 2	1075657	02/01/2015	251172	31/01/2015	\$3.424.710
AA 3	1075658	02/01/2015	249973	23/01/2015	\$421.128
AA 3	1075658	02/01/2015	255961	18/03/2015	\$2.628.528
AA 4	1075659	02/01/2015	249119	15/01/2015	\$474.408
AA 4	1075659	02/01/2015	249622	20/01/2015	\$5.061.930
AA 4	1075659	03/04/2017	250457	26/01/2015	\$1.663.536
AA 5	1075660	02/01/2015	249120	15/01/2015	\$1.296.912
AA 5	1075660	02/01/2015	249623	20/01/2015	\$2.040.180
AA 5	1075660	02/05/2016	250458	26/01/2015	\$895.872
AA 6	1075661	02/01/2015	249608	20/01/2015	\$1.798.956

Figura 2.63 Base de datos.

Como se evidencia en la base de datos el cliente AA 1, tiene 1 pedido con 13 facturas y el objetivo es encontrar la última factura con fecha de pedido 02/04/2016 y de esta misma manera para todos los cliente.

1. Lo primero que debemos hacer es crear dos columnas en la base de datos que llamaremos, Contador y Llave.

CLIENTE	Pedido	Fecha/Pedido	Contador	Llave	Num Factura	Fecha/Factura	Vlr Total Factura
AA 1	1075656	02/01/2015			249249	16/01/2015	\$467.376
AA 1	1075656	02/01/2015			249705	21/01/2015	\$6.735.792
AA 1	1075656	02/01/2015			250101	23/01/2015	\$5.550.108
AA 1	1075656	02/01/2015			250230	24/01/2015	\$2.678.856
AA 1	1075656	02/01/2015			250470	26/01/2015	\$17.848.734
AA 1	1075656	02/01/2015			250754	28/01/2015	\$12.036.102
AA 1	1075656	02/01/2015			250938	29/01/2015	\$6.517.782

Figura 2.64 Insertar columnas.

2. Creamos una segunda tabla donde se va a presentar la información, esta tabla debe tener los clientes únicos de nuestra base de datos:

CLIENTE	PEDIDO	CANTIDAD	FACTURA	FECHA FACTURA	VALOR
AA 1	1075656				
AA 2	1075657				
AA 3	1075658				
AA 4	1075659				
AA 5	1075660				
AA 6	1075661				
AA 7	1075662				
AA 8	1075663				
AA 9	1075664				
AA 10	1075665				

Figura 2.65 Clientes únicos (Informe).

3. Luego en la columna “Contador” se debe digitar la función CONTAR.SI para contar los pedidos y fijando la primera celda, para poder al arrastrar crear el conteo de todos los pedidos por cliente:

A	B	C	D	E	F	G	H
CLIENTE	Pedido	Fecha/Pedido	Contador	Llave	Num Factura	Fecha/Factura	Vlr Total Factura
AA 1	1075656	02/01/2015	1		249249	16/01/2015	\$467.376
AA 1	1075656	02/01/2015	2		249705	21/01/2015	\$6.735.792
AA 1	1075656	02/01/2015	3		250101	23/01/2015	\$5.550.108

Figura 2.66 CONTAR.SI.

4. Posteriormente en la columna “Llave” se debe concatenar el Pedido con la columna “Contador”, de esta manera nos crea una columna única para cada cliente:

CLIENTE	Pedido	Fecha/Pedido	Contador	Llave	Num Factura	Fecha/Factura	Vir Total Factura
AA 1	1075656	02/01/2015	1	10756561	249249	16/01/2015	\$467.376
AA 1	1075656	02/01/2015	2	10756562	249705	21/01/2015	\$6.735.792
AA 1	1075656	02/01/2015	3	10756563	250101	23/01/2015	\$5.550.108
AA 1	1075656	02/01/2015	4	10756564	250230	24/01/2015	\$2.678.856
AA 1	1075656	02/01/2015	5	10756565	250470	26/01/2015	\$17.848.734
AA 1	1075656	02/01/2015	6	10756566	250754	28/01/2015	\$12.036.102
AA 1	1075656	02/01/2015	7	10756567	250938	29/01/2015	\$6.517.782

Figura 2.67 Concatenar pedido y contador.

5. Luego debemos ir a la tabla del informe donde se encuentran los clientes únicos y en la columna “Cantidad” vamos a contar cuantas veces se encuentra el cliente en la base de datos y arrastramos:

CLIENTE	PEDIDO	CANTIDAD	FACTURA	FECHA FACTURA	VALOR
AA 1	1075656	=CONTAR.SI(A2:A274;J2)			
AA 2	1075657	CONTAR.SI(rango; criterio)			
AA 3	1075658				
AA 4	1075659				
AA 5	1075660				
AA 6	1075661				

Figura 2.68 Contar los pedidos del cliente.

6. Seguidamente en la columna “FACTURA” se hace un BUSCARV y en el argumento valor buscado se concatena el pedido con la cantidad y la matriz o tabla será nuestra base de datos desde la columna llave, (se repite con fecha factura y valor).

E	F	G	H	I	J	K	L	M	N	O
Llave	Num Factura	Fecha/Factura	Vir Total Factura		CLIENTE	PEDIDO	CANTIDAD	FACTURA	FECHA FACTURA	VALOR
10756561	249249	16/01/2015	\$467.376		AA 1	1075656	13	=BUSCARV(K2&L2;E:H;2;0)		
10756562	249705	21/01/2015	\$6.735.792		AA 2	1075657	1			
10756563	250101	23/01/2015	\$5.550.108		AA 3	1075658	2			
10756564	250230	24/01/2015	\$2.678.856		AA 4	1075659	3			
10756565	250470	26/01/2015	\$17.848.734		AA 5	1075660	3			
10756566	250754	28/01/2015	\$12.036.102		AA 6	1075661	3			
10756567	250938	29/01/2015	\$6.517.782		AA 7	1075662	4			
10756568	251194	31/01/2015	\$2.228.676		AA 8	1075663	1			
10756569	251585	04/02/2015	\$1.796.040		AA 9	1075664	4			
107565610	252173	10/02/2015	\$315.684		AA 10	1075665	4			

Figura 2.69 BUSCARV concatenando.

CLIENTE	Pedido	Fecha/Pedido	Contador	Llave	Num Factura	Fecha/Factura	Vlr Total Factura	CLIENTE	PEDIDO	CANTIDAD	FACTURA	FECHA FACTURA	VALOR
AA 1	1075656	02/01/2015	1	10756561	249249	16/01/2015	\$467.376	AA 1	1075656	13	253709	23/02/2015	\$ 433.368
AA 1	1075656	02/01/2015	2	10756562	249705	21/01/2015	\$6.735.792	AA 2	1075657	1	251172	31/01/2015	\$ 3.424.710
AA 1	1075656	02/01/2015	3	10756563	250101	23/01/2015	\$5.550.108	AA 3	1075658	2	255961	18/03/2015	\$ 2.628.528
AA 1	1075656	02/01/2015	4	10756564	250230	24/01/2015	\$2.678.856	AA 4	1075659	3	250457	26/01/2015	\$ 1.663.536
AA 1	1075656	02/01/2015	5	10756565	250470	26/01/2015	\$17.848.734	AA 5	1075660	3	250458	26/01/2015	\$ 895.872
AA 1	1075656	02/01/2015	6	10756566	250754	28/01/2015	\$12.036.102	AA 6	1075661	3	251213	31/01/2015	\$ 3.575.070
AA 1	1075656	02/01/2015	7	10756567	250938	29/01/2015	\$6.517.782	AA 7	1075662	4	251436	2/02/2015	\$ 1.265.460
AA 1	1075656	02/01/2015	8	10756568	251194	31/01/2015	\$2.228.676	AA 8	1075663	1	250803	28/01/2015	\$ 1.312.380
AA 1	1075656	02/01/2015	9	10756569	251585	04/02/2015	\$1.796.040	AA 9	1075664	4	251422	2/02/2015	\$ 2.096.082
AA 1	1075656	02/01/2015	10	107565610	252173	10/02/2015	\$315.684	AA 10	1075665	4	253254	18/02/2015	\$ 370.656
AA 1	1075656	02/01/2015	11	107565611	252443	12/02/2015	\$1.121.574	AA 11	1075666	2	251226	31/01/2015	\$ 907.596
AA 1	1075656	03/01/2015	12	107565612	252698	13/02/2015	\$295.812	AA 12	1075673	7	252517	12/02/2015	\$ 1.338.972
AA 1	1075656	02/04/2016	13	107565613	253709	23/02/2015	\$433.368	AA 13	1075674	1	250400	26/01/2015	\$ 4.078.632
AA 2	1075657	02/01/2015	1	10756571	251172	31/01/2015	\$3.424.710	AA 14	1075678	1	251159	31/01/2015	\$ 11.987.328

Figura 2.70 Resultado Final.

Se puede ver como el resultado obtenido es el último pedido para cada cliente, en el caso del cliente AA 1, nos trae como fecha de última factura el 23/02/2015, coincidiendo con lo registrado en la base transaccional, esta puede ser una gran herramienta cuando se va a hacer relacionamiento de clientes y tener toda la información con respecto a la última compra o pedido de estos.



HACKS: Encontrar el último registro de una base de datos.

2.14 El uso correcto de los comodines "*"

En Microsoft Excel se puede usar caracteres especiales para reemplazar cadenas de texto que no se encuentran correctamente definidas o estructuradas, por ejemplo, en la función BUSCARV se puede concatenar el valor buscado con el asterisco entre comillas para indicar que no tenga en cuenta el texto que se encuentre antes, después (o ambos) del argumento valor buscado.



HACKS: Uso del comodín "*" en la función BUSCARV.

Caso 1: Comodín a la derecha

Veamos a continuación el primer escenario de usar correctamente el comodín:



Alumno	Nota Examen Anterior
A001 - Pedro Fanco	7,5
A002 - Milena Ramirez	8,6
A003 - Lucas Jaramillo	5,4
A004 - Natalia Leal	3,9

Alumno	Nota Examen Actual	Nota Examen Anterior
A001	5,5	
A002	3,6	
A003	7,4	
A004	8,5	

Figura 2.71 Comodín 1.

Este caso nos solicita buscar la nota del examen anterior de los alumnos, quizás hasta aquí todo muy fácil y con la función BUSCARV se resolvería, pero cuando notamos que el dato valor buscado es el código del alumno y en la BD se encuentra el código unido al nombre del alumno, esto nos arrojaría un error.

Ahí es donde entra el potencial de los comodines, porque de lo contrario tocaría corregir manualmente los registros o anidar varias funciones para llegar al resultado esperado.



FUNCIÓN: =BUSCARV (B14&"*"; \$B\$6:\$C\$10 ; 2 ; 0)

Alumno	Nota Examen Anterior
A001 - Pedro Fanco	7,5
A002 - Milena Ramirez	8,6
A003 - Lucas Jaramillo	5,4
A004 - Natalia Leal	3,9

Alumno	Nota Examen Actual	Nota Examen Anterior
A001	5,5	=BUSCARV(B14&"*";\$B\$6:\$C\$10;2;0)
A002	3,6	
A003	7,4	
A004	8,5	

Figura 2.72 Fórmula con comodín 1.

De esta manera obtenemos el resultado sin necesidad de modificar los datos en la base de datos.

Hacks de un Experto en Excel
BUSCARV con comodín 1

Alumno	Nota Examen Anterior
A001 - Pedro Fanco	7,5
A002 - Milena Ramirez	8,6
A003 - Lucas Jaramillo	5,4
A004 - Natalia Leal	3,9

Alumno	Nota Examen Actual	Nota Examen Anterior
A001	5,5	7,5
A002	3,6	8,6
A003	7,4	5,4
A004	8,5	3,9

Figura 2.73 Resultado final.

Caso 2: Comodín a la izquierda

En el siguiente escenario se nos pide encontrar las cantidades y costo por producto, pero nos suministran solo el código del producto y en la matriz o tabla se encuentra todo completo el nombre con el código del producto:



NOMBRE PRODUCTO	CANTIDAD	COSTO	CODIGO	CANTIDAD	COSTO
NOVAMYL 10000 BG-Q10072	5	\$2.297.067,20	Q10154		
PEROXIDO AL 60% GRANEL-Q30070	23920	\$37.170.005,60	Q10300		
PEROXIDO AL 50% - GRANEL-Q30036	24210	\$33.464.030,40	Q10048		
ESPERASE 8.0 L-Q10098	50	\$4.606.409,50	Q30022		
ACIDO CITRICO-Q10019	1,5	\$3.744,57	Q10050		
GOMA XANTICA 200 MESH-Q10033	1,2	\$8.239,02	Q30022		
CLARA DE HUEVO ALTO BATIDO-Q10043	2,2	\$69.477,82	Q30074		
SORBATO DE POTASIO-Q10048	0,5	\$4.309,94	Q30001		
ALMIDON NATIVO DE PAPA (BOLSA GIRASO-Q10300)	1	\$2.160,59			

Figura 2.74 Comodín 2.



FUNCIÓN: =BUSCARV("*" & E7; \$A\$6:\$C\$30; 2; 0)

Como podemos ver el comodín ("*") se concatena antes del valor buscado y de esta forma nos arroja el resultado correcto.

NOMBRE PRODUCTO	CANTIDAD	COSTO	=BUSCARV("*" & E7; \$A\$6:\$C\$30; 2; 0)		
			CODIGO	CANTIDAD	COSTO
NOVAMYL 10000 BG-Q10072	5	\$2.297.067,20	Q10154	175	1.251.625
PEROXIDO AL 60% GRANEL-Q30070	23920	\$37.170.005,60	Q10300	1	2.161
PEROXIDO AL 50% - GRANEL-Q30036	24210	\$33.464.030,40	Q10048	1	4.310
ESPERASE 8.0 L-Q10098	50	\$4.606.409,50	Q30022	2.030	3.183.385
ACIDO CITRICO-Q10019	1,5	\$3.744,57	Q10050	1.000	3.421.210
GOMA XANTICA 200 MESH-Q10033	1,2	\$8.239,02	Q30022	2.030	3.183.385
CLARA DE HUEVO ALTO BATIDO-Q10043	2,2	\$69.477,82	Q30074	2.450	3.480.397
SORBATO DE POTASIO-Q10048	0,5	\$4.309,94	Q30001	2.400	3.592.152
ALMIDON NATIVO DE PAPA (BOLSA GIRASO-Q10300)	1	\$2.160,59			

Figura 2.75 Resultado final.

Caso 3: Comodín ambos lados.

Este caso nos presenta un grado mayor de dificultad, ya que como se evidencia en la imagen siguiente nos brinda el código para encontrar un valor, pero al revisar la matriz se encuentra el código en el intermedio, donde al lado izquierdo y derecho tiene diferente texto, por lo que debemos usar el comodín en ambos lados.



Figura 2.76 Comodín 3.



FUNCIÓN: =BUSCARV("*"&D7&"*", A:B; 2; 0)



Figura 2.77 Resultado final.

Como resultado se obtiene el dato correcto para cada código sin importar el texto que se tenga antes y después del valor buscado.

2.15 Tabular formulas a nivel profesional

Cuando ya manejas Microsoft Excel en tu día a día es muy importante evolucionar y mejorar en la parte de presentación de nuestras fórmulas y funciones, ya que si estas no se van a ocultar cualquier persona consumidora de los datos podrá visualizarla y esta parte también habla bien o mal del desarrollador del informe.

Es por esto que a continuación te damos un truco fácil, ágil y sencillo, pero que te dejará como todo un profesional a la hora de presentar tus fórmulas.



HACKS: Tabular las fórmulas y funciones con **Alt + Enter**

Una vez se ha terminado la fórmula o función o al momento de escribirla puedes oprimir la tecla Alt + Enter al mismo tiempo, esto debe hacer siempre que se haya terminado una línea de la fórmula.

 **Hacks de un Experto en Excel**
Tabular fórmulas

Tabla de Calificaciones			
1	Entre	30	Malo
31	Entre	60	Regular
61	Entre	90	Bueno
91	Entre	100	Excelente

CÓDIGO	TRABAJADOR	Puntaje	Calificación
1001	Cristian Osorio	23	Malo
1002	Rafael Roldan	7	Malo
1003	Nancy Pineda	82	Bueno
1004	Fredy Bolivar	33	Regular
1005	Mónica Carvajal	23	Malo
1006	Paola Gómez	19	Malo
1007	Elsa Correa	23	Malo

Figura 2.78 Caso tabular fórmulas.

En la imagen anterior nos muestran la fórmula que tiene aplicada la columna calificación, la cual con una función de SI anidada se logró el objetivo, sin embargo, cuando vamos a revisar la fórmula como quedó estructurada, podemos ver que su código es un poco largo y esto no facilita su correcta lectura, veamos:

 =+SI(D13<=\$D\$6;\$E\$6;SI(Y(D13>=\$B\$7;D13<=\$D\$7);\$E\$7;SI(Y(D13>=\$B\$8; D13<=\$D\$8);\$E\$8;SI(D13>=\$B\$9;\$E\$9;0))))

Figura 2.79 Fórmula aplicada.

Pero al usar el truco anterior, se podrá dar un formato más elegante y profesional a nuestras fórmulas y a su vez nuestros informes.

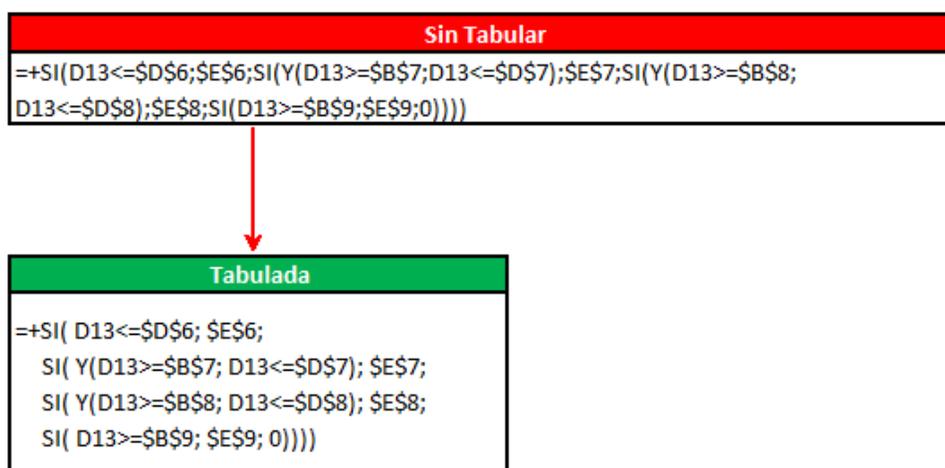


Figura 2.80 Fórmula tabulada.



FUNCIÓN: =+SI(D13<=\$D\$6; \$E\$6;

SI(Y(D13>=\$B\$7; D13<=\$D\$7); \$E\$7;

SI(Y(D13>=\$B\$8; D13<=\$D\$8); \$E\$8;

SI(D13>=\$B\$9; \$E\$9; 0))))

Mientras más larga sea la fórmula, mucho mejor se verá la tabulación o formateo de esta.

2.16 Nombrar rangos a un solo Clic

Cuando se requiere trabajar con rangos basados en los encabezados y filas, en ocasiones nos tomamos mucho tiempo si tenemos más de una columna o filas, por ejemplo, el caso siguiente:

**Hacks de un Experto en Excel**
Nombrar rangos a un Clic

VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO
MARIA	47	44	45	47	54	61
ANDREA	150	89	126	110	157	167
CARLOS	31	34	27	30	29	36
SIMON	63	62	54	57	57	77
MIGUEL	129	143	116	131	131	161
VANESSA	10	11	12	12	13	13
JOHANNA	50	66	69	73	71	101
FABIAN	7	9	8	9	10	12

Figura 2.81 Caso de rangos.

Se nos pide que debemos nombrar los rangos de cada mes y vendedor, si lo hacemos de manera mensual, tendríamos que seleccionar la columna de cada mes uno por uno e ir a la Opción Fórmulas → Segmento nombres definidos → Crear desde la selección.

Sin lugar a duda esto nos tomaría tiempo hacerlo, pero a continuación te daremos el truco para nombrar un rango completo a nivel de columna y filas con un solo Clic.



HACKS: Ctrl + Shift + F3 = Nombrar rangos definidos de manera automática y rápida.

1. Lo primero que debes hacer es seleccionar el rango de los datos que se quieren nombrar:

VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
MARIA	47	44	45	47	54	61	63	60	48	44	47	47
ANDREA	150	89	126	110	157	167	185	174	120	167	179	130
CARLOS	31	34	27	30	29	36	34	38	35	35	34	35
SIMON	63	62	54	57	57	77	67	87	64	72	68	83
MIGUEL	129	143	116	131	131	161	169	177	148	140	141	144
VANESSA	10	11	12	12	13	13	15	16	14	11	16	13
JOHANNA	50	66	69	73	71	101	102	112	88	85	77	85
FABIAN	7	9	8	9	10	12	13	13	14	12	13	11

Figura 2.82 Seleccionar rango.

2. Se oprime al mismo tiempo la tecla Ctrl + Shift + F3 y se le da aceptar al cuadro emergente que surge:

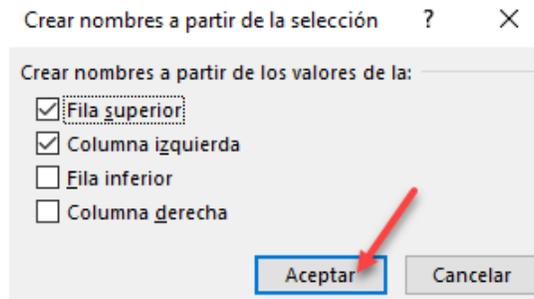


Figura 2.83 Ctrl + Shift + F3.

3. Al realizar esto podemos validar en la opción Fórmulas → Administrador de nombres, la lista que se creó a partir de este proceso:

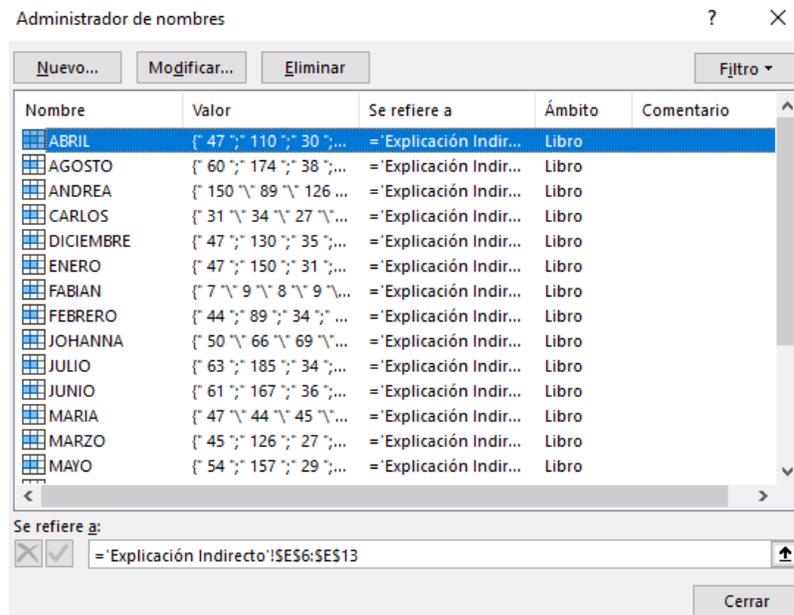


Figura 2.84 Administrador de nombres

2.17 Sacando el potencial del INDIRECTO

Teniendo claro el valor del truco anterior al nombrar con un solo Clic un rango de datos, vamos a utilizar esta información para ver el potencial que tiene la función INDIRECTO.

Devuelve la referencia especificada por una cadena de texto. Las referencias se evalúan de inmediato para presentar su contenido. Use INDIRECTO cuando desee cambiar la referencia a una celda en una fórmula sin cambiar la propia fórmula.



FUNCIÓN: =INDIRECTO(ref; [a1])

A continuación, se muestra una tabla de ventas por vendedor y mes, y si quisiéramos evaluar las ventas de un mes en específico para cada vendedor se tendría que hacer con la función INDICE y COINCIDIR anidadas, veamos este ejemplo:



FUNCIÓN:
=INDICE(\$A\$5:\$M\$13;COINCIDIR(\$B17;\$A\$5:\$A\$1048576;0);COINCIDIR(\$C\$16;\$5:\$5;0))

VENDEDOR	ENERO	FEBRERO	MARZO
MARIA	47	44	45
ANDREA	150	89	126
CARLOS	31	34	27
SIMON	63	62	54
MIGUEL	129	143	116
VANESSA	10	11	12
JOHANNA	50	66	69
FABIAN	7	9	8

=INDICE(\$A\$5:\$M\$13;COINCIDIR(\$B17;\$A\$5:\$A\$1048576;0);COINCIDIR(\$C\$16;\$5:\$5;0))

INDICE-COINCIDIR

MES	FEBRERO
MIGUEL	143
VANESSA	11
JOHANNA	66
FABIAN	9

Figura 2.85 Caso solución 1.

En el ejercicio anterior veíamos como se debe anidar 3 funciones para poder obtener el resultado, pero si hacemos un uso correcto de la función INDIRECTO y con la creación de los rangos como nombres podemos obtener información mucho más rápida.



HACKS: Utilizar función INDIRECTO de manera ágil y con rangos almacenados como nombres.

INDIRECTO	
MES	MAYO
MIGUEL	=INDIRECTO(F17) INDIRECTO(\$G\$16)
VANESSA	13
JOHANNA	71
FABIAN	10
MARIA	54
ANDREA	157
CARLOS	29
SIMON	57

Figura 2.86 Caso INDIRECTO.



FUNCIÓN: =INDIRECTO(F17) INDIRECTO(\$G\$16)



ADVERTENCIA: Es importante que crees los rangos como nombres, para realizarlo con la función INDIRECTO. (Ver antes truco 2.16)

De esta manera se crea un informe completamente dinámico y dependiente de rangos formulados y listas desplegables que dan una apareciendo mucho más profesional a tus informes.

2.18 Sacar valores UNICOS ordenados y automatizados

La forma más clásica de extraer los valores únicos de una columna era la siguiente:

Seleccionar el rango de los datos → ir a DATOS → herramientas de datos → quitar duplicados → seleccionar la columna quitar los duplicados → Aceptar.

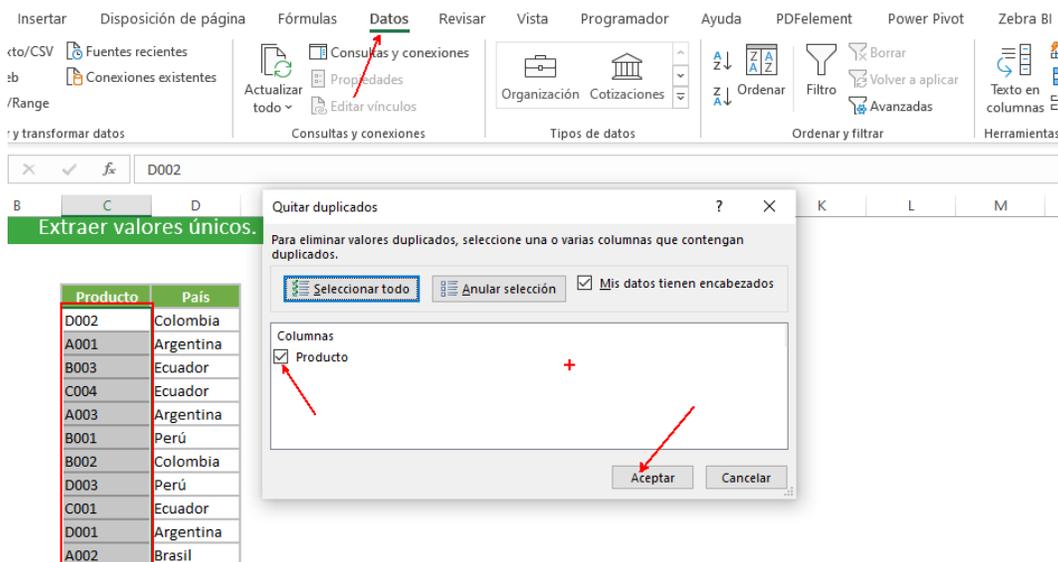


Figura 2.87 Quitar duplicados.

La acción anterior nos quitaba los valores que se repetían y los dejaba en el mismo rango, pero esta forma de hacerlo no es automatizada ya que, si ingreso otro dato repetido al rango, tendría que repetir el paso a paso para quitar ese duplicado.



HACKS: Utilizar la función **ORDENAR** y **UNICOS**.



ADVERTENCIA: Estas funciones solo se encuentran disponibles para usuarios con Office 365.



FUNCIÓN: =+ORDENAR(UNICOS(C7:C22))



Hacks de un Experto en Excel
Extraer valores únicos.

Producto	País
D002	Colombia
A001	Argentina
B003	Ecuador
C004	Ecuador
A003	Argentina
B001	Perú
B002	Colombia
D003	Perú
C001	Ecuador
D001	Argentina
A002	Brasil
C004	Ecuador
A003	Venezuela
B001	Brasil
B002	Argentina

=+ORDENAR(UNICOS(C7:C22))

Producto	País
A001	Argentina
A002	Brasil
A003	Colombia
B001	Ecuador
B002	Perú
B003	Venezuela
C001	
C004	
D001	
D002	
D003	

Figura 2.88 Quitar duplicados con UNICOS.

Lo maravilloso de estas funciones que nos permiten solo con seleccionar el rango de los valores y anidarlos entre ellas dos, tendremos un resultado mucho más profesional, ya que no solo nos deja los valores únicos, sino que también los ordena según sea el caso y adicionalmente queda automatizado, quiere decir que, si al rango inicial se ingresan valores nuevos, estos se incluirían solo con ampliar el rango a la función UNICOS y ORDENAR.

2.19 Búsquedas aproximadas

Cuando se nos da un valor el cual debe buscarse en un rango de datos y este no se encuentra exactamente igual, es necesario hacer una búsqueda aproximada, pero para solucionarlo es normal que los usuarios de Microsoft Excel acudan a utilizar la función SI anidada (*Ver significado), pero en este truco vamos a ver como de manera ágil y simple se podrá solucionar con una sola función.



NOTA: *Cuando se inserta una función como uno de los argumentos dentro de otra función en Excel, se le conoce como funciones de anidamiento o anidadas.

Veamos a continuación una solución al tema de búsquedas aproximadas por medio de la función SI anidada:

 Hacks de un Experto en Excel			
Búsquedas aproximadas			
NOMBRE	CEDULA	SALARIO	ESTADO
Abraham Dionisio Aguirre Tolivar	9415668	1.652.419	
Irene Eleonor Dominguez Wense	7135452	3.297.009	
Quiterio Isaías Josune Torrens	5947808	1.957.247	
Zuriñe Noa Wensel Roman	7909020	6.175.811	
Jerónimo Xenxo Guerrero Zamora	5103145	5.025.331	
Pura Karen Merorio Limas	8338043	4.856.649	
Adrián Silvestre Chavez Vidal	1579966	5.339.756	
Isis Katia Colombres Rojas	2718415	4.523.314	
Octavio Abraham Armesto Anaya	7060751	3.073.380	
Sandra Alcira Ribera Manilla	8480292	5.100.623	
Sansón Calixto Echegarre Villar	3534422	6.774.850	
Paulino Olivia Frey Rosa	7082479	3.608.045	

SALARIO	TIPO PAGO
1.500.000	NO PAGA
3.000.000	DECLARA
4.000.000	PAGA

Figura 2.89 Caso búsquedas aproximadas.

Para este caso vemos que nos dan un salario y se requiere definir el estado (tipo pago), pero como vemos los valores no son iguales, y tendría que hacerse por medio de una búsqueda aproximada, vamos a solucionarlo primeramente con la función SI.

Hacks de un Experto en Excel
Búsquedas aproximadas

=SI(C7<\$F\$8;\$G\$7;SI(Y(C7>=\$F\$8;C7<\$F\$9);\$G\$8;SI(C7>=\$F\$9;\$G\$9;0)))

NOMBRE	CEDULA	SALARIO	ESTADO	SALARIO	TIPO PAGO
Abraham Dionisio Aguirre Tolivar	9415668	1.652.419	NO PAGA	1.500.000	NO PAGA
Irene Eleonor Dominguez Wense	7135452	3.297.009	DECLARA	3.000.000	DECLARA
Quiterio Isaías Josune Torrens	5947808	1.957.247	NO PAGA	4.000.000	PAGA
Zuriñe Noa Wensel Roman	7909020	6.175.811	PAGA		
Jerónimo Xenxo Guerrero Zamora	5103145	5.025.331	PAGA		
Pura Karen Merorio Limas	8338043	4.856.649	PAGA		
Adrián Silvestre Chavez Vidal	1579966	5.339.756	PAGA		
Isis Katia Colombres Rojas	2718415	4.523.314	PAGA		
Octavio Abraham Armesto Anaya	7060751	3.073.380	DECLARA		
Sandra Alcira Ribera Manilla	8480292	5.100.623	PAGA		
Sansón Calixto Echegarre Villar	3534422	6.774.850	PAGA		
Paulino Olivia Frey Rosa	7082479	3.608.045	DECLARA		

Figura 2.90 Caso búsquedas aproximadas con SI.



FUNCIÓN:

=SI(C7<\$F\$8;\$G\$7;SI(Y(C7>=\$F\$8;C7<\$F\$9);\$G\$8;SI(C7>=\$F\$9;\$G\$9;0)))

Como se evidencia en la fórmula se tuvo que utilizar 3 funciones SI con un Y, anidadas para poder dar solución al caso planteado, es ahí donde entra a tener poder el siguiente truco de solución ante casos de búsquedas aproximadas.



HACKS: Búsquedas aproximadas con la función BUSCARV - Verdadero.

Quizás la función BUSCARV sea muy familiar para ti, y más cuando se nos pide encontrar información en diferentes bases de datos, pero hay un argumento que quizás no muchas veces cambiamos o usamos, ese argumento es el último [Rango], el cual nos pide el tipo de coincidencia que necesitamos, casi siempre usamos la coincidencia exacta ósea FALSO (0), pero para utilizar coincidencia aproximada (VERDADERO o 1) no es muy usual.

Veamos la solución al caso con la función BUSCARV Verdadero, sin necesidad de anidas más funciones ni alargar la fórmula:



Figura 2.91 Caso búsquedas aproximadas con BUSCARV.



FUNCIÓN: =BUSCARV(L7; \$O\$7:\$P\$9; 2; VERDADERO)

De esta manera tenemos solución al caso de una manera mucho más rápida, sin tener que anidar otras funciones y solo con cambiar el último argumento del BUSCARV que nos indica el tipo de coincidencia, que para este caso es aproximada, sin lugar a duda BUSCARV sigue siendo una excelente función donde al conocer su funcionalidad y aplicarla de manera correcta nos sacara de muchos problemas a la hora de formular nuestros informes.



NOTA: En el tipo de coincidencia se encuentra en el argumento [Rango] de la función BUSCARV, y en este se puede ingresar la palabra FALSO o "0" para una coincidencia exacta, o VERDADERO o 1 para una coincidencia aproximada.

2.20 Sumar valores no continuos

Realizar una suma de un rango de datos que se encuentra en una misma fila o columna, podrá ser una operación relativamente fácil, pero el reto viene cuando se nos pide sumar valores que se encuentran en columnas y filas diferentes, quizás muchos iniciarían la función (SUMA) y harían referencia a cada celda, pero que pasa cuando son más de 10 celdas que debo sumar. Esto nos llevaría mucho tiempo hacer referencia a cada celda para poder sumarla.

Veamos a continuación el siguiente caso:

Hacks de un Experto en Excel

Sumar valores no continuos

Año	Trimestre	Periodo	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2013	1	2013-01	1	37	3	5	3	24	9	1	9	43	46	32	44	32	32	2	31
2013	2	2013-02	2	25	34	1	18	17	24	2	16	47	37	33	11	36	8	26	
2013	3	2013-03	3	30	20	24	34	21	20	22	43	3	36	31	2	27	3		
2013	4	2013-04	4	43	44	36	47	18	42	44	19	33	5	9	21	18			
2014	1	2014-01	5	24	27	38	44	11	18	12	11	48	38	4	22				
2014	2	2014-02	6	17	10	36	34	42	10	30	47	36	15	2					
2014	3	2014-03	7	26	46	21	6	37	44	33	3	28	47						
2014	4	2014-04	8	40	27	41	24	11	10	50	2	46							
2015	1	2015-01	9	39	48	26	22	19	15	10	47								
2015	2	2015-02	10	17	39	14	32	8	7	11									?
2015	3	2015-03	11	10	13	19	5	24	36										
2015	4	2015-04	12	49	12	20	24	40											
2016	1	2016-01	13	5	44	32	30												
2016	2	2016-02	14	32	45	28													
2016	3	2016-03	15	33	13														
2016	4	2016-04	16	32															

Figura 2.92 Sumar valores no continuos.

Como podemos ver el caso anterior, nos indican que debemos sumar las celdas que se encuentran resaltadas en color rojo, pero estas celdas no se encuentran en una misma columna o fila y es ahí donde necesitamos establecer una fórmula que nos permite identificar cada celda y luego sumarla, veamos:

1. Lo primero que debemos hacer es nombrar las columnas, pero en número:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Año	Trimestre	Periodo	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2013	1	2013-01	1	37	3	5	3	24	9	1	9	43	46	32	44	32	32	2	31
2013	2	2013-02	2	25	34	1	18	17	24	2	16	47	37	33	11	36	8	26	
2013	3	2013-03	3	30	20	24	34	21	20	22	43	3	36	31	2	27	3		
2013	4	2013-04	4	43	44	36	47	18	42	44	19	33	5	9	21	18			

Figura 2.93 Nombrar columnas en número.

2. Posteriormente creamos una tabla aparte donde vamos a realizar los cálculos y la suma:

Fila	Columna	Dirección	Indirecto

Figura 2.94 Cuadro de cálculos.

3. Luego en la columna que llamamos fila, vamos a insertar la función FILA() y arrastramos hacia abajo, hasta la última fila donde tenemos datos:

Fila	Columna	Dirección	Indirecto
=FILA()			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			

Figura 2.95 Función FILA.

Esta función fila lo que hace es darnos el número de la fila actual, y como se arrastra nos indica el número del resto de filas.

4. En la columna que llamamos "Columna" vamos a hacer referencia al valor de la celda que tiene la última columna:

Hacks de un Experto en Excel
Sumar valores no continuos

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Fila	Columna	Dirección	Indirecto
1	37	3	5	3	24	9	1	9	43	46	32	44	32	32	2	31	7	=T5		
2	25	34	1	18	17	24	2	16	47	37	33	11	36	8	26		8			
3	30	20	24	34	21	20	22	43	3	36	31	2	27	3			9			
4	43	44	36	47	18	42	44	19	33	5	9	21	18				10			
5	24	27	38	44	11	18	12	11	48	38	4	22					11			

Figura 2.96 Referencia a última columna.

5. Posteriormente en las celdas inferiores vamos a restarle 1 al valor de la celda anterior y arrastramos:

Fila	Columna	Dirección	Indirecto
7	20		
8	+W7-1		
9			
10			
11			
12			
13			
14			
15			

Figura 2.97 Restar 1 celda.



FUNCIÓN: =DIRECCIÓN(fila-número, colum-número [, [ref-tipo] [, [A1-ref-tipo-indicador] [, nombre de la hoja]])

La función **DIRECCIÓN** es una función de búsqueda y referencia. Se usa para devolver la dirección de una celda expresada como un valor numérico.

6. Con esta función vamos a ingresarla en la columna “Dirección” y llamamos o hacemos referencia a las dos columnas anteriores, fila y columna los demás argumentos no son necesarios:

Fila	Columna	Dirección	Indirecto
7		=DIRECCION(V7;W7)	
8	19		
9	18		
10	17		
11	16		
12	15		
13	14		
14	13		
15	12		



Figura 2.98 Función Dirección.

7. Luego con la función INDIRECTO podemos hacer referencia a la celda dirección y esta nos traerá el valor que encuentre en cada celda:

Fila	Columna	Dirección	Indirecto
7	20	=INDIRECTO(X7)	
8	19	\$\$8	
9	18	\$R\$9	
10	17	\$Q\$10	

Figura 2.99 Función Indirecto.

Fila	Columna	Dirección	Indirecto
7	20	\$T\$7	31
8	19	\$\$8	26
9	18	\$R\$9	3
10	17	\$Q\$10	18
11	16	\$P\$11	22
12	15	\$O\$12	2
13	14	\$N\$13	47
14	13	\$M\$14	46
15	12	\$L\$15	47
16	11	\$K\$16	11
17	10	\$J\$17	36
18	9	\$I\$18	40
19	8	\$H\$19	30
20	7	\$G\$20	28
21	6	\$F\$21	13
22	5	\$E\$22	32

Figura 2.100 Datos.

De esta manera tenemos todos nuestros datos en una misma columna y listos para sumarlos:

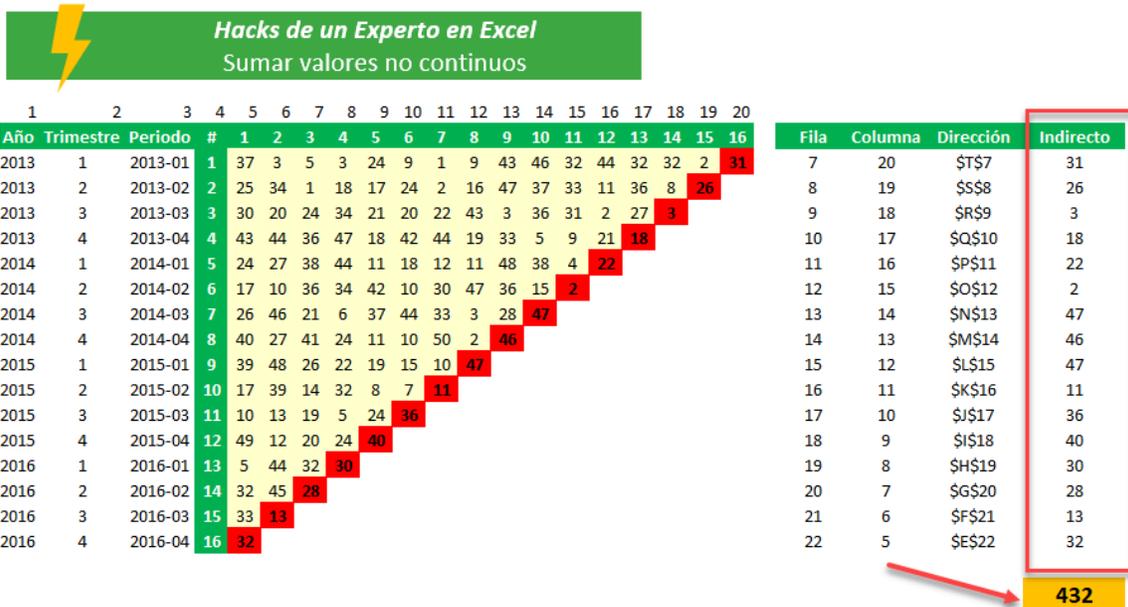


Figura 2.101 Solución.

Esta sería la solución para el caso planteado donde se nos pedía sumar los valor no continuos de un rango de datos.



HACKS: Sumar valores no continuos con la función FILA, DIRECCIÓN e INDIRECTO.

Capítulo 3: Informes y Gráficos

3.1 Gráfico de barras con lupa

Debemos analizar los datos de un periodo de tiempo, la clave de este es que será movable de tal forma que si deseas ver la información a un mayor o menor detalle solo sea cuestión de un par de clics y tener el resultado buscado.

Insertamos dos controles de formulario (Barra de desplazamiento) y los ubicamos en las variables de Meses y detalles, estas nos ayudarán a controlar el periodo de tiempo que queramos ver y la cantidad de ellos al tiempo.

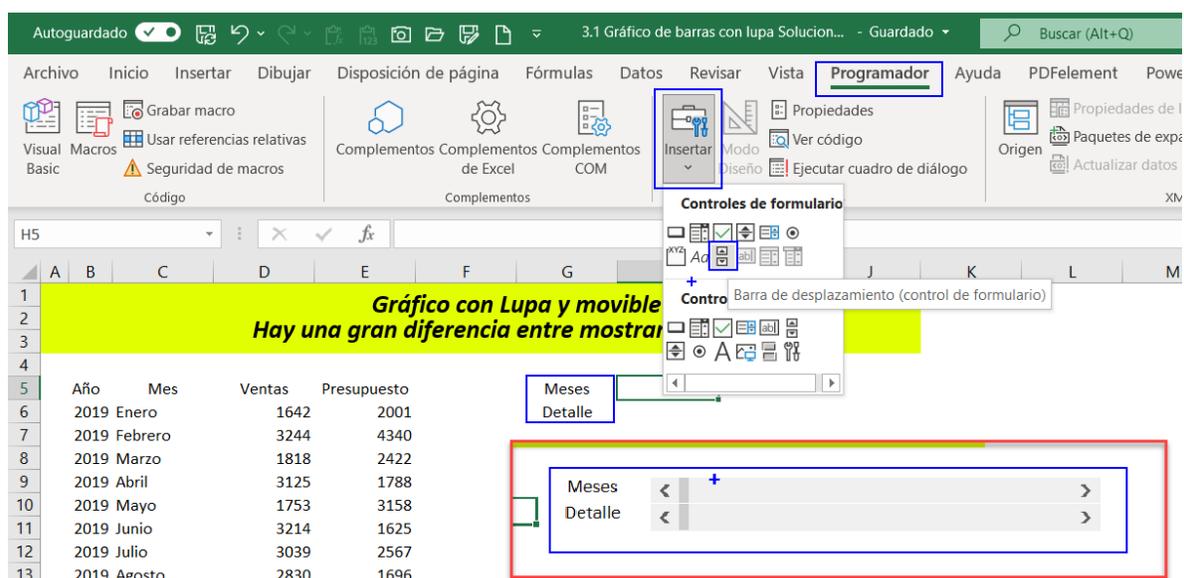


Figura 3.1 Insertar controles de formulario

Vamos a ver la cantidad de registros que tiene nuestra base de datos para configurar los controles de formulario, en este caso tenemos 24 registros.



NOTA: Los controles de formulario están disponibles desde la pestaña desarrollador o programador, para el uso de estos no es necesario que el libro este habilitado para macros.

Presionamos clic derecho sobre el control de formulario → Seleccionamos formato de control → Configuramos los parámetros de Valor mínimo: 0, Valor máximo: 24 e Incremento: 1, esto lo hacemos para las dos barras de desplazamiento insertadas.

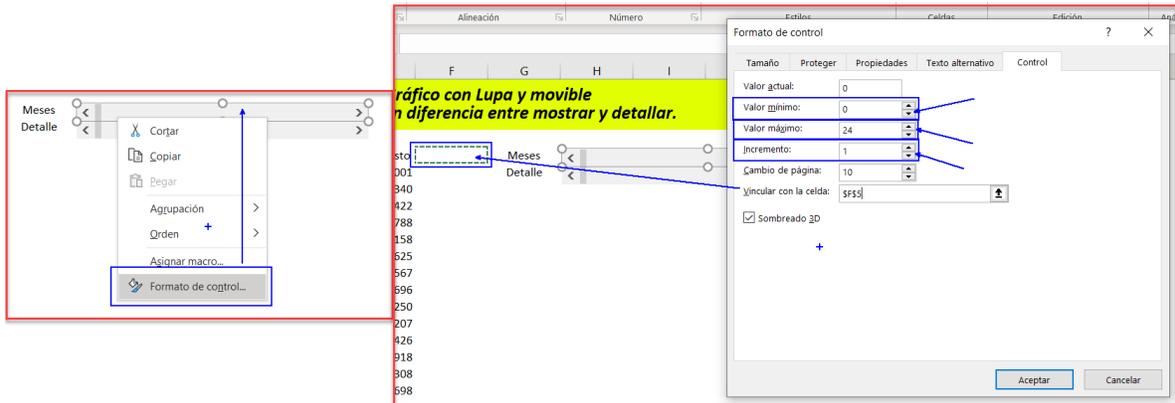


Figura 3.2 Configurar control de formulario

Vamos a insertar una nueva columna para crear un campo concatenado el cual nos servirá para ponerlo en las etiquetas del gráfico y así logramos que el nombre contenga el año y mes.

Año	Mes	Llave	Ventas	Presupuesto
2019	Enero	=C6&" - "&B6	1642	2001
2019	Febrero	Febrero - 2019	3244	4340
2019	Marzo	Marzo - 2019	1818	2422
2019	Abril	Abril - 2019	2125	1788

Figura 3.3 Concatenar campos



FUNCIÓN: =C6&" - "&B6

Vamos a seleccionar el rango de celdas con los campos Llave, Ventas y Presupuesto. Con estos y usando la opción de análisis rápido, vamos a insertar un gráfico de columnas agrupadas.

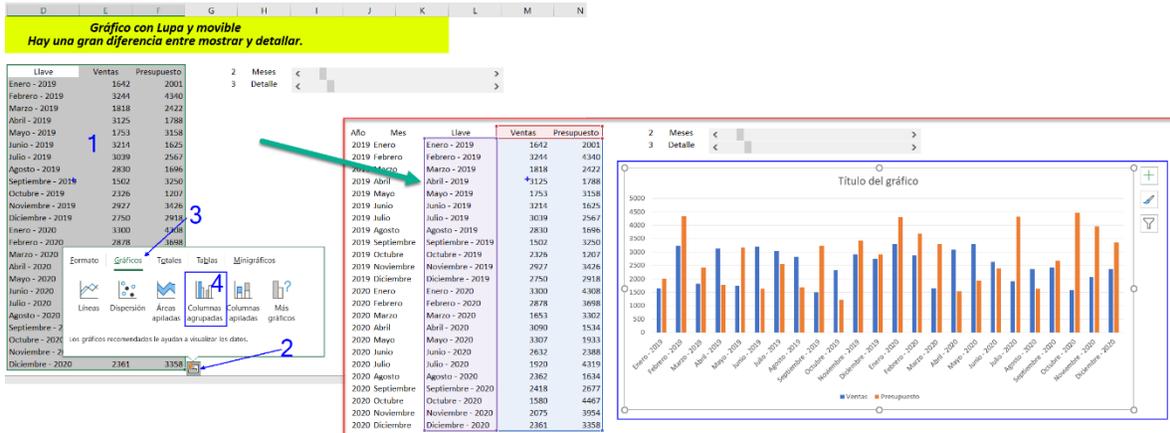


Figura 3.4 Insertar gráfico

Como podemos ver en este momento tenemos todos los años y todos los meses, lo que vamos a realizar a continuación nos permitirá visualizar un periodo de tiempo móvil.

En este momento vamos a crear los rangos dinámicos, y para esto vamos a crear nombres con la aplicación de la función DESREF.

Seleccionamos pestaña Fórmulas → Clic en Administrador de nombres → Al abrirse el cuadro de dialogo Administrador de nombres presionamos clic en Nuevo...

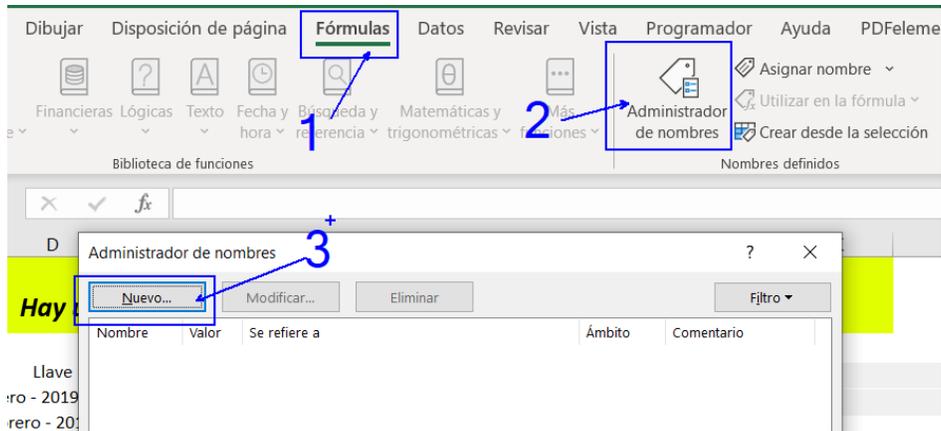


Figura 3.5 Crear nombres de rangos

Definimos el nombre del rango, seguido de esto en el campo “Se refiere a:” aplicamos la fórmula para detectar el rango de los meses de forma dinámica.



FUNCIÓN: =DESREF('1'!\$D\$6;'1'!\$G\$5;0;'1'!\$G\$6;1)

Esta fórmula se aplica para crear el rango de los meses.

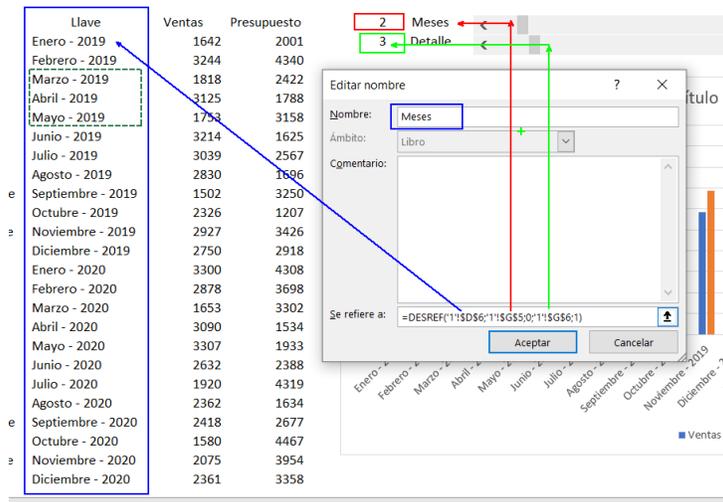


Figura 3.6 Crear rango dinámico con DESREF

Lo mismo realizaremos para los rangos de ventas y presupuesto, solo cambiamos el primer argumento de la función DESREF para el inicio de cada rango.

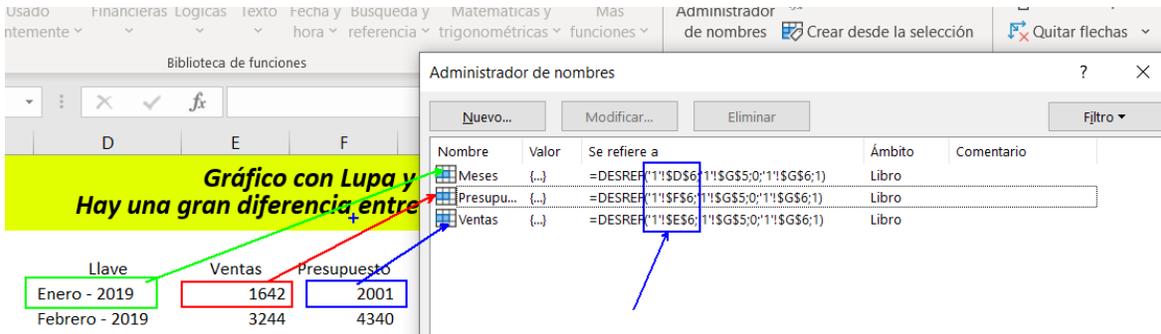


Figura 3.7 Crear rango dinámico con DESREF

Ahora para que nuestro informe, tome los nombres creados y editamos el origen de datos del gráfico con los nuevos nombres de rangos dinámicos.

Seleccionamos el gráfico → Clic en Diseño de gráfico → Clic en Seleccionar datos → Se nos muestra el cuadro de dialogo Seleccionar orígenes de datos.

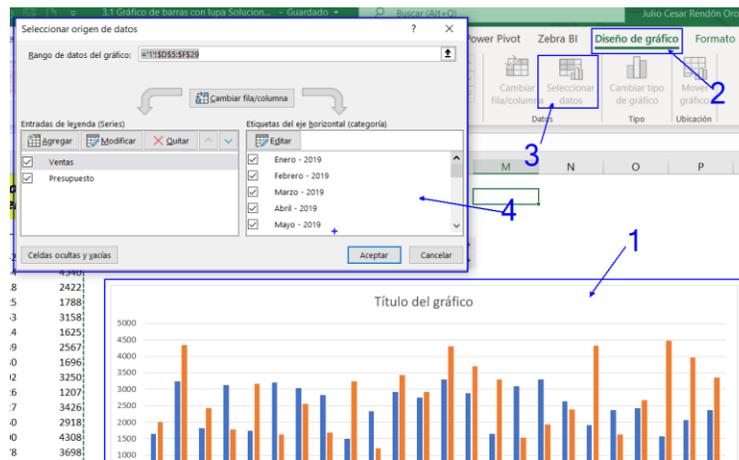


Figura 3.8 Cambiar origen de datos

Seleccionamos la serie a modificar en este caso Ventas → Clic en Modificar → Se abre el cuadro de dialogo Modificar serie, en este seleccionamos la casilla Valores de la serie y presionamos F3 para que se nos muestren los nombres de los rangos creados → Seguido de esto seleccionamos el nombre asignado al rango con los datos de la variable a modificar → Clic en Aceptar.

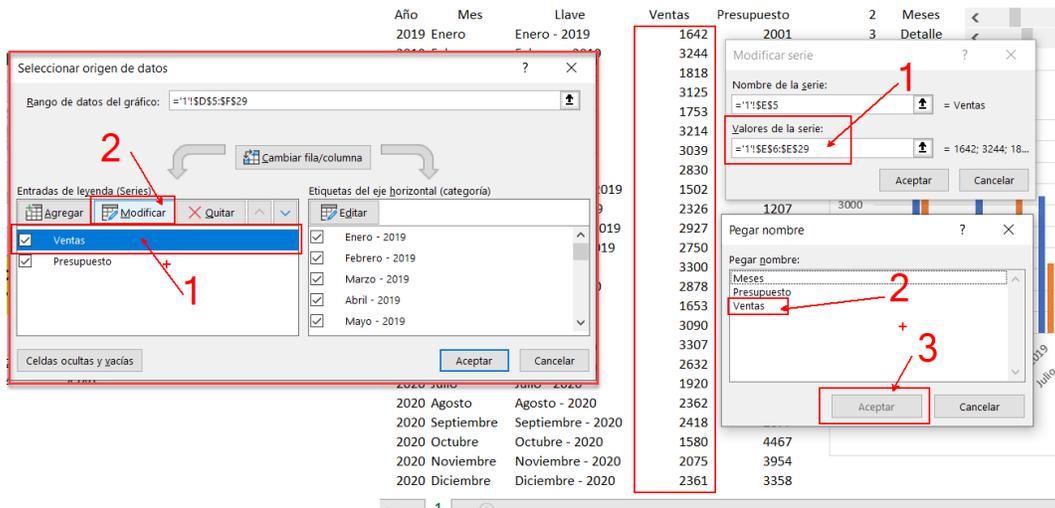


Figura 3.9 Asignar rango dinámico a gráfico



NOTA: Este mismo proceso se debe hacer para presupuesto y las etiquetas del eje horizontal.



ADVERTENCIA: Para el cambio del rango de la serie, debes tener presente que no puedes eliminar todo lo que está en esta casilla, solo debes borrar desde este signo “!” y poner el nombre dinámico. Se debe dejar el nombre de la Hoja y dar clic en Aceptar.

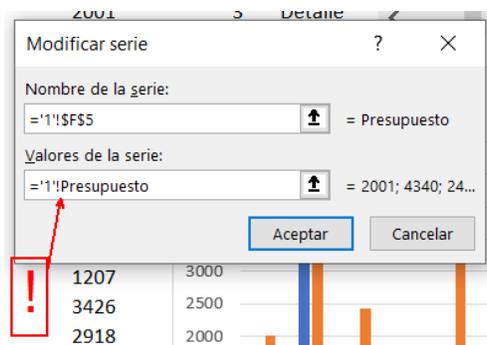


Figura 3.10 Cambiar origen de datos

Ahora con las barras de desplazamiento podemos definir qué meses y la cantidad de ellos que deseamos ver.

Gráfico con Lupa y movable
Hay una gran diferencia entre mostrar y detallar.



Figura 3.11 Informe en Excel

3.2 Anexar papeles de trabajo a informes

Al momento de entregar informes en la mayoría de las ocasiones hemos usado varios archivos para deducir o sacar las conclusiones que presentamos, estos regularmente se le llaman papeles de trabajo, son anexos que validan o respaldan nuestro informe y estas evidencias pueden ser archivos de Excel, Word, registros fotográficos o PDF, en cualquiera de los casos vamos a realizar el truco para que puedas anexarlo y quede insertado como documento dentro de tu informe y no tengas que enviarlo o compartirlo de forma independiente, adicional, el usuario final del informe solo con doble clic podrá abrir y ver este anexo.

En este caso vamos a anexar los papeles de trabajo (Archivos) a nuestro informe en el área demarcada.

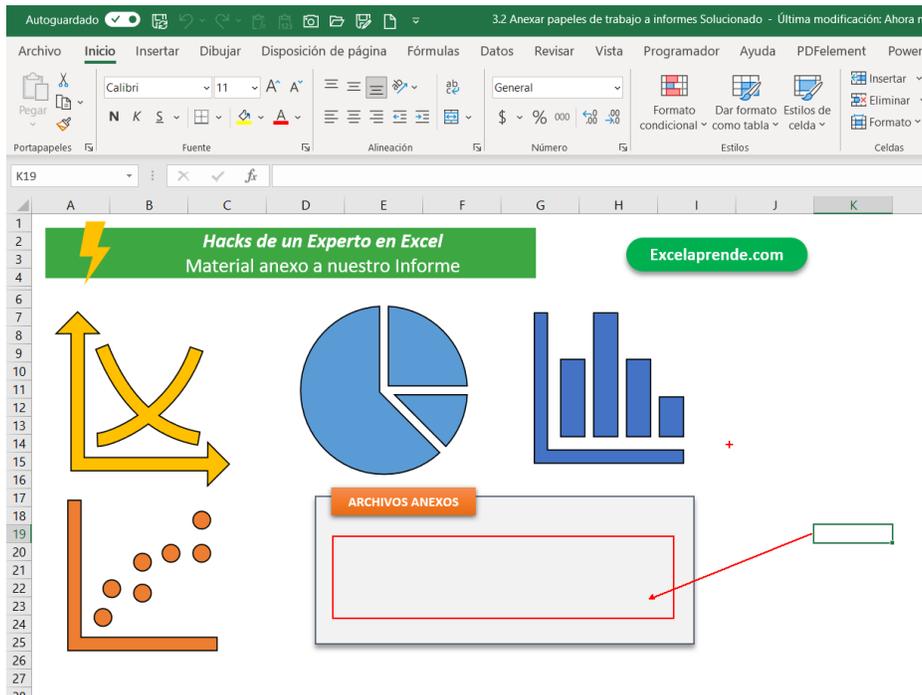


Figura 3.12 Informe para anexar archivos

Vamos a la pestaña Insertar → Comando Texto → Clic en Objeto

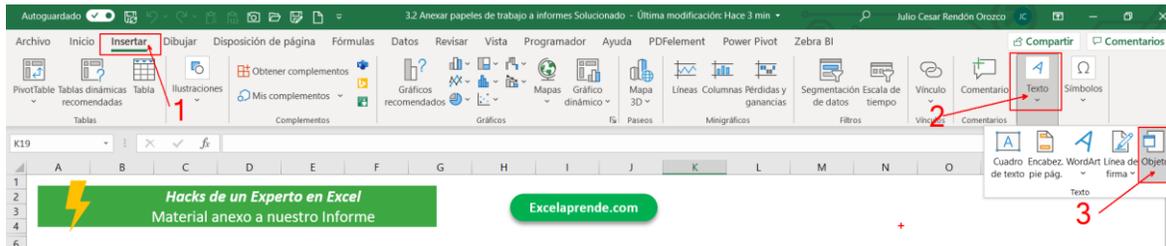


Figura 3.13 Abrir cuadro de dialogo Objeto

Seleccionamos la pestaña Crear de un archivo → Clic en Examinar → Buscamos y seleccionamos el archivo que deseamos insertar → Clic en Insertar.

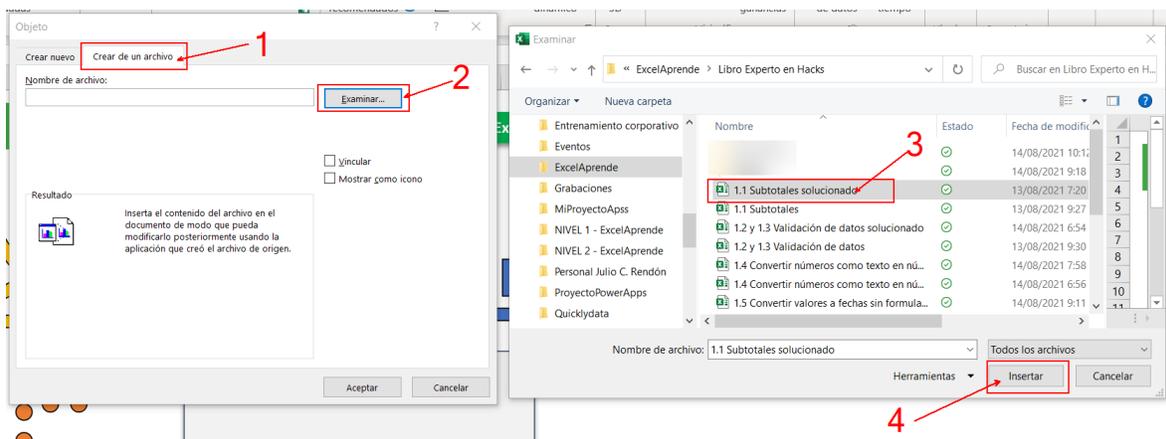


Figura 3.14 Insertar archivo para el informe

Una vez seleccionado, vamos a activar la casilla de Mostrar como icono, seleccionamos la opción cambiar icono y modificamos el nombre del archivo a mostrar en título, presionamos clic en Aceptar y Aceptar.

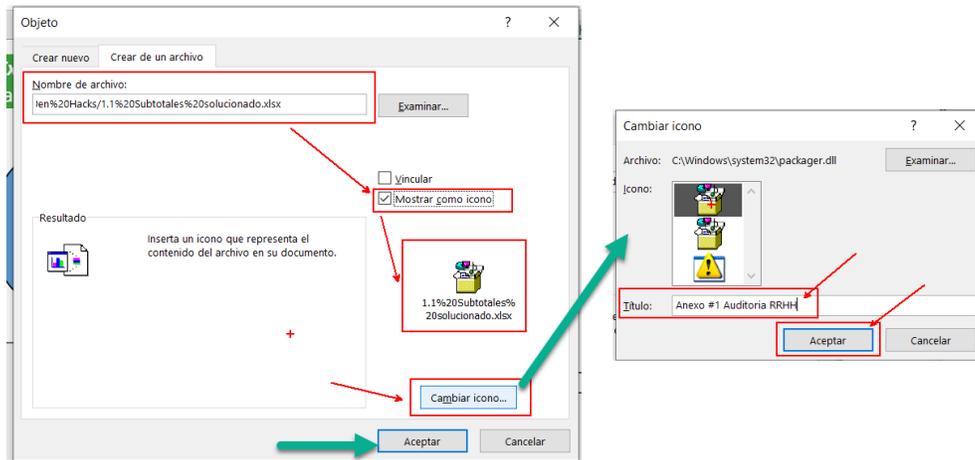


Figura 3.15 Cambiar título de archivo a insertar

Vamos a nuestro informe y ubicamos el archivo insertado, para abrirlo solo basta con dar doble clic para que este se muestre y podamos consumir la información o validar el contexto de las conclusiones de nuestro informe.



Figura 3.16 Archivo insertado como objeto

3.3 Validar en segmentación de datos la selección de dos campos

Cuando insertamos en un informe con segmentación de datos y queremos validar por ejemplo que solo se puedan seleccionar dos años, esto es algo que para muchos puede ser complejo y el truco que te vamos a mostrar a continuación permitirá que de forma fácil y práctica puedas lograrlo sin macros y sin desarrollos complejos.

En este caso encontramos la tabla con los datos, la segmentación de datos, la tabla dinámica con los valores de las ventas y las variaciones.

Vamos a controlar que solo se pueda ver la variación de dos años (periodos), para esto debemos seleccionar en la segmentación de datos dos de los años.



Figura 3.17 Informe de tabla dinámica



NOTA: En este caso vamos a usar la segmentación de año, pero puedes aplicar este truco en cualquier contexto donde requieras controlar que se permita seleccionar solo dos elementos de la segmentación de datos.

Para seleccionar dos campos en una segmentación, podemos usar control sostenido y presionar clic sobre los campos que deseamos dejar activados, en caso de que tengas disponible la opción selección múltiple la puedes activar y seleccionar también los dos campos.

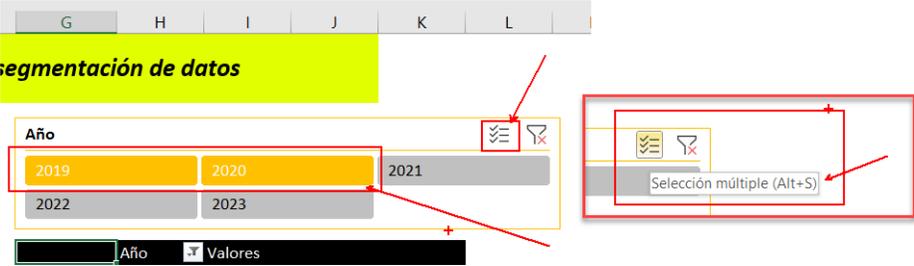


Figura 3.18 Seleccionar campos en segmentación de datos

Vamos a ocultar la columna que ha quedado en blanco para que nuestro informe se vea mucho más profesional y de esta forma podamos ver la información de los dos periodos de tiempo y la variación de una forma más limpia.

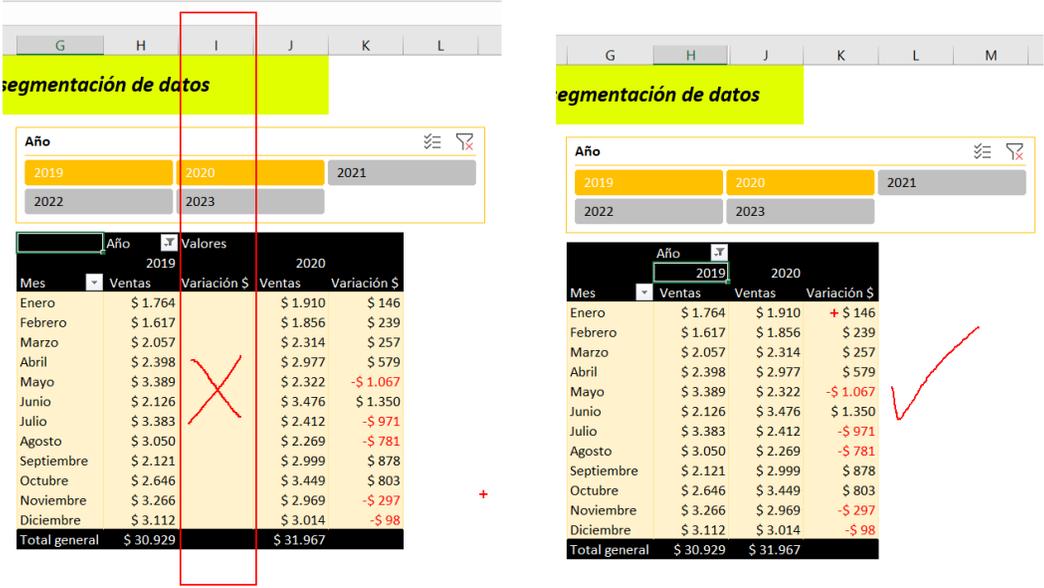


Figura 3.19 ocultar columna

Ahora tenemos nuestro informe con la estructura que estamos buscando para presentarlo de forma consistente.

Vamos a seleccionar al lado del encabezado de nuestra tabla dinámica un rango de 3 celdas, seguido de esto vamos a la pestaña inicio y presionamos clic sobre combinar y centrar, vamos a tener estas 3 celdas combinadas y nos servirá como control para evitar el desplazamiento o que el informe de tabla dinámica se pueda expandir.

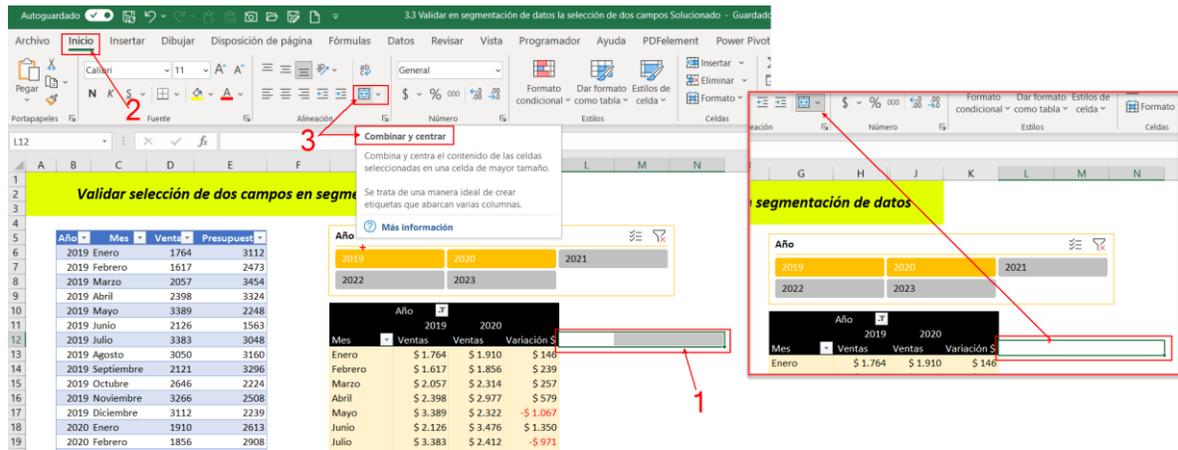


Figura 3.20 Combinar y centrar celdas

Si ahora intentas seleccionar un nuevo año, no la permitirá y en cambio te mostrara un mensaje en pantalla indicando que esta acción no se puede realizar, dado que contiene celdas combinadas y fue lo que acabamos de realizar.

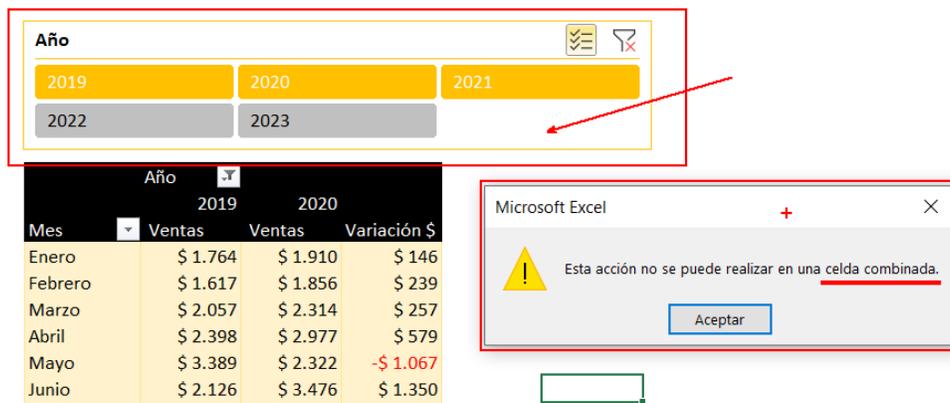


Figura 3.21 Segmentación validada

Al presionar aceptar quitará de la segmentación el periodo 3 y dejará solo dos.

Para poder cambiar de año o periodo vamos a dejar uno solo y luego seleccionar en nuevo, recuerda esto, lo puedes realizar con la opción de selección múltiple o dejando sostenido control para quitar la selección de uno de los campos y seleccionar uno nuevo.

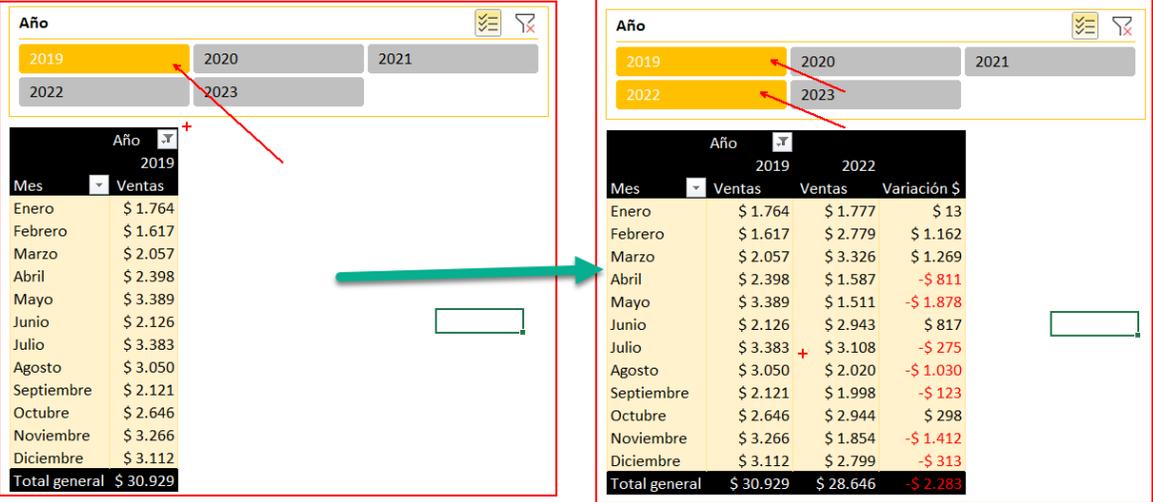


Figura 3.22 Segmentar un nuevo periodo

3.4 Resaltar cumplimiento de presupuesto en gráfico de barras

La forma más rápida y efectiva para mostrar un buen o mal resultado son los gráficos, por lo que en esta oportunidad crearemos las fórmulas y daremos la configuración a un gráfico de barras para que nos muestre en un periodo de tiempo (en meses) los que cumplen o no el presupuesto de ventas.

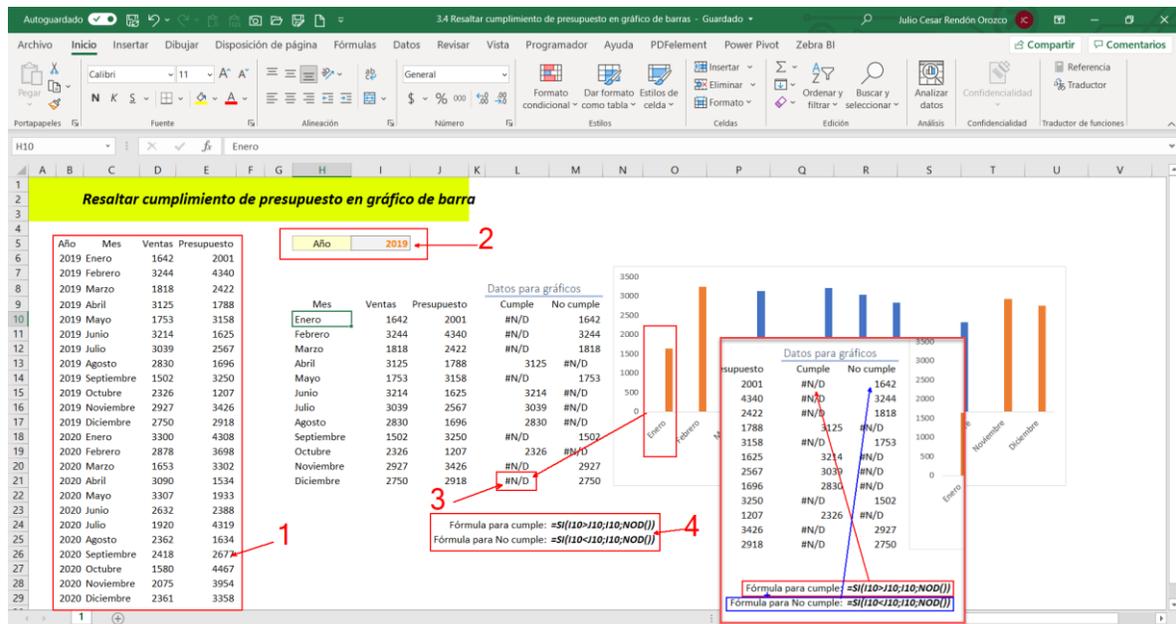


Figura 3.23 Contexto caso resaltar cumplimiento presupuesto ventas

Podemos observar los siguientes elementos:

1. Datos para evaluar los cumplimientos.
2. Lista desplegable para seleccionar años.
3. Usamos la función NOD() para generar un error a propósito donde no se cumpla el presupuesto, esto con la finalidad de que el gráfico no muestre la barra correspondiente en caso de no cumplir.

4. Usaremos la función SI para validar el cumplimiento del presupuesto con base a las ventas de cada mes.



FUNCIÓN: =SI(I10>J10;I10;NOD())

Si el valor de la venta es mayor que el valor del presupuesto, lo que buscamos es que coloque el valor de la venta en el campo Cumple para ser graficado, en caso contrario que nos muestre un error a propósito.



NOTA: Los datos seleccionados para la gráfica son los de cumple y no cumple, estos por medio de la formula aplicada nos muestran aquellos datos donde corresponden, ya sea que se ha cumplido o no el presupuesto.

Vamos a seleccionar una de las barras del gráfico, seguido de esto presionamos clic derecho sobre ella y se nos muestra el menú contextual en el que seleccionaremos Dar formato a serie de datos.



Figura 3.24 Abrir panel formato de serie de datos

Vamos a retirar los espacios de los datos que de momento están invisibles, aquellos que corresponden al resultado de la función NOD () y que no se muestran en la gráfica, distorsionan un poco la forma en que las barras se pueden ver.

Debemos ingresar el valor del 100% en el campo de Superposición de series, esto permite que las barras queden una sobre la otra. En nuestro caso al tener un solo dato para cada mes, veremos cómo queda una sola barra de color correspondiente al Cumple o No cumple mejor ubicada y centrada.

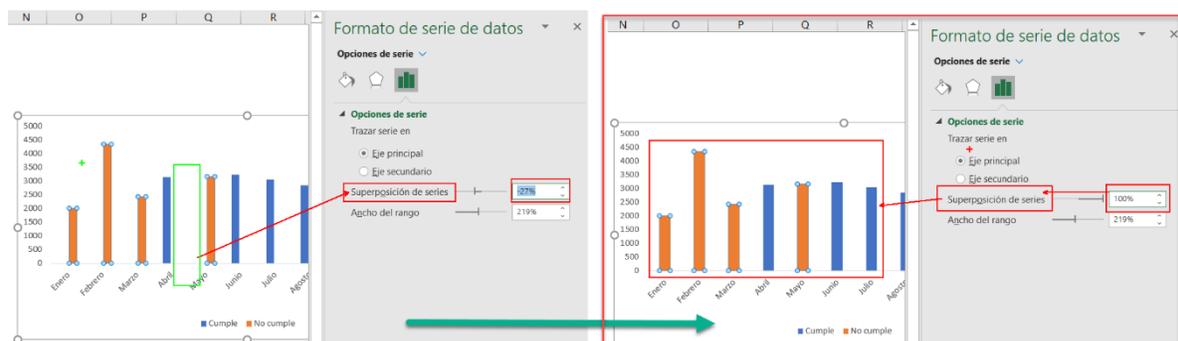


Figura 3.25 Superposición de series

Para mejorar nuestro informe gráfico, personalizaremos el Ancho del rango, en este mientras el número sea menor las barras quedarán más anchas, mientras sea más alto quedarán más delgadas, ya es cuestión de lo que mejor nos parezca para presentar nuestros datos.

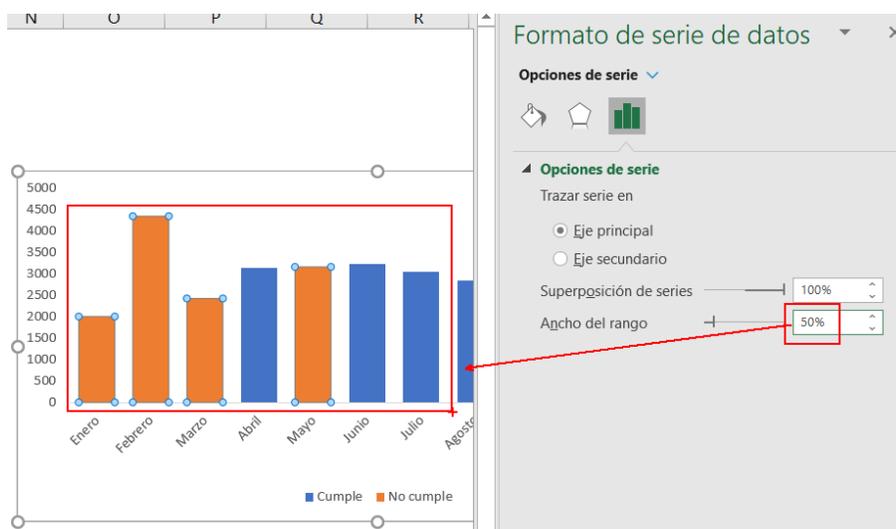


Figura 3.26 Ancho del rango

Podemos personalizar el color de las barras, solo basta con seleccionar una de las barras en este caso la de Cumple, seguido de esto ir a la opción “Opciones de serie”, en “Relleno” seleccionamos “Relleno Sólido” y escogemos el color que asignaremos.

Este mismo proceso lo aplicamos para las barras del No cumple.

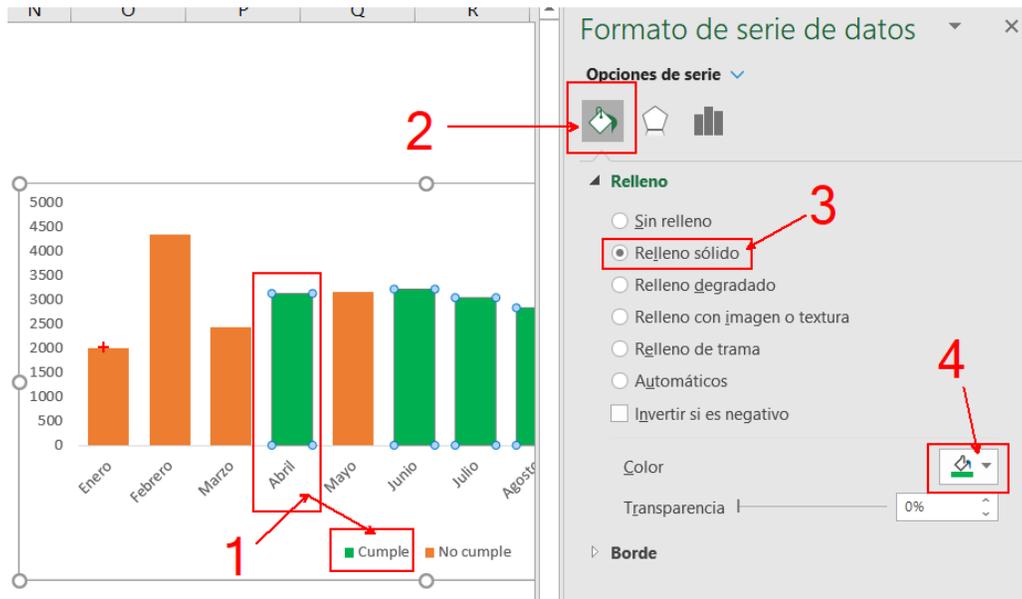


Figura 3.27 Personalizar relleno de serie

Activamos los elementos del gráfico para personalizar y dar mayor detalle a nuestro informe. En este caso activamos los títulos del gráfico y las etiquetas de datos, para esto es necesario seleccionar el gráfico y presionar clic en el + que aparece en la parte superior derecha, activamos las opciones mencionadas anteriormente y podemos ver los resultados.

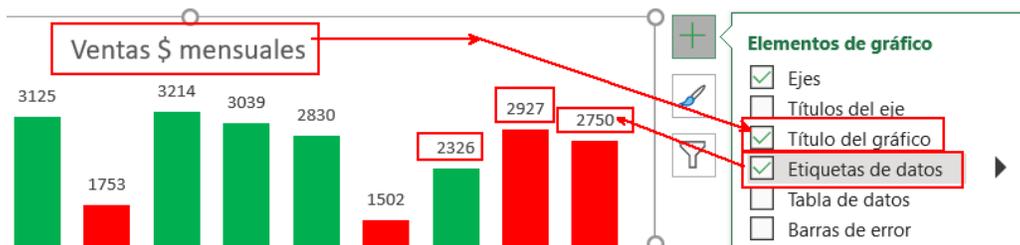


Figura 3.28 Activar elementos del gráfico

Finalmente, con este truco quedamos con un informe profesional que permite visualizar de color rojo aquellos meses en los que no se cumplió el presupuesto y en verde los que si se cumplió.

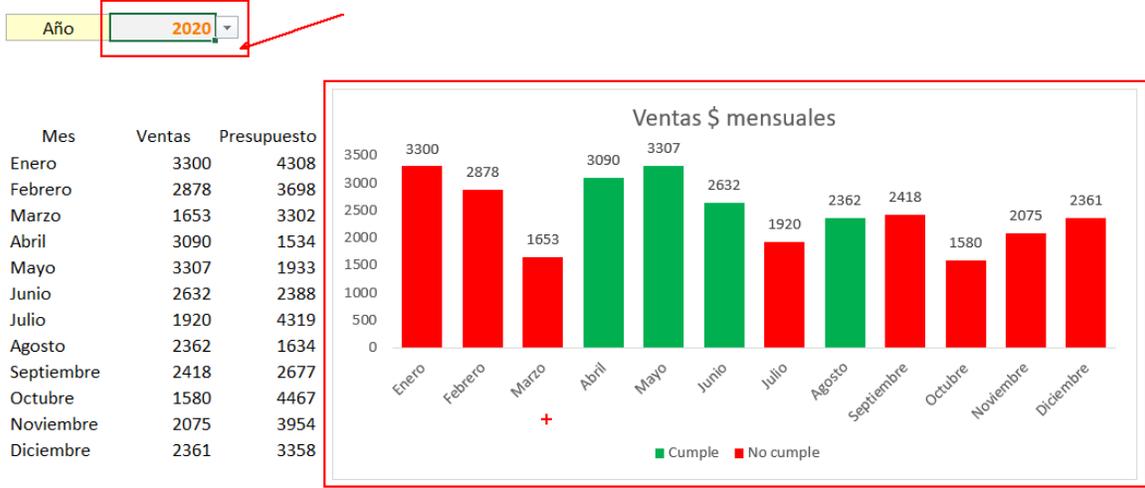


Figura 3.29 Presentar informe gráfico

Para efectos de la presentación es importante que el gráfico lo pongas sobre los campos formulados para que estos no sean visibles.

3.5 Tablas dinámicas con minigráficos

Agregaremos a un informe de tabla dinámica un elemento visible que permite un análisis rápido de la información, en este caso estamos hablando de los minigráficos.

Vamos a agregarlo de tal forma que estos parezcan parte de la tabla dinámica, en este caso usaremos el gráfico de Columnas para ver qué país es mejor en cada año, por lo que te recomendamos retirar el total para filas de la tabla dinámica.

Selecciona la celda N12 donde insertaremos el prime minigráfico → Pestaña insertar → Clic en el grupo minigráficos en Columnas, se nos mostrará el cuadro de dialogo Crear Minigráficos.

The screenshot shows the Excel interface with the 'Insertar' ribbon active. The 'Columnas' chart type is selected in the 'Minigráficos' group. A pivot table is displayed with the following data:

Año	Mes	Venta	Presupuest	País	Genero cliente
2019	Enero	1764	3112	Venezuela	Mujer
2019	Febrero	1617	2473	Colombia	Hombre
2019	Marzo	2057	3454	Argentina	Mujer
2019	Abril	2398	3324	Colombia	Mujer
2019	Mayo	3389	2248	Colombia	Mujer
2019	Junio	2126	1563	España	Hombre
2019	Julio	3383	3048	Colombia	Hombre
2019	Agosto	3050	3160	Colombia	Mujer
2019	Septiembre	2121	3296	Venezuela	Mujer
2019	Octubre	2646	2224	Colombia	Hombre
2019	Noviembre	3266	2508	España	Hombre
2019	Diciembre	3112	2239	España	Mujer
2020	Enero	1910	2613	Venezuela	Mujer

The small chart overlaid on the pivot table shows sales data for different countries (Argentina, Colombia, España, Venezuela) across years (2019-2023). The chart is a column chart with the following data:

Año	Argentina	Colombia	España	Venezuela
2019	\$ 2.057	\$ 16.483	\$ 8.504	\$ 3.885
2020	\$ 10.645	\$ 5.291	\$ 14.121	\$ 1.910
2021	\$ 5.086	\$ 8.647	\$ 6.038	\$ 8.152
2022	\$ 2.799	\$ 9.307	\$ 5.385	\$ 11.155
2023	\$ 4.592	\$ 4.062	\$ 7.208	\$ 11.363
Total gen.	\$ 25.179	\$ 43.790	\$ 41.256	\$ 36.465

Figura 3.30 Insertar minigráfico

Seleccionamos el rango de celdas para generar el minigráfico → Clic en Aceptar.

The screenshot shows the 'Crear Minigráficos' dialog box. The 'Rango de datos' field is set to 'J13:M13' and the 'Ubicación' field is set to '\$N\$13'. The 'Aceptar' button is highlighted.

Figura 3.31 Insertar minigráfico

Vamos a resaltar la barra más alta del minigráfico usando un color diferente a las demás barras.

Seleccionamos el minigráfico creado → vamos a la pestaña que se habilita Minigráfico →

Clic en Color de marcador → Clic en Punto Alto → Selecciona el color de tu preferencia.

The screenshot shows the Excel interface with the 'Minigráfico' ribbon active. The ribbon includes options for 'Color de minigráfico', 'Color de marcador', 'Puntos negativos', 'Marcadores', 'Punto alto', 'Punto bajo', 'Primer punto', and 'Último punto'. A dropdown menu for 'Color de marcador' is open, showing 'Colores del tema' and 'Colores estándar'. A red arrow labeled '1' points to a mini-chart icon in the table below. Another red arrow labeled '2' points to the 'Minigráfico' ribbon. A red arrow labeled '3' points to the 'Color de marcador' dropdown. A red arrow labeled '4' points to the 'Punto alto' option. A red arrow labeled '5' points to a color selection in the 'Colores estándar' palette.

Ventas		País			
Año	Argentina	Colombia	España	Venezuela	
2019	\$ 2.057	\$ 16.483	\$ 8.504	\$ 3.885	
2020	\$ 10.645	\$ 5.291	\$ 14.121	\$ 1.910	
2021	\$ 5.086	\$ 8.647	\$ 6.038	\$ 8.152	
2022	\$ 2.799	\$ 9.307	\$ 5.385	\$ 11.155	
2023	\$ 4.592	\$ 4.062	\$ 7.208	\$ 11.363	
Total gen	\$ 25.179	\$ 43.790	\$ 41.256	\$ 36.465	

Figura 3.32 Personalizar punto más alto de minigráficos



NOTA: Puedes personalizar el valor más bajo, el primer punto y último punto.

También puedes tomar uno de los estilos personalizados que nos ofrece Excel por defecto.

Vamos a arrastrar el minigráfico para el resto de los años y el total → seguido colocamos el título al nuevo campo → Seleccionamos el campo al lado izquierdo para tomar el formato → Clic en la pestaña inicio → Clic en copiar formato → ahora damos clic en el rango de celdas del nuevo campo y este asumirá el formato que lo hará parte del informe de tabla dinámica.

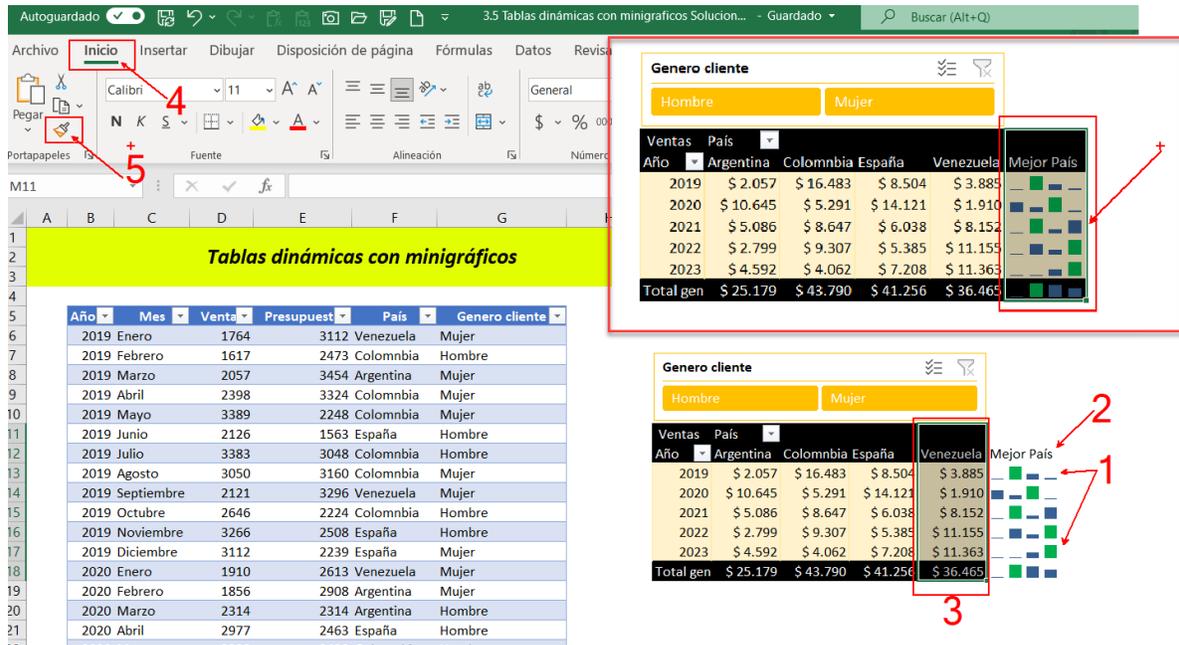


Figura 3.33 Copiar formato de celdas

Si aplicamos un filtro por medio de la segmentación de datos, veremos como los minigráficos reflejan los nuevos valores y no se muestran de forma independiente.



Figura 3.34 Aplicar filtro en segmentación

3.6 Gráfico de barras para presentar variaciones de VENTAS y PRESUPUESTO

Modificaremos la presentación estándar de los datos en un gráfico de barras, convertiremos las barras en una especie de termómetro que visualmente nos indique en vamos bien o vamos mal.

A continuación, veras el estándar y como regularmente las personas podrían presentar estos datos, vamos a agregar dos campos adicionales y las fórmulas que nos permitirán personalizar y lograr una presentación más profesional.

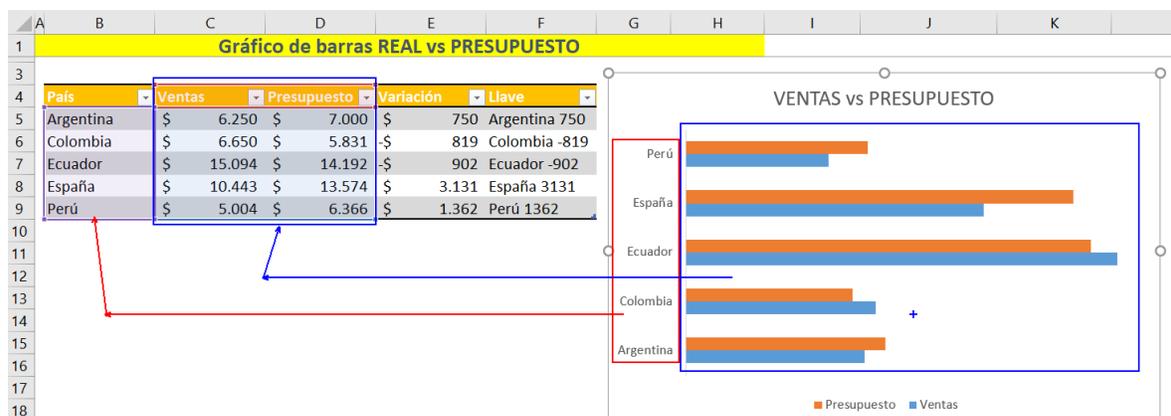


Figura 3.35 Gráfico de barras



FUNCIÓN: Variación =[@Presupuesto]-[@Ventas]

Llave =[@País]&" "&[@Variación]

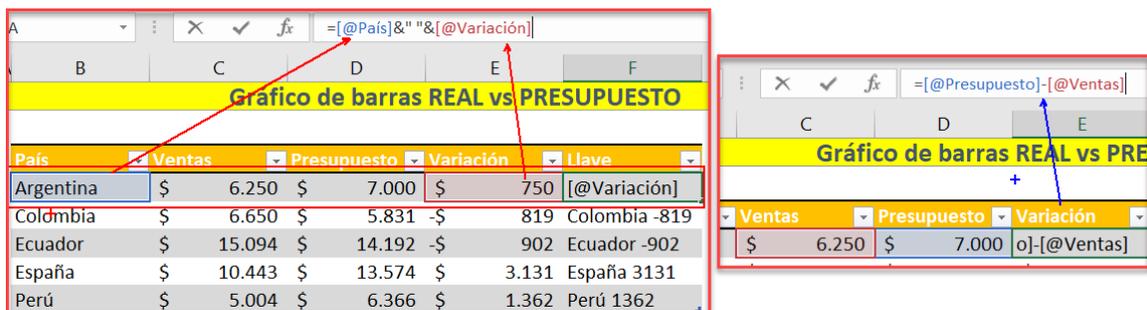


Figura 3.36 Formulas aplicadas

Te estarás preguntado para que hayamos incluido el campo Llave, este es uno de los campos fundamentales para personalizar nuestro gráfico.



NOTA: Con el campo llave vamos a personalizar los datos de la serie del eje vertical, con el cual mostraremos al lado del nombre de cada país el valor de la variación.

Seleccionamos el gráfico → Pestaña Diseño de gráfico → Seleccionar datos → En el cuadro de dialogo Seleccionar datos de Origen presionamos clic en Editar → Seleccionamos el nuevo rango e inmediatamente veremos cómo los nombres de país ahora tienen la variación concatenada.

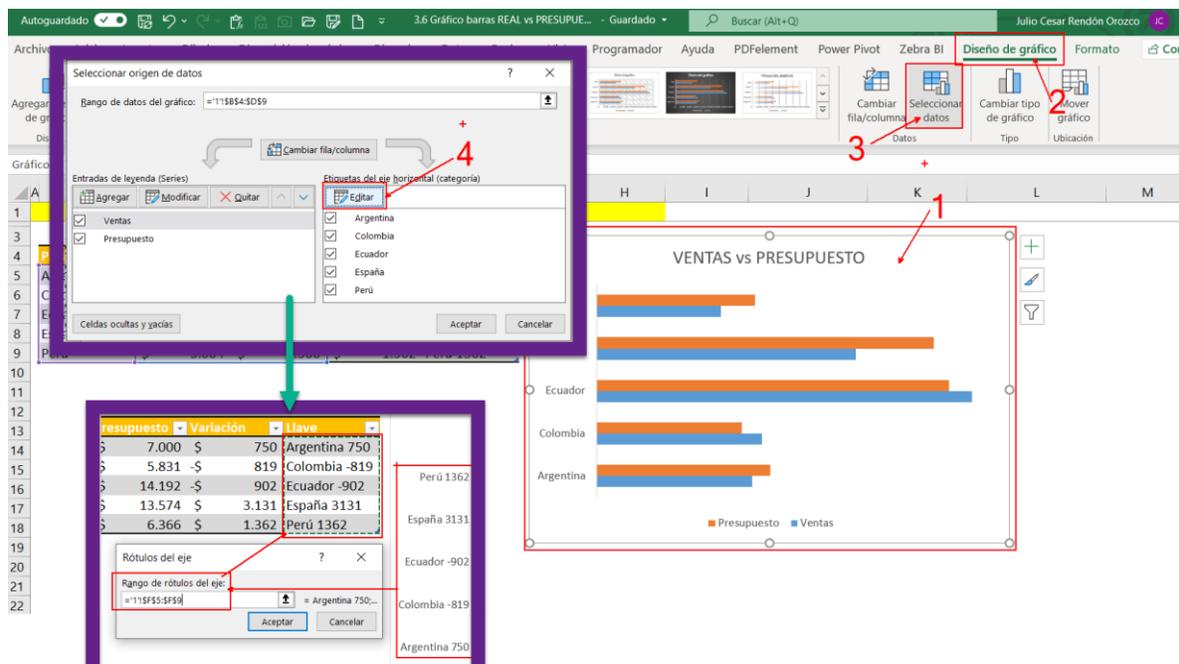


Figura 3.37 Personalizar serie de etiquetas

Vamos a abrir formato de serie de datos del presupuesto, para personalizarlo.

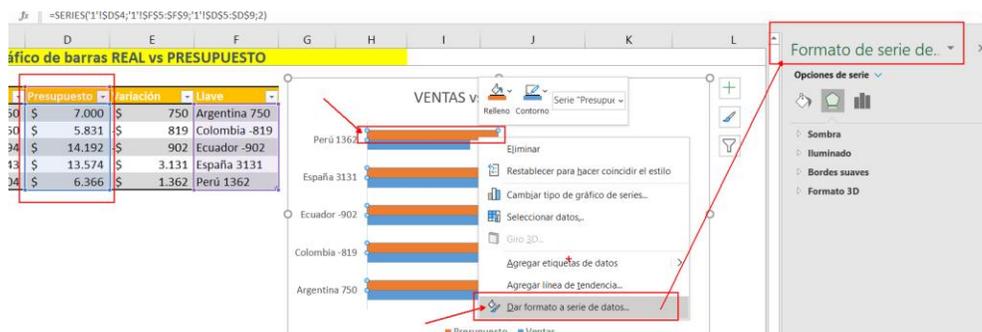


Figura 3.38 Formato de serie de datos

Seleccionamos Relleno y línea → Borde y definimos línea sólida → Seleccionamos el color del relleno que tenga la barra de presupuesto → ancho lo vamos a poner en 3 para que pueda resaltar cuando el presupuesto sea menor que las ventas.

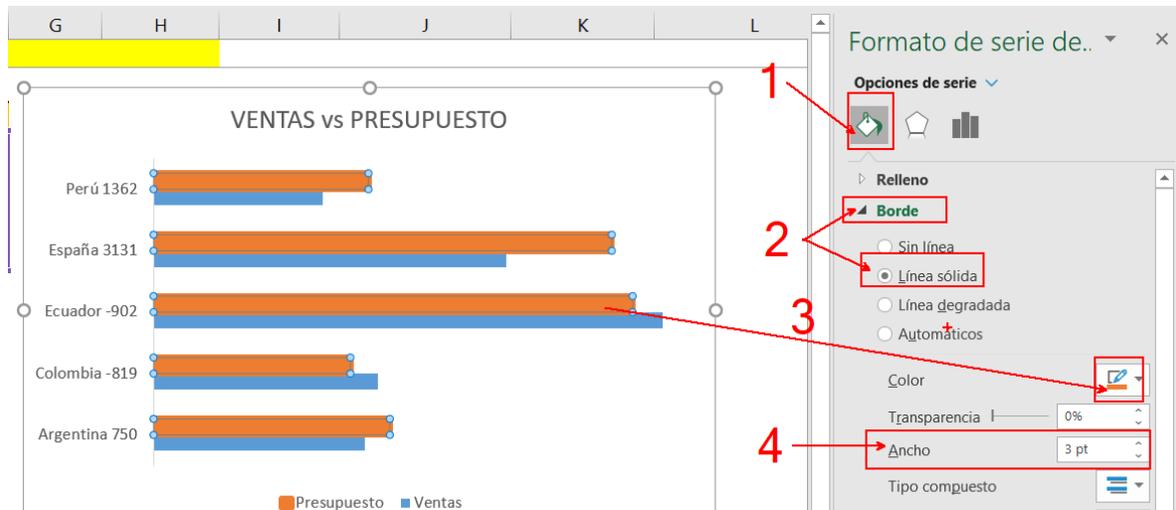


Figura 3.39 Configurar borde de barra de gráfico

Seguido de esto vamos a las opciones de serie → Super posición de serie 100%, lo que permite que tengamos una barra sobre la otra → Ancho del rango al 50% para que las barras queden más gruesas.

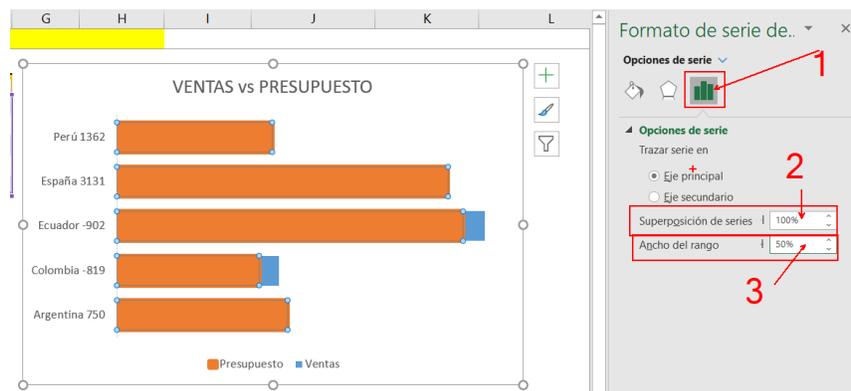


Figura 3.40 Configurar superposición y ancho del rango



ADVERTENCIA: En este momento la barra del presupuesto está por encima de la de ventas, lo que no permite visualizar bien nuestro informe, debemos proceder y ponerla debajo para que el efecto del borde más ancho nos de la visión del cumplimiento o avance con base al presupuesto.

Seleccionamos el gráfico → Diseño de gráfico → Seleccionar datos → Seleccionamos en este caso Ventas → Con los botones para bajar y subir las series vamos a garantizar que ventas quede debajo de presupuesto → clic en Aceptar → podemos ver como ahora la barra de presupuesto está por debajo, permitiendo analizar los cumplimientos.

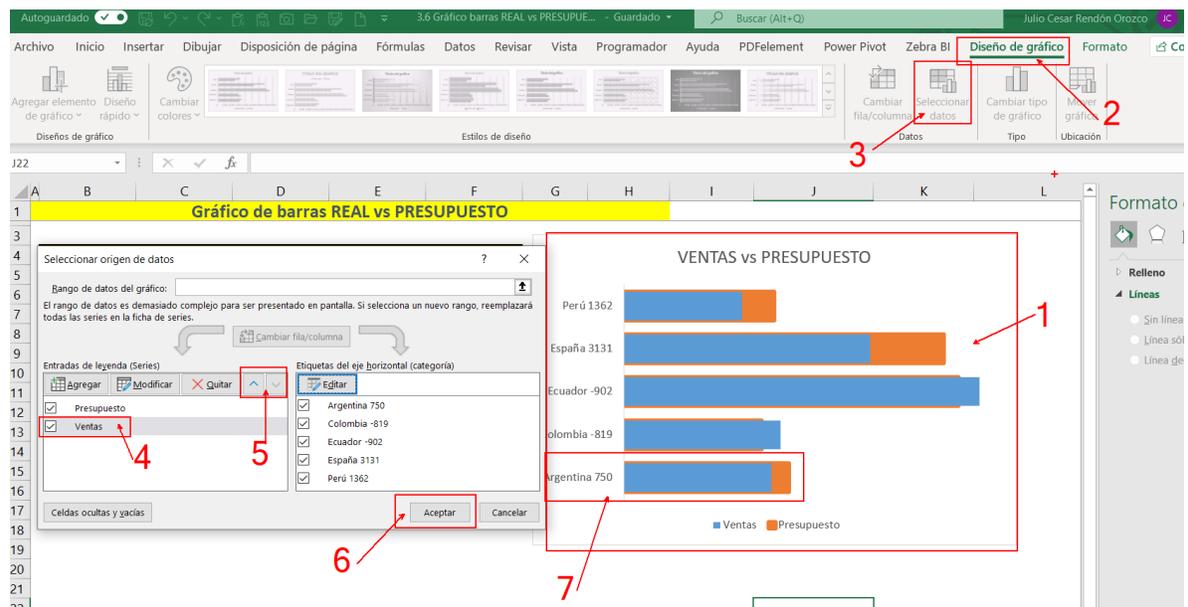


Figura 3.41 Ordenar serie de datos para gráficos

3.7 Resaltar registro con base a valor de una lista desplegable

Luego de tener un resumen de nuestros datos, queremos revisar la información por el mes el cual está en este contexto en una lista y queremos que al seleccionar un mes se pueda resaltar de un color toda la fila de registros del mes seleccionado.

En el caso a continuación se debe resaltar el mes de Oct y facilitar que podamos visualizar la información rápidamente.

Mes	Argentina	Colombia	Ecuador	España	Perú	Total general
Ene	3381	0	2640	7905	1790	15716
Feb	1552	2011	3257	1881	5238	13939
Mar	3364	3063	2868	5873	0	15168
Abr	2239	5244	4738	0	3162	15383
May	0	2525	7749	3291	1648	15213
Jun	8373	0	1571	3446	2426	15816
Jul	0	2039	5942	1989	4235	14205
Ago	0	4095	2547	2495	2617	11754
Sep	2037	3676	2609	2217	0	10539
Oct	2732	3168	4811	0	2643	13354
Nov	2448	3179	6286	0	2938	14851
Dic	0	8825	0	3492	2687	15004
Total general	26126	37825	45018	32589	29384	170942

Figura 3.42 Contexto resaltar fila del mes seleccionado

Seleccionamos el rango de celdas de nuestros registros → Pestaña Inicio → Formato condicional → Nueva Regla → se abre el cuadro de dialogo Nueva regla de formato.

Reglas para resaltar celdas

Reglas para valores superiores e inferiores

Barras de datos

Escalas de color

Conjuntos de iconos

Nueva regla...

Borrar reglas

Administrar reglas...

Figura 3.43 Aplicar formato condicional

Seleccionamos “Utilice una fórmula que determine las celdas para aplicar formato” → Ingresamos la formula $=\$B7=\$C\$4$ la cual deja como constante el valor de la celda $\$C\4 donde seleccionamos el mes a revisar y se fija la columna $\$B7$ para comparar los datos contra este mes → Aplicamos el formato que deseamos ver reflejado en los registros → Clic en Aceptar.

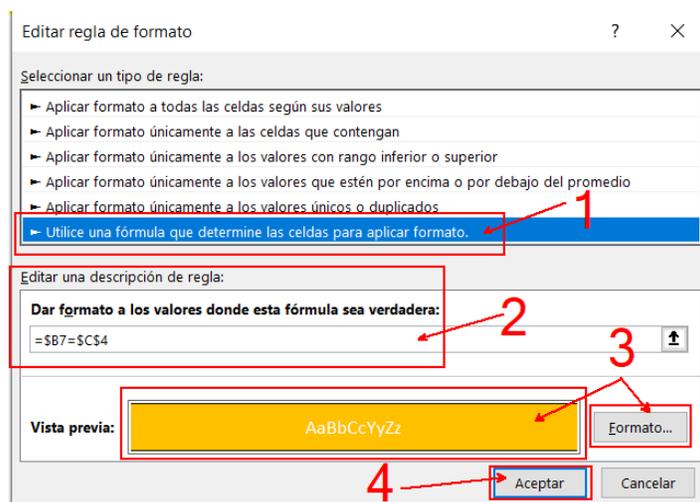


Figura 3.44 Aplicar formato condicional

Ahora podemos ver como se resalta el registro para el mes seleccionado con el formato que hemos configurado con el formato condicional.

Mes +	Argentina	Colombia	Ecuador	España	Perú	Total general
Ene	3381	0	2640	7905	1790	15716
Feb	1552	2011	3257	1881	5238	13939
Mar	3364	3063	2868	5873	0	15168
Abr	2239	5244	4738	0	3162	15383
May	0	2525	7749	3291	1648	15213
Jun	8373	0	1571	3446	2426	15816
Jul	0	2039	5942	1989	4235	14205
Ago	0	4095	2547	2495	2617	11754
Sep	2037	3676	2609	2217	0	10539
Oct	2732	3168	4811	0	2643	13354
Nov	2448	3179	6286	0	2938	14851
Dic	0	8825	0	3492	2687	15004
Total general	26126	37825	45018	32589	29384	170942

Figura 3.45 Tabla con formato condicional

3.8 Expresar cifras en los informes en miles y millones

Cuando presentamos informes y las cifras se requieren que sean expresadas en miles o millones y deseamos conservar el valor inicial, pero en cambio definir por medio de formato que se pueda visualizar en lo que definamos como miles o millones.

Esto es posible gracias al formato de celda y en esta oportunidad aprenderemos como podemos aplicarlo y sacar provecho de este para lograr este efecto visual en los valores de nuestros informes.

Seleccionamos las celdas donde queremos aplicar el formato de celdas personalizado → presionamos el atajo Control + 1 del teclado alfanumérico y se nos abre el cuadro de dialogo formato de celdas → Seleccionamos Personalizada → buscamos el formato \$ #.##0;-\$ #.##0 el cual vamos a tomar como referencia para llevar los valores a la expresión que buscamos → vemos la Muestra como esta de momento → procedemos a cambiarla.

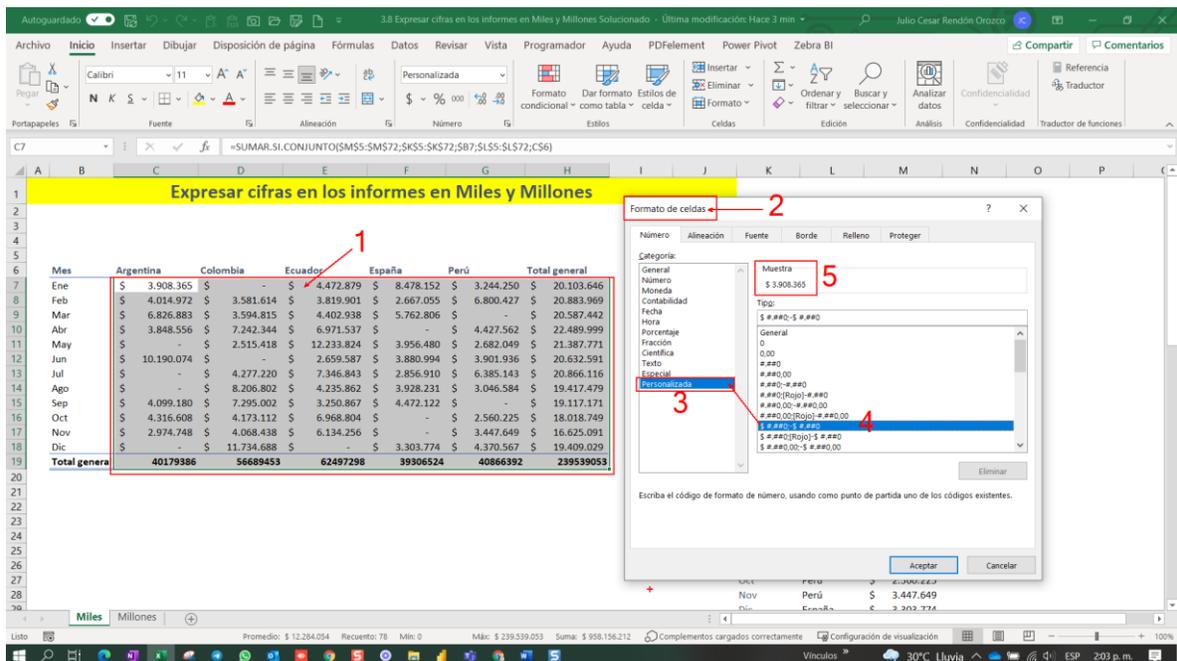


Figura 3.46 Formato de celdas

Solo basta con aplicar **\$ #.##0. "K";-\$ #.##0. "K"** para que las cifras se expresen en Miles, al agregar un punto (.) y la letra "K" tenemos el resultado que vemos a continuación.

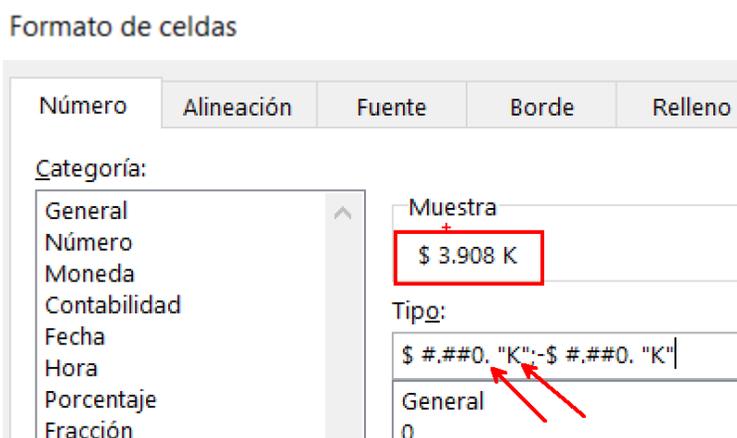


Figura 3.47 Formato personalizado para Miles

Para ver las cifras expresadas en millones, el formato que debemos aplicar es

\$ #.##0.. "M";-\$ #.##0.. "M"

Si queremos que el valor quede con decimales **\$ #.##0,00.. "M";-\$ #.##0,00.. "M"**

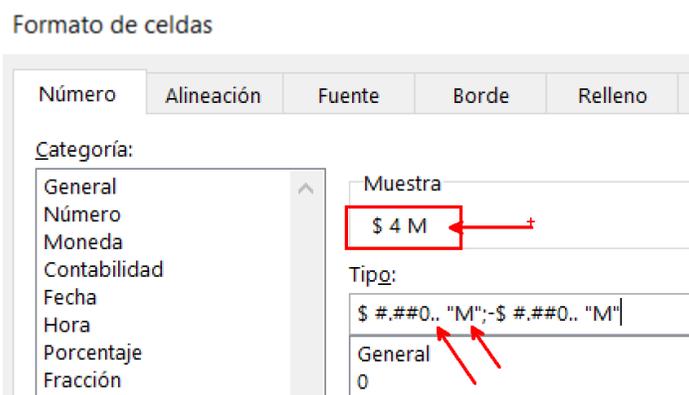


Figura 3.48 Formato personalizado para Millones

3.9 Copiar Estilos de segmentación de datos de un informe a otro

Al momento de crear informes personalizados y detallados creamos elementos personalizados, entre ellos la segmentación de datos por lo que, si queremos ese mismo estilo en otro informe y no gastar el tiempo que ya dedicamos a crear esta primera y solo pasarlo de un libro a otro, debes aplicar este truco.

Revisemos nuestro libro CON ESTILO y vemos que tenemos una segmentación personalizada y es la que deseamos pasar al nuevo informe.

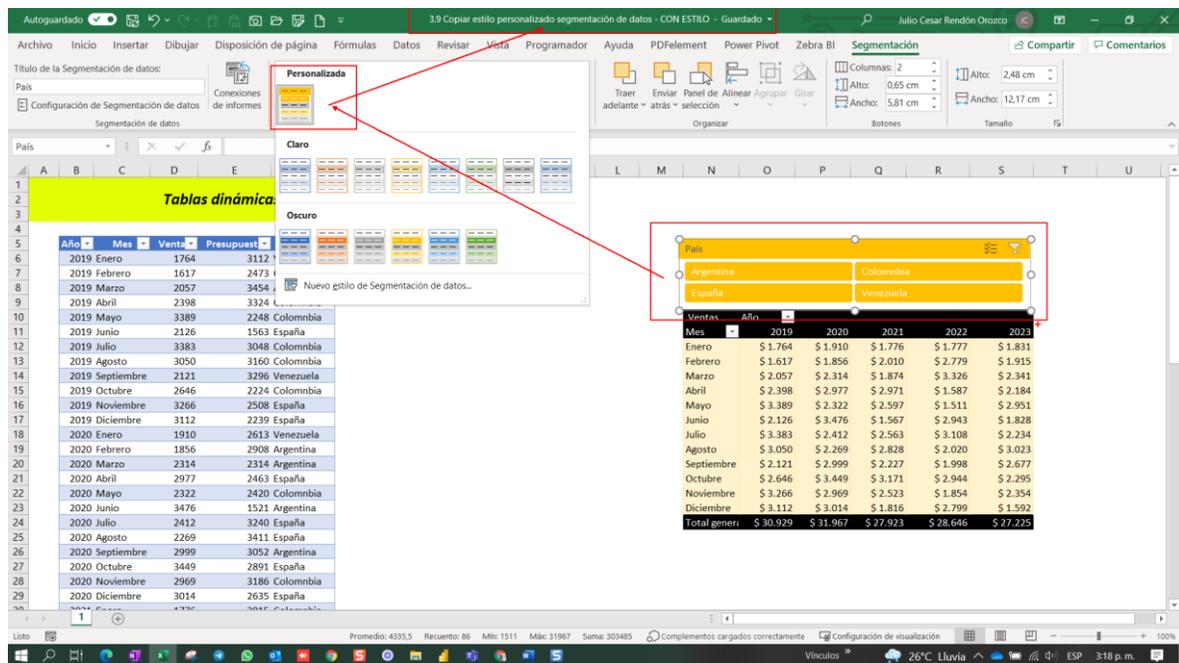


Figura 3.49 Segmentación de datos personalizada

Como podemos ver en este nuevo libro no contamos con un estilo personalizado para nuestra segmentación de datos.

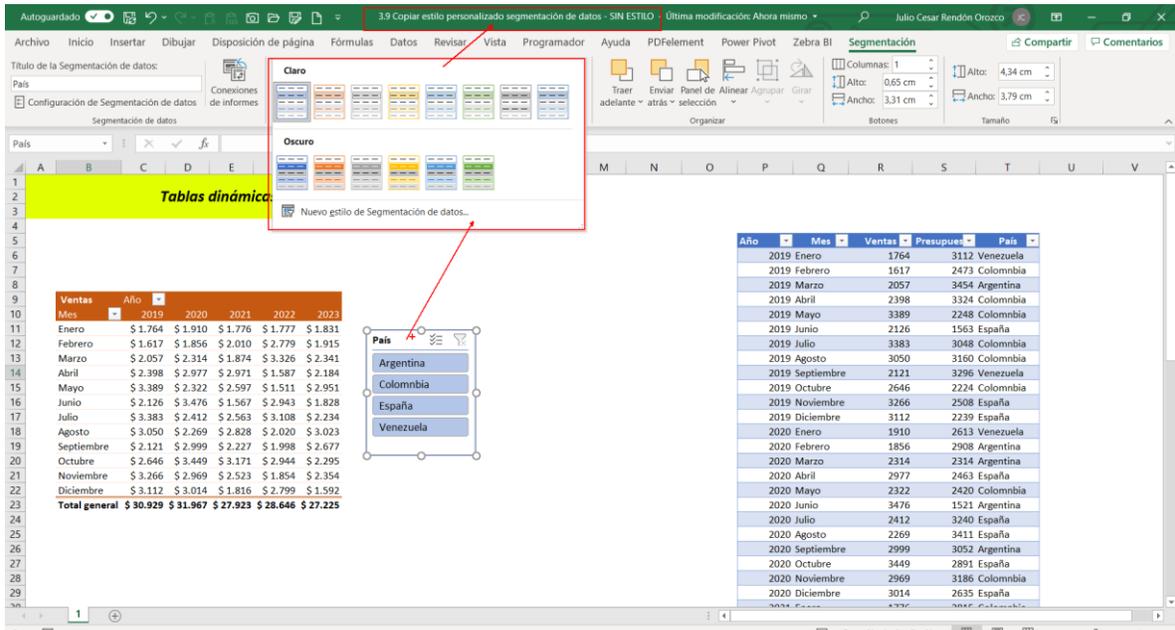


Figura 3.50 Segmentación de datos estándar

Debes ir al libro con la segmentación personalizada, copiarla e ir al libro donde la deseas llevar y pegarla.

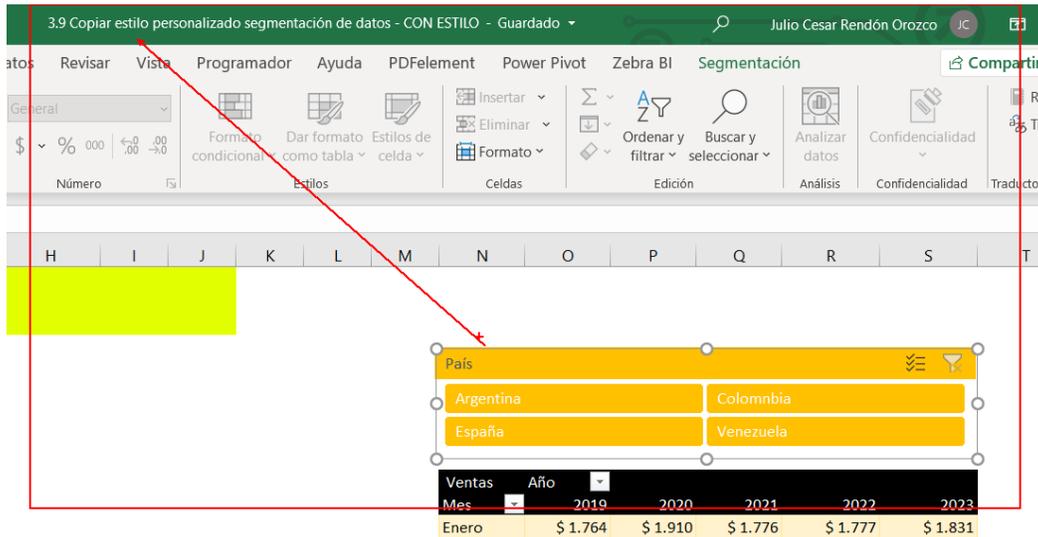


Figura 3.51 Copiar segmentación de datos

Al momento de pegarla vamos y revisamos que se nos muestre el estilo personalizado y procedemos a borrar la segmentación que nos ayudó a importar el estilo.

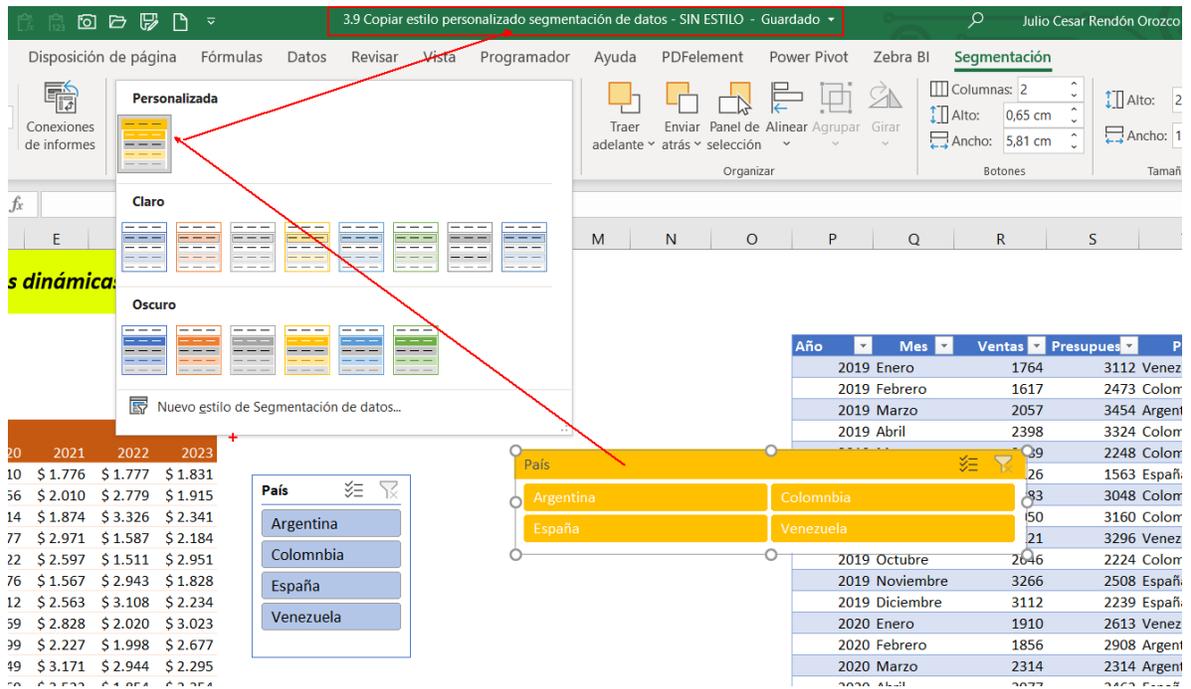


Figura 3.52 Copiar segmentación de datos

Ya podemos usarla para la segmentación del informe nuevo.

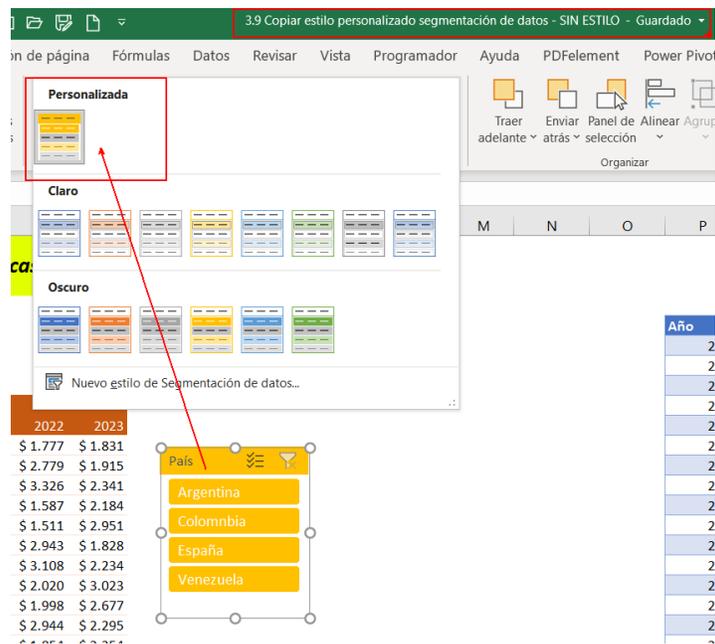


Figura 3.53 Usar estilo personalizado de segmentación de datos

3.10 Gráficos premium para ahorrar tiempo

En el transcurrir de los años hemos desarrollado decenas de informes y encontrado múltiples herramientas que facilitan la construcción de estos, por eso el truco en esta ocasión es recomendarte una de las herramientas que sin duda alguna ofrecen grandes características y de la cual somos Partner.

<https://zebrabi.com/partners/dataice/>

Ahora vamos a la acción y sorpréndete de como con pocos clics podemos generar un informe de alto impacto.

Seleccionamos el rango con los datos de ventas y presupuesto → Pestaña Zebra BI → Desviación → Actual más varianza integrado → Columna →

Con solo 4 clics observa el resultado, un gráfico donde podemos ver el comportamiento y variaciones.

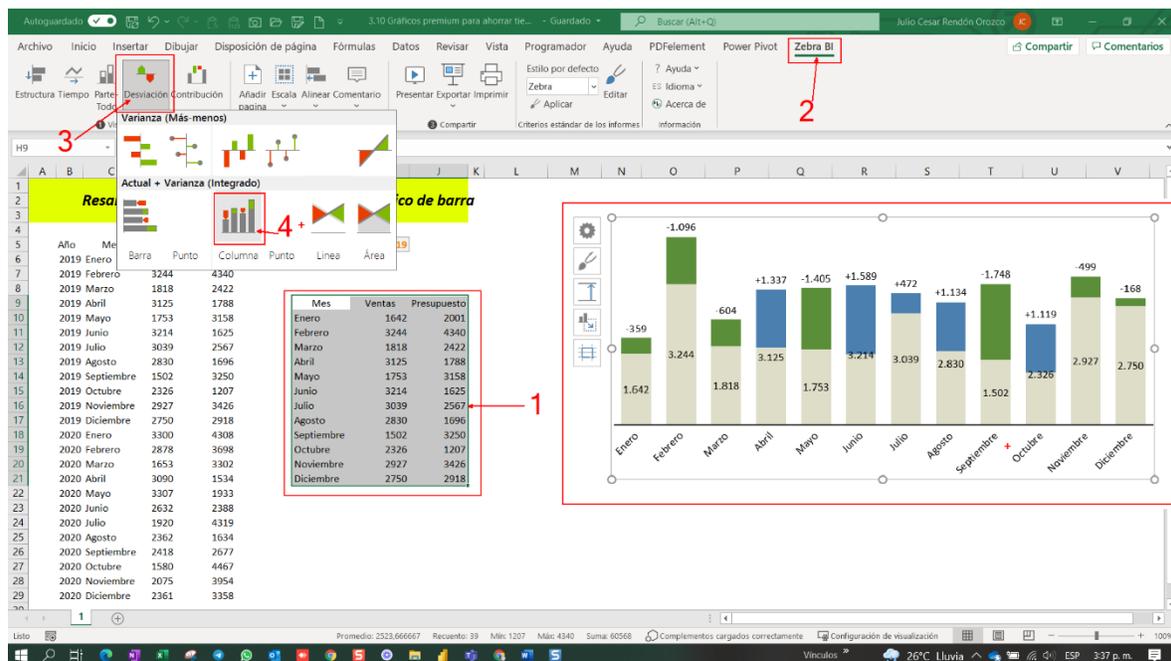


Figura 3.54 Zebra Excel

Podemos personalizarlo para darle un toque personal a nuestro estilo y con solo 4 clics más el resultado es impactante y sorprendente.

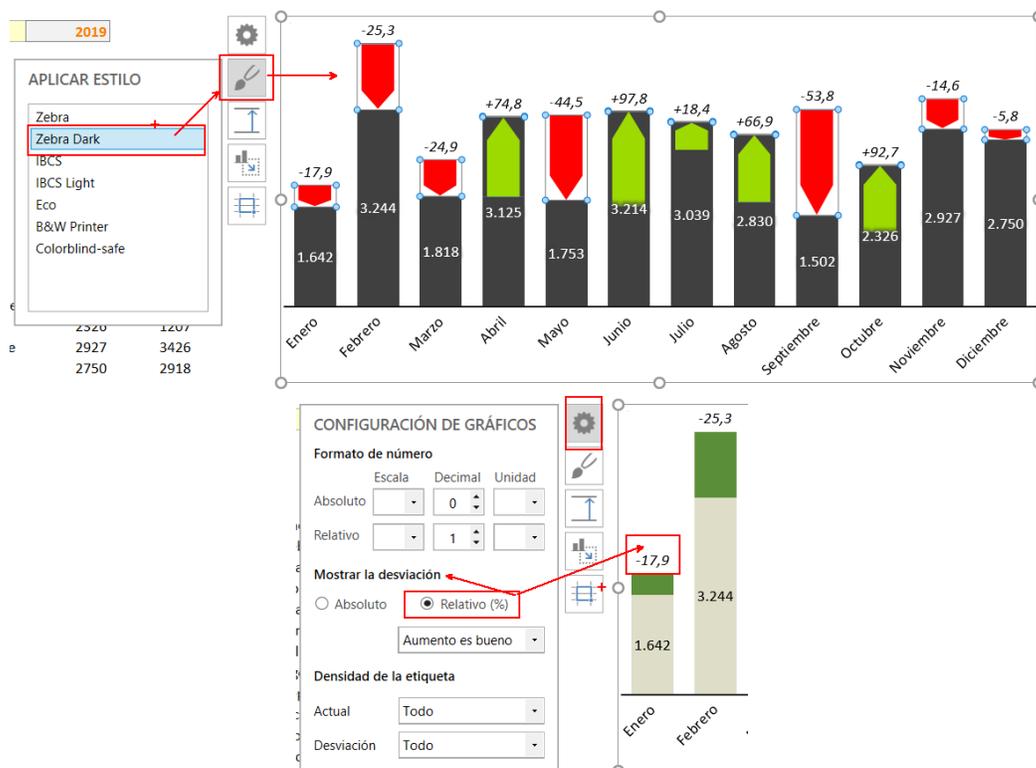


Figura 3.55 Personalizar gráfico de zebra

Capítulo 4: Excel VBA

Visual Basic for Applications es un lenguaje de programación de Microsoft que nos permite crear las macros para nuestros proyectos en Excel, lo que aprenderemos en este capítulo es como usar trucos que los desarrolladores de VBA utilizamos para optimizar nuestros procesos, tiempos, trabajos y mucho más al momento de crear desarrollos y/o aplicaciones que lleguen a otro nivel en tiempos, organización, efectividad y optimización de procesos.



NOTA: Visual Basic for Applications se incorporó con su primera versión desde 1992, de ahí realiza mejoras hasta la versión 6.0 lanzada en 1998, Microsoft le dio soporte a esta misma versión hasta el 2008 y actualmente es la que se sigue usando hoy en día.

El contenido que verán durante este capítulo tendrá como objetivo mejorar sus habilidades al momento de realizar desarrollos y aprender trucos de VBA que permitan mejorar su rendimiento y calidad de sus trabajos.

Para ingresar a Visual Basic realizamos la siguiente acción:

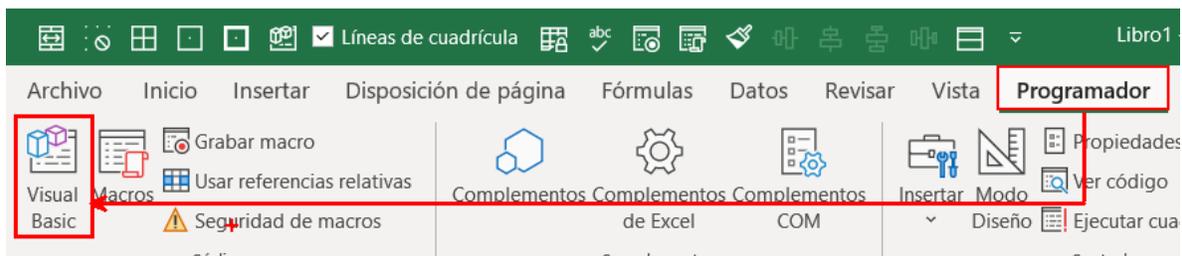


Figura 4.1 Abrir Visual Basic

Pestaña programador → Visual Basic

Si no tienes la pestaña programador activa, realiza la siguiente acción:

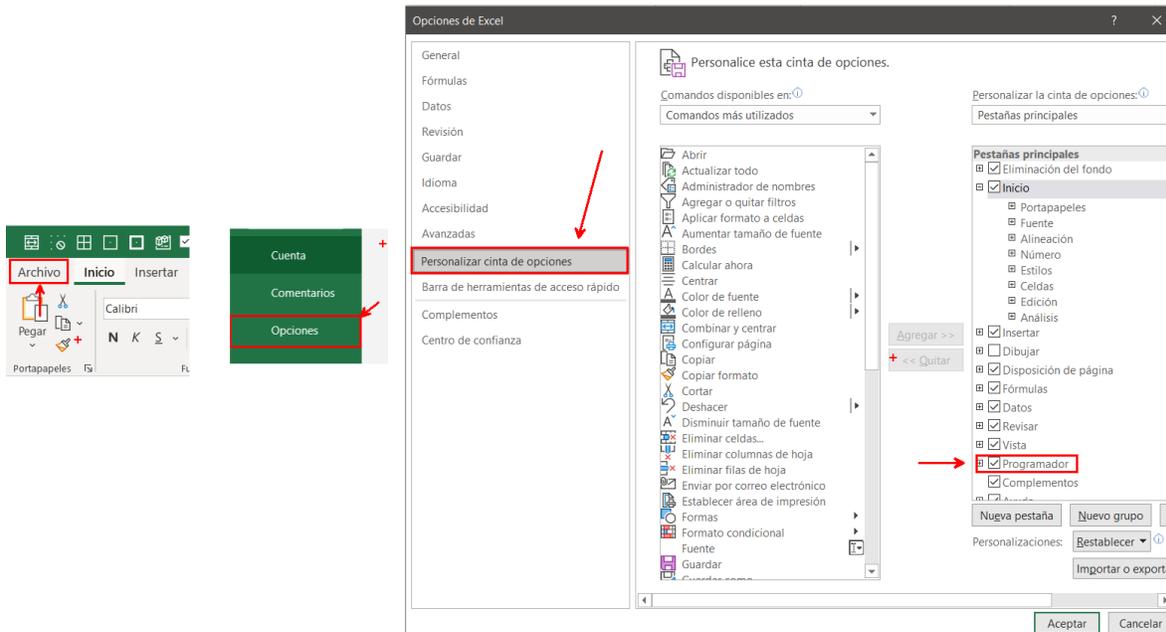


Figura 4.2 Activar pestaña Programador

Archivo → Opciones → Personalizar cinta de opciones → Activa programador

Recuerda que según tu versión de Excel la puedes encontrar con otro nombre, ya sea Programador o desarrollador.

4.1 Duplicar macros entre proyectos método 1

Durante los proyectos que realizamos en VBA, es común tener que repetir macros, es por ello por lo que muchos de los que trabajamos desarrollos buscamos la manera de optimizar los tiempos de trabajo a través de pequeños trucos que nos permite la interfaz de Visual Basic.

En este primer truco veremos cómo replicar módulos solo arrastrando de un proyecto a otro, este podría ser el método más rápido siempre y cuando tengamos el archivo de origen con la macro y el archivo destino donde vamos a pasar la macro.



NOTA: los archivos para este truco lo encontrarán en “carpeta 4 - Archivos de VBA.”

Nos dirigimos a “Carpeta 4 - Archivos de VBA” → Truco 1 - Duplicar macros entre proyectos método 1, luego abrimos el libro llamado “Libro 1 - Origen” y abrimos también “Libro 2 - Destino”.

Una vez abiertos los libros, ingresamos a Visual Basic y miramos nuestra ventana de proyectos para ver el módulo que deseamos replicar de forma rápida y fácil.

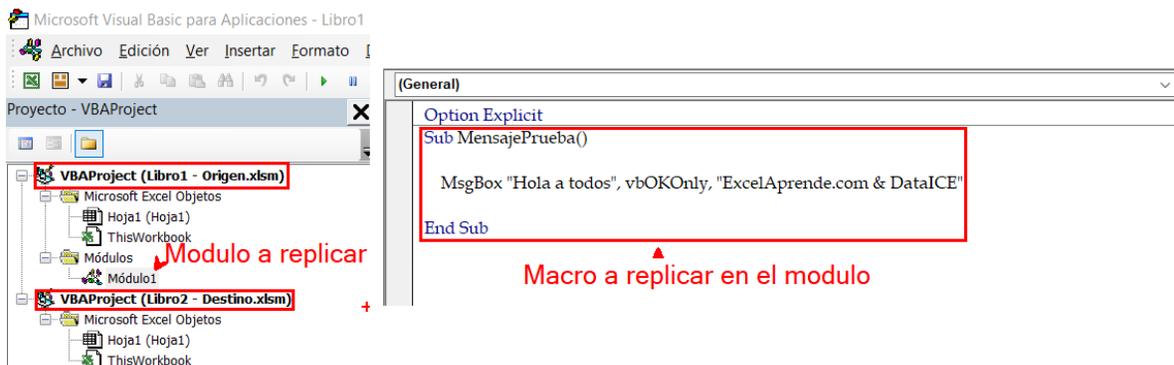


Figura 4.3 Libros en Visual Basic y módulo con macro a replicar

Una vez abierto visual con nuestros dos libros los cuales vamos a usar para el ejercicio, vamos a darle clic sostenido al módulo que deseamos replicar del libro 1 y lo vamos a arrastrar al libro 2 para que este se copie con la macro en el destino.

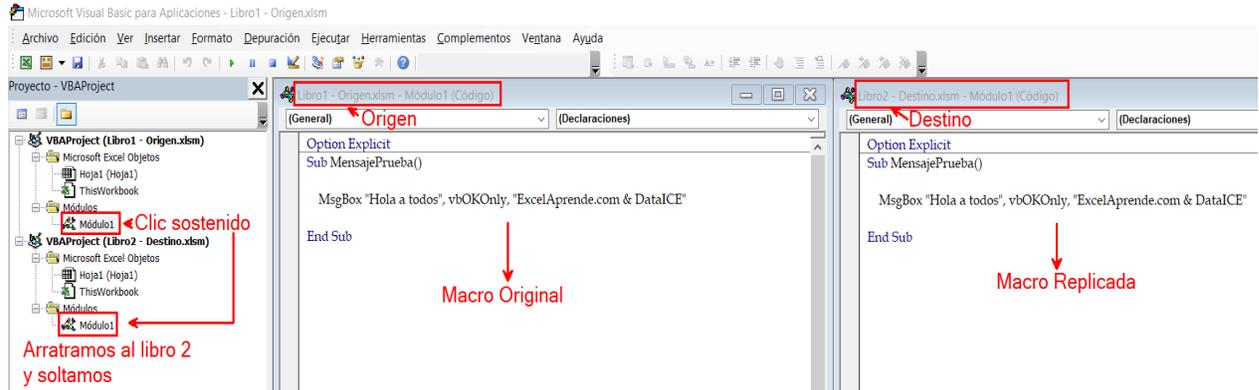


Figura 4.4 Módulo con macro replicada



HACKS: Una vez realizada esta acción, tendremos como resultado una copia del módulo con todas las macros que ya tenía creadas dentro de este, permitiéndonos simplificar tiempos de redacción en los códigos y errores.

Puedes hacer esto también con los formularios y este se duplicará con todos los objetos que tiene internamente y sus macros.

Este es un truco muy eficaz para no tener que escribir nuevamente y solo a través de una acción puedes lograr tener de manera instantánea la información que necesitas copiar en nuevos proyectos.

4.2 Duplicar macros entre proyectos método 2

En este truco veremos cómo replicar nuestros códigos cuando no tenemos el archivo de origen como en el truco pasado, por lo que aprenderás cómo exportar tus macros como archivos, esto nos permite guardar las macros para poder reutilizarlos en otros proyectos sin necesidad de estar en nuestro computador o con desarrollos que contengan las macros para replicar, brindándonos mayor flexibilidad para poder llevar nuestros códigos a cualquier lugar sea en USB, una nube, disco externo, etc...

Para este ejercicio iremos a nuestra segunda carpeta llamada “Truco 2 - Duplicar macros entre proyectos método 2” y vamos a abrir ambos libros nuevamente y nos dirigimos a Visual Basic.

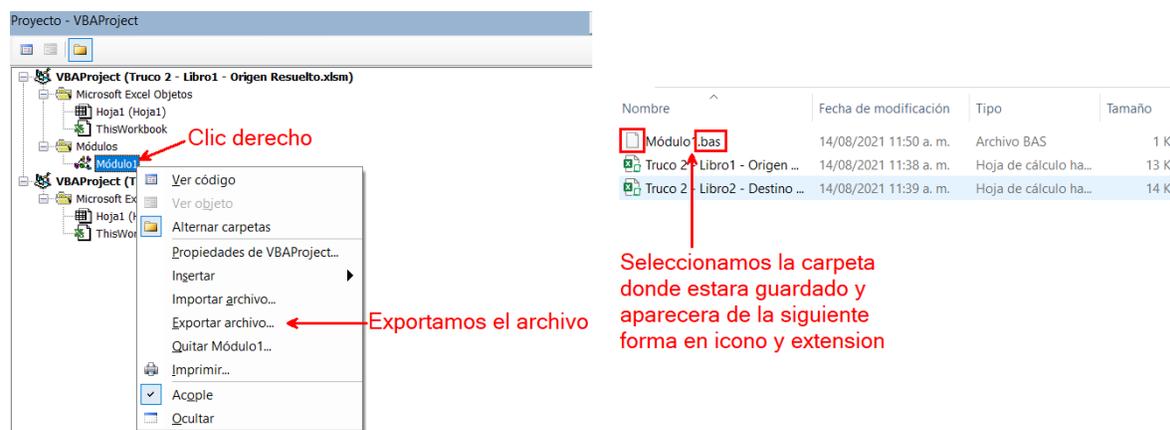


Figura 4.5 Exportar módulos a una carpeta

Seleccionamos el módulo y le damos clic derecho y luego lo guardamos en la carpeta que deseamos.

Ahora viene lo interesante, nosotros deseamos importar este archivo que tenemos guardado al Excel de destino el cual será nuestro nuevo proyecto, para esto existen 2 formas de importarlo.

Forma 1:

Vamos a nuestro libro de destino, abrimos nuestro proyecto y realizamos la siguiente acción:

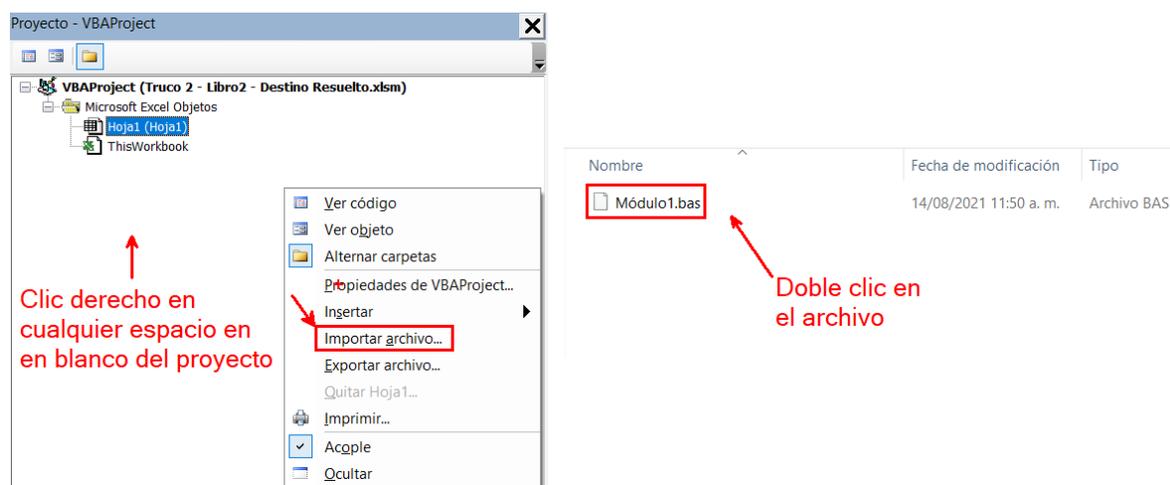


Figura 4.6 Importar módulos desde Visual Basic

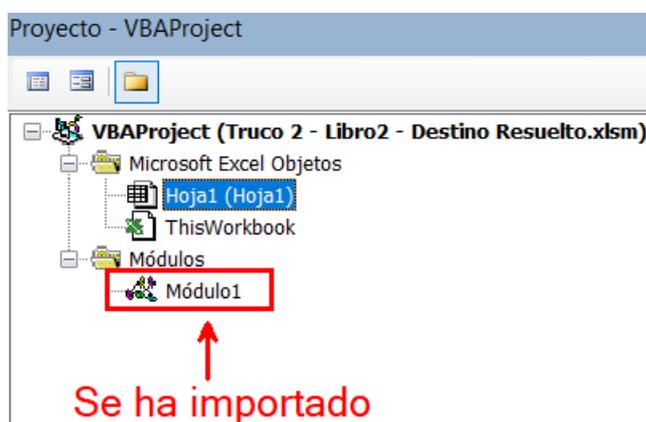


Figura 4.7 Módulo importado

Este truco nos permite tener más facilidad para poder tener nuestras macros almacenadas en algún medio alcanzable como un dispositivo o una nube para poder reutilizarlas en cualquier lugar y ahorrarnos tiempos en nuestros proyectos.

Forma 2:

Si deseas importar de forma rápida tus archivos sin necesidad de dar clic derecho, importar y buscar en las carpetas tu macro, puedes realizar un arrastre desde la carpeta a la interfaz de Visual Basic, para esto mira la siguiente imagen.

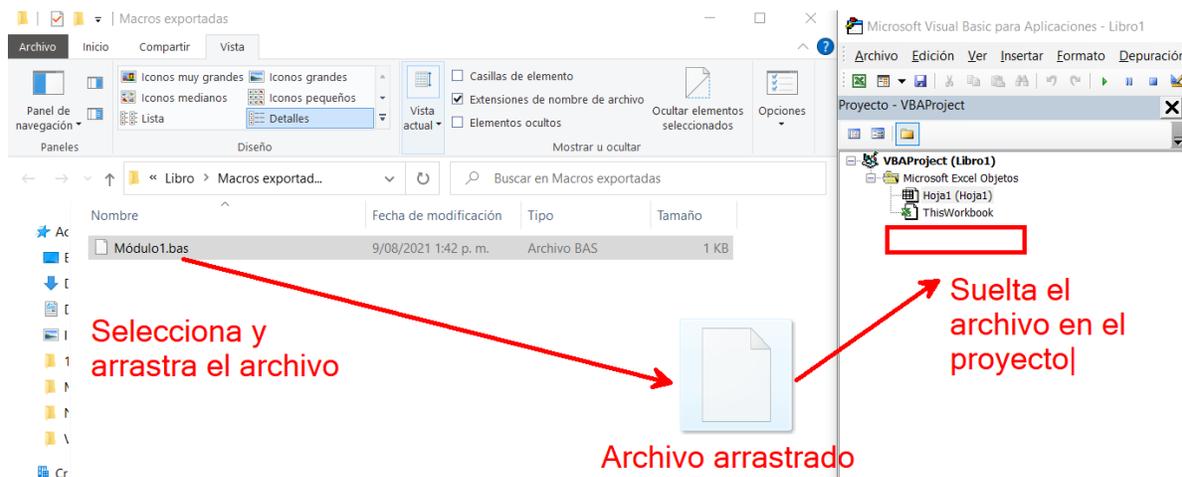


Figura 4.9 Importar macro con arrastre desde una carpeta

Este último truco es una mezcla del arrastre más importación, en este caso lo haces desde la carpeta al proyecto importando de una forma ágil y eficaz.



HACKS: Cuando realizamos desarrollos, muchas veces nos vemos en la situación de tener que repetir o replicar macros que ya hemos realizado en un pasado, y a veces podemos perder tiempo al no conocer estos trucos, hacemos que nuestros trabajos sean largos, dispendiosos y con la posibilidad de cometer errores por no recordar muchas veces qué hicimos en códigos anteriores, estos 2 trucos de duplicación de macros te ayudarán a crear desarrollos con menor tiempo.

4.3 Variables obligatorias en los proyectos

Las variables en VBA son muy importantes, esta no solo permite identificar si estas son numéricas, de texto, fecha, objetos entre otros, sino que nos permiten hacer que nuestros proyectos tengan muchos más beneficios, para esto veremos el siguiente truco que nos ayudaran a mejorar nuestro uso de las variables.

Al momento de realizar nuestras variables, comúnmente no las declaramos y más si estamos iniciando o estamos acostumbrados a no hacerlo, esta falta de declaración hace que nuestros proyectos se vean afectado muchas veces por funcionamiento y peso, por lo que se recomienda usar la siguiente declaración antes de iniciar cualquier macro.

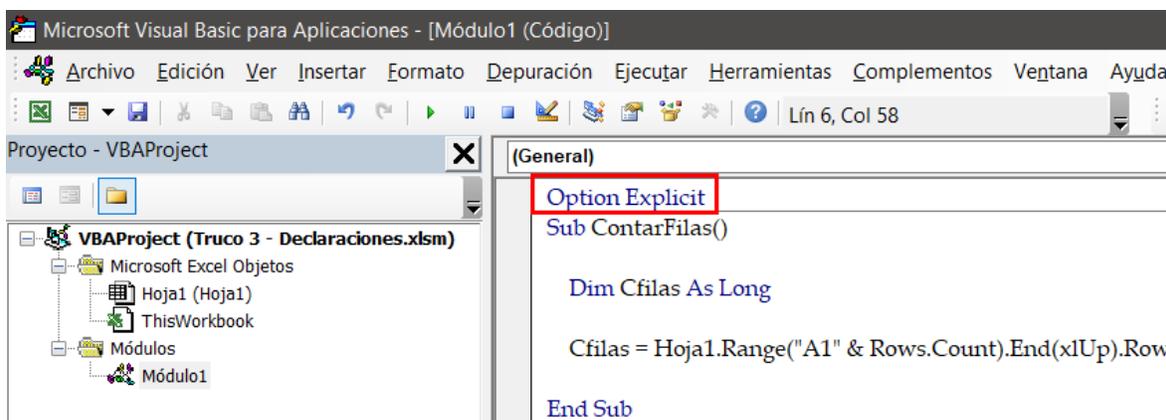


Figura 4.10 Palabras clave para volver la declaración de variables obligatoria



DECLARACIÓN: Option Explicit

Esta declaración nos obliga siempre a declarar las variables en nuestros proyectos.



HACKS: Escribir "Option Explicit" en nuestros proyectos nos permite hacer que nuestras macros por obligación deban tener todas sus variables declaradas, de lo contrario Visual Basic nos alertará si nos hace falta declarar alguna de estas, esto es un buen truco para hacer una buena práctica en nuestro desarrollo en macros y prevenir un mal funcionamiento de este.

Existe una forma si deseas no escribir esta declaración manualmente, sino que por defecto ya aparezca al inicio de nuestros trabajos y es la siguiente:

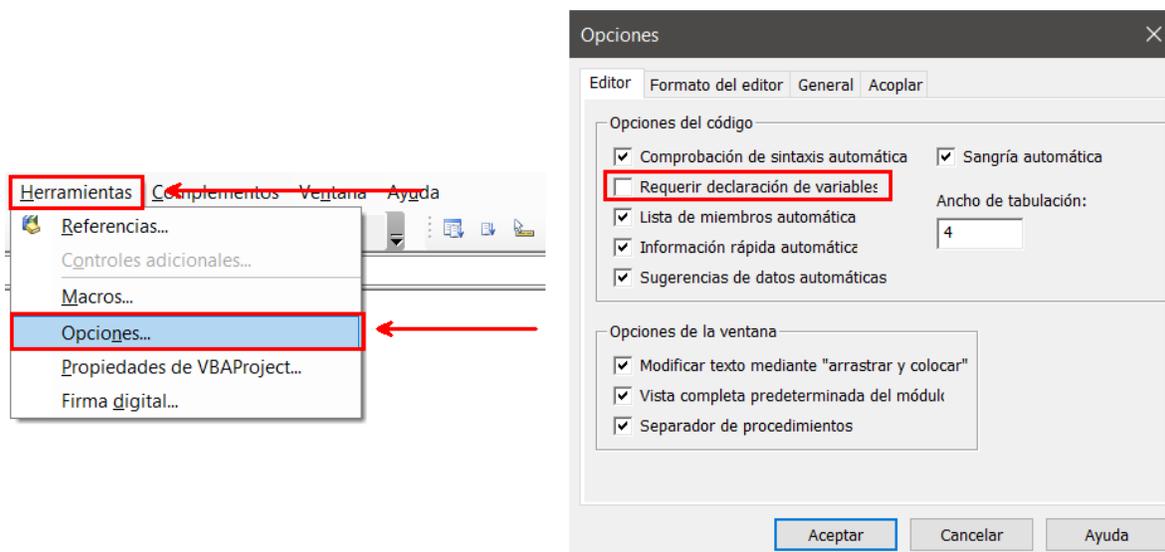


Figura 4.11 Activar declaración de variables

Nos dirigimos a Herramientas → Opciones → Requerir declaración de variables

Si en dado caso no declaramos la variable nos aparecería una alerta como esta al momento de ejecutar la macro y nos indicará la variable no declarada:

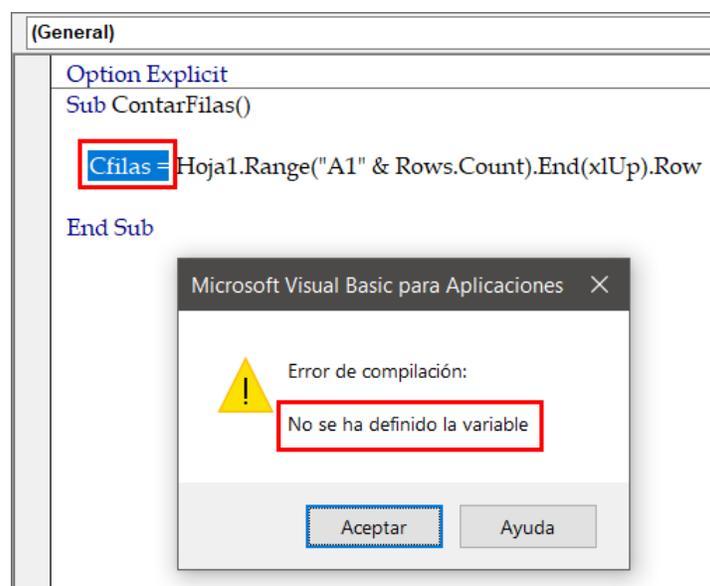


Figura 4.12 Variable no declarada

4.4 Atajos de escritura al declarar variables

Muchas veces cuando no declaramos variables, nos toma tiempo tener que escribir una y otra vez cada que la requerimos en nuestro código, por lo que la declaración de variables nos trae un truco al momento de escribirla.

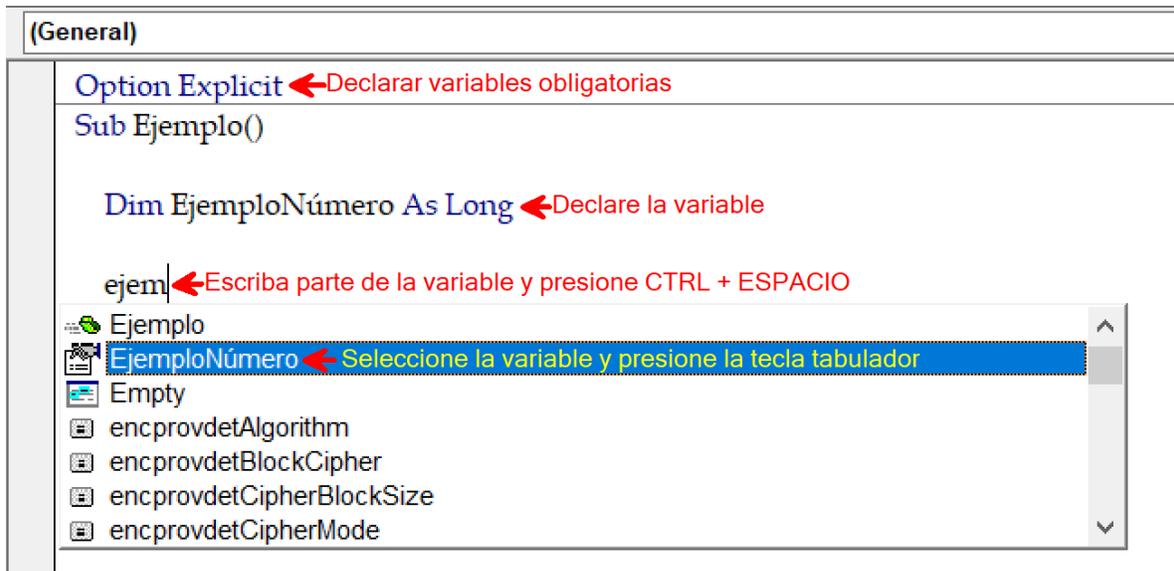


Figura 4.13 Autocompletar escritura de variables



HACKS: Para que una variable pueda funcionar en tus macros, este debe escribirse igual a como se declaró, gracias a este truco logramos autocompletar la escritura de estas reduciendo tiempo y errores de redacción al momento de utilizarlas, esto es algo que nos ha ayudado mucho en nuestros trabajos y que deseamos que ustedes lo aprendan, esto es algo pequeño pero poderoso y de uso continuo.

4.5 Evitar parpadeo de pantalla

Cuando ejecutamos macros es muy común ver que la pantalla genera parpadeos, como lo es saltar entre hojas, desplazarse en una hoja, cargar datos entre otras cosas, esto puede llegar a ser molesto visualmente, pero lo que muchas de las personas no saben es que el parpadeo solo es algo superficial, lo que realmente genera esos cargues de pantalla es que tu macro sea más lenta en la ejecución, pero con la propiedad que veremos a continuación podemos hacer que los tiempo de ejecución se reduzcan en un 30% o más según la densidad de los códigos.

```
(General)
Option Explicit
Sub ContarFilas()
    Application.ScreenUpdating = False ←Desactivar pantalla
    'Ingresa la macro que vas a ejecutar
    Application.ScreenUpdating = True ←Activar pantalla
End Sub
```

Figura 4.14 Propiedad Application.ScreenUpdating



PROPIEDAD: Application.ScreenUpdating

Nos permite desactivar y activar los cargues de pantalla durante la ejecución de una macro.

Cómo podrás observar en la figura 4.14, luego de nuestra propiedad damos la instrucción de **False** (Falso) el cual indica que esta propiedad no va a estar activa y por eso mencionamos la acción de desactivar y **True** (Verdadero) donde indica que la propiedad esta activa.



ADVERTENCIA: Recuerda siempre volver a activar los cargues de pantallas o este también podrá afectar tu libro de trabajo.



HACKS: Para una correcta ejecución de esta propiedad, debes entender que tiene dos estados, uno para desactivar los cargues de pantalla y otro para activarlos nuevamente, siempre que vayas a realizar una macro y desees poner esta propiedad como truco para optimizar tus tiempos de ejecución deber primero desactivar los cargues de pantalla para que este ayude a la macro ser más rápida, y al final de la macro una vez escritas las acciones, debes volver a activar los cargues de pantalla, sino la aplicación de Excel continuará con este efecto activado.

4.6 Evita colapsarte por cálculos automáticos

Cuando ya hemos trabajado con Excel, a muchos nos pasa que se nos hace normal el hecho de que las fórmulas y las funciones se actualicen por sí solas, al grado que cuando el Excel carga y se vuelve lento lo vemos como algo normal y decimos “esperemos que cargue”, esto en Visual Basic nos afecta mucho en las ejecuciones de macros, ya que si estamos cargando información entre hojas o generando datos masivos, las fórmulas y funciones afectaran el desempeño de nuestra macro por que se estarán recalculando cada que la ejecutemos o cada que se cumpla un ciclo si tenemos bucles, para esto aprenderemos un truco que nos permite correr las macros sin interrupciones que cuando la macro acabe, solo ahí se calcules la formulas y funciones.

```
(General)
Option Explicit
Sub ApagarCalculos()
    Application.Calculation = xlCalculationManual ←Activamos calculos manuales
    'Macro que se va a ejecutar
    Application.Calculation = xlCalculationAutomatic ←Activamos calculos automaticos
End Sub
```

Figura 4.15 Propiedad Application.Calculation



PROPIEDAD: Application.Calculation

Nos permite desactivar y activar los cálculos automáticos de tu libro de Excel mientras se ejecuta la macro.

En la figura 4.15 podemos observar que la propiedad tiene una instrucción de **xlCalcultionManual** (Calculo manual) el cual indica que no va a estar activa para que Excel realice cálculos de las fórmulas y funciones hasta cambiar el estado a automático el cual se seria indicando **xlCalculationAutomatic** (Calculo automático) en la propiedad.



ADVERTENCIA: Recuerda siempre volver a activar los cálculos automáticos ya que si no lo haces tus formulas y funciones no se calcularán cuando las escribas en Excel.



HACKS: En este truco podrás ver una mejora considerable en los rendimientos de las ejecuciones de las macros, desde un 30% a 80% de reducción en tiempos si tu archivo posee muchas formulas y funciones.

4.7 Eventos que obstaculizan las macros

Existen acciones en Excel que generan alertas de confirmación o avisos con el objetivo de evitar que pierdas información o algo importante.



NOTA: algunos ejemplos comunes son:

- Al guardar: confirmación de guardado.
- Al salir del libro: alerta de guardado
- Al eliminar una hoja de Excel: alerta de eliminación de la hoja

Cuando realizamos nuestras macros y creamos acciones que pueden cruzarse con estos eventos pueden afectar la ejecución de esta, ¿cómo? las alertas de estos eventos interfieren y la macro solo continuará una vez cerremos estas alertas, por lo que usaremos un truco a través de otra propiedad para evitar estas interrupciones.

```
(General)
Option Explicit
Sub AnularEventos()
    Application.EnableEvents = False ← Desactivar eventos
    ' Macro a ejecutar
    Application.EnableEvents = True ← Activar eventos
End Sub
```

Figura 4.16 Propiedad Application.EnableEvents



PROPIEDAD: Application.EnableEvents

Nos permite desactivar y activar las alertas de los eventos de tu libro de Excel mientras se ejecuta la macro.

En la figura 4.16 podemos observar que la propiedad tiene una instrucción de **False** (Falso) el cual indica que esta propiedad no va a estar activa para que los eventos interrumpen la macro hasta cambiar el estado a **True** (Verdadero) en la propiedad.



HACKS: Este truco te ayudará mucho en aquellas macros las cuales se vean afectadas por eventos, optimizando su rendimiento y operación sin ningún tipo de interrupción.

4.8 Alertas para tus macros

A veces nuestras macros pueden ser muy útiles, pero en ocasiones nos pueden jugar una mala pasada cuando las activamos por accidente y nos crea o borra cosas por accidente, es por eso por lo que debemos aprender algunos trucos para controlar las ejecuciones de nuestras macros, por ejemplo, un mensaje cuando estas terminen para saber si ya realizo la acción o hacer una validación para mirar si deseamos o no activar la macro, para este truco aprenderemos lo siguiente.



FUNCIÓN: MsgBox

Esta función nos permite crear mensajes de alerta en nuestras macros, ya sea para avisarnos que han finalizado, si deseamos iniciarlas o ayudarnos a entender que algo nos falta para ejecutar la macro entre otras cosas.

Como puedes ver, los MsgBox son mensajes que nos ayudan en nuestras macros, pero cómo es la sintaxis de este y cuales funciones tiene en su ejecución.

Esta sintaxis se compone 5 argumentos, pero solo nos enfocaremos en las primeras 3 las cuales son:

```
(General)  
Option Explicit  
Sub Mensajes()  
    MsgBox "Hola a todos", vbOKOnly, "ExcelAprende.com & DataICE"  
End Sub
```

Mensaje ↑
Boton o signo ↑
Titulo de la alerta ↑

ExcelAprende.com & DataICE ×
Titulo →
Hola a todos ← Texto
Boton → Aceptar

Figura 4.17 Función MsgBox

Texto: en este deberás poner el mensaje que deseas que las personas lean cuando se despliegue la alerta.



NOTA: Los textos siempre deben estar entre comillas, sino te van a generar un error.

Botones: en la parte de botones podemos personalizar nuestro MsgBox para activar diferentes botones y señales que nos ayuden con las alertas y los mensajes, para esto veremos los siguientes ejemplos de la biblioteca de Visual Basic para crear:

Argumento	Tipo	Imagen
vbOKOnly	Botón	
vbOKCancel	Botón	
vbAbortRetryIgnore	Botón	
vbYesNoCancel	Botón	
vbYesNo	Botón	
vbRetryCancel	Botón	
vbCritical	Señal	
vbQuestion	Señal	
vbExclamation	Señal	
vbInformation	Señal	

En este argumento podemos utilizar un truco muy sencillo para poder mezclar los botones con las señales al mismo tiempo y para esto usaremos la tecla "+" de la siguiente forma.



Figura 4.18 MsgBox con señal + botón

El último argumento que veríamos es el título, este lo encontramos en la parte superior del cuadro de mensaje, normalmente lo usamos para hacer referencia a algo o a la marca de quien desarrolla el trabajo en Excel VBA.

Ejercicio: Crear un MsgBox que nos valide con si o no una acción, la cual en este caso será crear otro MsgBox según la selección y dar un mensaje.



NOTA: Para realizar MsgBox con validación debes conocer o manejar el tema de condicionales como lo es el IF, ElseIF y el End IF para este ejercicio.

Para este ejercicio haremos una mezcla de trucos mencionados anteriormente cómo lo es la declaración de una variable ya cual contendrá la respuesta de la validación, luego esta variable la pasaremos por una pregunta de si o no con el MsgBox, y finalizaremos con el condicional IF para identificar cual acción debemos tomar, la cual se verá representada por un mensaje de MsgBox, veremos en la siguiente figura la macro requerida para esta acción:

```
(General)
Option Explicit
Sub ValidaciónMsgBox()

    Dim Validador As String ← Declaramos la variable

    Validador = MsgBox("¿Deseas realizar esta acción?", vbQuestion + vbYesNo, "ExcelAprende.com")
    ↑ Cargamos la variable con la respuesta
    ↑ MsgBox con la pregunta

    If Validador = vbYes Then
        ↑ Validamos si la respuesta es "Si"
        MsgBox "Indicó que desea continuar, se ejecutará la macro."
        ↑ De ser "Si", activamos esta macro
    ElseIf Validador = vbNo Then
        ↑ Validamos si la respuesta es "No"
        MsgBox "Indicó que no desea continuar, se cancela la ejecución la macro."
        ↑ De ser "No", activamos esta macro
    End If
End Sub
```

Figura 4.19 Ejercicio de validación



NOTA: Podrás encontrar el archivo con la macro en la Carpeta 4 – Archivos de VBA → Truco 7 – MsgBox → Truco 7 - Validación con MsgBox



HACKS: Cómo has visto, este truco no solo es para crear mensajes en tus macros, también son buenos auxiliares o apoyos que permiten tener un mayor control de tus ejecuciones.

4.9 Mensajes sin interrumpir la macro

He tenido varias preguntas de como ver el progreso de una macro en ejecución sabiendo que si usamos un MsgBox el cual es un cuadro de mensaje va a interrumpir la macro y la idea es que esta sea continua, para esto usaremos un truco que a veces pocos conocemos y es el uso de la siguiente propiedad.



PROPIEDAD: Application.StatusBar

Esta propiedad nos permite ver en la barra inferior izquierda los mensajes que designemos desde las macros sin interrumpir la ejecución de estas.

```
(General)
Option Explicit
Sub BarraDeEstado()
    Application.StatusBar = "Hola a todos, saludos desde ExcelAprende.com & DataICE"
End Sub
```

Indicamos la barra de estado

Indicamos el mensaje a mostrar en la barra

23				
24				
25				
26				
27				
28				
29				

Hoja1

Hola a todos, saludos desde ExcelAprende.com & DataICE

Figura 4.20 Barra de estado (StatusBar)

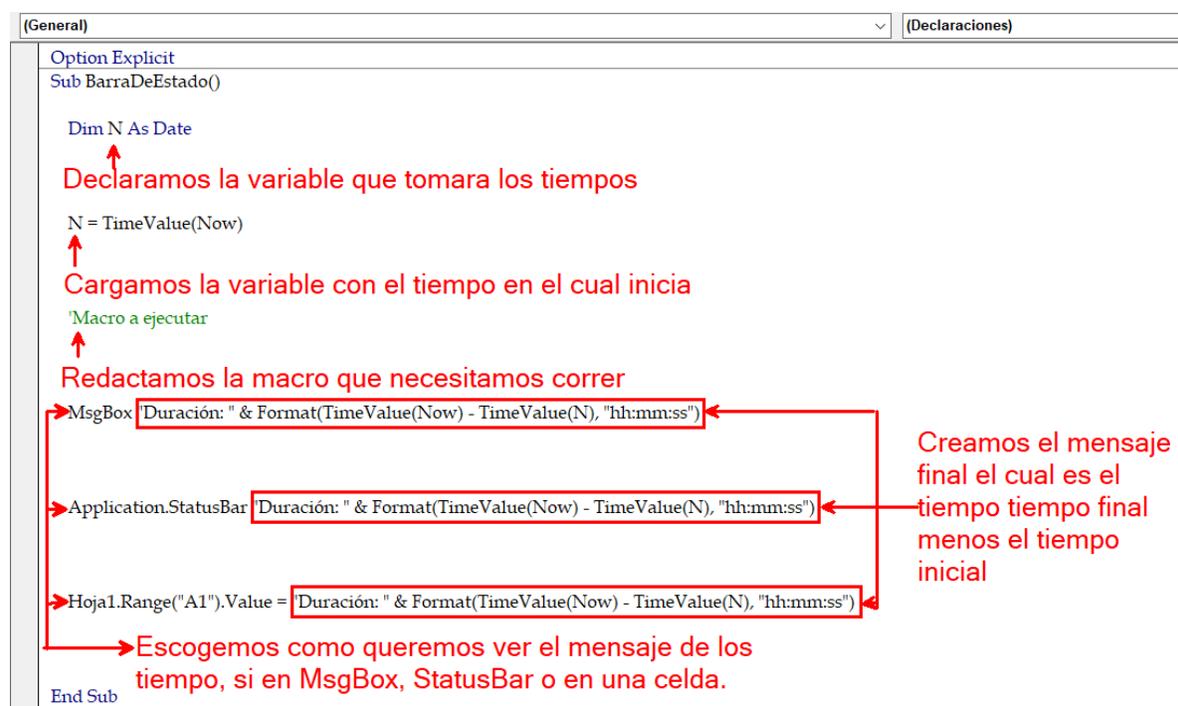


HACKS: Con esta propiedad la podemos usar para saber el estado de un proceso, o podemos indicar si ha finalizado o sigue en ejecución, estos son pequeños trucos que permiten al desarrollador brindar apoyos visuales dentro de su desarrollo hacia sus clientes o el mismo, este puede llegar a ser muy útil, pero depende de la creatividad de la persona que lo use.

4.10 Conoce los tiempos de tus macros

Una de las mejores ayudas que podrás tener en tus proyectos es crear temporizadores en tus macros, para que te indiquen al final del proceso cuanto fue el tiempo en procesar toda la acción, este truco tiene muchas ventajas para el desarrollador para saber cuánto demora y si realiza optimizaciones saber si en realidad está mejorando tiempos o no y para un cliente puede ser muy útil, porque si una macro pueden aumentar o disminuir según la cantidad de datos a procesar, entonces el cliente ya podrá saber cuáles van a ser sus tiempos de espera y poder anticipar acciones o tareas mientras este se ejecuta.

Para crear este temporizador, realizaremos la siguiente macro:



```
Option Explicit
Sub BarraDeEstado()

    Dim N As Date
    N = TimeValue(Now)
    'Macro a ejecutar
    MsgBox "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Application.StatusBar "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Hoja1.Range("A1").Value = "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
End Sub
```

Declaramos la variable que tomara los tiempos

Cargamos la variable con el tiempo en el cual inicia

Redactamos la macro que necesitamos correr

Creamos el mensaje final el cual es el tiempo tiempo final menos el tiempo inicial

Escogemos como queremos ver el mensaje de los tiempo, si en MsgBox, StatusBar o en una celda.

Figura 4.21 Macro temporizadora



HACKS: Con este truco podrás conocer realmente cuanto demoran tus macros y cuando las optimices cuanto tiempo estas reduciendo, y ahora tus clientes podrán saber cuánto tardan sus automatizaciones para organizar sus tiempos en sus trabajos.

4.11 Aprende a buscar palabras con atajos

Cuando trabajamos con Visual Basic, cualquier ayuda en tiempo es vital, es por eso por lo que aprenderemos unos trucos de búsqueda rápida en nuestras macros para que te vuelvas más rápido en tus trabajos.

Este es un truco muy práctico cuando empezamos a tener macros muy extensas y necesitamos realizar búsquedas rápidas y para esto necesitamos entender que este también va a ser dependiente de una característica que ya veremos.



ATAJO: Buscar en Visual Basic

Para activar la búsqueda debes presionar las teclas Ctrl + F

Una vez ingresado el atajo, deberás aparecer el siguiente cuadro:

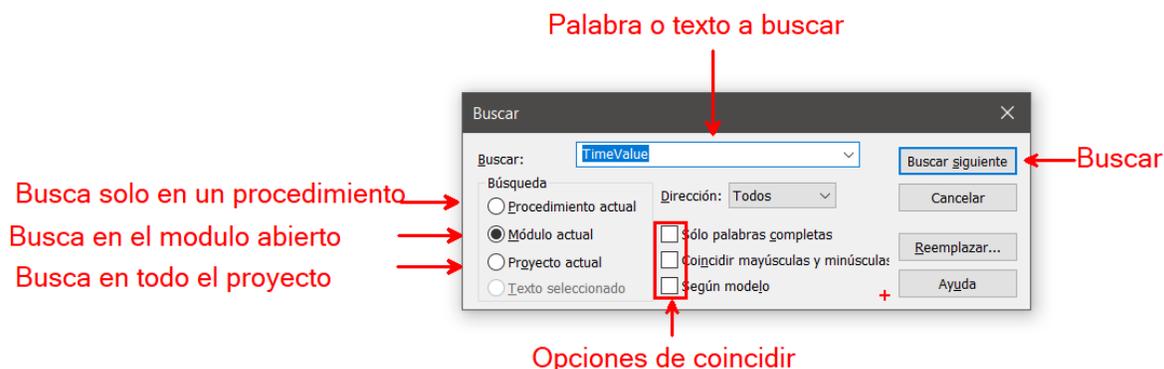


Figura 4.22 Ventana de buscar en VBA

A través de esta ventana podrás buscar los que necesitas, es muy útil al momento de realizar búsquedas, pero si necesitas realizar la búsqueda sin tantas vueltas el siguiente truco te va a encantar.



ATAJO: Buscar en Visual Basic con atajo

Para activar la búsqueda rápida debes presionar las teclas Ctrl + F3 sobre la palabra que necesitas buscar, luego presionando nuevamente F3 varias veces te podrás desplazar por todas las palabras que estas buscando.

```
(General) BarraDeEstado
Option Explicit
Sub BarraDeEstado()

    Dim N As Date

    N = TimeValue(Now)

    'Macro a ejecutar
    MsgBox Duración & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Application.StatusBar "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Hoja1.Range("A1").Value = "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")

End Sub
Sub BarraDeEstado2()

    Dim N As Date

    N = TimeValue(Now)

    'Macro a ejecutar
    MsgBox "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Application.StatusBar "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")
    Hoja1.Range("A1").Value = "Duración: " & Format(TimeValue(Now) - TimeValue(N), "hh:mm:ss")

End Sub
```

Figura 4.23 Atajo de búsqueda con teclado



ADVERTENCIA: Este atajo tiene una limitante y es que depende a como quede configurado la ventana de búsqueda de CTRL + F, es decir, si está configurado para solo buscar en el procedimiento, el módulo o el proyecto solo lo hará en esa especificación.



HACKS: Cuando vamos a realizar cambios o buscamos una palabra específica para repasar nuestra macro, estos trucos de búsqueda se vuelven muy prácticos y rápidos para realizar dicha acción.

4.12 Realiza pruebas a los resultados

Cuando trabajamos con VBA, es común que ejecutemos la macro y luego miremos si todo sale bien, pero aquí tendrás un truco con el cual podrás revisar los resultados sin necesidad de salir de la interfaz de Visual, es truco lo usamos los desarrolladores para tratar de mitigar errores y evaluar que nuestras macros llegan al resultado que buscamos, por eso el día de hoy aprenderás a manejar la **ventana inmediato**.

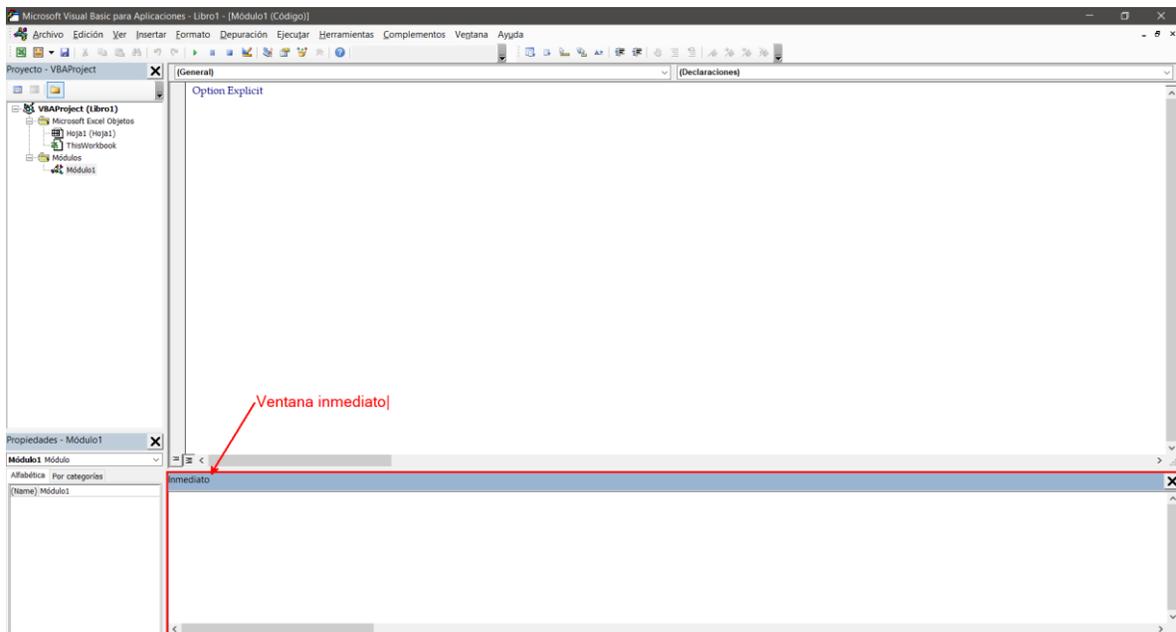


Figura 4.24 Ventana -inmediato

Esta ventana se puede abrir rápidamente usando atajos de teclado con la combinación de teclas **CTRL + G**, te estarás preguntando, ¿este atajo no es para guardar?, en visual el atajo de guardado es **CTRL + S** por lo tanto el otro atajo mencionado se designó para abrir la ventana que estaremos trabajando en este truco.

La ventana inmediata tiene muchas ventajas de uso que nos permite analizar si nuestras macros llegan a los resultados esperados y te estarás preguntando cómo, vamos a ver 2 formas rápidas y prácticas de su uso.

Forma # 1:

Para no tener que ejecutar la macro y ver si nos genera error, podemos copiar los códigos que deseamos evaluar y pegarlos luego de haber escrito "?" antes del código, luego presionamos la tecla **ENTER** y nos traerá el resultado de la macro, para una mejor orientación veamos la figura

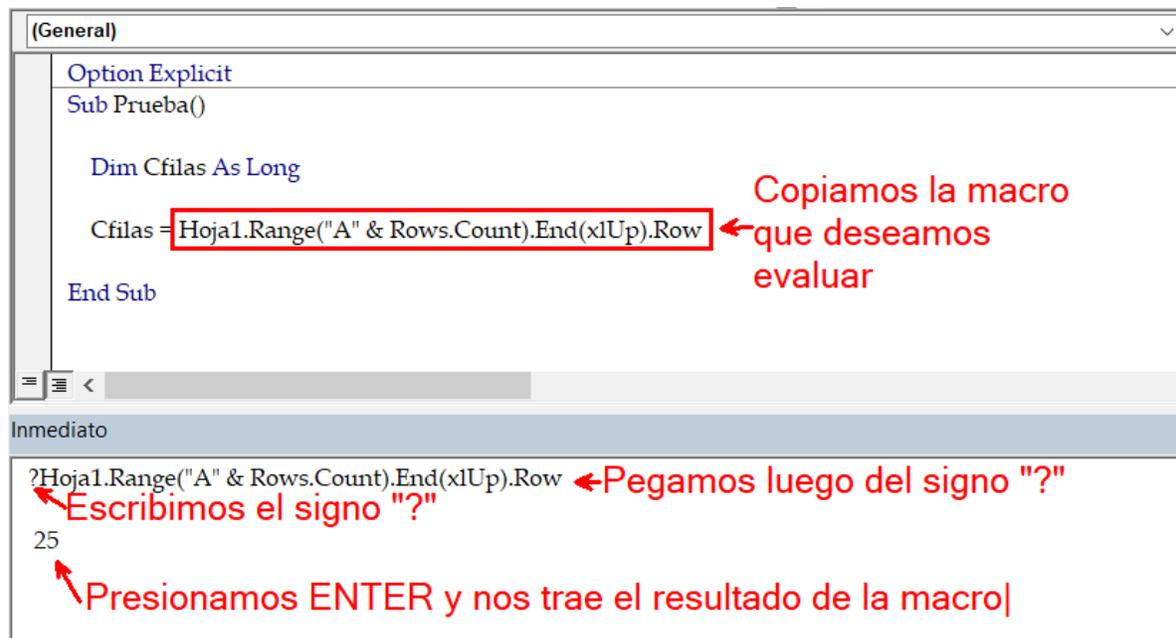


Figura 4.25 Evaluar resultado de una macro

Esta es una forma de validar primero si lo que estas creado en tus proyectos es lo que deseas ver luego en una hoja de Excel, lo que tratamos de hacer es intentar mitigar errores al máximo, este es un ejemplo muy sencillo, pero cuando estés programan y desees validar en toda tu macro varios puntos clave que puede afectar tu código, ahí será cuando este truco te generará mucho valor.

Ahora bien, también existe otra forma de imprimir los resultados antes de indicar que los desees exportar a otra ubicación del libro, y este te será muy útil para validar que las variables que usas si carguen los datos correctos, para esto usaremos la siguiente instrucción:



INSTRUCCIÓN: Debug.print

Esta instrucción nos permite imprimir en la ventana inmediato el resultado de la variable que le indiquemos, este nos facilita trayendo cada una de las variables que le indiquemos a la instrucción.

The screenshot displays the VBA editor with a macro named 'Sub Prueba()'. The code includes two 'Debug.Print' statements. Red arrows and labels identify the components of these statements: 'Variable 1' points to the variable 'Cfilas1', 'Variable 2' points to 'Cfilas2', and 'Instruccion' points to the 'Debug.Print' keyword. The Immediate window at the bottom shows the output: 'Filas de la columna A: 25' and 'Filas de la columna B: 10'. A red bracket and arrow labeled 'Resultados de las impresiones|' point to these two lines of output.

```
Option Explicit
Sub Prueba()

    Dim Cfilas1, Cfilas2 As Long

    Cfilas1 = Hoja1.Range("A" & Rows.Count).End(xlUp).Row

    Debug.Print "Filas de la columna A: " & Cfilas1

    Cfilas2 = Hoja1.Range("B" & Rows.Count).End(xlUp).Row

    Debug.Print "Filas de la columna B: " & Cfilas2

End Sub
```

Immediato

Filas de la columna A: 25
Filas de la columna B: 10

Resultados de las impresiones|

Figura 4.26 Debug.Print



HACKS: Gracias a este truco logramos analizar nuestro desarrollo de una forma rápida y ágil para evitar errores de información dentro de nuestras macros y reduciendo a futuro temas de soportes por estos errores comunes al programar.

4.13 Inspecciona tus variables

En este truco veremos algo muy similar a la **Ventana Inmediato**, pero en este caso se llamará la **Ventana Locales**, esta nos permite tener aún más detalles de nuestros trabajos y variables que usamos dentro de estas, a diferencia de la ventana de inmediato que imprime un dato debajo del otro, esta ventana actualiza el valor de la variable y la especificación de esta.



NOTA: La **Ventana Locales** no reemplaza la **Ventana Inmediato**, según lo que requieras podrás usar una de las dos, si deseas ver el consecutivo de resultados de las variables te servirá más inmediato, si solo deseas ver cuál es el último valor cargado en una variable, te servirá más locales, es decir ambas tienen muchos beneficios, depende de que requieres en el momento.

Por otro lado, la Ventana Locales trae una interfaz con columnas que también nos ayuda ver con mayor detalle de la información que estamos trabajando, para ver un ejemplo de esta miremos la figura 4.27

Expresión	Valor	Tipo
Módulo1	Vacio	Módulo1/Módulo1
Nombre	Vacio	Variant/Empty
Fecha	Vacio	Variant/Empty
Validador	-	String
CFilas	Vacio	Variant/Empty
i	0	Long

Aqui veras las variables de tu proyecto

Aqui podras ver el valor que tiene cada variable, de no haber cargado aun esta aparecera vacio

En esta parte podras encontrar tu variable como ha sido declarada

Figura 4.27 Ventana Locales

Gracias a esta ventana podemos identificar no solo las variables declaradas sino también las no declaradas, ¿qué nos permite eso?, empecemos por la optimización de los pesos, si has trabajado variables anteriormente, sabrás que según la declaración esta puede tener mayor o menos peso en tu proyecto, al nosotros no declarar las variables, el sistema de visual toma estos datos como Variant la cual es la más pesada de las declaraciones, si identificamos estos detalles en nuestros proyectos, vamos a poder cambiar y declarar las variables de forma correcta y evitar que nuestro trabajo se vuelva muy pesado.

Por otro lado, cuando estamos revisando el código ya sea con la tecla F5 que es para ejecutar la macro o la tecla F8 que es para ver paso a paso nuestra macro mientras se ejecuta, la ventana locales nos va a mostrar como las variables se van actualizando y que valores nos van cargando y así revisar si todo marcha bien o lo valores están errados.



HACKS: Cómo lo vez la ventana locales nos permite identificar rápidamente variables dentro de los procedimientos que llevamos, quizás en proyectos pequeños no lo veamos muy útil, pero cuando empiezas con varios módulos, múltiples macros y formularios le encontrarás mucha utilidad.

4.14 Valida errores con depuración y compilar

Hemos visto el tema de cómo revisar errores y como mitigarlos de la mejor manera posible, pero aún tenemos un truco más que te ayudara a revisar rápidamente si tu macro posee errores o no antes de ejecutarla sea por escribir mal una instrucción o por no declarar una variable o por algo más

Este truco se a través de la siguiente ruta: Depuración → Compilar VBA Project

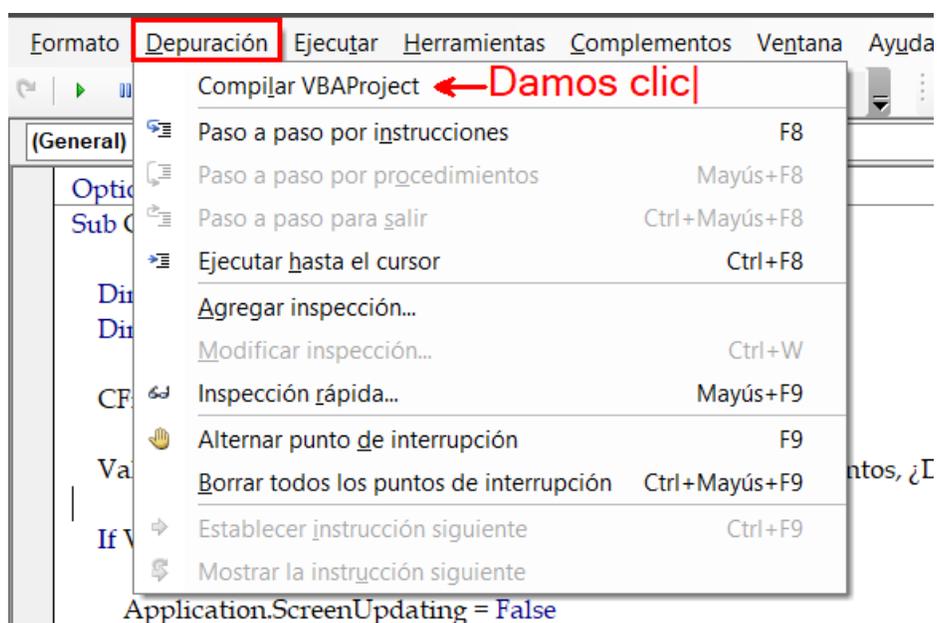


Figura 4.27 Compilar VBA Project



HACKS: Este truco nos ha sacado a muchos de pequeños problemas detectando la ubicación del problema que nuestra macro puede poseer, siendo honesto esto es uno de los primeros trucos que deberías aprender cuando inicias a programar con VBA para evitarte muchos dolores de cabeza

4.15 El poder del filtro avanzado en búsquedas

Filtrar es un término muy común en Excel, donde la mayoría de quienes trabajamos con esta herramienta solemos escuchar o usar continuamente, pero con este truco que verás no solo realizarás un filtro, sino que podrás llevar los datos filtrados a otra ubicación, para este truco realizaremos un paso a paso de que debes realizar para alcanzar este proceso con éxito.



NOTA: El archivo de este ejercicio lo podrán encontrar en la Carpeta 4 - Archivos de VBA → Truco 15 - Filtro Avanzado

Para realizar este filtro primero debemos tener en cuenta los siguientes datos antes de realizar la macro para este truco, estos son:

- Base de datos
- Criterios del filtro
- Ubicación de los datos a exportar con el filtro

The screenshot shows an Excel spreadsheet with a data table and a macro button. The data table has the following columns: ENTIDAD DETENCION, NOMBRE ENTIDAD, DOCUMENTO, TIPO_DOCUME NTO_FK, PRIMER_APELLI DO, SEGUNDO_APELLIDO, PRIMER_NOMBI RE, SEGUNDO_NOMBRE, and Administradora. The macro button is labeled 'Filtro Avanzado PRO'. Red arrows point to the criteria and the destination of the filtered data.

ENTIDAD DETENCION	NOMBRE ENTIDAD	DOCUMENTO	TIPO_DOCUME NTO_FK	PRIMER_APELLI DO	SEGUNDO_APELLIDO	PRIMER_NOMBI RE	SEGUNDO_NOMBRE	Administradora
899999003								
899999003	ERIO DE DEFENSA NA	9845	C	xxx	ssss	ssss	dddd	manza
899999098	ARA DE REPRESENTA	5130	C	klkik	kkk	klki	klkl	Protección
830040256	SPITAL MILITAR CENT	9659	C	xxx	ssss	ssss	dddd	Protección
899999003	ERIO DE DEFENSA NA	8390	C	xxx	ssss	ssss	dddd	Protección
800165860	CCIONAL DE ADMINIS	6999	C	klkik	kkk	klki	klkl	Protección
899999001	NIO DE EDUCACION N	7584	C	xxx	ssss	ssss	dddd	Protección
830115395	MBIENTE Y DESARRC	8154	C	xxx	ssss	ssss	dddd	Protección
899999090	DE HACIENDA Y CRED	6839	C	klkik	kkk	klki	klkl	Protección
899999118	TIRO DE LAS FUERZAS	5355	C	xxx	ssss	ssss	dddd	Protección
890900184	IMIENTOS DEL NARE S	5175	C	xxx	ssss	ssss	dddd	Protección
800197384	IMPUESTOS Y ADUAN	7683	C	klkik	kkk	klki	klkl	Protección
899999090	DE HACIENDA Y CRED	8342	C	xxx	ssss	ssss	dddd	Protección
899999003	ERIO DE DEFENSA NA	8814	C	xxx	ssss	ssss	dddd	Protección
890001339	MUNICIPIO DE FILAND	6549	C	klkik	kkk	klki	klkl	Protección
9.00891E+11	RIA DE EDUCACION DI	5886	C	xxx	ssss	ssss	dddd	Protección
825002037	E EDUCACION DPITAL	9393	C	xxx	ssss	ssss	dddd	Protección
800128835	VO GENERAL DE LA N	8497	C	klkik	kkk	klki	klkl	Protección
899999003	ERIO DE DEFENSA NA	9223	C	xxx	ssss	ssss	dddd	Protección
890399027	CIA DEL VALLE DEL CA	8408	C	xxx	ssss	ssss	dddd	Protección
890399029	MENTO DEL VALLE DE	6677	C	klkik	kkk	klki	klkl	Protección
899999055	HISTERIO DE TRANSPO	5708	C	xxx	ssss	ssss	dddd	Protección
890900000	ERIO DE DEFENSA NA	5012	C	xxx	ssss	ssss	dddd	Protección

Figura 4.28 Hoja con los datos requeridos

Teniendo ya los datos para realizar el filtro avanzado nos dirigimos a Visual Basic y realizamos el siguiente código:

```

(General) FiltroAvanzadoPRO
Option Explicit
Sub FiltroAvanzadoPRO()
Dim CantidadRegistros As String, pregunta As String

pregunta = MsgBox("Se dispone hacer el filtro avanzado, quiere continuar", vbYesNo, "Excelaprende.com")

If pregunta = vbNo Then
MsgBox "Proceso cancelado", vbInformation, "Excelaprende.com"
Exit Sub
Else

'Primero elimino lo que ya esta como resultado del filtro anterior
Columns("Q:Y").Select
Selection.Clear

'Luego me paro en la A6 para tomar todo el rango de datos nuevos para el filtro avanzado
Range("A6").Select
Application.CutCopyMode = False

'Valido cuantos campos o registros tengo para que el filtro los tome, ESTO CON BASE AL CAMPO NIT
CantidadRegistros = Application.WorksheetFunction.CountA(Hoja6.Range("A6:A1048576")) + 5

Range("A6:I" + LTrim(CantidadRegistros) & "AdvancedFilter Action:=xlFilterCopy CriteriaRange:=Range(" &
"A2:I3"), CopyToRange:=Range("Q6"), Unique:=False)

MsgBox "Listo, terminado el proceso", vbInformation, "Excelaprende.com"
End If
End Sub

```

Validamos si queremos iniciar el filtro
 De ser "No", se cancela el procedimiento
 Para evitar sobreponer informacion en la ubicacion destino, limpiamos lo que hay
 Nos ubicamos en la posicion inicial de la base de datos
 Validamos la cantidad de datos
 Selecciona el rango de la base
 Realizamos la accion de copiado para el filtro
 seleccionamos el rango de criterios para el filtro
 Rango destino de los datos filtrados

Figura 4.28 Macro para el filtro avanzado

Con esta macro realizaremos la acción de hacer filtros avanzados, para explicar un poco el paso a paso del código y poder entenderlo lee el siguiente paso a paso:

1. Declaramos las variables



DECLARACIÓN:

Dim CantidadRegistros As String, pregunta As String

2. Creamos un validador con MsgBox para evitar que se ejecute la macro por error en el botón donde la tengamos asignada.

**MACRO:**

```
pregunta = MsgBox ("Se dispone a hacer el filtro avanzado, quiere continuar",  
vbYesNo, "Excelaprende.com")
```

3. Creamos una condición donde indicamos que, si no deseamos accionarlo, que este cancele el procedimiento, pero de ser afirmativa la respuesta entonces que continúe con la ejecución de la macro.

**MACRO:**

```
If pregunta = vbNo Then  
  
MsgBox "Proceso cancelado", vbInformation, "Excelaprende.com"  
  
Exit Sub  
  
Else
```

4. Cómo los datos filtrados se van a cargar en una ubicación que le hemos asignado cada vez que ejecutemos la macro, esta se va a sobreponer los datos y es por lo que debemos realizar primero una limpieza de la ubicación.

**MACRO:**

```
Columns ("Q: Y"). Select  
  
Selection.Clear
```

5. Ahora que ya hemos limpiado, nos dirigimos a la ubicación inicial de la base "A6".

**MACRO:**

```
Range("A6"). Select  
  
Application.CutCopyMode = False
```

6. Ya, por último, realizaremos la importación de la información filtrada con la última parte de la macro.



MACRO:

```
CantidadRegistros=Application.WorksheetFunction.CountA(Hoja6.Range("A6:A1048576")) + 5  
  
Range("A6:I"+LTrim(CantidadRegistros)).AdvancedFilterAction:=xlFilterCopy,CriteriaRange:=Range("A2:I3"), CopyToRange:=Range("Q6"), Unique:=False
```

7. Para cerrar, mostramos un MsgBox que nos informe que el procedimiento a finalizado.



MACRO:

```
MsgBox "Listo, terminado el proceso", vbInformation, "Excelaprende.com"  
  
End If  
  
End Sub
```



ADVERTENCIA: Para que el filtro sea funcional, la posición final donde se lleva los datos debe tener la misma cantidad en columnas que la base inicial, para que lo tengas en cuenta al momento de usar este truco.



HACKS: Si logras dominar los filtros avanzados, este será una de tus mejores herramientas para la extracción de datos, no solo por ser una macro de filtros, sino porque dentro de las macros que realizan búsquedas y cargue de datos en otras ubicaciones este método es de los más rápidos en tiempos de ejecución.

4.16 Mejora tus búsquedas con Array

Array es una de las formas más rápidas de analizar datos con Visual Basic, este permite que los datos que buscamos o deseamos cargar en algún lado sea mucho más rápido que otras macros convencionales, leyendo lo anterior te estarás preguntando por que Array es tan rápido y que realiza para tener esa efectividad, ya lo veremos a continuación.

Lo primero que debemos entender es que Array trabaja con rangos de datos, como tablas o rangos de celdas, la diferencia a otros métodos de selección de rangos, Array realiza una acción y es que los datos seleccionados como rangos no los trabaja desde la ubicación de Excel sino que los carga cómo una tabla en segundo plano dentro de visual, suena loco pero así es, Array prácticamente está cargando de alguna forma los datos internamente, permitiendo realizar acciones de búsqueda más rápidas en cuanto a tiempos de ejecución.

Para entender este truco, realizaremos un ejemplo basándonos en la tabla que usamos en el truco de filtro avanzado donde exportaremos unos datos de una ubicación a otra, la diferencia con este filtro es que los criterios estarán en el código y no en Excel.

Inicialmente se realizará macro la cual se usa comúnmente contra el Array, y usaremos el truco del temporizador para medir los tiempos de ejecución y así saber cuánto tarda cada acción.

Macro 1

```
(General)
Option Explicit
Sub ExportarDatos1()

'Limpiamos los datos de la base destino si existen
Hoja2.Range("A1").CurrentRegion.Offset(1).ClearContents ←Limpiamos la unicacion
                                                         de destino

'Contador de tiempo

Dim Temporizador As Date ←Activamos el temporizador
Temporizador = TimeValue(Now)

'Definimos el rango de la información

Dim Rango As Range ←Creamos el rango que vamos
Set Rango = Hoja1.Range("A1").CurrentRegion a usar

'Iniciamos la exportación de datos

Dim I, Y As Long
Y = 2
For I = 2 To Rango.Rows.Count ←Macro con ciclo y condicional
    If Hoja1.Cells(I, 1) = "899999003" Then para ir creando la base filtrada
        'Copiar usando selecciones fila por fila segun el criterio
        Hoja1.Activate de filtro
        Rango.Rows(I).Select
        Selection.Copy
        Hoja2.Activate
        Hoja2.Range("A" & Y).Select
        Selection.PasteSpecial xlPasteValues
        Y = Y + 1
    End If
Next I
|
'Cerramos el contador del tiempo
|
MsgBox Format(TimeValue(Now) - TimeValue(Temporizador), "hh:mm:ss")
|
End Sub
```

Figura 4.29 Macro usada comúnmente para exportar datos filtrados

El tiempo de ejecución fue de:

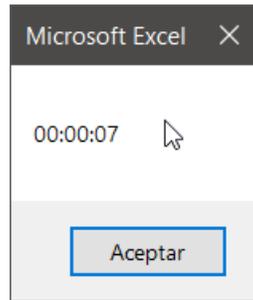


Figura 4.29 Macro usada comúnmente para exportar datos filtrados

Ahora veremos cómo está estructurada la macro con el Array:

```
(General)
Option Explicit
Sub ExportarDatos6()

'Limpiamos los datos de la base destino si existen
Hoja2.Range("A1").CurrentRegion.Offset(1).ClearContents

'Contador de tiempo
Dim Temporizador As Date
Temporizador = TimeValue(Now)

'Definimos el rango de la información
Dim Arr As Variant
Arr = Hoja1.Range("A1").CurrentRegion.Value

'Iniciamos la exportación de datos
Dim I, Y, J As Long
Y = 2
For I = LBound(Arr, 1) To UBound(Arr, 1)
    If Arr(I, 1) = "899999003" Then
        For J = LBound(Arr, 2) To UBound(Arr, 2)
            Hoja2.Cells(Y, J) = Arr(I, J)
        Next J
        Y = Y + 1
    End If
Next I

'Cerramos el contador del tiempo
MsgBox Format(TimeValue(Now) - TimeValue(Temporizador), "hh:mm:ss")

End Sub
```

← Limpiamos los datos del destino

← Iniciamos el temporizador

← Creamos la base dentro del Array

← Macro con ciclo y condicional que recorre los datos cargados en el Array

← Cerramos el procedimiento finalizando el temporizador

Figura 4.30 Macro con Array para filtros

El tiempo de ejecución fue de:

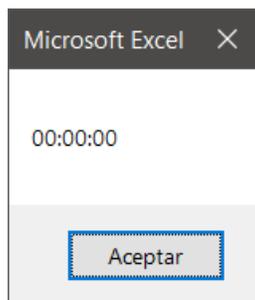


Figura 4.31 Macro con Array para filtros

Como podemos observar esta es una base con solo 770 filas en datos, la primera macro nos arrojó 7 segundos al momento de ejecutar mientras que con el Array se demoró 0 a 1 segundos, mostrando la gran diferencia que posee el uso de esta función.



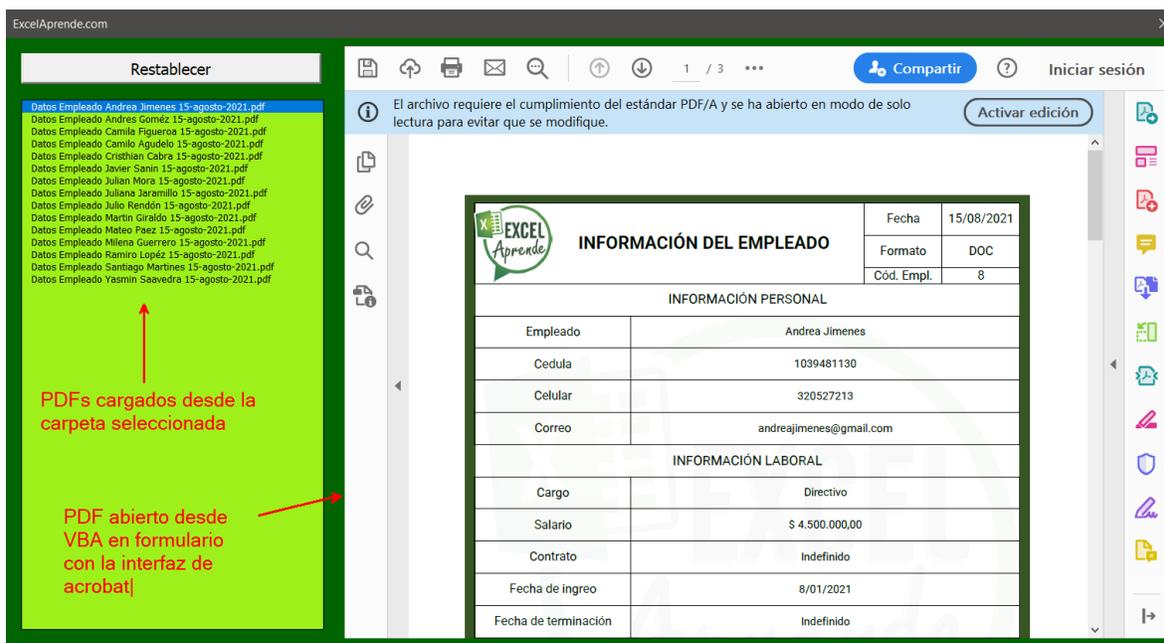
HACKS: Este truco lo he utilizado en una tabla de más de 44 mil datos en filas, usando la primera macro común, solo extrayendo 200 datos filtrados de 8000+ que debía sacar se demoraba aproximadamente 12 segundos, solo 200 datos, mientras que con Array me sacaba los más de 8000+ datos filtrados en 1 segundo, con esto podemos dimensionar un poco el poder de este truco.

4.17 Cómo cargar PDFs en un formulario

Hace un tiempo acá he tuve que buscar una solución para la búsqueda de PDFs desde Excel, pero con el objetivo de abrirlo internamente desde VBA y que no se abriera externamente, ¿Cuál era su propósito? Solicitaron un aplicativo basado en Excel, donde tuviese la opción de centralizar la mayor cantidad de acciones posibles sin necesidad de salir del aplicativo, adicional por encontrar de forma muy rápida los documentos solicitados para auditorias, seguimientos entre otras cosas.

En este truco aprenderás como cargar de una carpeta los PDFs que se encuentren ahí con la posibilidad de abrirlo desde un formulario en VBA.

Primero veremos un ejemplo del trabajo que realizaremos, miremos la figura 4.32



Restablecer

Datos Empleado Andres Gomez 15-agosto-2021.pdf
Datos Empleado Camila Figueroa 15-agosto-2021.pdf
Datos Empleado Camilo Agudelo 15-agosto-2021.pdf
Datos Empleado Cristhian Cabra 15-agosto-2021.pdf
Datos Empleado Javier Sanin 15-agosto-2021.pdf
Datos Empleado Julian Mora 15-agosto-2021.pdf
Datos Empleado Juliana Jaramillo 15-agosto-2021.pdf
Datos Empleado Julio Rendón 15-agosto-2021.pdf
Datos Empleado Martin Giraldo 15-agosto-2021.pdf
Datos Empleado Mateo Paez 15-agosto-2021.pdf
Datos Empleado Milena Guerrero 15-agosto-2021.pdf
Datos Empleado Ramiro Lopez 15-agosto-2021.pdf
Datos Empleado Santiago Martinez 15-agosto-2021.pdf
Datos Empleado Yasmin Saavedra 15-agosto-2021.pdf

PDFs cargados desde la carpeta seleccionada

PDF abierto desde VBA en formulario con la interfaz de acrobat

ExcelAprende.com

Compartir Iniciar sesión

El archivo requiere el cumplimiento del estándar PDF/A y se ha abierto en modo de solo lectura para evitar que se modifique. Activar edición

INFORMACIÓN DEL EMPLEADO		Fecha	15/08/2021
		Formato	DOC
		Cód. Empl.	8
INFORMACIÓN PERSONAL			
Empleado	Andrea Jimenes		
Cedula	1039481130		
Celular	320527213		
Correo	andreaJimenes@gmail.com		
INFORMACIÓN LABORAL			
Cargo	Directivo		
Salario	\$ 4.500.000,00		
Contrato	Indefinido		
Fecha de ingreso	8/01/2021		
Fecha de terminación	Indefinido		

Figura 4.32 Formulario con PDFs y PDF abierto

Como puedes observar, este es un formulario creado con Visual Basic, el cual te permite traer PDFs en formato de lista y luego abrirlos con la interfaz de acrobat en Excel, este es un truco que me ha ayudado con clientes ayudándolos a traer rápidamente documentación para auditorias, seguimientos, reportes o informes que deben presentar y han usado Excel como una herramienta administrativa.

Ahora veremos un paso a paso de cómo realizar este formulario, primero nos dirigimos a la interfaz de visual y creamos un formulario.

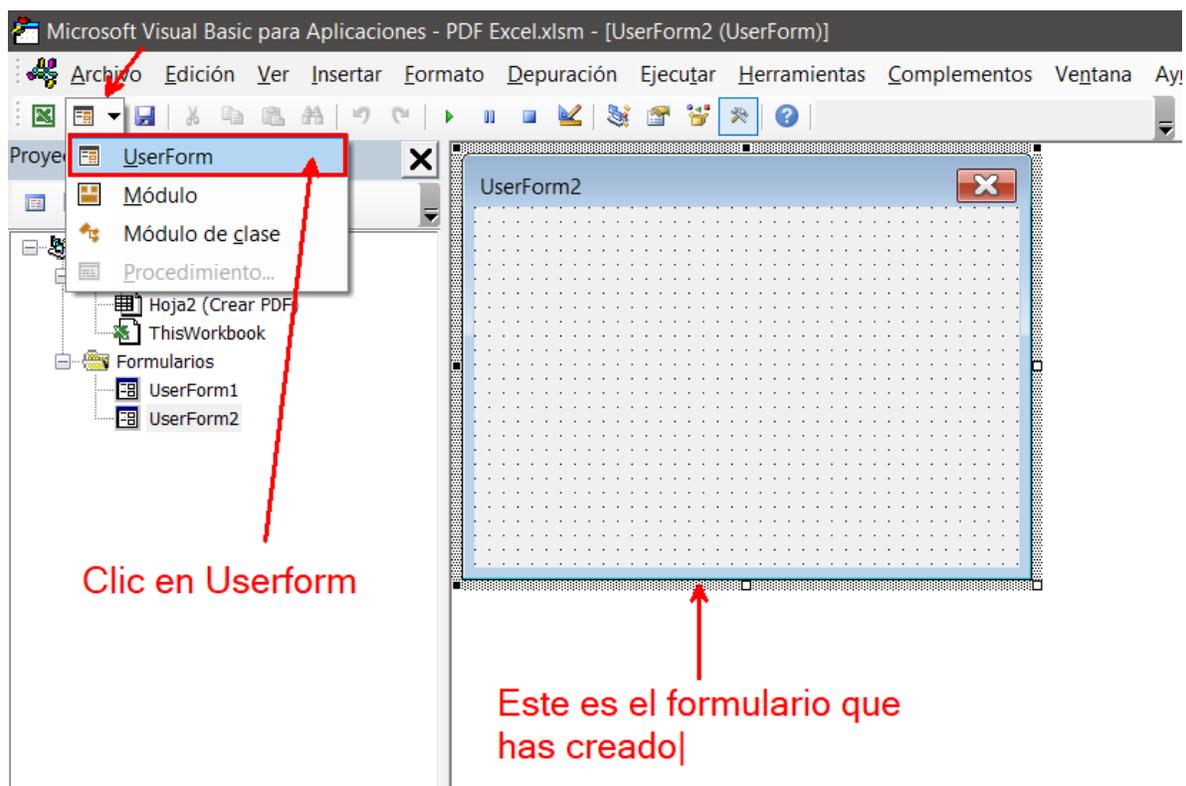


Figura 4.33 Crear un formulario

Ahora bien, vamos a crear los objetos del formulario, para esto vas a ir a las herramientas de formulario y vas a crear lo siguiente:

- Un botón
- Un cuadro de lista
- Un WebBrowser

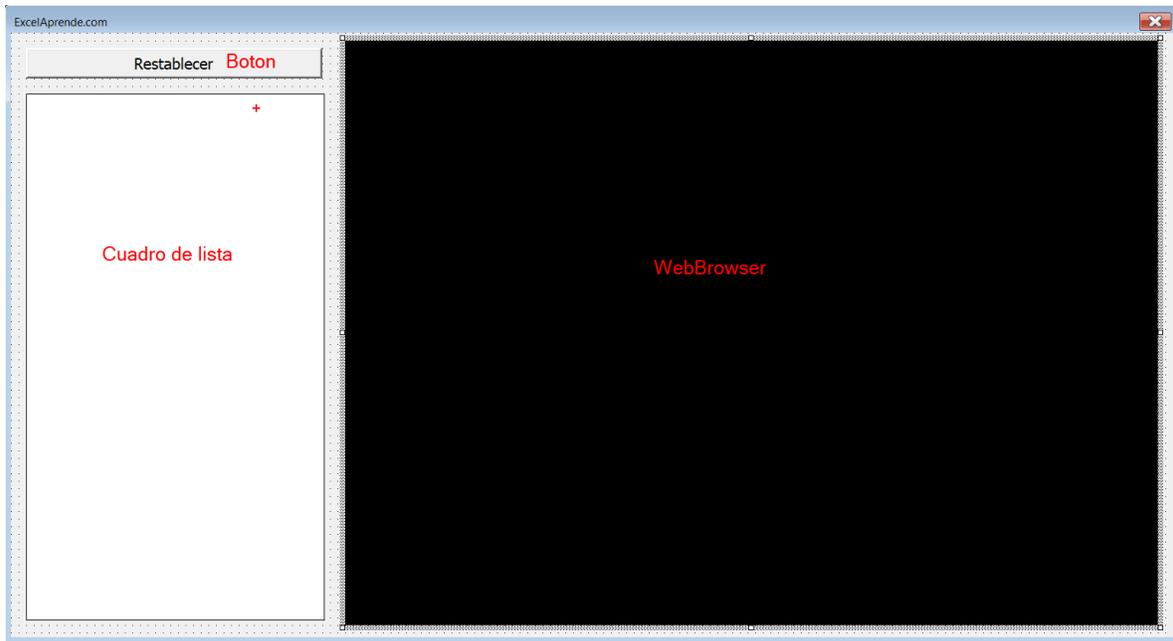


Figura 4.34 Diseño del formulario con los objetos

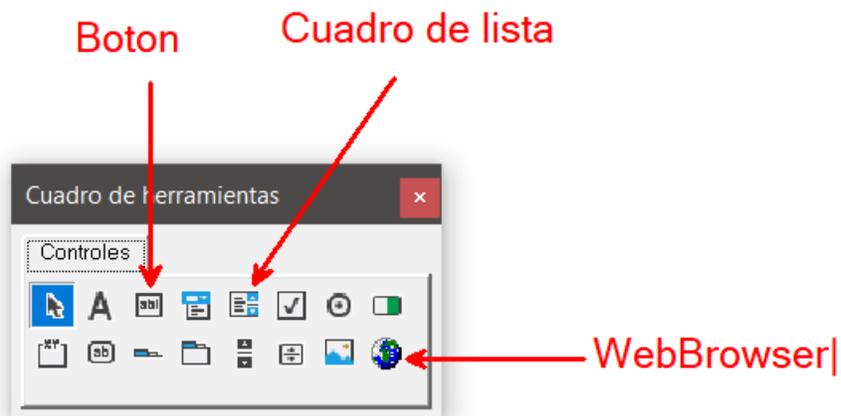


Figura 4.35 Cuadro de herramientas y objetos para el formulario

Una vez realizado el formulario iniciaremos con los códigos para que estos sean funcionales, ahora bien, recuerda que cuando trabajamos con formularios tenemos varios eventos, por lo que el primero que debemos de usar es el de Activate, este nos cargara los primeros datos del formulario los cuales son los archivos de PDF de la carpeta que tengamos indicada como ruta, luego tendremos otra macro con el cuadro de lista, en esta nos permitirá cargar con doble clic el archivo de PDF que escojamos, por último tendremos la macro del botón la cual nos cierra el archivo de PDF abierto.

```

(General)
Option Explicit
Dim Carpeta As String ←Variables publicas que trabajan entre procedimientos
Dim Pdfs As String
Private Sub UserForm_Activate() ←Procedimiento 1, activacion del formulario

    On Error Resume Next
    Carpeta = "C:\Users\Cristhian\Desktop\Libro\Carpeta 4 - Archivos de VBA\Truco 16 - Buscar PDF"
    ChDir (Carpeta)
    Pdfs = Dir("*.pdf") ← Ruta de la carpeta con los PDFs

    Do
        Me.ListBox1.AddItem Pdfs ← Busca los archivos con extension .pdf
        Pdfs = Dir() ← Carga los PDFs en el cuadro de lista
    Loop Until Pdfs = ""

End Sub

Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean) ←Procedimiento 2,
    abrir PDF

    If Me.ListBox1.ListIndex = -1 Then
        MsgBox "No existen datos a seleccionar.", , "ExcelAprende.com"
        Exit Sub ← Valida si hay archivos, si no hay te alerta
    Else
        Me.WebBrowser1.Navigate Carpeta & Me.ListBox1.List(Me.ListBox1.ListIndex, 0)
    End If ← Carga el archivo seleccionado en el
    WebBrowser

End Sub

Private Sub Restablece_Click() ←Procedimiento 3, reestablecer el WebBrowser

    Me.ListBox1.ListIndex = -1
    Me.WebBrowser1.Navigate "" ← Quita la seleccion del PDF y restaura el
    WebBrowser

End Sub

```

Figura 4.36 Macro para buscar PDFs y abrirlos con formulario



ADVERTENCIA: El WebBrowser no está activo en la ventana de herramientas, debes activarlo primero

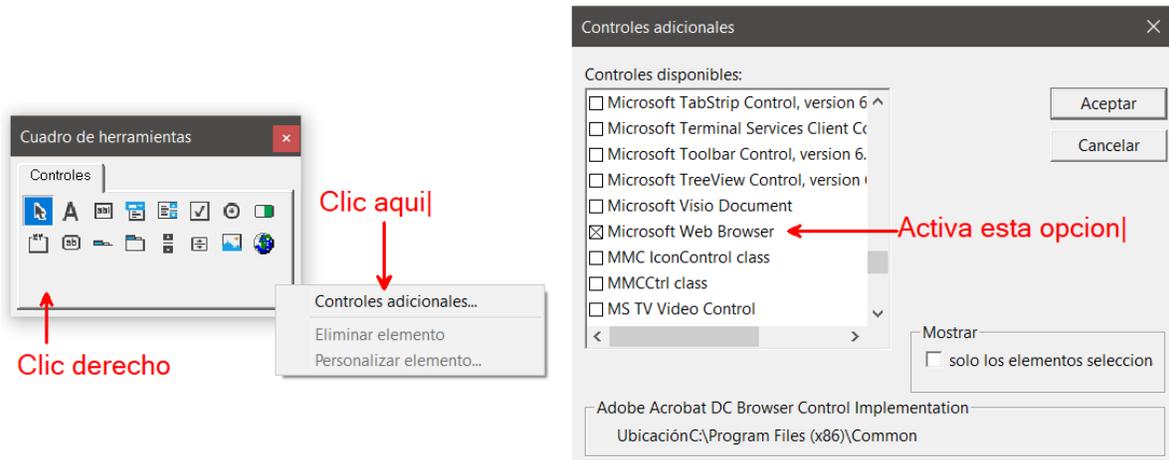


Figura 4.36 Cómo activar el WebBrowser



HACKS: Con este truco puedes optimizar el control documental en algunos de tus desarrollos, logrando administrar información que esta externalizada de tu Excel en PDF, se sorprenderán de lo útil que es este truco.

4.18 Autocompletar palabras en VBA

Cuando trabajas en Visual Basic y estamos iniciando o ya tenemos algo de experiencia, es muy común que muchos no conozcan los atajos de autocompletado para escribir las macros de forma más rápida, estos atajos no permiten reducir tiempos de redacción lo evitar errores de escritura, este truco será corto, pero nunca lo olvidarás.

En VBA existe una forma de autocompletar instrucciones, variables, nombres entre otros tipos de escritura, para esto no hace falta que estén después de un punto como vemos a veces en nuestras macros con el tema de métodos y propiedades que salen en automático para ayudarnos a completar la macro.



ATAJO:

CTRL + ESPACIO

En la siguiente figura verás este atajo aplicado a diferentes partes de una macro.

The figure illustrates the application of the Ctrl+Space shortcut in the VBA IDE. It shows four examples of autocompletion:

- Traer un modulo:** In the Properties Window, typing 'men' results in 'mensaje' being suggested.
- Autocompletar objetos:** In the Properties Window, typing 'app' results in 'Application' being suggested.
- Autocompletar nombre de forms:** In the Properties Window, typing 'user' results in 'UserForm1' being suggested.
- Traer macro de otro modulo:** In the Code Window, typing 'mensaje.' results in 'mensaje1' being suggested.

Figura 4.37 Atajos de autocompletar y que puedes autocompletar

4.19 Buscar definiciones y última posición

Cuando trabajamos entre módulos y tu proyecto se crece, a veces es complejo poder recordar la ubicación de las macros que estas llamando o usando entre módulos, con este truco vas a poder encontrar muy rápido tus macros llamadas entre otras macros y vas a ahorrar mucho tiempo de trabajo.

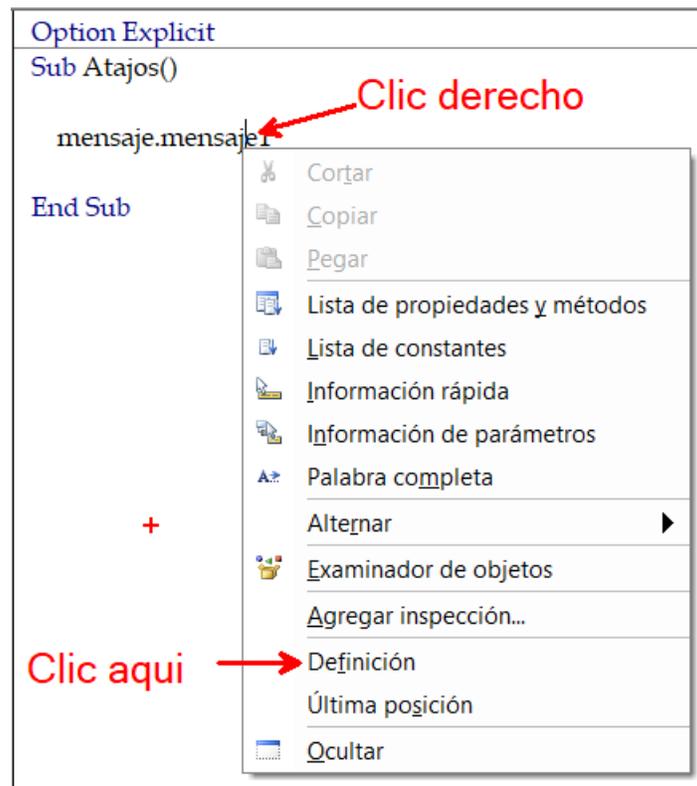


Figura 4.38 Como activar definición

“**Definición**” nos ayuda a ir donde está la macro que estamos llamando, y así revisar que hemos redactado por si requerimos hacer ajustes, cambios o solo una revisión, por otro lado, también puedes usar **Última posición** la cual nos permitirá ir a la macro donde finaliza la acción de la macro llamada.

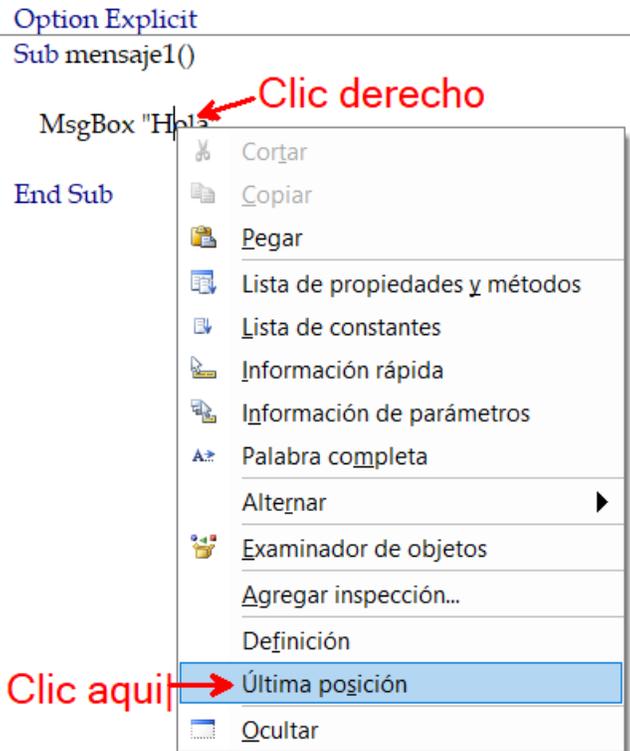


Figura 4.39 Última posición



HACKS: Gracias a este truco he logrado realizar soportes de mis desarrollos de forma muy rápida, ya que solo con dar clic derecho puedo rastrear las macros dependientes de otras para ver que está fallando o de donde provienen las acciones del código.

4.20 Application.InputBox

Con este truco podrás sacarle el mayor provecho al InputBox, normalmente la mayoría de nosotros usamos InputBox que sería lo mismo que VBA.InputBox la cual solo nos permite ingresar valores, pero no muchos usan esta otra forma de InputBox llamada Application.Inputbox, aunque ambas son muy similares, la segunda opción la cual hace referencia al truco que aprenderás hoy, trae más alternativas de uso las cuales serán de mucha utilidad en tus trabajos.

Un truco que podrás sacarle mucho provecho de este InputBox es que podrás escoger rangos dinámicos para tus análisis, procesos o extracción de datos de forma muy flexible, cómo has visto en otros trucos como lo es filtrar avanzado, el Array entre otras macros que puedan usar rangos, esta función te podrá ayudar a potenciar estos trucos ya que es común que las bases, matrices o rangos cambien de ubicación, de tamaño o de datos, gracias a este podrás seleccionar el rango nuevo sin ningún problema y evitar entrar al código una y otra vez para actualizar la ubicación de los datos.

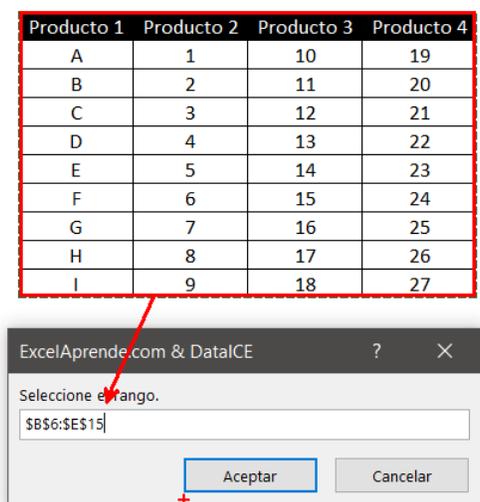


Figura 4.39 Selección de rangos con InputBox

Para poder crear esta función debes realizar la siguiente macro y a partir de ahí podrás usar el rango seleccionado como mejor te parezca dentro de tus macros.

Option Explicit

Sub Rangos()

Dim Rg As Range

Set Rg = Application.InputBox("Seleccione el rango.", "ExcelAprende.com & DataICE", Type:=8)

End Sub

Establece la variable para que sea igual a la selección del InputBox

Declara una variable de tipo rango

Funcion

Mensaje

Titulo

Tipo 8 es selección de rango

Figura 4.39 Última posición

También puedes usar en vez de rango y variable "Range" usar Array con variable "Variant".



HACKS: Con este truco lograras reducir tiempos de trabajo cuando necesites un rango dentro de tus macros, siempre y cuando estas sean cambiantes en su estructura.

Por otro lado, esta función tiene otros tipos de uso para que puedas darles otros usos.

4.21 Bonus 3 palabras mágicas para buscar

Durante la realización de macros, se aprende que hay 3 formas de programar con VBA y estas son:



Figura 4.40 3 formas de programar en VBA

Las formas más comunes para la programación se basan en la grabadora de macros y en la búsqueda de macros, son pocos los que desarrollan a través de escritura, pero aun así se puede decir que el mejor truco y manera de trabajar en VBA es a través de la búsqueda de macros, pero existe un truco que los desarrolladores de VBA usan para que sus búsquedas sean más efectivas.



NOTA: Es recomendable que sepas un poco de inglés, eso te ayudará a encontrar más información y macros poderosas.

Cuando vamos a buscar en Google, debemos usar 3 palabras mágicas antes de cualquier búsqueda, y estas son **“VBA code to”**, con estas 3 palabras puedes realizar búsquedas muy específicas que te ayudarán a reducir tiempos de búsquedas para hallar los códigos que necesitas, lo único que te recomendamos es que debes tener al menos un nivel básico de inglés para que puedas sacarle el mayor provecho a este truco, ya que muchas de estas búsquedas se realizan en esta lengua, ya sea por variedad de resultados o porque visual trabaja normalmente en inglés.

Google podría volverse tu mejor amigo de programación si aprendes a realizar búsquedas inteligentes y bien formuladas con las tres palabras mágicas, realmente no necesitas mucho más para poder empezar a encontrar y reforzar tus conocimientos de macros.

¿Cuáles son las ventajas de usar este tipo de búsqueda? Esta normalmente nos permite encontrar macros funcionales de fuentes confiables las cuales traen consigo espacios no solo de la explicación sino también de la macro y documentos de estudio para poder profundizar.

Por otro lado, al igual que cuando grabamos una macro, debemos empezar a pulir y depurar los códigos que no vamos a utilizar en nuestro proyecto y aquí es cuando usas tu experiencia en la escritura, como saber aprovechar el gran truco de la búsqueda combinando tu conocimiento en VBA para acoplar las macros que encuentres



HACKS: Con este último truco, aunque no está enfocado a un trabajo en VBA, es un excelente truco para que pueda ayudarte en tu día a día de tus desarrollos, dándote soluciones prácticas y casi inmediatas.

Capítulo 5: Hacks Power Query

Power Query es una herramienta de inteligencia de negocios para el tratamiento de datos, lo que se conoce como proceso ETL (Extracción, Transformación y Carga de datos), en este libro no abarcamos la explicación del manejo de esta herramienta ya que tenemos otros recursos para esto; este libro es para usuarios que vienen trabajando o explorando Power Query, ya que con estos grandes trucos vas a poder automatizar lo automatizado, si, tal cual como leíste, parece una redundancia pero al fin esto es lo que hace el poder de las consultas, automatizar procesos, ¿pero que tal si te damos trucos para mejorar esa automatización?.



NOTA: Power Query viene como complemento para Excel 2010 y 2013, a partir de 2016 ya viene predeterminado a partir de la versión 2016, 2019 y Office 365.

En esta sesión vamos a usar office 365.

Cada Hacks plasmado en este capítulo salen debido a las múltiples consultorías que hacemos como equipo en las diferentes empresas que ayudamos, así que son casos reales que aumentaran tu productividad y como no, tener esos AS bajo la manga que en algún juego con tus datos debes sacar a lucir.

5.1 Consolidar todas las tablas de un libro de Excel en el mismo archivo

Algo muy común es tener varias tablas u hojas en Excel, consolidarlas es la tarea de muchos analistas, pero este no es el reto, el objetivo de este truco consiste no solo en consolidar dicha información, también que este proceso quede automatizado, es decir, si se ingresa una nueva tabla esta información se anexe de manera automática al consolidado.



NOTA: En la carpeta 5. Archivos Power Query se encuentra el material de trabajo

Nos dirigimos al archivo 1. *Consolidar Tablas en Excel*

Lo primero que debemos revisar es que la información este como tabla o rango definido, como lo vimos en trucos anteriores.

Como validamos esto:

Nos ubicamos en una celda que contenga datos → Pestaña Diseño de tabla

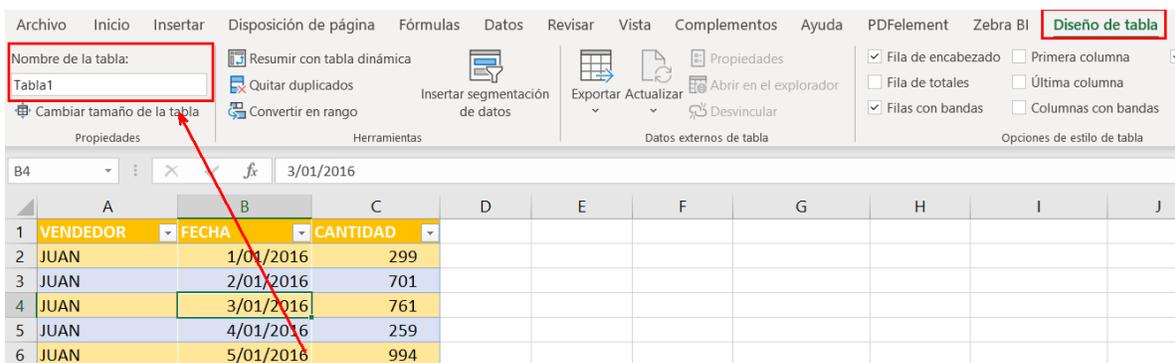


Figura 5.1 Validar información este como tabla.

Si la información no esta como tabla, se debe convertir a tabla o a rango definido.

Seleccionar datos → Insertar → Tabla o con el atajo CTRL + T

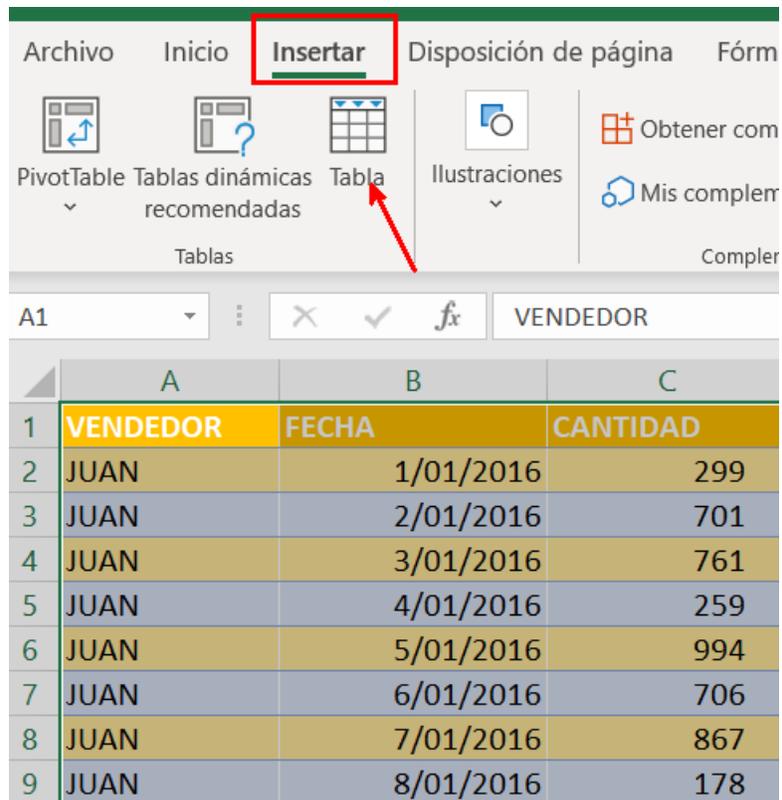


Figura 5.2 Convertir datos a tabla.

Después de validar cada una de las hojas y verificar que los datos estén como tabla, procedemos a aplicar este truco.

Ir a Power Query

Vamos a la pestaña Datos → Obtener datos → Desde otras fuentes → Consulta en blanco

Ver la figura 5.3

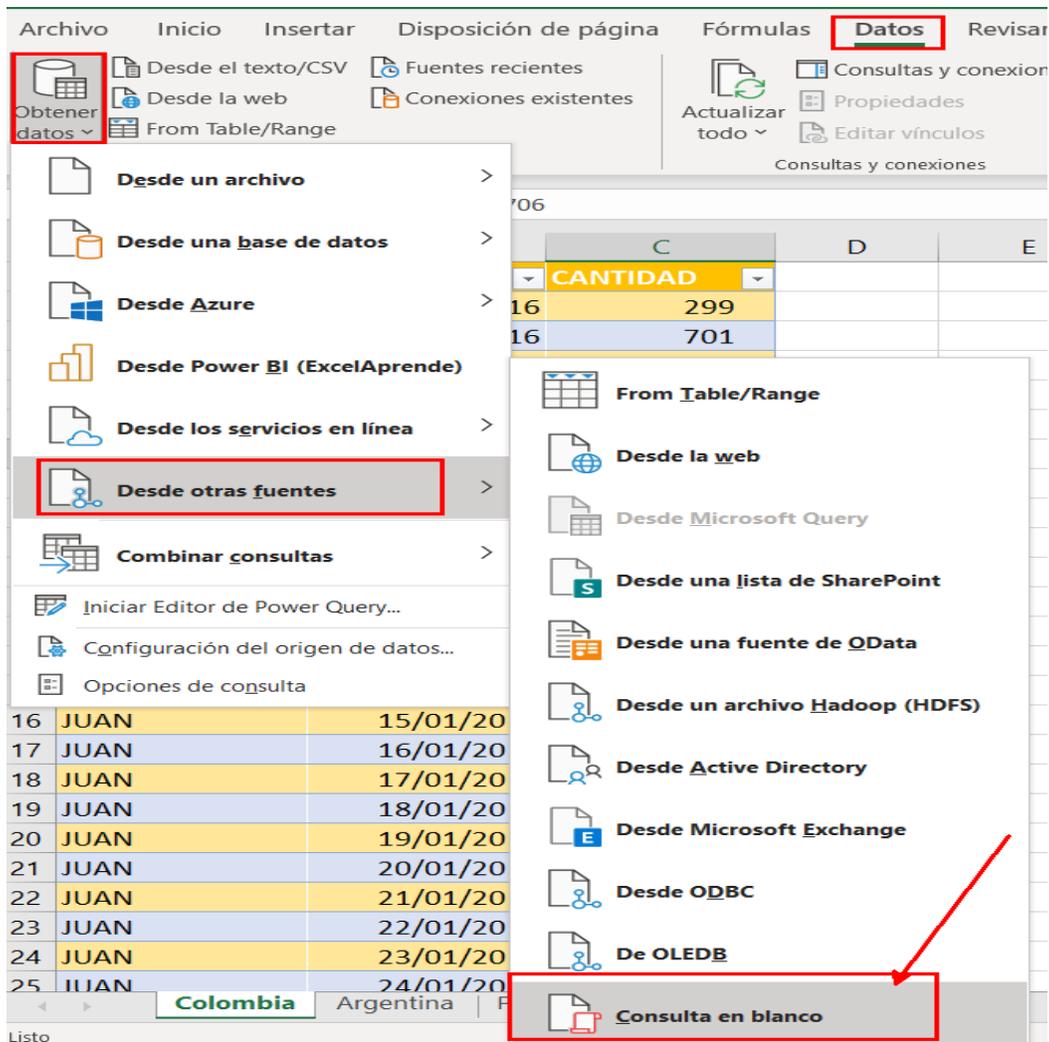


Figura 5.3 Ir a Power Query.

Estando en el navegador de consultas de Power Query, validamos que nuestra barra de fórmula este activa

Pestaña Vista → Habilitar barra de fórmulas

Luego ingresamos la siguiente función:

= Excel.CurrentWorkbook()

y para finalizar enter, el resultado es el siguiente. Ver figura 5.4

	Content	Name
1	Table	Tabla1
2	Table	Tabla2
3	Table	Tabla4
4	Table	Tabla5

Figura 5.4 Función consolidar tablas.



FUNCIÓN: *Excel.Workbook()*

Función para consolidar tablas y rango en Excel

Este es el resultado de esta función del lenguaje M. Trae como resultado todas las tablas y rangos definimos que encuentra activos en el libro de trabajo.

En la columna *Content* damos Clic para expandir los datos y traer la información.

Buscar columnas para expandir

Expandir Agregar

- (Seleccionar todas las columnas)
- VENDEDOR
- FECHA
- CANTIDAD

Usar el nombre de columna original como prefijo

⚠ La lista puede estar incompleta. [Cargar más](#)

Aceptar Cancelar

Figura 5.5 Expandir información.

Una buena práctica es deshabilitar *Usar el nombre de columna original como prefijo*, para que el nombre de la columna Content no esté como prefijo, sino quedaría de la siguiente manera: Content VENDEDOR Content FECHA

Para terminar, vamos a Inicio → Cerrar y aplicar

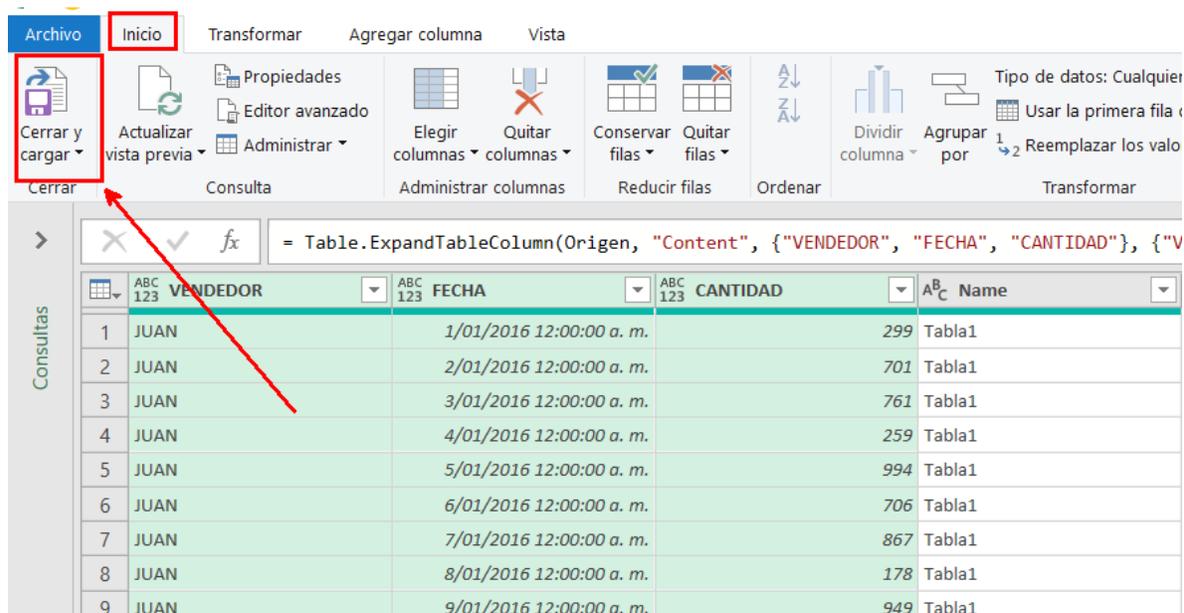


Figura 5.6 Cargar información a Excel.

De esta manera tenemos todas las tablas cargadas y consolidadas en Excel

¿Lo sabías? Es algo increíble, solo con una función poder tener la información lista para ser analizada, espero te hayas emocionado tanto como nosotros cuando lo aprendimos.

Revisamos la información en Excel, validemos que todo este correcto. Vamos hasta la última fila de la consulta.

2734	MIGUEL	43037	702	Tabla5
2735	MIGUEL	43038	128	Tabla5
2736	MIGUEL	43039	1305	Tabla5
2737	MIGUEL	43040	188	Tabla5
2738	MIGUEL	43041	133	Tabla5
2739	MIGUEL	43042	217	Tabla5
2740	MIGUEL	43043	634	Tabla5
2741	MIGUEL	43044	737	Tabla5
2742	MIGUEL	43045	596	Tabla5
2743	MIGUEL	43046	180	Tabla5
2744	MIGUEL	43047	208	Tabla5
2745				Consulta1!DatosExternos_1
2746				

Figura 5.7 Validar información en Excel.

Como lo enseña la figura 5.7 se cargaron 2.744 registros, pertenecientes a las 4 tablas y hojas que están el libro de Excel.

Antes de terminar con la emoción y saltar en un solo pie de saber esto, debemos hacer un ajuste de suma importancia, no pierdan de vista ese registro que se llama *Consulta1! Datos Externos_!*

Que sucede si actualizamos la consulta, los datos deben seguir siendo los mismos ya que no hemos ingresado más registros o tablas nuevas, pero lastimosamente no es así.

Nos ubicamos en una celda de la tabla → Clic derecho actualizar

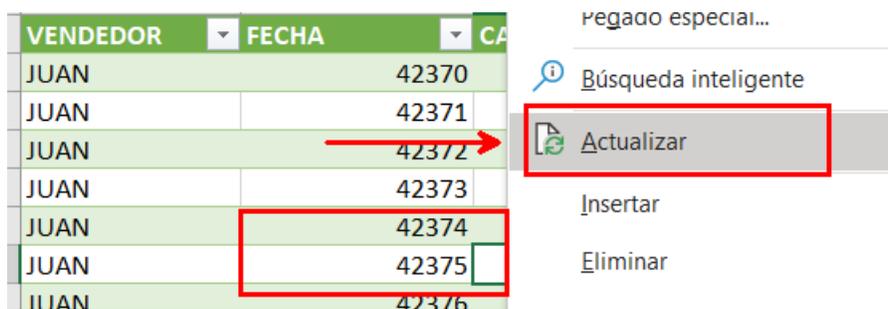


Figura 5.8 Actualizar consulta.

Volvamos a revisar nuestra información, vamos a hasta la última fila de la tabla.

5477	MIGUEL	43036	634	Tabla5
5478	MIGUEL	43037	702	Tabla5
5479	MIGUEL	43038	128	Tabla5
5480	MIGUEL	43039	1305	Tabla5
5481	MIGUEL	43040	188	Tabla5
5482	MIGUEL	43041	133	Tabla5
5483	MIGUEL	43042	217	Tabla5
5484	MIGUEL	43043	634	Tabla5
5485	MIGUEL	43044	737	Tabla5
5486	MIGUEL	43045	596	Tabla5
5487	MIGUEL	43046	180	Tabla5
5488	MIGUEL	43047	208	Tabla5

Consultas | Conexiones

1 consulta

Consulta1

Se cargaron 5.487 filas.

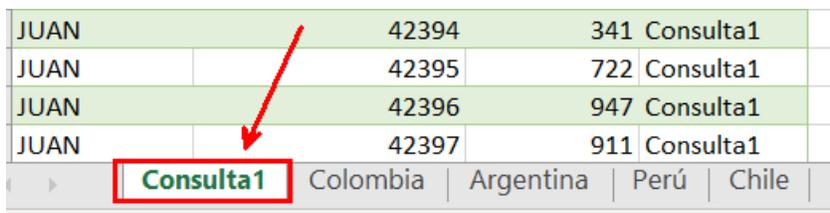
Figura 5.9 Tabla actualizada.

¿Qué sucedió?, la tabla duplico la información, validemos el filtro de la última columna Name

Figura 5.10 Consulta1 agregada.

Efectivamente se agrega nueva información llamada *Consulta1*, fue la culpable de duplicar la información.

Esto tiene toda la razón, ya que recuerda que la función =Excel.Workbook() consolida todas las tablas del libro, y si revisas, cuando cargamos los datos de Power Query, automáticamente se agrega una nueva hoja en Excel.



JUAN	42394	341	Consulta1
JUAN	42395	722	Consulta1
JUAN	42396	947	Consulta1
JUAN	42397	911	Consulta1

Consulta1 Colombia Argentina Perú Chile

Figura 5.11 Tabla nueva en el libro.

Esto hacer que, al actualizar la consulta, la función vuelva y recorra todas las tablas y las anexe, al encontrar una nueva, la añade.

Editar consultas

Para solucionar este inconveniente vamos a editar esta consulta, para filtrar el campo Name, que no contenga *Consulta1*.

Clic derecho en consultas y conexiones → Editar

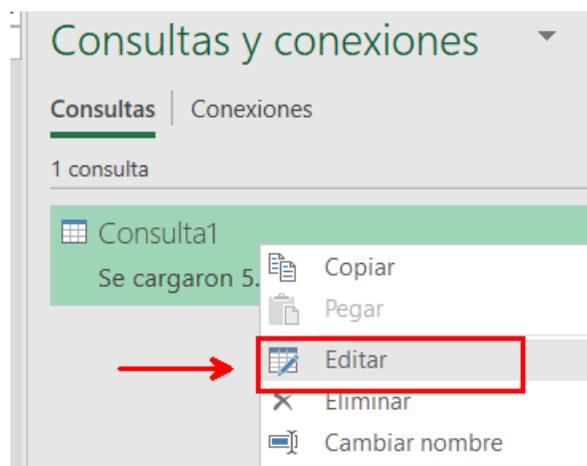


Figura 5.12 Editar consultas.

Luego de estar en Power Query, filtramos el campo *Name* y deshabilitamos Consulta1.

Recuerda darle en Cargar *más* para habilitar todos los campos. Ver figura 5.13

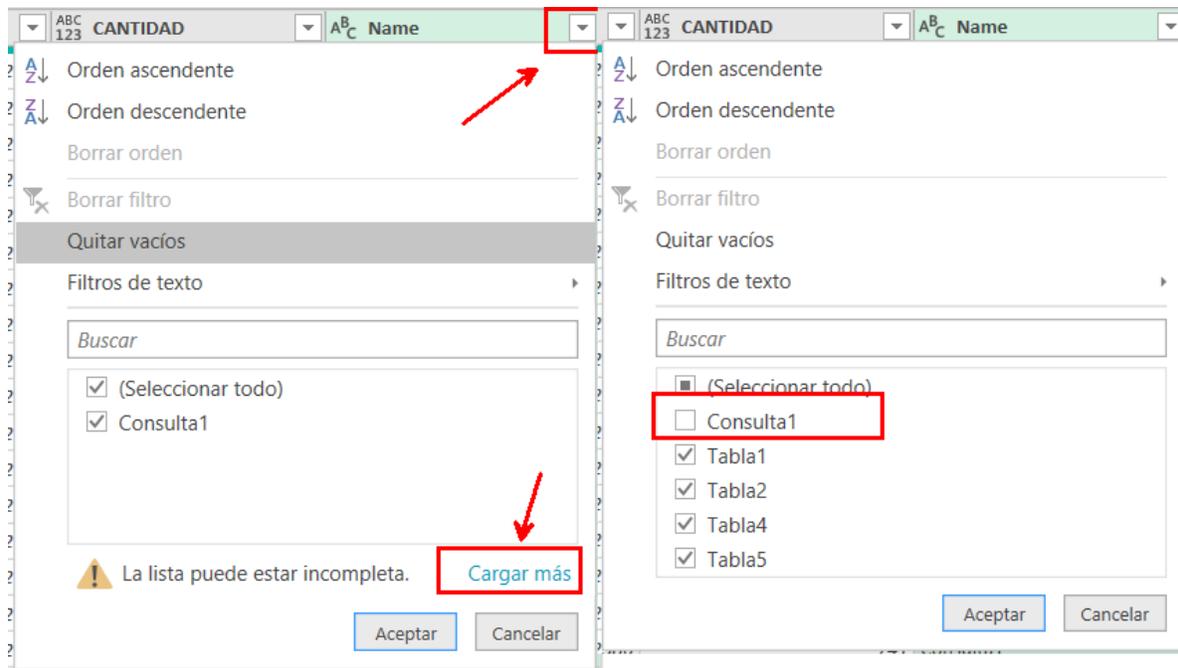


Figura 5.13 Deshabilitar campo Consulta1.

Volvemos a la pestaña Inicio → Cerrar y cargar

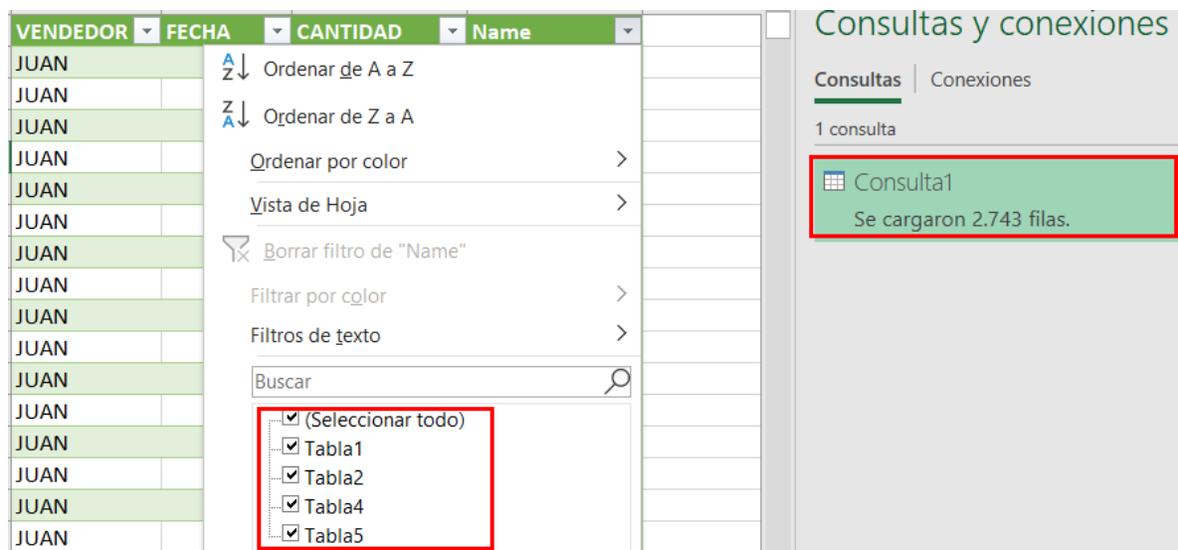


Figura 5.14 Datos cargados correctamente.

Y así es como solucionamos este inconveniente de tener la información duplicada.

Aprovechar el nombre de las Tablas

En el libro tenemos 4 hojas, que hacen parte de diferentes países, pero el nombre del país solo podemos identificar en la hoja, no como parte de la tabla, y así en el consolidado es difícil saber cada información a que país pertenece.

Debemos sacar ventaja a los nombres de las tablas, y renombrarlas según el país correspondiente.

Nos ubicamos en la hoja Colombia → Seleccionamos una celda de la tabla → Diseño → Nombre de la tabla → Renombramos

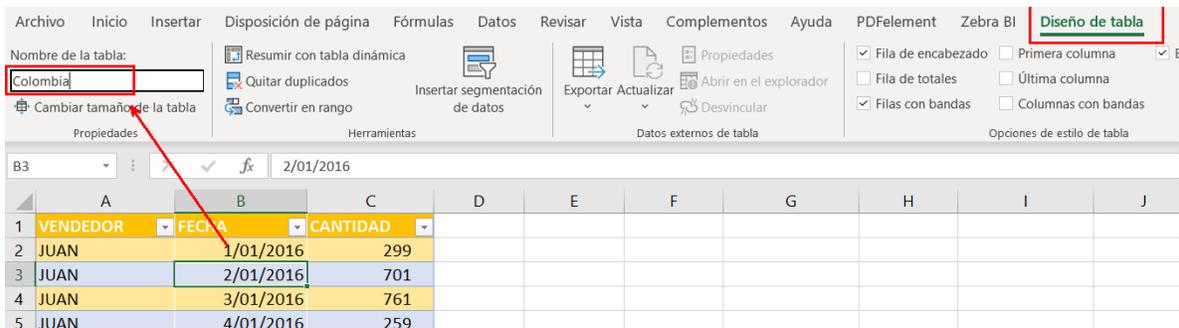


Figura 5.15 Renombrar tabla.

Este mismo proceso lo repetimos para los países Argentina, Perú y Chile.

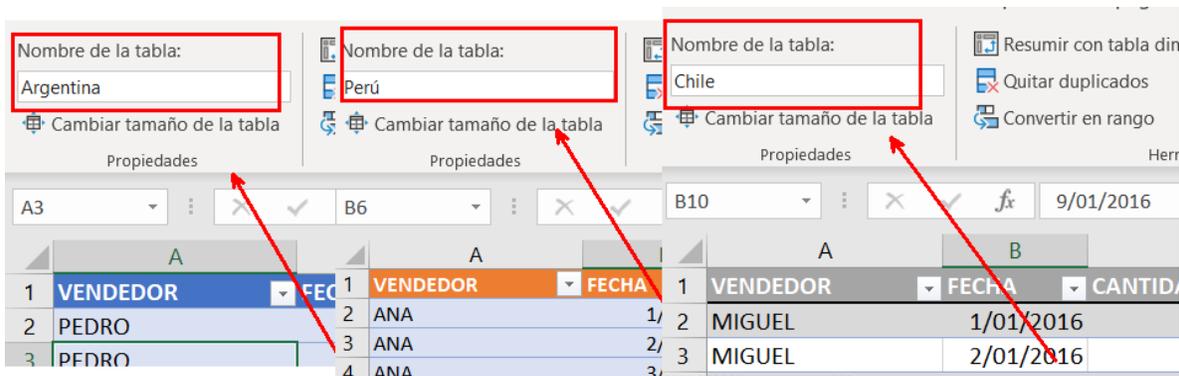


Figura 5.16 Renombrar tablas de otros países.

Volvemos a la hoja *Consulta1* → Clic derecho en la tabla → Actualizar y ahora tenemos el nombre del país como campo en los datos y vemos la información según figura 5.17.

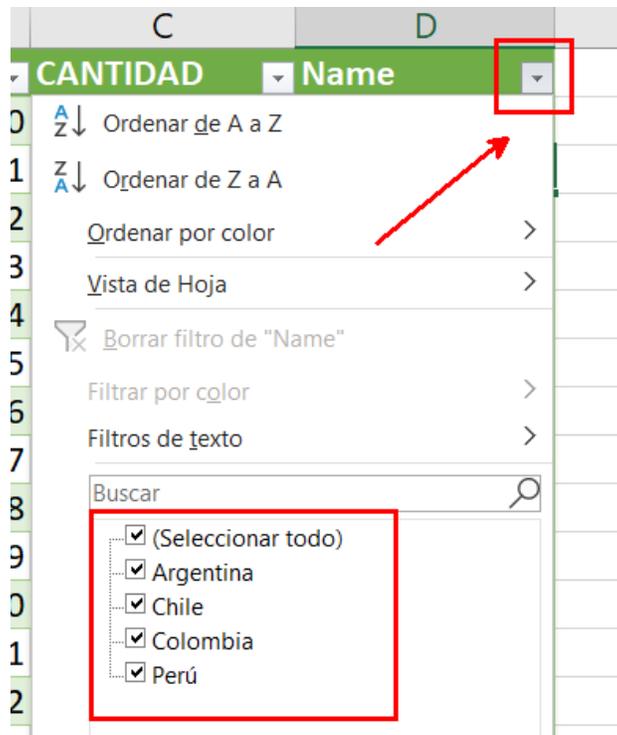


Figura 5.17 Nombre los países como campo.



ADVERTENCIA: Para sacar provecho de este gran truco debemos tener la misma estructura en cada tabla

Qué tal si probamos su funcionamiento, creemos una hoja nueva con nombre de país Brasil, con pocos datos para hacer la prueba, inventa datos y hagamos juntos la prueba.

Nombre de la tabla: Brasil

Resumir con tabla dinámica
 Quitar duplicados
 Convertir en rango
 Insertar segmentación de datos

Cambiar tamaño de la tabla

Propiedades Herramientas

C2 622

	A	B	C	D
1	VENDEDOR	FECHA	CANTIDAD	
2	DIDIER	1/01/2021	622	
3	DIDIER	2/01/2021	994	
4	DIDIER	3/01/2021	586	
5	DIDIER	4/01/2021	857	
6	DIDIER	5/01/2021	993	
7	DIDIER	6/01/2021	260	
8	DIDIER	7/01/2021	481	
9	DIDIER	8/01/2021	613	
10	DIDIER	9/01/2021	931	
11				
12				

Consulta1 Colombia Argentina Perú Chile Brasil

Figura 5.18 Anexar nueva tabla Brasil.

Volvemos a la hoja Consulta1 y damos Clic derecho Actualizar.

La figura 5.19 muestra que efectivamente los datos de la tabla nueva que anexamos hacen parte del consolidado.

Y así con este tip tenemos un proceso automatizado a un Clic.

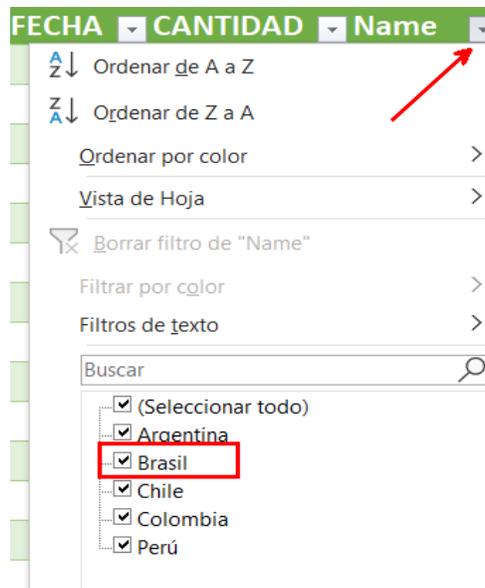


Figura 5.19 Datos de Brasil anexados.



HACKS: Hemos finalizado nuestro primer truco de Power Query, consolidar todas las tablas en mismo archivo, que los disfrutes.

5.2 Ruta dinámica

Cuando se consolida archivos desde una carpeta mediante Power Query, y esta conexión es local, la ruta mediante esta redireccionada se debe cambiar directamente desde el editor de consultas si otra persona deseara trabajar con este archivo y hacer cambios en Power Query tendría que entrar directamente a la interfaz y hacer sus ediciones, lo que significa que este usuario debe saber la herramienta de Power Query. Como analistas y desarrolladores de este proceso debemos garantizar la mejor experiencia posible para nuestros clientes internos y externos.

Este Hacks trata de cómo mejorar esta experiencia, solo la persona debería ingresar en una celda de Excel la ruta donde se encuentran los archivos y actualizar y listo, problema solucionado.

Para iniciar, abrimos el archivo de Excel llamado *2. Ruta Dinámica*, el cual ya tiene una conexión desde una carpeta con archivos de Excel.

Vamos al panel de consultas y conexiones → Clic derecho Editar

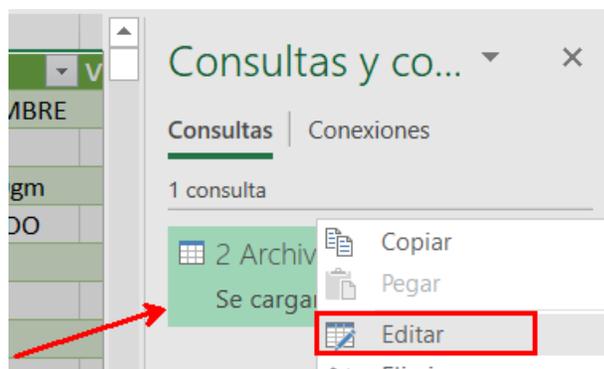


Figura 5.20 Editar consulta conexión carpeta.

Al ingresar al editor de consultas, te va a salir algo como esto:

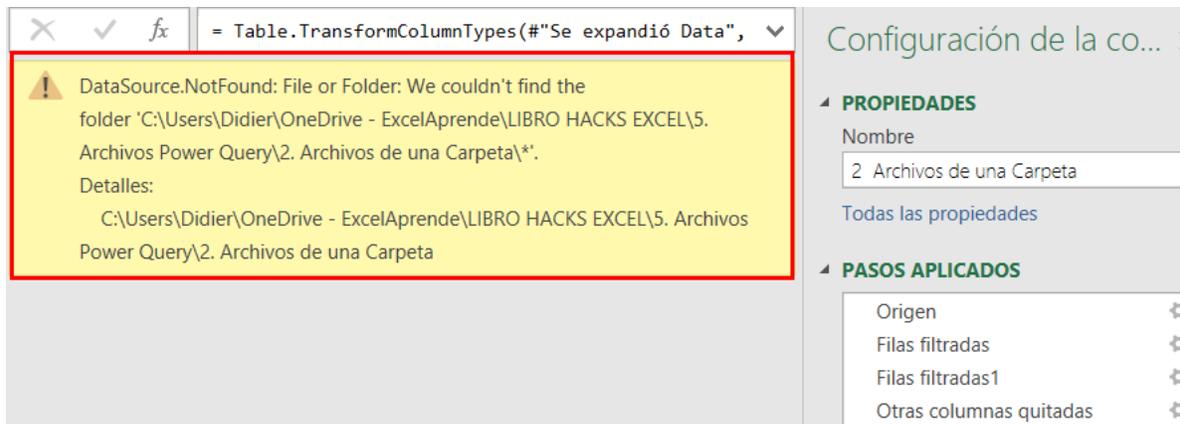


Figura 5.21 Error en editor de consultas.

Evidentemente hay un error, lo que sucede es que la ruta de donde están los archivos no fue encontrada, es lógico, tú no tienes acceso a esta ruta local, es un error muy común, ya sea por cambio de ruta o cambio de nombre en la carpeta.

Para que soluciones esto de manera manual, debes permanecer ahí y nos dirigimos a la configuración de consultas, el primer paso llamado *Origen* → Editar

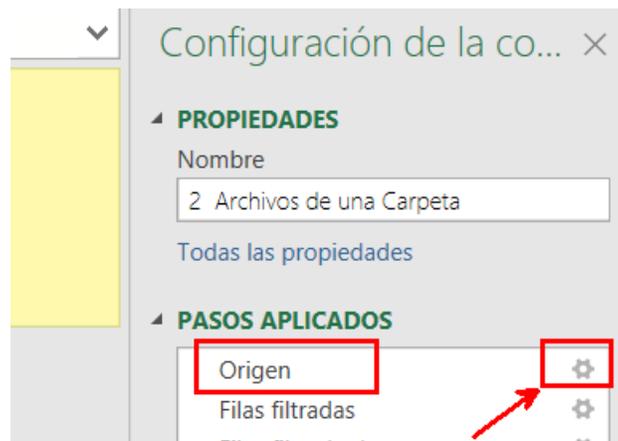


Figura 5.22 Cambiar ruta de archivos.

Se nos habilita una ventana donde ingresamos la nueva de los archivos o presionamos en examinar para redireccionar y buscar la carpeta.

Carpeta

Ruta a la carpeta

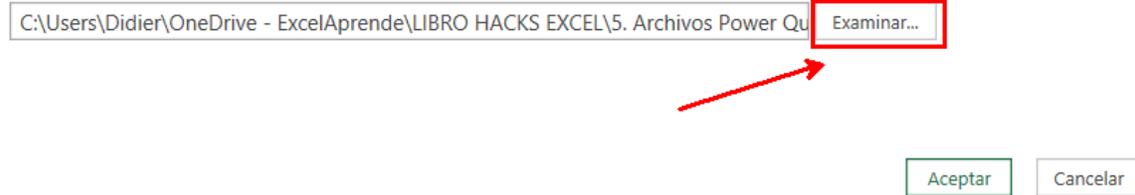


Figura 5.23 Pegar nueva ruta.

Aceptamos y tenemos de nuevos nuestros datos para ser editados, transformados y cargados.

Pero como notamos, es un proceso muy manual y para un usuario final, se vuelve hasta técnico o complejo realizarlo, podría dañarte el proceso que llevas, para evitar esto vamos a crear esta gran truco.

Volvamos a Excel → Inicio → Cargar y cerrar

Creamos una hoja nueva y vamos a nombrar un encabezado llamado ruta

Debajo del encabezado pegamos la ruta de la carpeta donde se encuentran tus archivos de Excel.

Y por último convertimos esto a tabla y habilitamos que la tabla tiene encabezado.

Ver figura 5.24

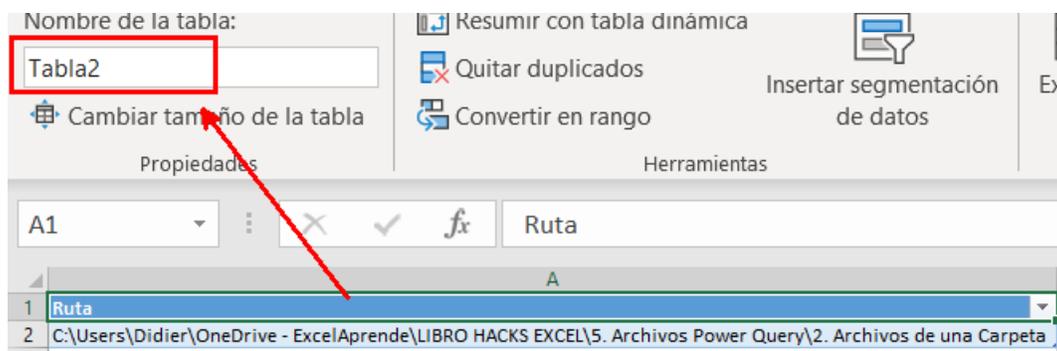


Figura 5.24 Tabla Creada para ruta dinámica.

Luego de crear la tabla con un solo registro donde se encuentra la ruta de la carpeta, debemos, cargar esta tabla a Power Query.

Vamos a la pestaña Datos → Grupo obtener y transformar datos → de una tabla o rango (From Table/Range)

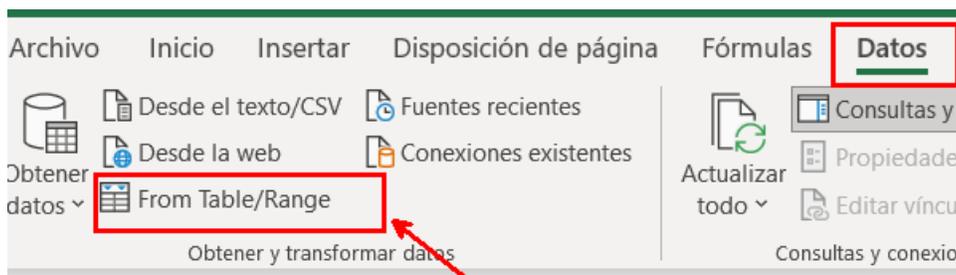


Figura 5.25 Cargar tabla a Power Query.

Power Query muestra esta tabla como una consulta, llamada Tabla2, para un mejor orden cambiamos el nombre de la consulta a → Ruta

Clic derecho encima de la consulta → Cambiar nombre → Ruta

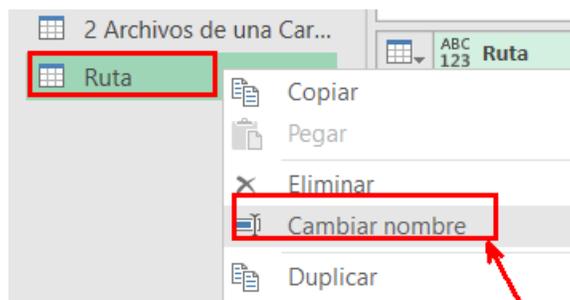


Figura 5.26 Cambiar nombre consulta Ruta.

El reto para lograr esta automatización es podemos quedarnos solo con el registro donde se encuentra la ruta, es decir, debemos quitar el encabezado de la tabla y que solo me muestre este registro.

Vamos a la pestaña Transformar → Convertir en lista →

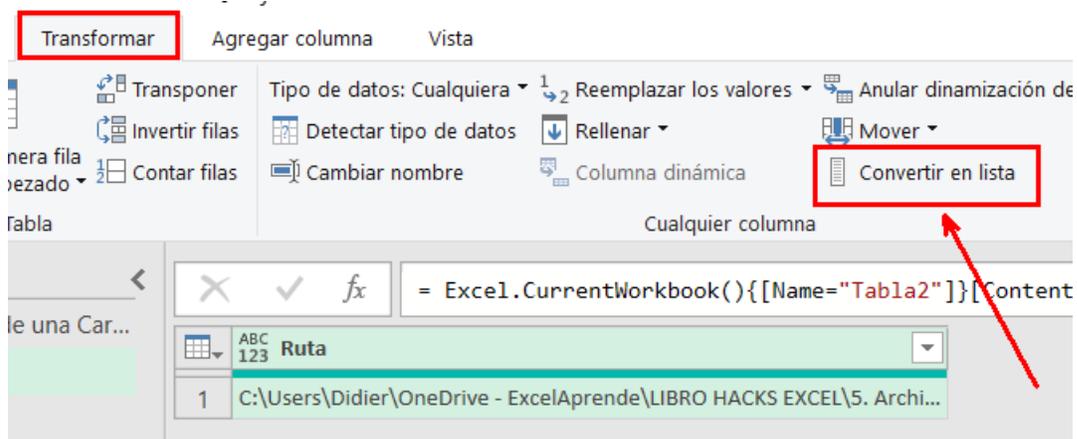


Figura 5.27 Convertir en lista.

Estamos a un solo paso de obtener solo el registro de la ruta.

Nos ubicamos en la barra de fórmula → Vamos al final de dicha fórmula → y abrimos llaves **{0}**

Ver figura 5.28

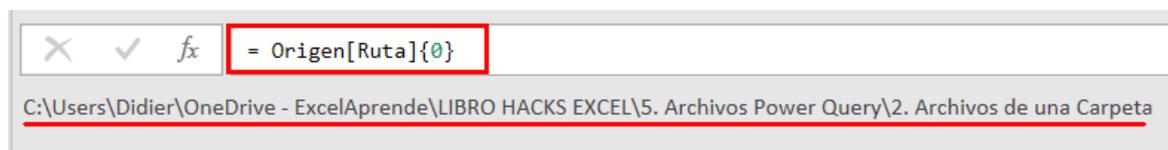


Figura 5.28 Extraer el primer registro de una consulta.

Te estarás preguntando, que significa eso {0}, pues Power Query trabaja en base cero, lo que significa que el primero registro no es el número 1, es el cero, y las llaves significa traer un registro de una lista.

Por último, es ir al código que genera Power Query de manera automático, lo que llamamos el lenguaje M y parametrizar que la ruta no sea constante, sino dinámica, que dependa de la celda que está en Excel.

Clic derecho en la consulta *2 Archivos de una Carpeta* → Editor Avanzado

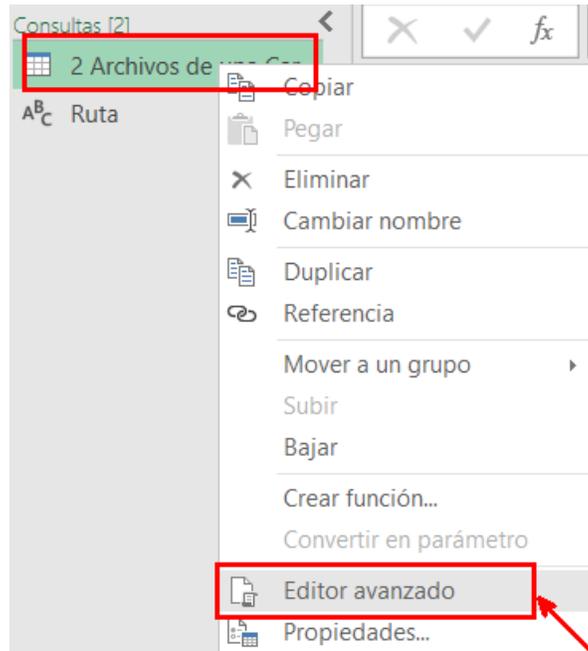


Figura 5.29 Habilitar editor avanzado.

Y con mucho cuidado, casi que con pinzas vamos a eliminar la ruta que está entre comillas e ingresamos o llamamos la consulta llamada "Ruta". Como se muestra en la figura 5.30

2 Archivos de una Carpeta

```
let
  Origen = Folder.Files("Ruta"),
  #"Filas filtradas" = Table.SelectRows(Origen, each not Text.StartsWith([Name], "~")),
  #"Filas filtradas1" = Table.SelectRows(#"Filas filtradas", each [Extension] = ".xlsx"),
  #"Otras columnas quitadas" = Table.SelectColumns(#"Filas filtradas1", {"Content"}),
  #"Personalizada agregada" = Table.AddColumn(#"Otras columnas quitadas", "Personalizado", each
  #"Columnas quitadas" = Table.RemoveColumns(#"Personalizada agregada", {"Content"}),
  #"Se expandió Personalizado" = Table.ExpandTableColumn(#"Columnas quitadas", "Personalizado",
  #"Otras columnas quitadas1" = Table.SelectColumns(#"Se expandió Personalizado", {"Data"}),
  #"Se expandió Data" = Table.ExpandTableColumn(#"Otras columnas quitadas1", "Data", {"Años",
  #"Tipo cambiado" = Table.TransformColumnTypes(#"Se expandió Data", {"VENTAS", Int64.Type})
in
  #"Tipo cambiado"
```

Figura 5.30 Código ajustado.

Ahora con este nuevo código ajustado, damos en la tecla Listo → Pestaña inicio → Cerrar y cargar

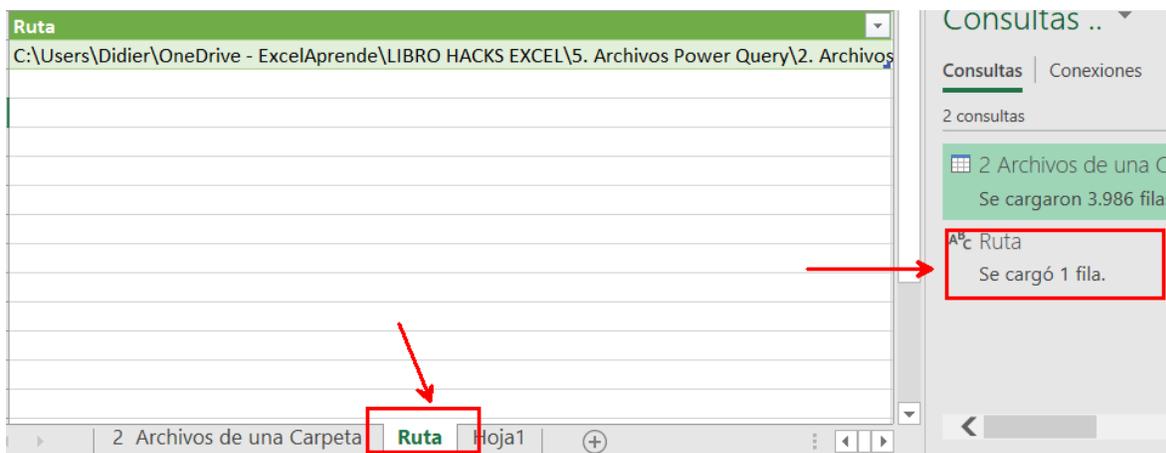


Figura 5.31 Consulta cargada.

Aparece una nueva hoja, llamada Ruta, mientras llegamos a los siguientes trucos, vamos a ocultar esta hoja para evitar confusiones

Ahora probemos su funcionamiento, vamos a la Hoja 1 e ingresamos una nueva ruta de donde se encuentran los archivos

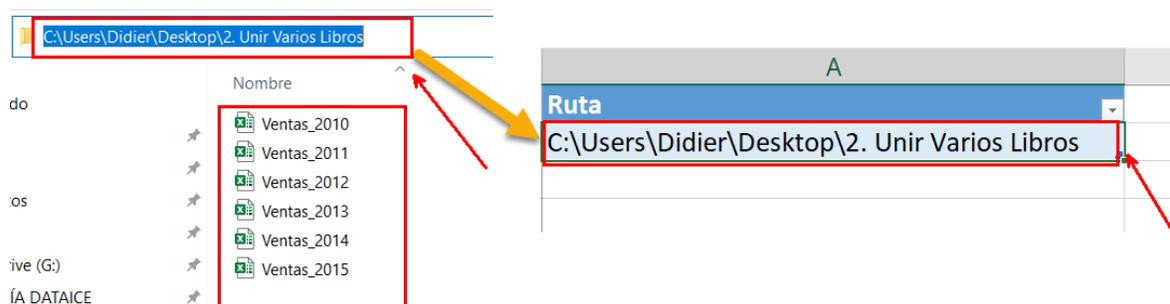


Figura 5.32 Copiar y pegar ruta.

Regresamos a la hoja donde se encuentran nuestros datos consolidados.

Pestaña Datos → Actualizar todo

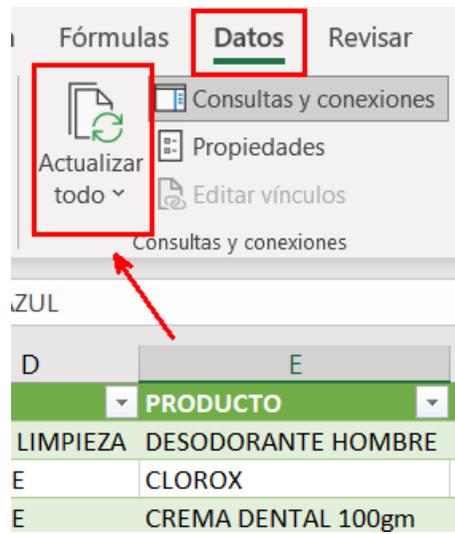


Figura 5.33 Actualizar consultas.



HACKS: Con este truco puedes lograr cosas sorprendentes, garantizas trabajo en equipo, evitar errores en cambios manuales de ruta y automatización de direcciones locales.

5.3 Segmentar o filtrar información mediante listas en Excel

Cuando trabajamos con Power Query es muy constante filtrar información en las consultas, aunque se conecta a las fuentes de información o bases de datos completas, no siempre se requiere todos los datos descargados a Excel, por ende, se hacen filtros para segmentar la información, aunque se hace desde el editor de Power Query y es una buena práctica no es muy útil cuando requiero traer información de múltiples ítems o campos.

Abrimos el archivo *3.1 Filtros*, y vemos que hay una cantidad de registros significativos de la base de datos, ya que se encuentran toda la data.

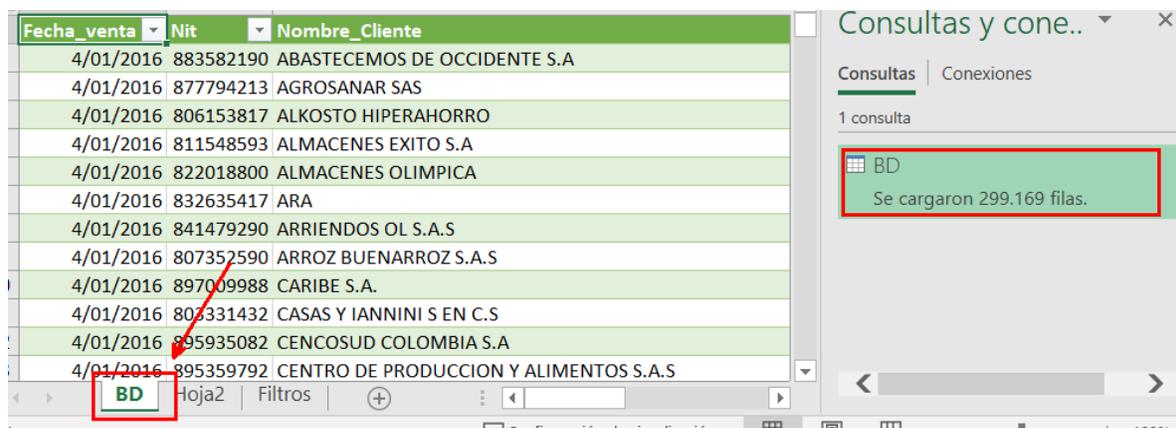


Figura 5.34 Base de datos completa.

Si queremos filtrar la información, por ejemplo, por ciudad debemos ir a la consulta BD y dar Clic derecho editar. Ver figura 5.12

Estando ubicados en el editor de Power Query, nos desplazamos a la columna *Municipio*

Y podemos realizar un filtro para llevar solo los datos estrictamente necesarios, por ejemplo, solo queremos seleccionar los municipios de BELLO Y APARTADÓ.

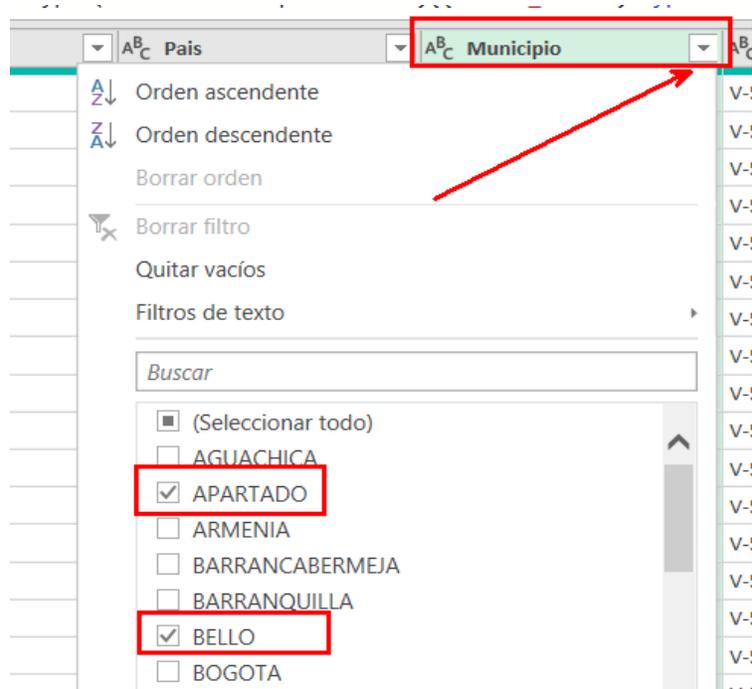


Figura 5.35 Filtrar municipios.

Damos aceptar → Inicio → Cerrar y cargar

Al llevar la nueva data a Excel, observamos claramente que la data se redujo significativamente, pasó de 299.169 registros a 1.879 filas.

Fecha_venta	Nit	Nombre_Cliente	Pais	Municipio	Cod_ve
4/01/2016	877794213	AGROSANAR SAS	COLOMBIA	APARTADO	V-5702
4/01/2016	806153817	ALKOSTO HIPERAHORRO	COLOMBIA	BELLO	V-5703
4/01/2016	801187872	RESTAURANTE CHEN JIAN LAI	COLOMBIA	APARTADO	V-5721
4/01/2016	856617987	RESTAURANTE HUFEN MEI	COLOMBIA	BELLO	V-5722
5/01/2016	843538728	HOTEL CABO DE LA VELA	COLOMBIA	APARTADO	V-5710
5/01/2016	812419431	HOTEL CALASANZ REAL	COLOMBIA	BELLO	V-5711
6/01/2016	890105895	HOLIDAY INN EXPRESS & SUITES M	COLOMBIA	APARTADO	V-5729
6/01/2016	854970747	HOSTAL CASA RAM	COLOMBIA	BELLO	V-5730
7/01/2016	806282673	LA ECONOMIA	COLOMBIA	APARTADO	V-5718
7/01/2016	898177269	LA GRAN ESQUINA	COLOMBIA	BELLO	V-5719
8/01/2016	840717872	AUTOSERVICIOS EL CALIDOSO	COLOMBIA	APARTADO	V-5707
8/01/2016	835946444	AUTOSERVICIOS ESTRELLA DE ORI	COLOMBIA	BELLO	V-5708

Consultas ...

Consultas | Conexiones

1 consulta

BD
Se cargaron 1.879 filas.

Figura 5.36 Base de datos filtrada.

Esto es una buena práctica y es lo normal que se hace o como se hace, pero que sucede si tenemos una cantidad de municipios que se deban filtrar o aún mas interesante, que cada persona, usuario o desarrollador pueda traer y cambiar de manera automática.

Volvamos a nuestro archivo, y en la hoja *Municipios*, hay una tabla llamada Filtros, la cual queremos usar para filtrar nuestra Data a partir de estos registros que tenemos allí.

Carguemos esta tabla a Power Query → Datos → Obtener y transformar datos → De una tabla o rango. Ver figura 5.25

Vamos a repetir el proceso antes visto, de convertir esta tabla como lista.

Pestaña Transformar → Convertir lista

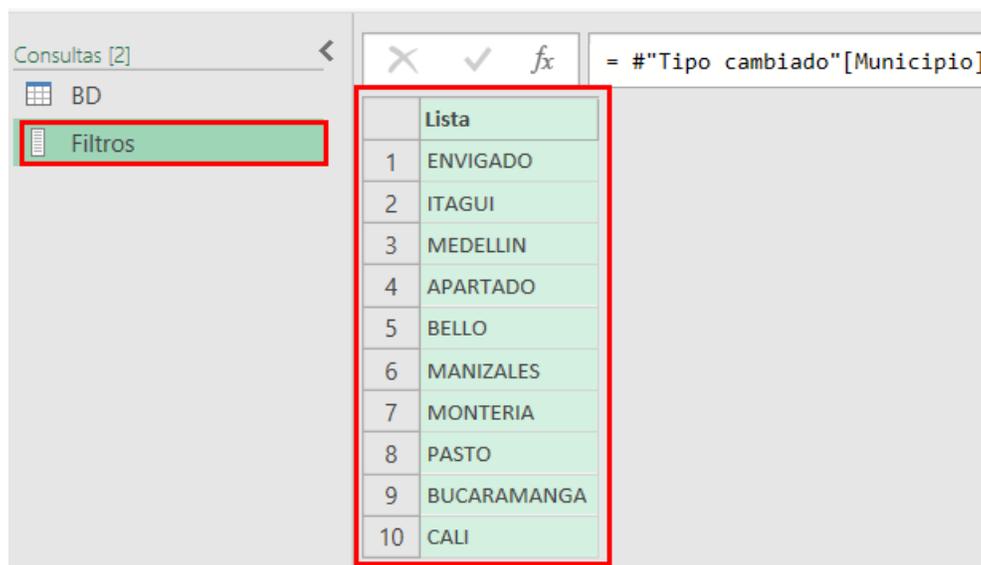


Figura 5.37 Tabla convertida a lista.

Volvemos a la consulta BD, para implementar una fórmula que valide contra la columna Municipio y la lista Filtros.

Antes de entrar a este detalle, borremos el paso aplicado de filtrar los municipios manuales, como vimos anteriormente, ver figura 5.35.

Consulta BD → Propiedades de consultas → Borrar paso → Filas filtradas

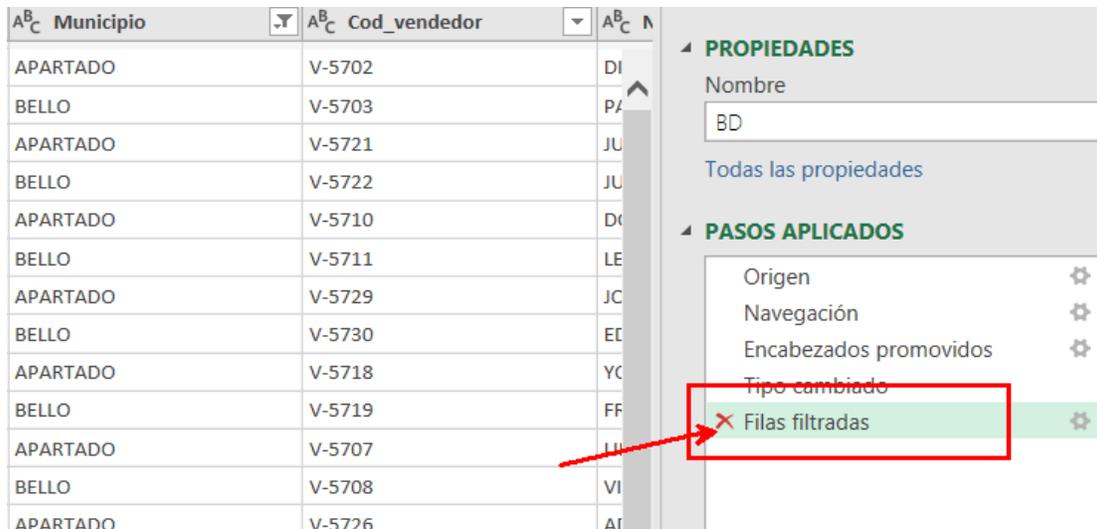


Figura 5.38 Borrar paso aplicado.

Vamos a crear una columna personalizada, donde validamos los municipios que están en la BD y en la lista Filtros.

Columnas personalizadas

Pestaña Agregar columna → Columna personalizada

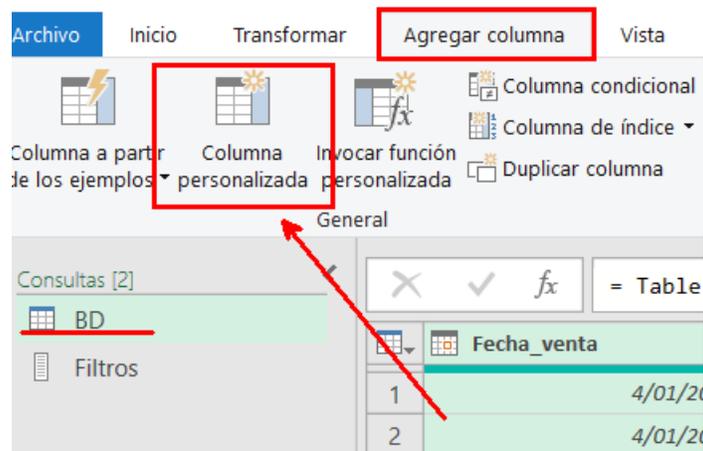


Figura 5.39 Agregar columna personalizada.

En el cuadro de dialogo que se habilita, vamos a ingresar la fórmula que me válida si cada registro que esta en la columna Municipios de la tabla BD se encuentra en la lista Filtros.

Aceptamos y nos crea una columna personalizada con verdaderos y falsos (TRUE /FALSE), lo cual ya podemos intuir que si dice TRUE (Verdadero) es que, porque este registro hace parte de la lista, lo cual lo debemos incluir. Ver figura 5.41.

Por último, realizamos un filtro en esta columna personalizada.

Columna Personalizada → Filtro y selecciono solo TRUE

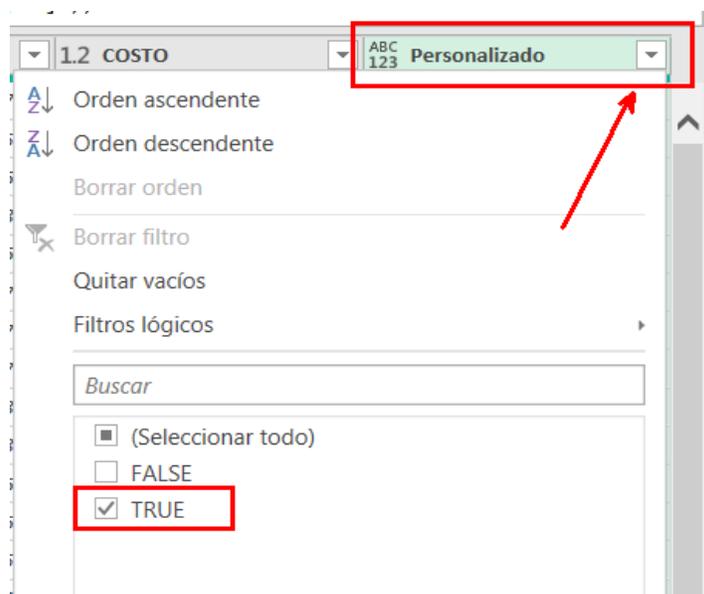


Figura 5.42 Filtrar solo TRUE.

Volvemos a Inicio → Cerrar y cargar

En Excel validamos que solo y exclusivamente me carga los Municipios que están en la tabla Filtros.

Adicional me crear una consulta u hoja nueva con estas listas, mientras llegas el truco de cargar datos como solo conexión, ocultamos dicha hoja y vemos nuestra información se la siguiente manera:

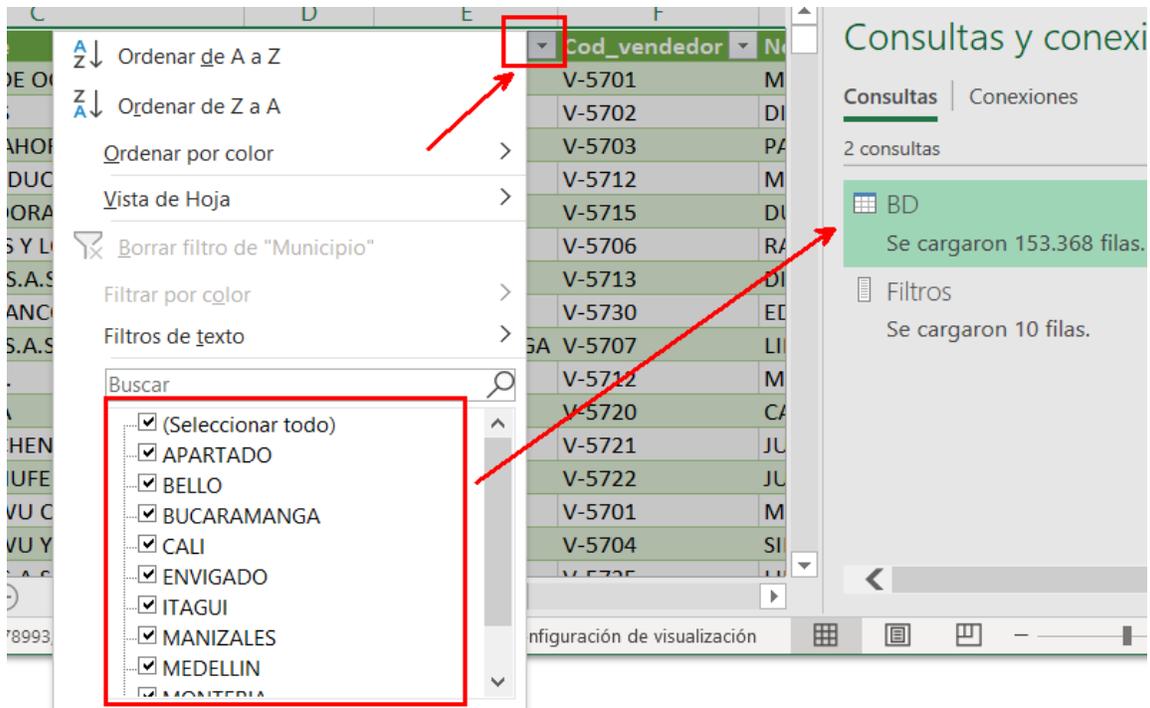


Figura 5.43 Tabla filtrada con varios municipios.



ADVERTENCIA: Este es un proceso un poco pesado, ya que se válida cada fila en el campo Municipio si se encuentra en la lista de Filtros, se debe contar con buenas características en el equipo o PC en que se corra este proceso.



NOTA: La lista de municipios puede estar en el mismo libro de Excel, o en un libro aparte, o ser parte de una lista de Share Point o en OneDrive o cualquier base de datos



HACKS: Este truco vale oro como la famoso frase de cajón, ya que permite segmentar información siendo controlada directamente desde Excel.

5.4 Crear tabla calendario automática en Power Query

Las tablas calendarios son un insumo muy relevante cuando se trabajaban con modelos de datos, es casi imposible no trabajar con estas tablas ya que tienen demasiadas ventajas, en cuanto a relación de tablas, fórmulas para inteligencia de tiempo y demás.

En Excel se pueden crear estas tablas de manera manual, es decir, pones una fecha inicial, y hacer uno relleno hasta que fecha se quiere llegar, de ahí extraer los demás campos como año, mes, día etc., es válido realizar esta operación, pero Power Query trae el poder para que esta tabla se cree de manera automática y lo mejor que el usuario final o desarrollador tenga la posibilidad de ingresar desde y hasta donde quiere ver las fechas y todas las columnas de año, trimestre, mes, semana, día etc. se creen de manera continua.

Vamos al archivo 4. *Tabla Calendario*, y vemos en Hoja 1 ya dos tablas predeterminadas de fecha inicial y fecha final, estas tablas ya están cargadas en Power Query y convertidas en listas, ver figura 5.25 y 5.27 para repasar como se lleva una tabla a Power Query y se convierte en lista.

Vamos a crear una consulta nueva que a partir de estas dos tablas se genere una lista de fechas.

Vamos a la pestaña Datos → Obtener y transformar → Desde otras fuentes → Consulta en blanco.

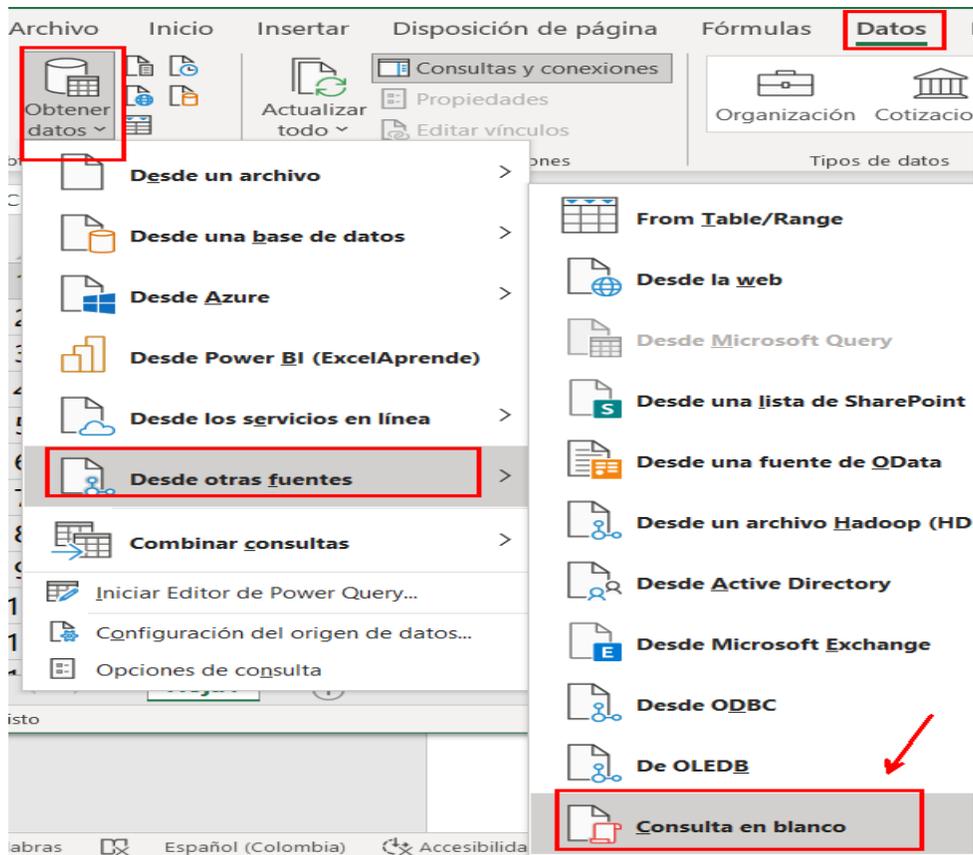


Figura 5.44 Crear consulta en blanco.

En el editor de Power Query, la barra de fórmulas se activa de manera automática, lista para ingresar esta función que nos devuelve como resultado una lista de fechas.



FUNCIÓN: List.Dates()

Esta es una función de lista.

=List.Dates, tiene 3 parámetros, la fecha inicial, el número de registros que va contener esta fecha y a qué nivel de periodo va crecer, día, hora, minuto y segundo.

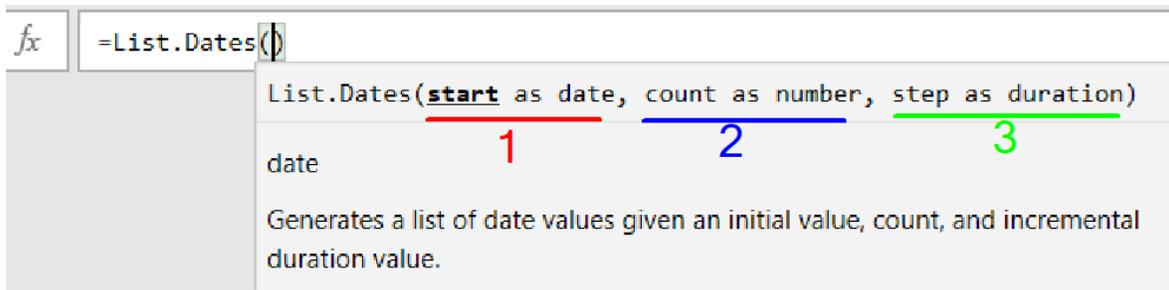


Figura 5.45 Sintaxis función List.Dates.

1. **Start as date:** es la fecha inicial, y de ser un campo fecha, ya está cargada en Power Query
2. **Count as number:** es un número entero, y es el número de registros que va a contener dicha lista, si queremos que sea dinámico, debemos crear un cálculo entre la fecha inicial y la fecha final
3. **Step as duration:** con qué frecuencia va crecer, día, hora, minuto o segundo.

Con estos argumentos claro procedamos a realizar un ejemplo de esta lista.

En la barra de fórmula ingresamos la fecha inicial, pongamos como número de registro el número 10 para este primer ejemplo, y en el 3 argumento, debemos llamar la funcionalidad de #duration(1,0,0,0)

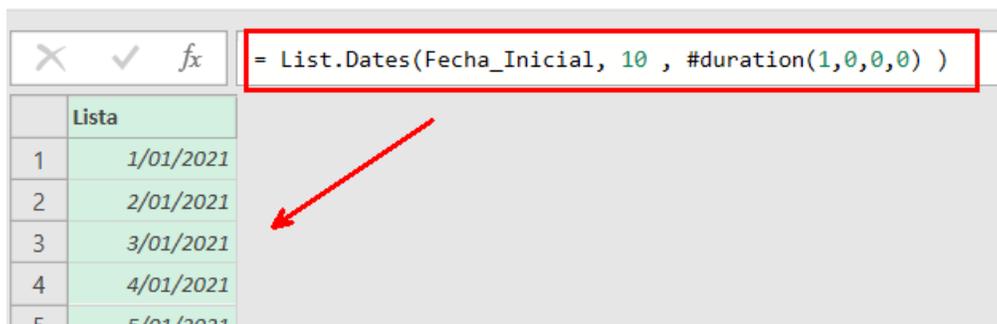


Figura 5.46 Lista de fechas.

Lo único raro que te puede parecer es este #duration(1,0,0,0), esto es Power Query, el tipo de dato en cada argumento juega un papel muy muy importante, lo más normal era ingresar

solo el número 1 para indicarle que es día, pero no, debes prefijarle el #duration, para identificar la duración entre fechas.

Hemos logrado lo más difícil, solo queda calcular este número 10 de manera dinámica, es simple, restar la fecha fin de la fecha inicio.

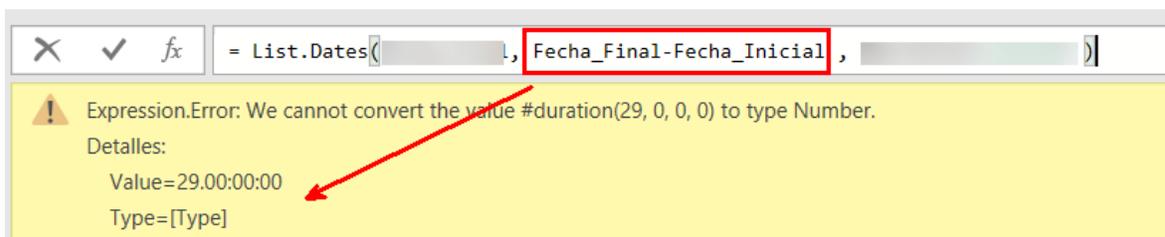


Figura 5.47 Calcular días entre fechas.

¿Algo hicimos mal? Pues no, simplemente es Power Query, este error se debe al tipo de datos, es lo que nos esta diciendo, que no puede convertir un resultado de tipo duración a número entero, y es que el tipo de datos en Power Query es un dolor de cabeza para todo desarrollador de esta herramienta, en el Hacks 6, trataremos esto con mayor detalle.

Si lo que nos pide Power Query es que este resultado tiene que ser estrictamente un número entero, sencillo, por medio de una función, convertimos este valor a este tipo de datos solicitado, y le añadimos un +1 para que nos tome las fechas completas.

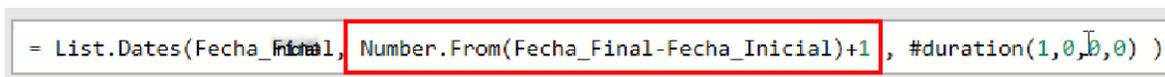


Figura 5.48 Convertir datos a número entero.

Number.From(), convierte cualquier dato a número entero y si damos enter tenemos una lista de fechas correctas.

Esta consulta es una lista, debemos convertirla en la tabla, cambiar el nombre de la consulta a calendario.

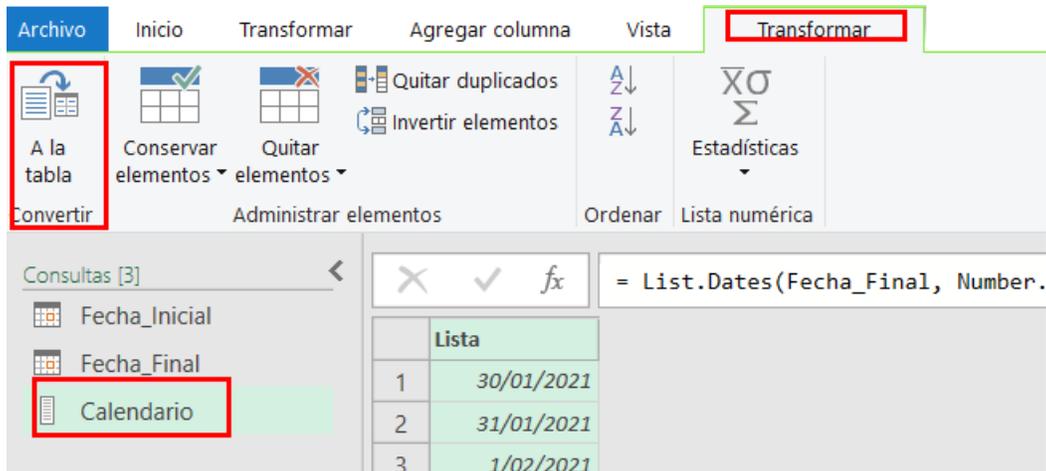


Figura 5.49 Convertir a tabla una lista.

Damos aceptar en el siguiente cuadro de dialogo.

Para sacar mayor provecho de las funciones que trae Power Query en cuestión de fechas, convertimos el tipo de datos a Fecha y cambiamos su nombre.

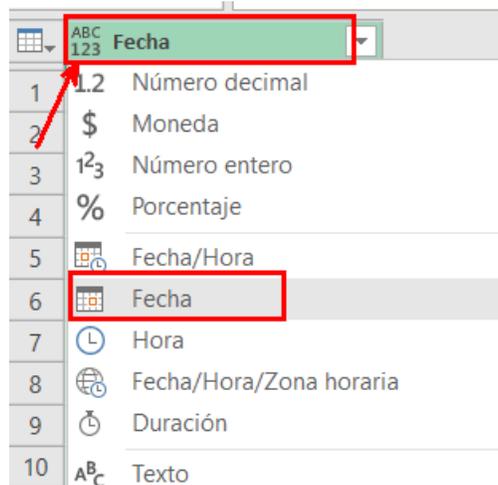


Figura 5.50 Convertir a fecha tipo de dato.

Lo siguiente es un juego de agregación de columnas, ya que gracias al potencial de Power Query, podemos desglosar esta fecha al nivel deseado (año, trimestre, mes, semana, día etc.)

Seleccionamos la columna Fecha → Pestaña Agregar columna → Grupo de Fecha y hora → Fecha

Y contamos con un portafolio de opciones para esto.

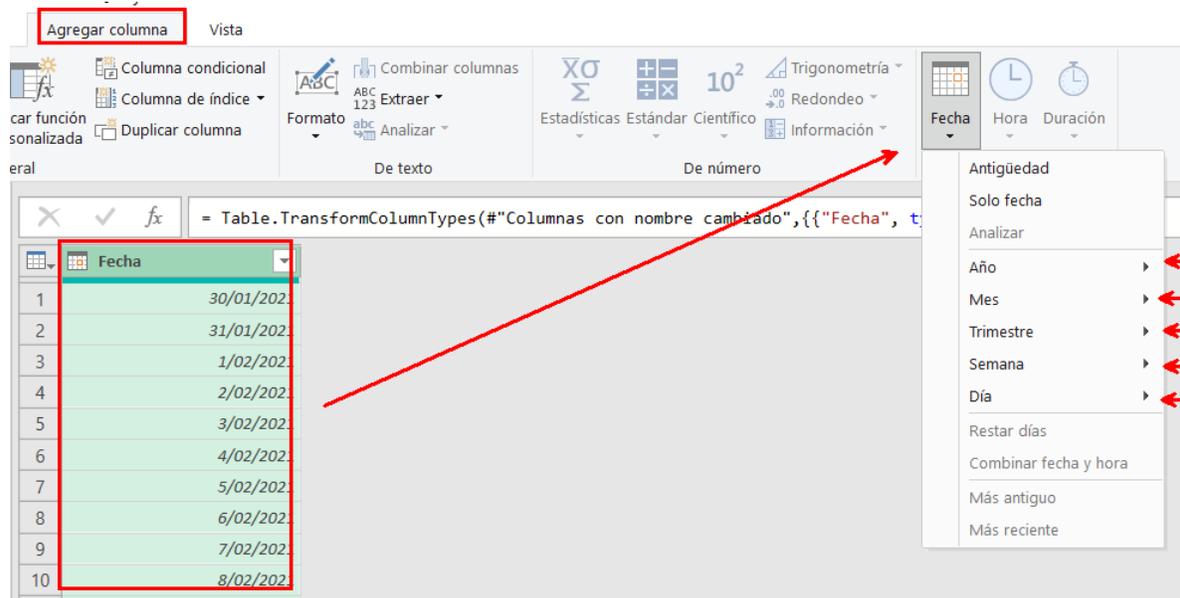


Figura 5.51 Extraer campos a la fecha.



ADVERTENCIA: Para que se habilite esta opción de traer o agregar más campos a la fecha, el tipo de dato de ser fecha.

Si das clic en año, no solo tenemos la opción de extraer el año, también el inicio de este año, el fin de este año, esto es válido para saber cuántos días faltan o van corrido a partir de una fecha.

Lo mismo sucede con las demás opciones, a esto llamamos inteligencia de tiempo, ya que a unos pocos clic tienes análisis a más detalle con periodos.

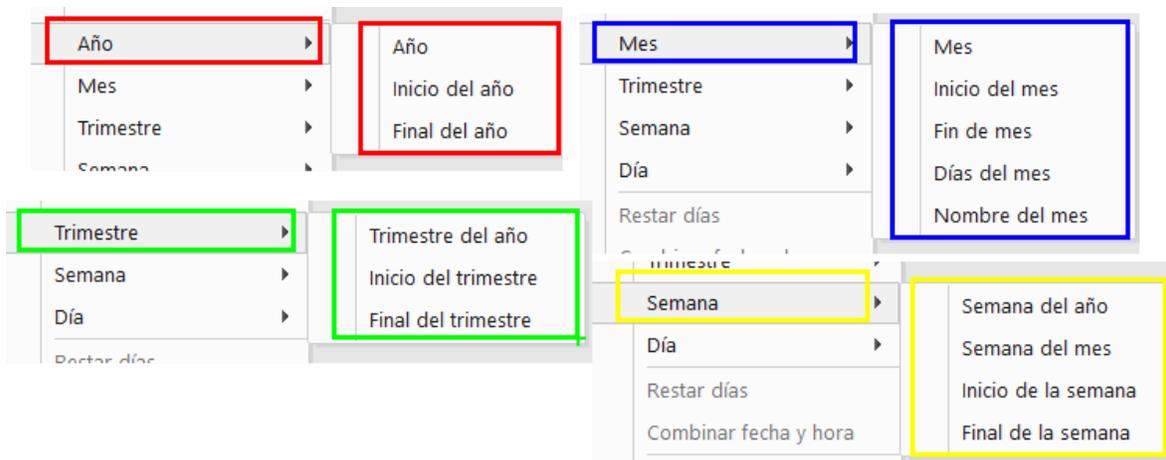


Figura 5.52 Opciones para desagregar fecha.

Practicemos, señalemos el campo fecha y pongamos el año, luego el mes, el trimestre, la semana y el número del día.

Fecha	Año	Mes	Nombre del mes	Trimestre	Semana del año	Día
30/01/2021	2021	1	enero	1	5	30
31/01/2021	2021	1	enero	1	6	31
1/02/2021	2021	2	febrero	1	6	1
2/02/2021	2021	2	febrero	1	6	2
3/02/2021	2021	2	febrero	1	6	3
4/02/2021	2021	2	febrero	1	6	4
5/02/2021	2021	2	febrero	1	6	5
6/02/2021	2021	2	febrero	1	6	6
7/02/2021	2021	2	febrero	1	7	7
8/02/2021	2021	2	febrero	1	7	8
9/02/2021	2021	2	febrero	1	7	9
10/02/2021	2021	2	febrero	1	7	10
11/02/2021	2021	2	febrero	1	7	11
12/02/2021	2021	2	febrero	1	7	12

Figura 5.53 Tabla Calendario.



NOTA: Para anexas más columnas, siempre se debe tener seleccionada la columna fecha.

Vamos a la pestaña Inicio → Cerrar y Cargar

Puedes probar cambiando los rangos de fecha desde el Excel e ir a la pestaña Datos y Actualizar todo



HACKS: Más que un truco es una herramienta que todo usuario de Excel debe tener a la mano, ya que permite crear fechas dinámicas con toda su desagregación.

5.5 Identificar campos faltantes entre dos tablas a un clic

Realizar cruces entre tablas es tan frecuente como realizar la función SUMA en Excel, ya que siempre necesitamos datos de otras tablas, hojas o libros, hay demasiados métodos para esto, especialmente en Excel, funciones tales como BUSCARV, ahora BUSCARX, INDICE etc.

En Power Query este proceso lo podemos realizar por combinar consultas, en este punto no es nuestra intención explicar todo este proceso, ya que en nuestro primer libro de Inteligencia de Negocios con Excel y Power BI lo explicamos a detalle y en demás jornadas que hacemos en vivo; este Hacks se trata exclusivamente se cómo traer información que falta o no se encuentra entre una tabla y otra.

Pongamos esto en contexto, imaginemos tenemos una base de datos de movimientos contables y otra tabla con el PUC (Plan Únicos de Cuentas), necesitamos saber que cuentas faltan crear en la tabla PUC que tuvieron movimiento y estas cuentas no se han creado. Establecer este control cada que se alimente la base de movimientos es de mucha utilidad, ya que a un clic vamos a tener este resultado.

Abrimos el archivo *5 Datos Faltantes*, hay dos consultas creadas, en Power Query, para veas de que te hablo, solo ve a la Pestaña datos → Consultas y conexiones.

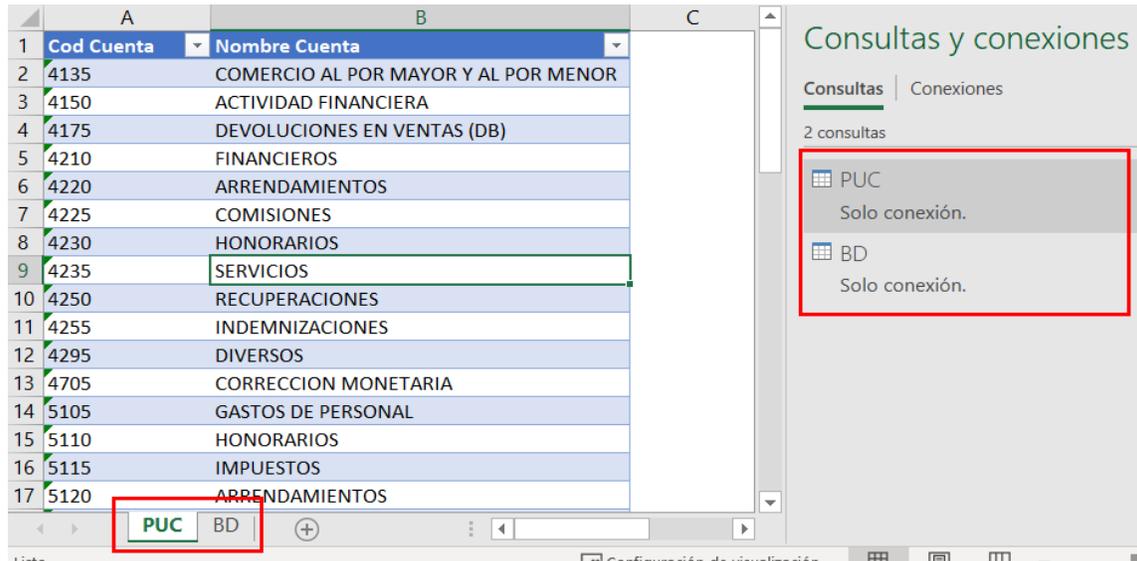


Figura 5.54 Consultas creadas.

Estas consultas están como solo conexión, truco que veremos en el Hacks 7 de este capítulo, por ahora nos concentramos en nuestra tarea de saber qué información falta de una tabla a otra.

Procedemos a dar clic derecho en una de estas consultas → Editar, para ir a Power Query.

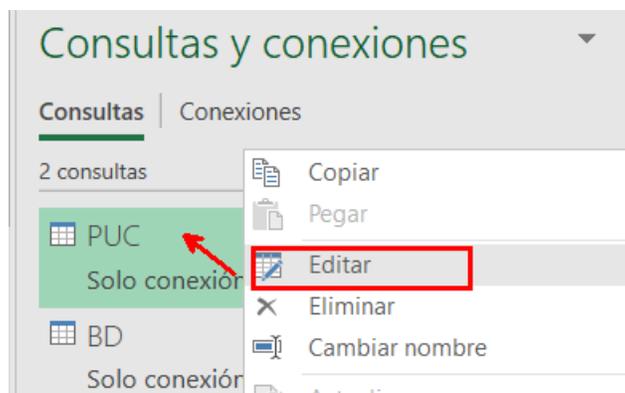


Figura 5.55 Editar consultas.

En el editor vamos a usar una de las grandes características que tiene Power Query, y es combinar consultas.

Con la consulta BD seleccionada → Inicio → Grupo Combinar → Combinar Consultas → Combinar consultas para crear una nueva.

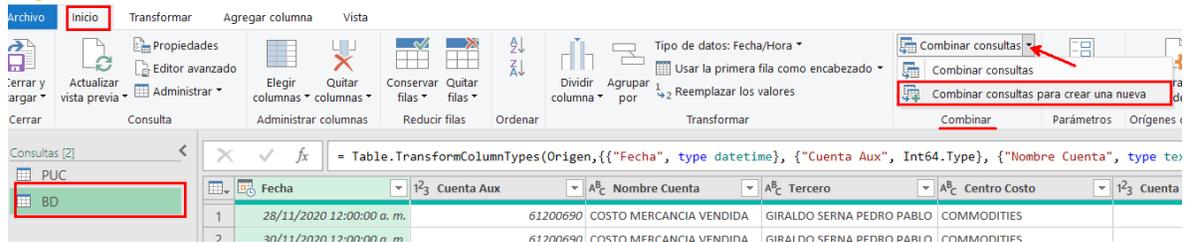


Figura 5.56 Crear nueva combinación.

En el cuadro de diálogo de la nueva ventana, debemos seleccionar la tabla PUC, es con la que se va a cruzar la información.

Combinar

Seleccione tablas y columnas coincidentes para crear una tabla combinada.

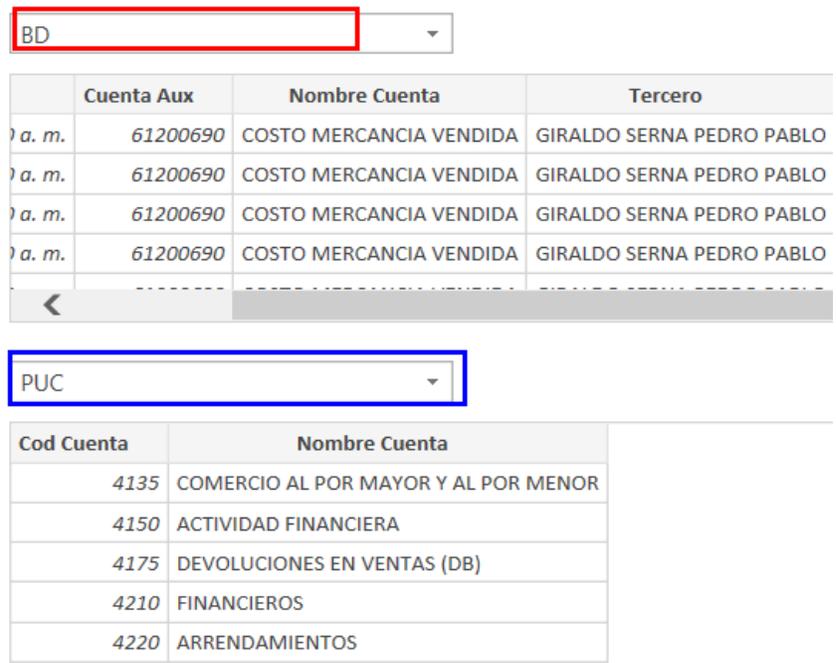


Figura 5.57 seleccionar consultas.

Seleccionamos los campos en común que están entre las dos tablas, para este caso sería la cuenta.

Power Query válida de manera automática el número de coincidencias que encuentra en las dos consultas. Ver figura 5.58

BD

	Cuenta Aux	Nombre Cuenta	Tercero	Centro Costo	Cuenta
a. m.	61200690	COSTO MERCANCIA VENDIDA	GIRALDO SERNA PEDRO PABLO	COMMODITIES	6120
a. m.	61200690	COSTO MERCANCIA VENDIDA	GIRALDO SERNA PEDRO PABLO	COMMODITIES	6120
a. m.	61200690	COSTO MERCANCIA VENDIDA	GIRALDO SERNA PEDRO PABLO	COMMODITIES	6120
a. m.	61200690	COSTO MERCANCIA VENDIDA	GIRALDO SERNA PEDRO PABLO	COMMODITIES	6120

PUC

Cod Cuenta	Nombre Cuenta
4135	COMERCIO AL POR MAYOR Y AL POR MENOR
4150	ACTIVIDAD FINANCIERA
4175	DEVOLUCIONES EN VENTAS (DB)
4210	FINANCIEROS
4220	ARRENDAMIENTOS

Tipo de combinación
 Externa izquierda (todas de la primera, coincidencias...)

Use las coincidencias aproximadas para comparar la combinación.

> Opciones de coincidencia aproximada

✓ La selección coincide con 5495 de 13003 filas de la primera tabla.

Aceptar

Figura 5.58 Número de registros que coinciden.



ADVERTENCIA: Si no se encuentra ninguna coincidencia y usted sabe que existen, lo más seguro es el tipo de dato, es decir, la columna **cuenta** debe estar en texto o en número iguales en las dos tablas.

La clave está en esto, si damos aceptar, el resultado son todos los datos de la consulta BD más los campos que se encuentra en el PUC, este es el comportamiento normal que usa todo usuario, pero pocos acceden a las otras características de *Tipo de combinación*.

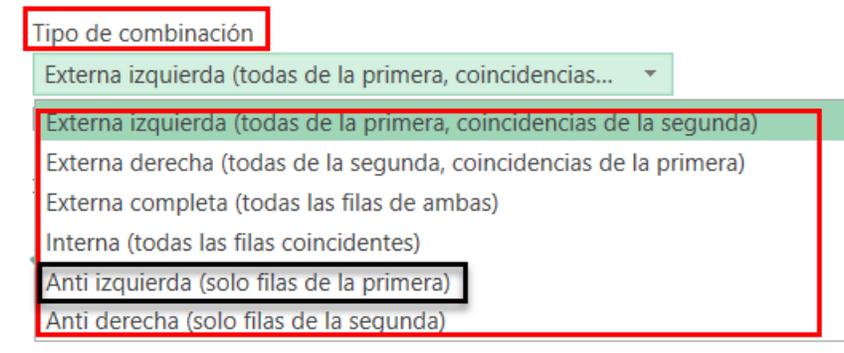


Figura 5.59 Tipo de combinación.

La opción que nos permite saber que cuentas están en movimiento y aún no están creadas en el PUC, es *Anti izquierda (solo filas de la primera)*.

Damos Aceptar, solo y únicamente trae estos registros de cuenta NO encontrados en el PUC.

Para un mejor orden vamos a eliminar las demás columnas aparte de cuenta, la idea es tener solo el control de aquellas cuentas que debemos crear.

Clic derecho en columna Cuenta → Quitar otras columnas

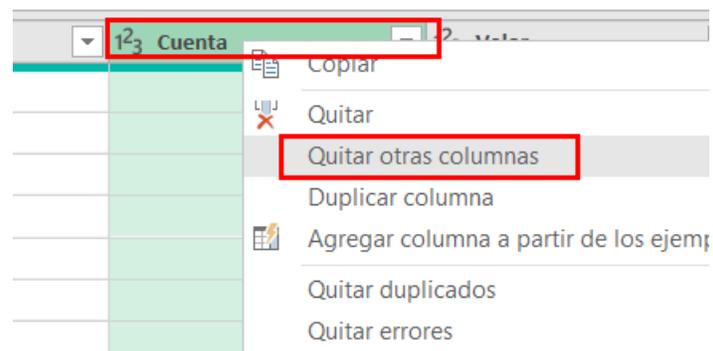


Figura 5.60 Quitar Otras columnas.

Para dejar los registros únicos, quitamos los duplicados de esta columna y ya tenemos lo deseado.

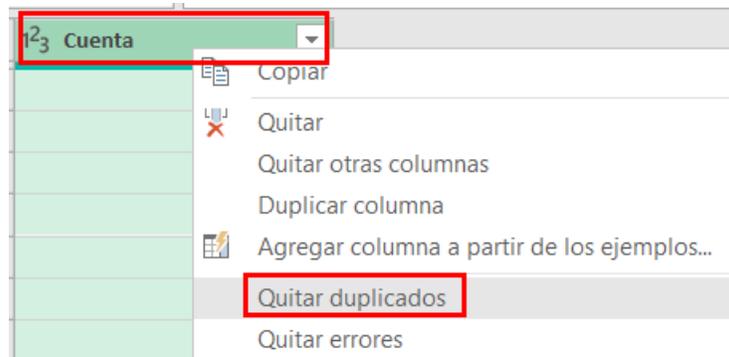


Figura 5.61 Quitar duplicados.

Por último cambiamos el nombre de la consulta → “Cuentas faltantes”

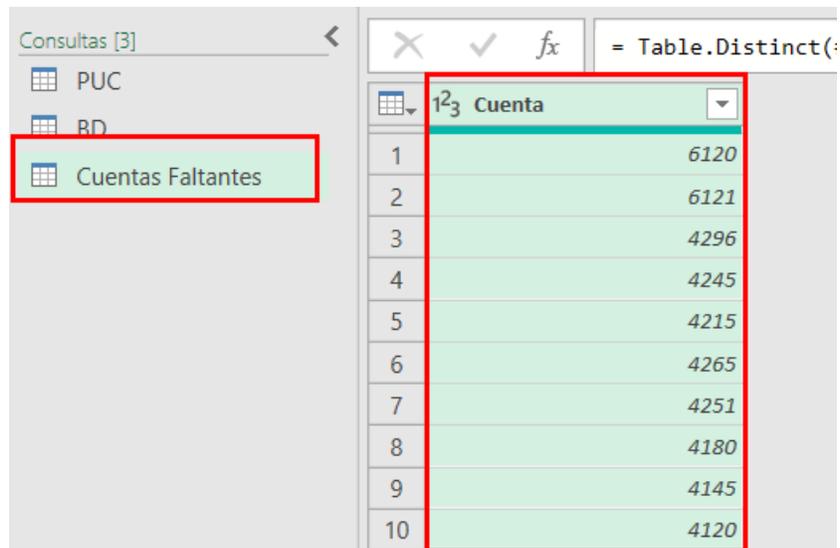


Figura 5.62 Cuentas únicas faltantes.

Pestaña Inicio → Cerrar y cargar.

Ya contamos con 10 registros en Excel, pertenecientes a las cuentas que no se han ingresado al PUC.

Volvamos al Excel para ver el detalle de este caso.

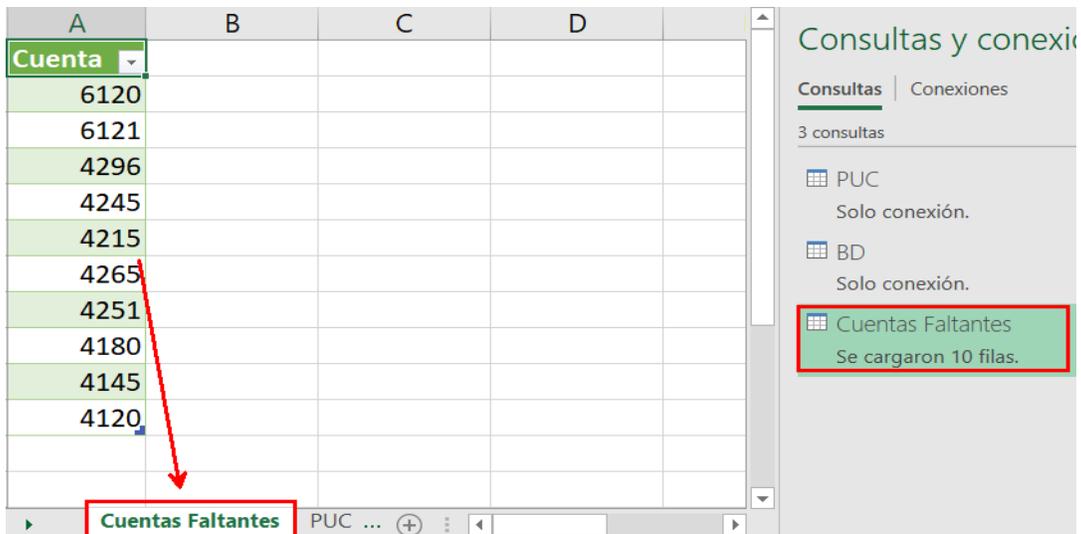


Figura 5.63 Tabla de cuentas faltantes en Excel.

Para probar su funcionamiento, simplemente vamos a la hoja PUC e ingresamos la cuenta 6120 y su nombre.

Volvemos a la hoja “Cuentas Faltantes” clic derecho actualizar, y listo, la saca de la lista.

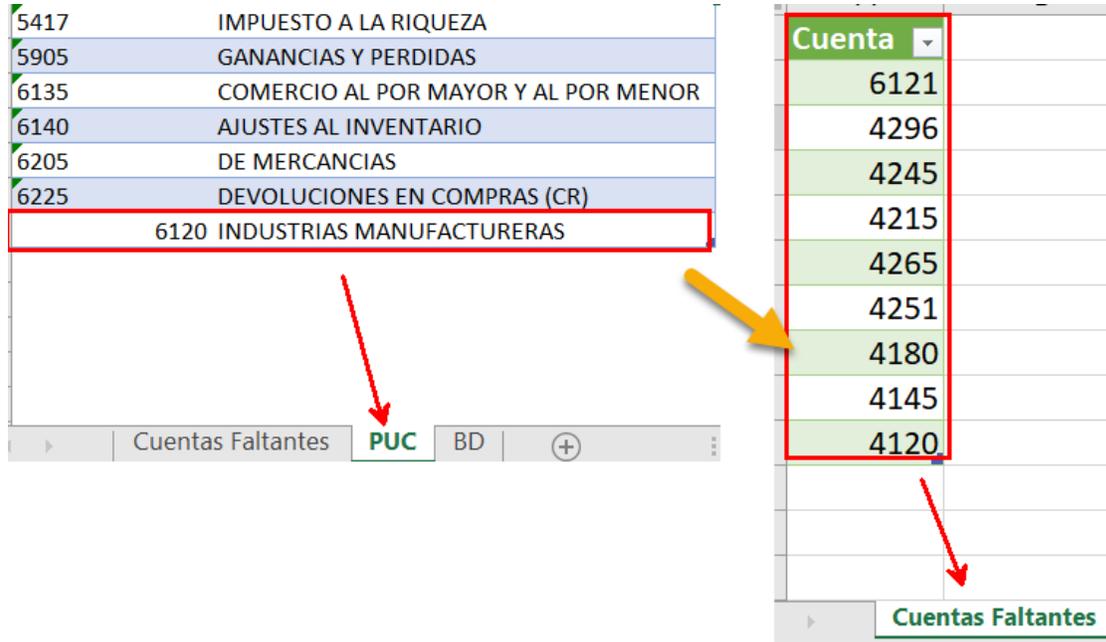


Figura 5.64 Cuenta creada y actualizada.



HACKS: Como decimos en nuestros países latinos, la saca del estadio, te imaginas el mar de posibilidades que se nos abre con esto, como por ejemplo saber que clientes están en una base y cuáles no, que productos están en inventario y no se han movido, conciliaciones bancarias etc.

5.6 Desactiva el encabezado y tipo cambiado automático y te evitas dolores de cabeza

Uno de los grandes retos de trabajar con Power Query es el tipo de datos que contiene cada columna, es lo que conocemos en Excel como formato de celda, cuando se asigna si es número, fecha, porcentaje etc. En Power Query este tipo de dato no es tan trivial en su manejo, ya que se debe tener algunas consideraciones antes de, si le asignas a una columna tipo de dato número entero, pero en esta columna hay registros en blanco, o un dato como texto, generar un error y puede dañarte todo el proceso ETL de esta consulta.

Que dolor de cabeza es esto, con nuestra experiencia del día a día en desarrollos, enseñanzas y aplicaciones lo vemos, como usuarios se desesperan tanto al tratar de encontrar estos errores que su cabeza llega al límite de casi “explotar”.

En esta sesión vas a evitar mucho de estos tipos de dolores, solo deshabilitando el tipo cambiado y el encabezado promovido, que no lo detecte de manera automático.

Veamos el contexto en una conexión de un archivo de Excel.

Vamos al libro de Excel 6.1 Conexión Países, tenemos cargada una tabla desde un libro → Vamos a la pestaña Datos → Consultas y conexiones → Clic derecho editar. Ver figura 5.55

Siempre que se carga un archivo de Excel a Power Query, se generan 4 pasos de manera automático, dos ellos son obligatorios, los otros 2 NO.

A ^B C VENDEDOR	FECHA	1 ² 3 CANTIDAD
JUAN	1/01/2016	
JUAN	2/01/2016	
JUAN	3/01/2016	
JUAN	4/01/2016	
JUAN	5/01/2016	
JUAN	6/01/2016	
JUAN	7/01/2016	
JUAN	8/01/2016	
JUAN	9/01/2016	
JUAN	10/01/2016	

PROPIEDADES

Nombre

Todas las propiedades

PASOS APLICADOS

- Origen ⚙
- Navegación ⚙
- Encabezados promovidos ⚙
- Tipo cambiado ⚙

Figura 5.65 Consulta de un libro de Excel.

Lo pasos aplicados de Origen y Navegación hacen referencia al libro y a la hoja del libro de Excel. Encabezados promovidos y Tipo cambiado son la identificación de las columnas y el tipo de dato.

Si notan ya cada columna tiene un tipo de dato asignado de manera automática, al parecer no tiene ningún inconveniente esto hasta el momento, pero que sucede si la columna VENDEDOR, cambia de nombre en la fuente original a COMERCIAL por ejemplo.

Veamos. Abrimos el archivo *6. Base Países*, y renombramos esta columna.



ADVERTENCIA: Para poder realizar este cambio de nombre se debe cerrar o salir del editor de Power Query, ya que no permite abrir mas libros de Excel cuando estamos allí.



NOTA: Para realizar este paso a paso, debes redireccionar la ruta de la conexión en Power Query a tu libro de Excel local

Cerramos el libro con este cambio, y volvemos al editor de consultas de la conexión anterior.

Nos sale esto:

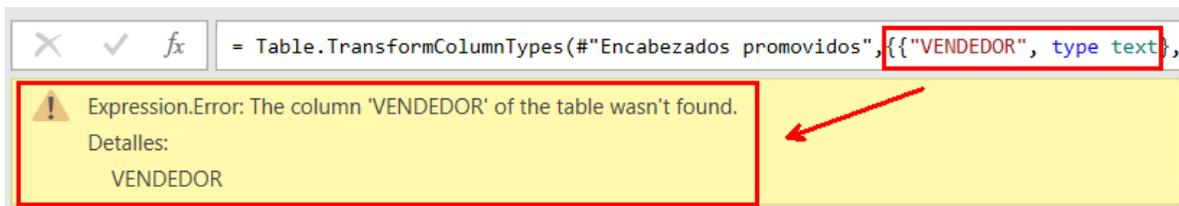


Figura 5.66 Error por cambio de nombre en columna.

Traduciendo el error, la columna VENDEDOR no fue encontrada, esto sucede ya que el tipo cambiado se le fue asignado a una columna antes existente llamada VENDEDOR, y ahora se llama COMERCIAL.

Y este es uno de los múltiples escenarios que trae el tipo cambiado automático, como podemos corregir esto, hay varias formas, una de ellas es desde la barra de fórmula como esta señalada en la figura 5.66, cambiar VENDEDOR por COMERCIAL. Procedamos.

	COMERCIAL	FECHA	CANTIDAD	PAÍS
1	JUAN	1/01/2016	299	Colombia
2	JUAN	2/01/2016	701	Colombia
3	JUAN	3/01/2016	761	Colombia
4	JUAN	4/01/2016	259	Colombia
5	JUAN	5/01/2016	994	Colombia
6	JUAN	6/01/2016	706	Colombia
7	JUAN	7/01/2016	867	Colombia
8	JUAN	8/01/2016	178	Colombia

Figura 5.67 Corregir error en columna cambiada.

El error se corrige de manera muy manual, editando en la barra de fórmulas por el nuevo nombre, pero este proceso es delicado y tedioso hacer esto todo el tiempo.

Cambiamos la configuración para que no promueva los encabezados ni identifique el tipo cambiado.

Dentro del editor de Power Query vamos a la Pestaña Archivo → Opciones y configuración → Opciones de consulta → Carga de datos → Detección de tipos.

En la carga de Datos → Habilitamos → No detectar nunca los encabezados y tipos de columna para los orígenes no estructurados

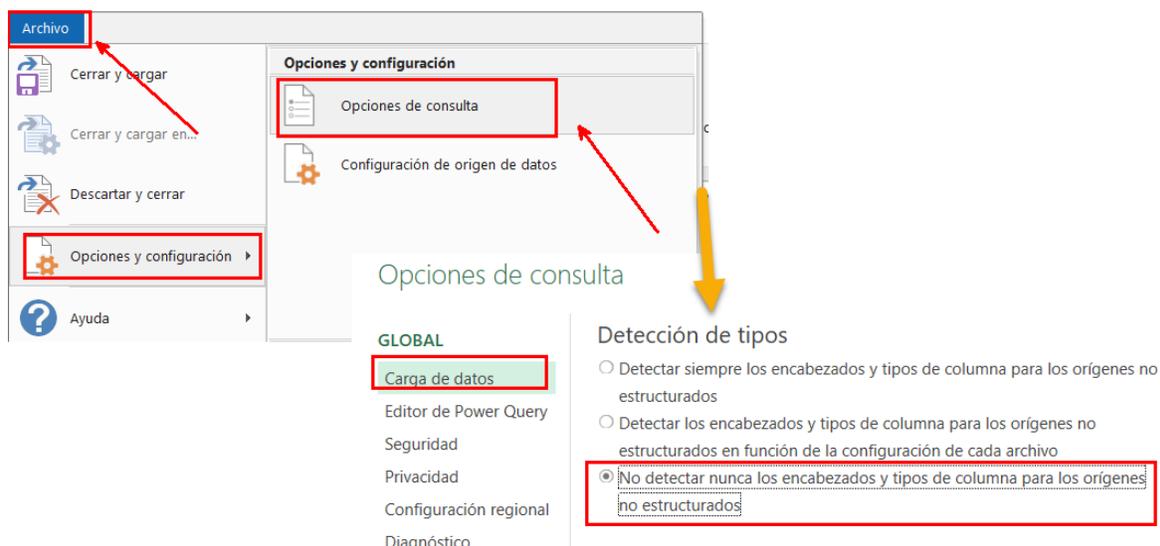


Figura 5.68 Cambiar detección de tipos.



NOTA: Esta opción solo está habilitada para Office 365 y Power BI Desktop

Probemos como es funcionamiento, Aceptar.

Desde el mismo editor vamos a cargar el mismo archivo de Excel.

Pestaña Inicio → Nueva Consulta → Nuevo origen → Archivo Excel

Buscamos la ruta donde se encuentra este libro o conectarse a cualquier libro que se considere para hacer dicha prueba.

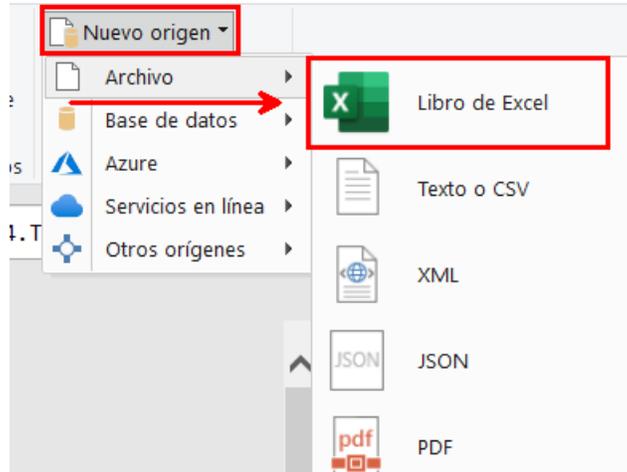


Figura 5.69 Conectarse a libro de Excel.

Señalando cualquier hoja del libro, damos aceptar, y vemos que solo en la consulta de Power Query para esta nueva carga no solo hay dos aplicados, los otros los elimina.

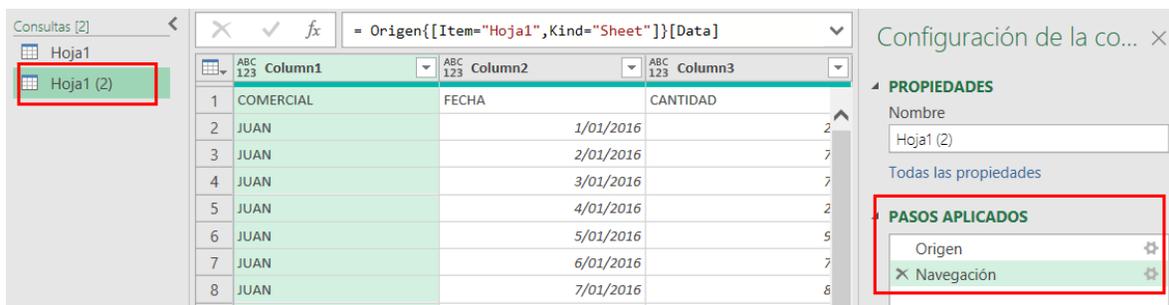


Figura 5.70 Encabezado promovido y tipo cambiado no detectado.

Si vamos a promover los encabezados, simplemente vamos a transformar y usar la primera fila como encabezado.



HACKS: Más que un truco es una muy buena práctica cuando se realizan proceso de carga de información, y evitar este tipo de errores, tener presente que cuando se cargan datos de tablas o bases de datos estructurados Power Query reconoce los encabezados, pero no asigna el tipo cambiado, que es el meollo del asunto.

5.7 Como trabajar como millones de registros en Excel con solo conexión

No es un secreto que Excel baja el rendimiento cuando su número de filas sobrepasa los 100 mil registros o un poco más, y peor aún si tenemos una cantidad de fórmulas aplicadas, puede salir este mensaje aburridor de “Excel no responde” y perder tiempo y hasta horas de trabajo porque no guarda los cambios o una cantidad de inconvenientes más.

Excel cuenta con 1.048.576 filas, que sucede si hay una o varias filas más?, no se puede, hasta hace poco era una limitante, ahora con Power Query podemos trabajar con millones de registros sin necesidad de tener los datos físicos, trabajar con solo conexión.

Con el libro *7. Datos Importados*, hay una conexión directa a un libro que contiene más de medio millón de registros y múltiples columnas.



A	B	C	D
Codigo Cuenta	Nombre Cuenta	Fecha	Documento
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	CONSIGNACIÓN
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	CONSIGNACIÓN
110505001	CAJA GENERAL	43467	FACTURA DE VENTA
110505001	CAJA GENERAL	43467	FACTURA DE VENTA

Figura 5.71 Carga de registros importados.

Cargar esta información de manera directa hace que el archivo tenga un peso importante, y si tiene fórmulas peor, la solución es llevar este tipo de carga a solo conexión, ya que si nos conectamos a 2 o más archivos como estos, estamos pasando la cantidad de registros permitidos por Excel.

En el panel de consultas y conexiones → Clic derecho → Cargar en

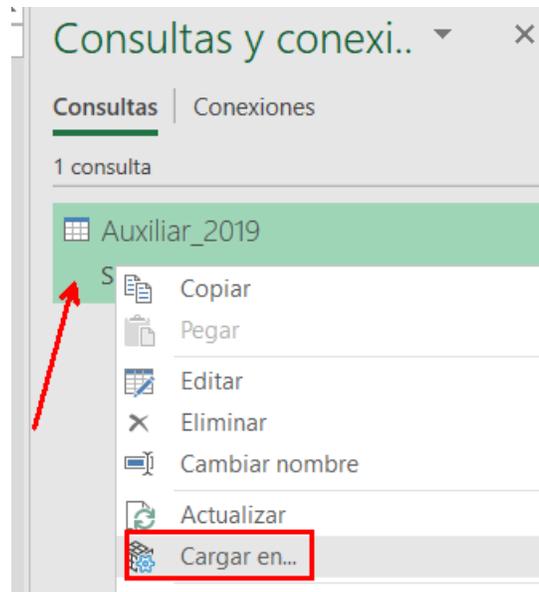


Figura 5.72 Cambiar tipo de carga.

Se nos habilita una ventana más para seleccionar el tipo de carga que deseamos llevar nuestros datos.

Seleccionamos → Crear únicamente la conexión

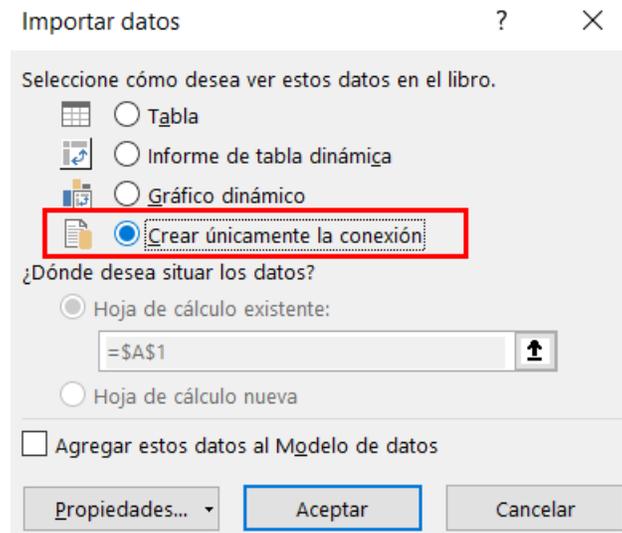


Figura 5.73 Cargar datos en solo conexión.



NOTA: Excel nos brinda varias opciones para cargar la información, incluso podemos llevar directamente los datos a una tabla dinámica.

Aceptar, y nos sale un mensaje de confirmación, que se eliminará la tabla de datos físicamente. Le indicamos que sí.

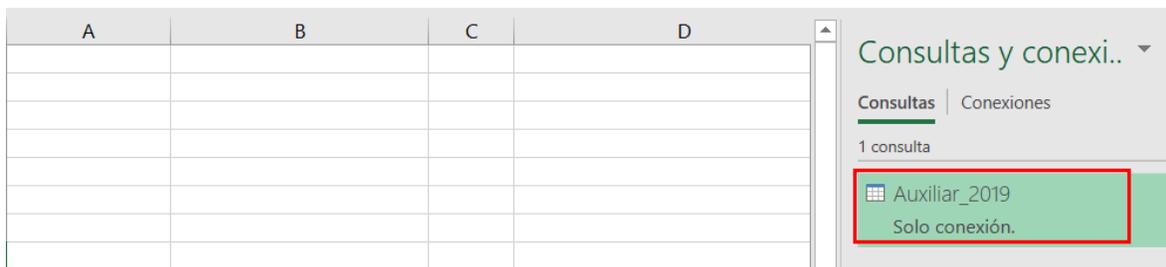


Figura 5.74 Datos cargados como solo conexión.

La hoja de Excel ya está en blanco y la consulta como solo conexión.

Usar datos de solo conexión

Lo siguiente es como consumimos estos datos para realizar informes. Si esta información se quiere mostrar en una tabla dinámica, se puede dar clic derecho en cargar en Tabla dinámica, o desde la pestaña Insertar → Tabla dinámica.

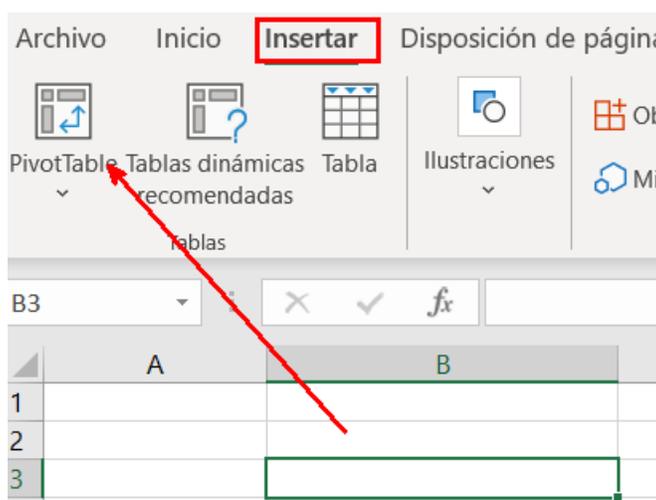


Figura 5.75 Insertar Tabla dinámica.

Hay una opción poca usada, y es utilizar la fuente de datos externa, y allí elegimos la conexión de datos.

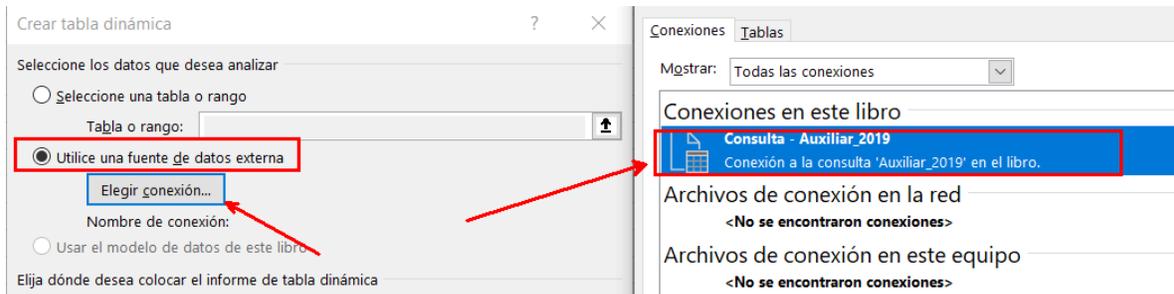


Figura 5.76 Conexión a fuente externa.

Abrimos los datos y damos Aceptar.

Se crea una copia en la consulta y una tabla dinámica lista para crear reportes sin necesidad de tener los datos en una tabla física.

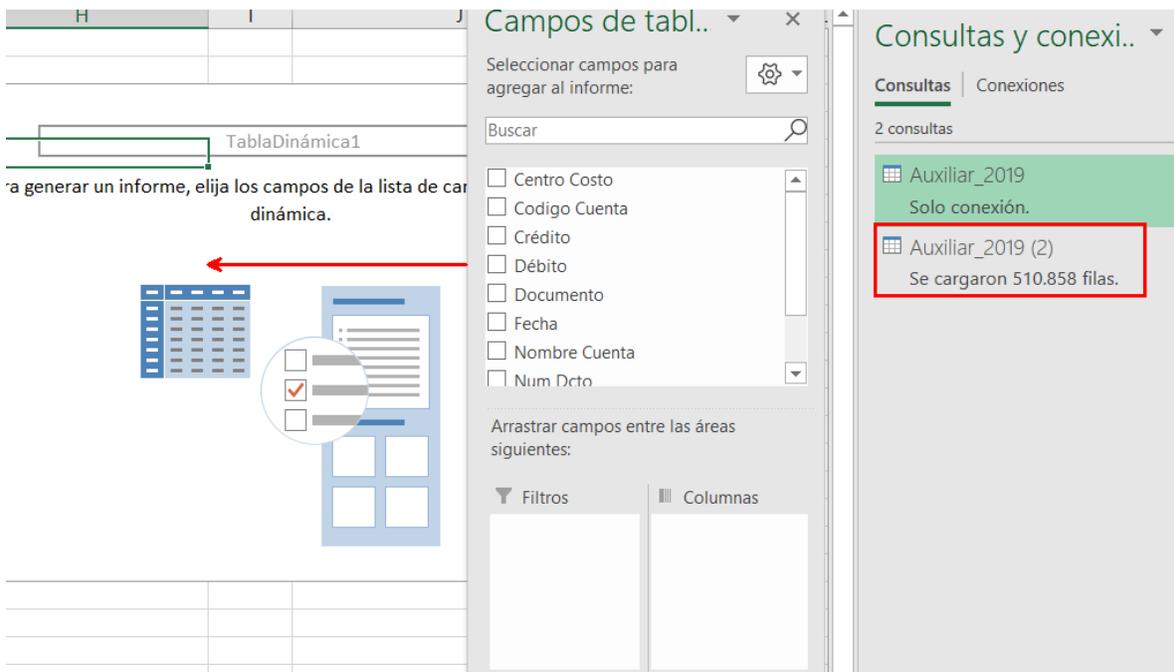


Figura 5.77 Crear tabla dinámica mediante solo conexión.



NOTA: Este proceso de conectarnos a una conexión existente lo enseñamos, ya que no todas las personas cuentan con Office 365 o una versión más actualizada, si cuentas con office 365 puedes directamente cargar tus datos a una tabla dinámica y así duplicar consultas



HACKS: Implementando este truco solucionamos el problema si los datos exceden el límite permitido por Excel, y el rendimiento y peso de los datos va a mejorar exponencialmente.

5.8 Eliminar errores al inicio, te dará un buen final

Trabajar con errores en Excel es muy común, ya sea que no se encuentra un registro, se elimina una columna que dependía una fórmula, se haga una división entre cero y entre otros, darles manejo a estos problemas en Excel es relativamente fácil, usando la función SI.ERROR para que no sean visibles, pero en Power Query este tipo de errores es intolerable en esta herramienta, por ende si no se le da un buen manejo al inicio, tendremos un mal final.

Empecemos con un libro de Excel en blanco y cargamos los datos que se encuentran en el archivo 8. *Balances*.

Pestaña Datos → Obtener datos → Desde un archivo → Desde un libro

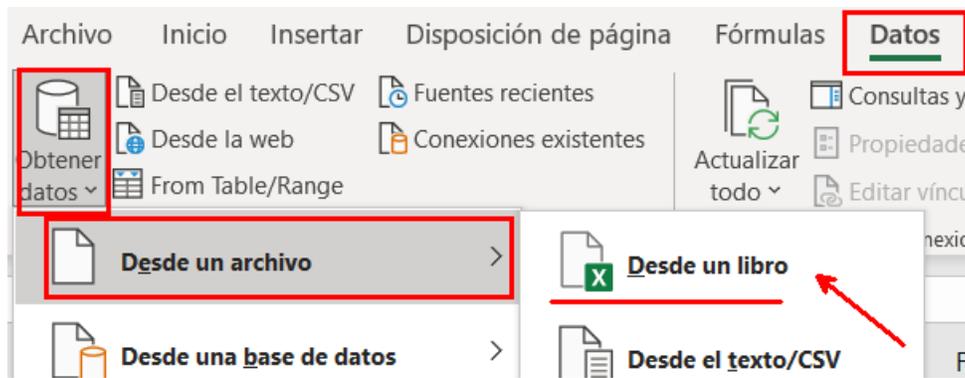


Figura 5.78 Conectarse a un libro de Excel.

Buscamos el archivo de Balances → Importar → Seleccionamos la hoja Datos → Transformar datos.

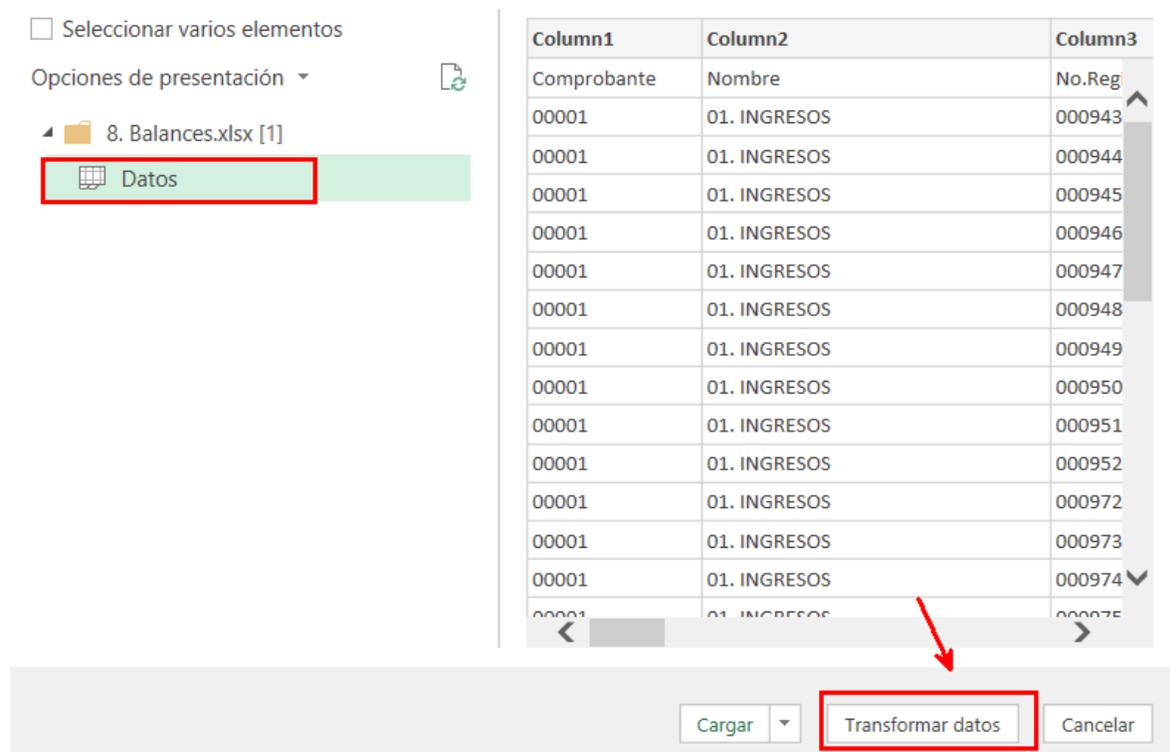


Figura 5.79 Cargar la hoja Datos.

Ya en el editor de Power Query y si deshabilitaste la detección de Encabezados promovidos y de tipo cambiado, los anexamos para ver que sucede.

Pestaña Transformar → Usar la primera fila como encabezado

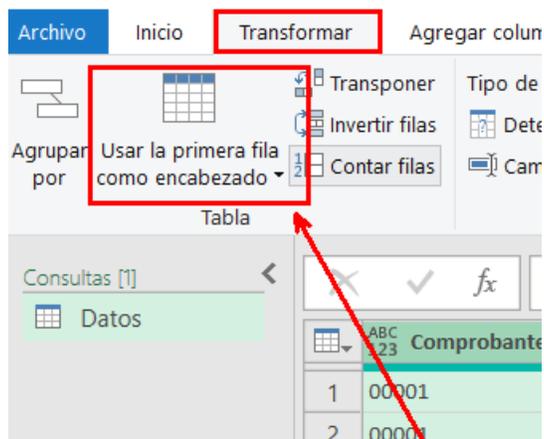


Figura 5.80 Usar la primera fila como encabezado.

Luego le añadimos el tipo cambiado a todas las columnas, ¿se deben hacer una a una? Es una muy buena pregunta, pues no, para eso nos atrevimos a escribir este libro que tengas siempre tu AS bajo la manga.

Seleccionando solo una columna de la consulta vamos a darle: **CTRL + A**

Y Bingo, todas sus columnas seleccionadas en un segundo.

En la misma pestaña Transformar → Detectar tipo de datos.

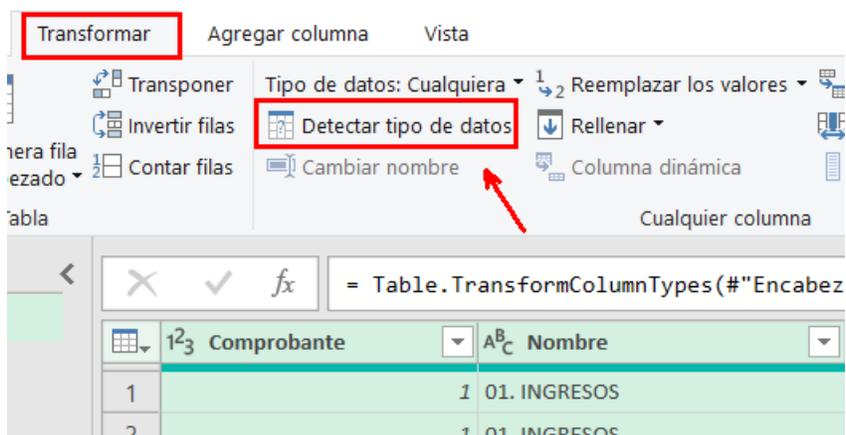


Figura 5.81 Detectar tipo de dato.

Si tienes office 365, puedes observar que una de las columnas llamada *Documento*, tiene la barra de color rojo, lo que significa que tiene errores, esto pasa ya que el tipo cambiado se detectó de manera automática y Power Query hace un recorrido fila a fila y según el muestreo que tome de los datos, si hay más números que textos, asigna a esta columna como número entero, tal como lo vimos en el Hacks 5.6.

Volvemos a comprobar que este tipo cambiado es un dolor de cabeza cuando se asigna de manera automática.



NOTA: Para una mejor práctica, asigna el tipo cambiado al final, y solo a las columnas de tipo fecha y número.

Eliminemos este paso de tipo cambiado.

ABC 123 Comprobante	ABC 123 Nombre	ABC 123 No.Registros	ABC 123 Documento	ABC 123 Docto. Referencia
00001	01. INGRESOS	000943	000002572	FT0000717
00001	01. INGRESOS	000944	000002572	FT0000744
00001	01. INGRESOS	000945	000002572	FT0000745
00001	01. INGRESOS	000946	000002572	FT0000774
00001	01. INGRESOS	000947	000002572	000000000
00001	01. INGRESOS	000948	000002572	000000000
00001	01. INGRESOS	000949	000002572	000000000
00001	01. INGRESOS	000950	000002572	000000000
00001	01. INGRESOS	000951	000002572	000000000
00001	01. INGRESOS	000952	000002572	000002572
00001	01. INGRESOS	000972	000002573	FT0000766
00001	01. INGRESOS	000973	000002573	000000000

Figura 5.82 Paso tipo cambiado eliminado

La columna *Documento* esta de color verde nuevamente y todas las demás de igual manera.

Para acortarte la curva de aprendizaje todavía hay errores, así es, aunque no se vean, conociendo la información y el tipo de archivo que estamos manejando hay un error al final de las filas, pero aún Power Query no lo detecta, se solapa, lo podemos identificar si hacemos un filtro en la columna *Nombre* → Filtrar → Cargar más.

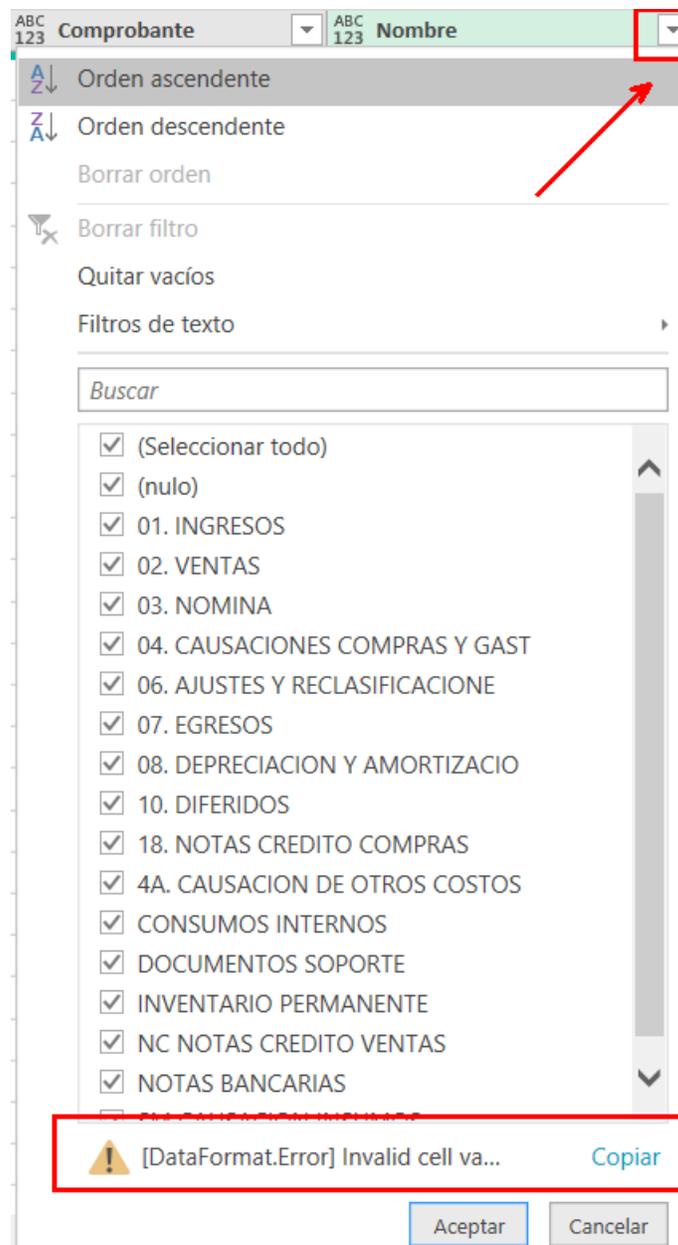


Figura 5.83 Filtro en columna nombre

Hay una advertencia al final, según la figura 5.83, si copiamos y pegamos este error:

[DataFormat.Error] Invalid cell value '#N/A'.

Hay una celda que es inválida según el tipo de error, y si no conocemos esto seguimos trabajando y realizando más procesos de transformación y vamos a tener un mal final.

Tener estos errores en la Data tiene muchas desventajas, una de ellas es que no nos muestra toda la granularidad de registros en las columnas, en caso de querer hacer un filtro.

La solución o el truco es quitar estos errores antes de iniciar nuestra transformación, créenos pasamos horas enteras para poder identificar esto y traértelo para que lo tengas en minutos.

Nos ubicamos en la columna Nombre → Clic derecho → Quitar errores

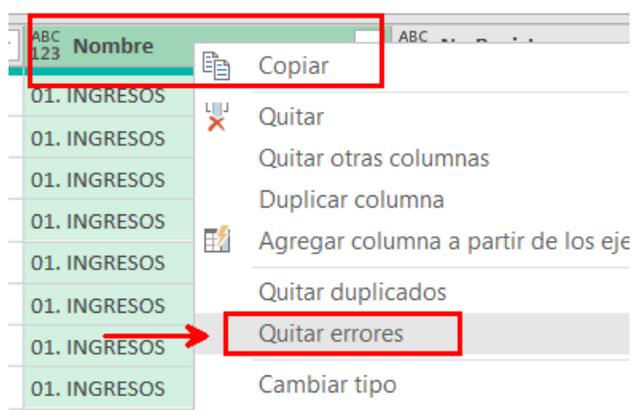


Figura 5.84 Quitar errores

Puedes continuar con la demás transformación de datos.

Puedes validar filtrando nuevamente la columna Nombre, realizando el filtro y cargar más.

La figura 5.85 muestra una columna limpia de errores. Y lo mejor es que trabajar con Power Query no elimina solo el error para esta columna, sino para todas las demás.

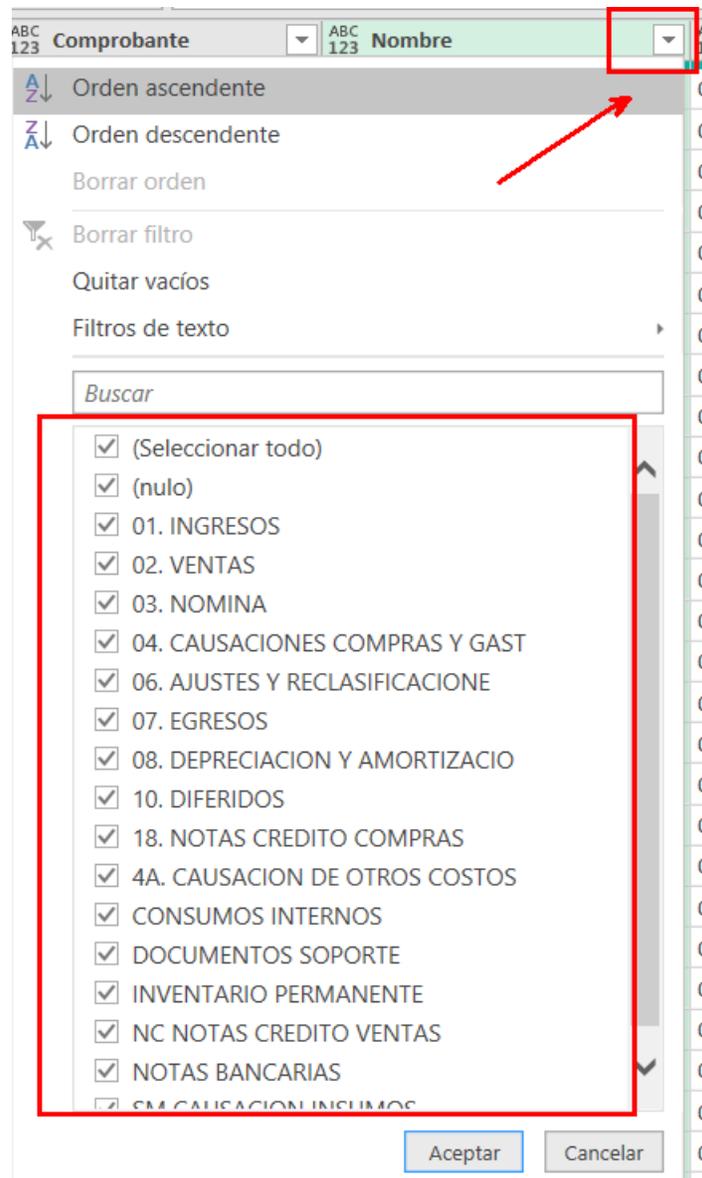


Figura 5.85 Errores quitados



HACKS: Eliminar los errores al inicio te dará siempre un buen final, adicional estos errores se solapan, y luego encontrarlos después de haber hecho decenas de pasos más es algo tedioso

5.9 Ahorra pasos de transformación con este gran truco

El proceso ETL está compuesto de múltiples aplicaciones en la transformación de datos, cada que se genera un clic, una funcionalidad, un cambio de nombre en una columna, un tipo de dato o alguna fórmula, se realiza un nuevo código en Power Query y ya hace parte del proceso, y cada paso se refrescará en la actualización, correrá como parte de la consulta y por ende va a consumir memoria en el PC.

Para nuestro ejercicio tenemos una conexión realizada a un libro de Excel, con encabezados promovidos y tipo cambiado.

El libro esta en la carpeta, nombrado *9.1 Ahorrar Pasos Aplicados*.

Pestaña datos → Consultas y conexiones → Clic derecho en la consulta → Editar.

	Nombre Vendedor	Departamento	Municipio	Fecha Venta	Código Producto	Nombre Producto	Tipo Cliente
1	JORGE HERNAN ROMERO TAPAS...	5	5001	1/02/2021	RE-58836-C-324972	CERVEZA CORONA EXTRA x 355ml	CADENA NACIONAL
2	RAUL FLOREZ RAMIREZ	5	5045	1/01/2021	AL-43000-O-426557	OF.CHILE CON CARNE FRIJOL REF X 300G	CADENA REGIONAL
3	DIEGO MORA POSADA	11	11001	1/02/2021	HA-92539-A-394787	AVENA EN HOJUELA	DISTRIBUIDOR
4	DUBER ANTONIO RIOS	8	8758	1/01/2021	AL-43000-O-426557	OF.CHILE CON CARNE FRIJOL REF X 300G	DISTRIBUIDOR
5	JULIO CESAR VILLEGAS ARANZAZU	27	27001	1/03/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	HORECAS
6	JULIO CESAR VILLEGAS ARANZAZU	27	27001	1/01/2021	CO-48295-M-312675	MAGGI C.GALLINA CUBOS x48 x 528 gr	DISTRIBUIDOR
7	RAUL FLOREZ RAMIREZ	5	5045	1/02/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	CADENA NACIONAL
8	RAUL FLOREZ RAMIREZ	5	5045	1/01/2021	AL-43000-T-670430	TORTILLA FAJITA X 240G	CADENA REGIONAL
9	RAUL FLOREZ RAMIREZ	5	5045	1/03/2021	AL-43000-T-670430	TORTILLA FAJITA X 240G	CADENA REGIONAL
10	NILTON CESAR PATINO MUNDOZ	44	44001	1/03/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	HORECAS

Figura 5.86 Consulta cargada

Iniciemos aplicando varios pasos, para que identifiques a que quieres llegar.

Seleccionamos la columna código de producto → Agregar columna → Extraer-> Primeros Caracteres.

Vamos a extraer los dos primeros dígitos.

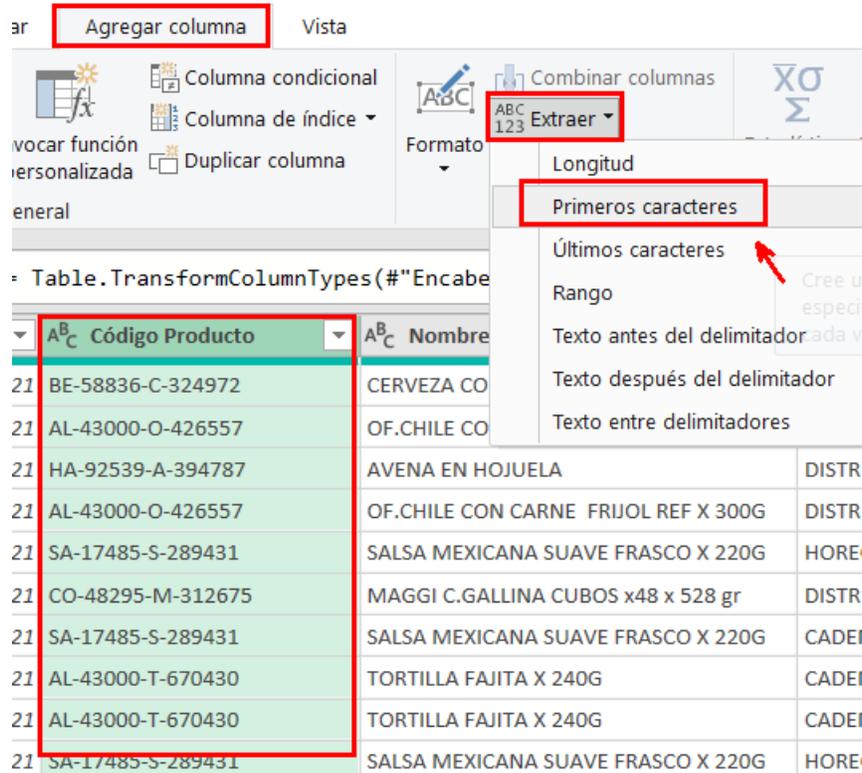


Figura 5.87 Extraer primeros caracteres

Ingresamos el número 2 y aceptamos.



Figura 5.88 Columna con 2 caracteres

Se añadió un nuevo paso llamado: *Primeros caracteres insertados*.

Luego debemos cambiar el nombre de la columna a: Código

Clic derecho en el encabezado → Cambiar nombre → Código

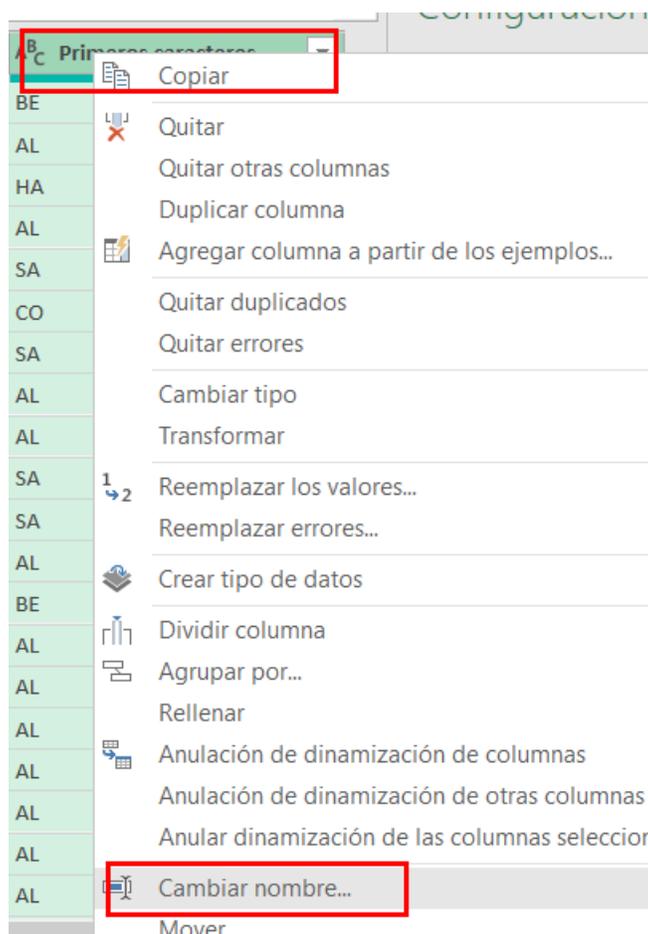


Figura 5.89 Cambiar nombre

Se inserto un nuevo paso aplicado.

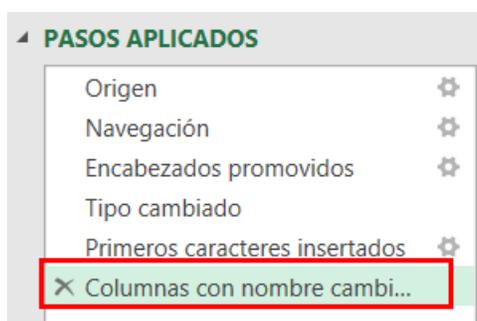
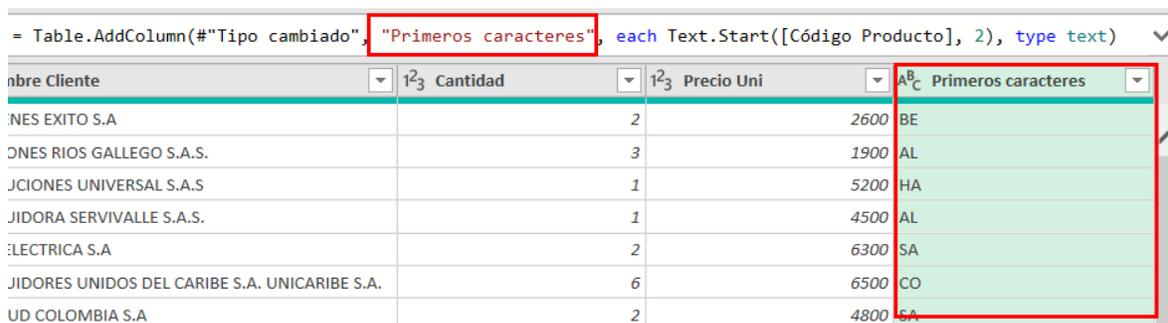


Figura 5.90 Nuevo paso aplicado

Cada que se use una funcionalidad en Power Query lo guarda como registro, una especie de grabadora en cada paso.

Es lo normal que se haga, el inconveniente se encuentra en que, si esto lo hacemos con diferentes columnas, la cantidad de pasos aplicados es enorme, y si el proceso es con una base de datos muy grande el proceso de actualización va a ser cada vez más lento.

Así que como podemos ahórranos estos pasos aplicados, es muy sencillo, borremos el último paso aplicado, y vamos a sacar provecho a la barra de fórmula.

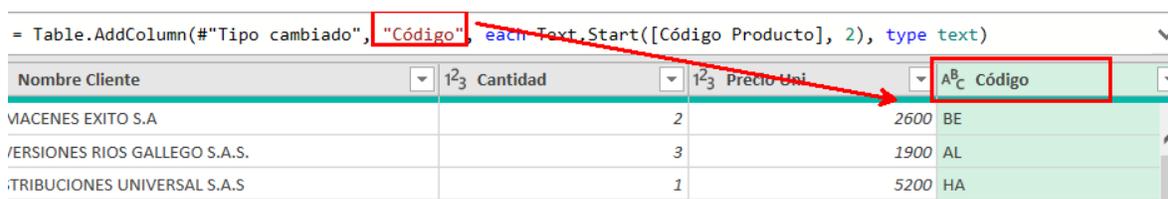


The screenshot shows the Power Query formula bar with the following formula: `= Table.AddColumn("#Tipo cambiado", "Primeros caracteres", each Text.Start([Código Producto], 2), type text)`. The text "Primeros caracteres" is highlighted with a red box. Below the formula bar is a table with the following data:

Nombre Cliente	Cantidad	Precio Uni	Primeros caracteres
MACENES EXITO S.A	2	2600	BE
VERSIONES RIOS GALLEGO S.A.S.	3	1900	AL
TRIBUCIONES UNIVERSAL S.A.S	1	5200	HA
TRIDORA SERVIVALLE S.A.S.	1	4500	AL
TRILECTRICA S.A	2	6300	SA
TRIDORES UNIDOS DEL CARIBE S.A. UNICARIBE S.A.	6	6500	CO
TRID COLOMBIA S.A	2	4800	SA

Figura 5.91 Paso aplicado primeros caracteres en barra de fórmula

La palabra “Primeros caracteres” esta entre comillas, es un nombre que la herramienta le asigna por defecto, y si le cambiamos el nombre desde allí con mucha cautela, pues no tendríamos que añadir un nuevo paso a la consulta.



The screenshot shows the Power Query formula bar with the following formula: `= Table.AddColumn("#Tipo cambiado", "Código", each Text.Start([Código Producto], 2), type text)`. The text "Código" is highlighted with a red box. Below the formula bar is a table with the following data:

Nombre Cliente	Cantidad	Precio Uni	Código
MACENES EXITO S.A	2	2600	BE
VERSIONES RIOS GALLEGO S.A.S.	3	1900	AL
TRIBUCIONES UNIVERSAL S.A.S	1	5200	HA

Figura 5.92 Cambiar nombre en barra de fórmula

Con este sencillo truco, ya hemos ahorrado un paso aplicado, hasta este punto puede ser algo muy trivial, que sucede si por ejemplo realizas una multiplicación entre precio y cantidad, para calcular la venta, luego cambiar el nombre de la columna, y asignar el tipo de dato, estamos ahorrando ca vez más pasos.



NOTA: Para realizar este proceso, debemos tener habilitado la barra de fórmula, en la pestaña vista, seleccionamos Barra de fórmula

Otra aplicación que tiene saber editar la barra de fórmula es si necesitamos ajustar tipo de datos identificados de manera automática, por ejemplo, si miras el archivo original que se encuentra en la carpeta 5 Archivos Power Query, el libro 9. BD Pasos, las columnas *Departamento* y *Municipio* se encuentran diferentes a como están en Power Query.

Departamento	Municipio	Departamento	Municipio
05	05001	5	5001
05	05045	5	5045
11	11001	11	11001
08	08758	8	8758
27	27001	27	27001
27	27001	27	27001
05	05045	5	5045
05	05045	5	5045
44	44001	44	44001
27	27001	27	27001
17	17380	17	17380
17	17001	17	17001
05	05045	5	5045
44	44001	44	44001
05	05045	5	5045
17	17380	17	17380

Figura 5.93 Columnas en Excel y Power Query

¿Ya notaste la diferencia? El archivo original, las columnas *Departamento* y *Municipio* contienen unos ceros al inicio, Power Query los elimina para convertirlo a número entero. Ya no podríamos recuperar esto al final de los pasos, incluso puedes intentar poniendo el tipo de dato a texto, y aún así no se recupera.

Departamento	Municipio
5	5001
5	5045
11	11001
8	8758
27	27001
27	27001
5	5045
5	5045
5	5045
44	44001
27	27001
17	17000

Configuración de la CO...

PROPIEDADES
Nombre
Hoja1
Todas las propiedades

PASOS APLICADOS
Origen
Navegación
Encabezados promovidos
Tipo cambiado
Primeros caracteres insertados
X Tipo cambiado1

Figura 5.94 Convertir columna a texto

Intento fallido, porque hay un tipo cambiado antes de este último y allí fue donde se eliminaron los ceros. Eliminemos este último paso.

Damos clic en Tipo cambiado → Barra de fórmulas → cambiamos Int64.Type por → type text

isFormColumnTypes(#"Encabezados promovidos",{"Nombre Vendedor", type text}, {"Departamento", type text}, {"Municipio", type text})

Departamento	Municipio	Fecha Venta	Código Producto	Nombre Producto	Tipo
05	05001	1/02/2021	BE-58836-C-324972	CERVEZA CORONA EXTRA x 355ml	CADE
05	05045	1/01/2021	AL-43000-O-426557	OF.CHILE CON CARNE FRIJOL REF X 300G	CADE
11	11001	1/02/2021	HA-92539-A-394787	AVENA EN HOJUELA	DISTR
08	08758	1/01/2021	AL-43000-O-426557	OF.CHILE CON CARNE FRIJOL REF X 300G	DISTR
27	27001	1/03/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	HORE
27	27001	1/01/2021	CO-48295-M-312675	MAGGI C.GALLINA CUBOS x48 x 528 gr	DISTR
05	05045	1/02/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	CADE
05	05045	1/01/2021	AL-43000-T-670430	TORTILLA FAJITA X 240G	CADE
05	05045	1/03/2021	AL-43000-T-670430	TORTILLA FAJITA X 240G	CADE
44	44001	1/03/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	HORE
27	27001	1/01/2021	SA-17485-S-289431	SALSA MEXICANA SUAVE FRASCO X 220G	DISTR
17	17380	1/02/2021	AL-43000-N-188756	NACHOS NATURALES X 180G	HORE

Configuración...

PROPIEDADES
Nombre
Hoja1
Todas las propiedades

PASOS APLICADOS
Origen
Navegación
Encabezados pro...
X Tipo cambiado
Primeros caracter...

Figura 5.95 Editar tipo cambiado



HACKS: Editando la barra de fórmulas ahorramos pasos aplicados y procesos incensarios en nuestras transformaciones de datos.

5.10 Refrescar consultas automáticas con Excel y VBA

Hasta este punto y en todos los trucos vistos hemos actualizado las consultas de manera manual, dando clic derecho sobre una de las tablas actualizar o en la pestaña Datos → Actualizar todo.

Es el proceso normal, solo que si queremos brindar una mejor experiencia al usuario debemos programar una actualización automática o por medio de un botón programado en Visual Basic (VBA).

Actualización Automática

En Excel existe varias maneras de que las consultas se refresquen de manera automática, ya sea al abrir el archivo y con cierta periodicidad.

Vamos al archivo 10. Actualizar consultas → Pestaña Datos → Consultas y conexiones.

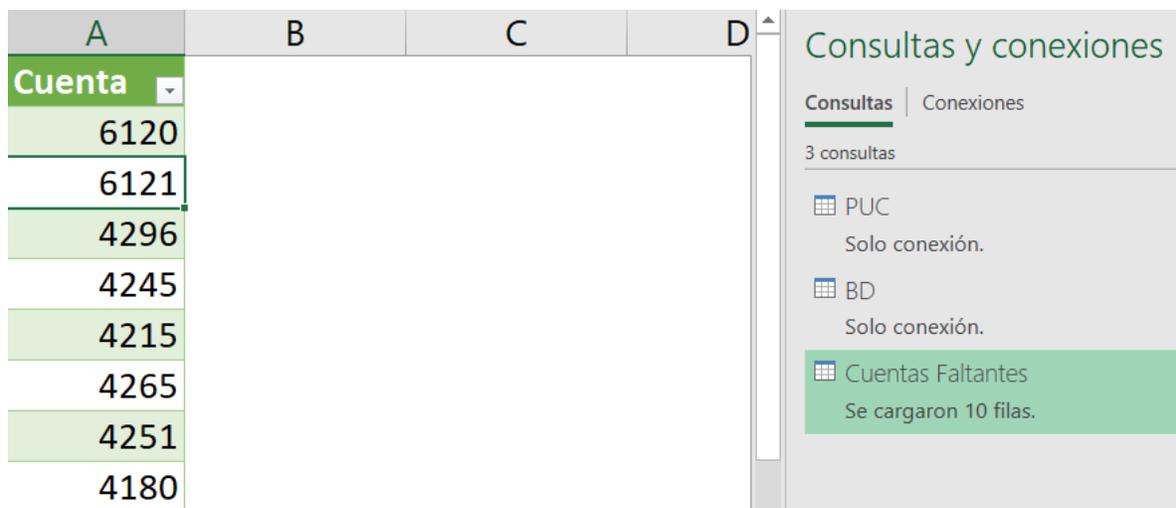


Figura 5.96 Consultas en Excel

Estas consultas pertenecen al proceso realizado en el truco de Datos faltantes.

Nos ubicamos en una de las consultas → Clic derecho → Propiedades

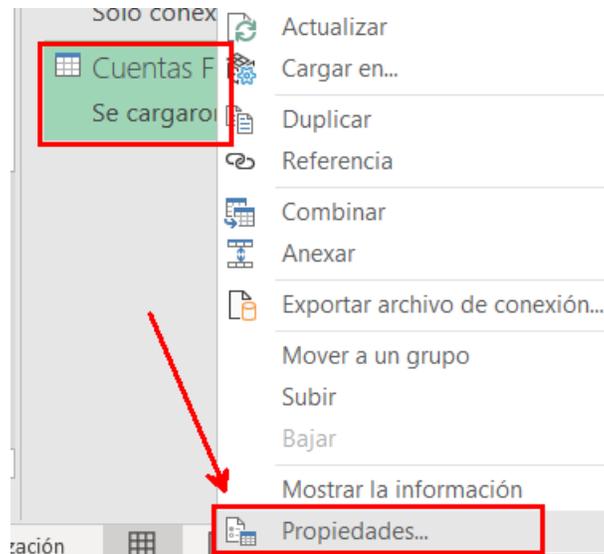


Figura 5.97 Propiedades de las consultas

Y en el cuadro de dialogo podemos programar para que las consultas se actualicen cada que se abra el archivo o según un número de minutos.

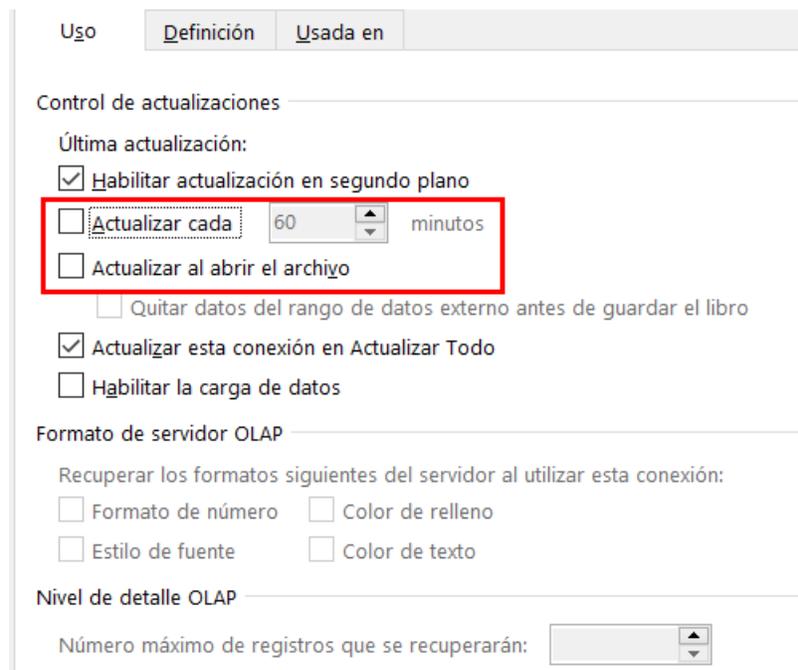


Figura 5.98 Programar actualización de consulta



NOTA: Este tipo de actualización se recomienda cuando los datos que se manejan son muy recurrentes o de corto periodo de análisis.

Actualización con VBA

Visual Basic o más conocidas como las macros, nos pueden ayudar a darle una mejor experiencia al usuario para actualizar las consultas existentes en el libro.

Para programar esta macro, estando en el archivo vamos a presionar la combinación → ALT + F11

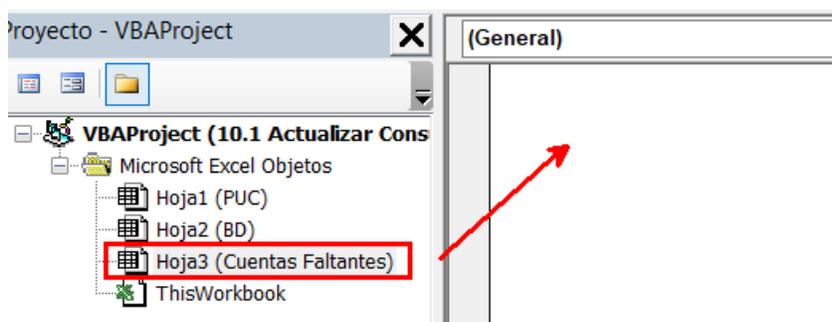


Figura 5.99 Ventana programación VBA

Damos doble clic en la hoja o módulo donde ingresamos este corto código, tal como se estudio en el capítulo 4: Hacks de Macros.

```
Sub Actualizacion()  
  
ActiveWorkbook.Connections ("Consulta - Cuentas Faltantes").Refresh  
  
|  
  
End Sub
```

Figura 5.100 Código actualizar consulta

Es un código muy sencillo, actualiza la consulta del libro la cual se quiere ejecutar, lo único que cambia para que se actualice las demás consultas es el nombre de la consulta.

```

Sub Actualizacion()

ActiveWorkbook.Connections("Consulta - Cuentas Faltantes").Refresh

ActiveWorkbook.Connections("Consulta - PUC").Refresh

ActiveWorkbook.Connections("Consulta - BD").Refresh

End Sub

```

Figura 5.101 Código completo actualizar consultas

Si deseamos una mejor experiencia podemos incluir en el código un mensaje de confirmación de actualización.

```

Sub Actualizacion()

ActiveWorkbook.Connections("Consulta - Cuentas Faltantes").Refresh

ActiveWorkbook.Connections("Consulta - PUC").Refresh

ActiveWorkbook.Connections("Consulta - BD").Refresh

MsgBox "Consultas Actualizadas", vbInformation, "Excelaprende_Dataice"

End Sub

```

Figura 5.102 Código con mensaje de confirmación

Salimos del editor de Visual Basic e insertamos un botón en Excel para asignarle esta macro y el usuario lo pueda correr.

Pestaña Programador → Insertar → Botón de formulario

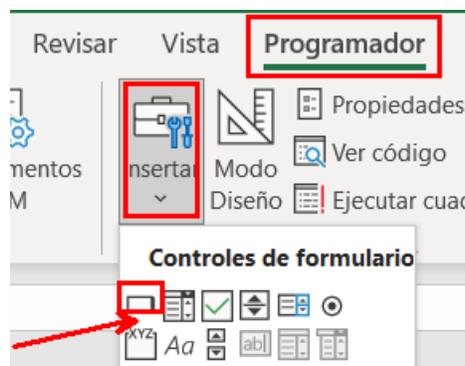


Figura 5.103 Insertar botón de formulario

Insertamos el botón en la hoja de Excel y en el cuadro de dialogo que se habilita le asignamos la macro.

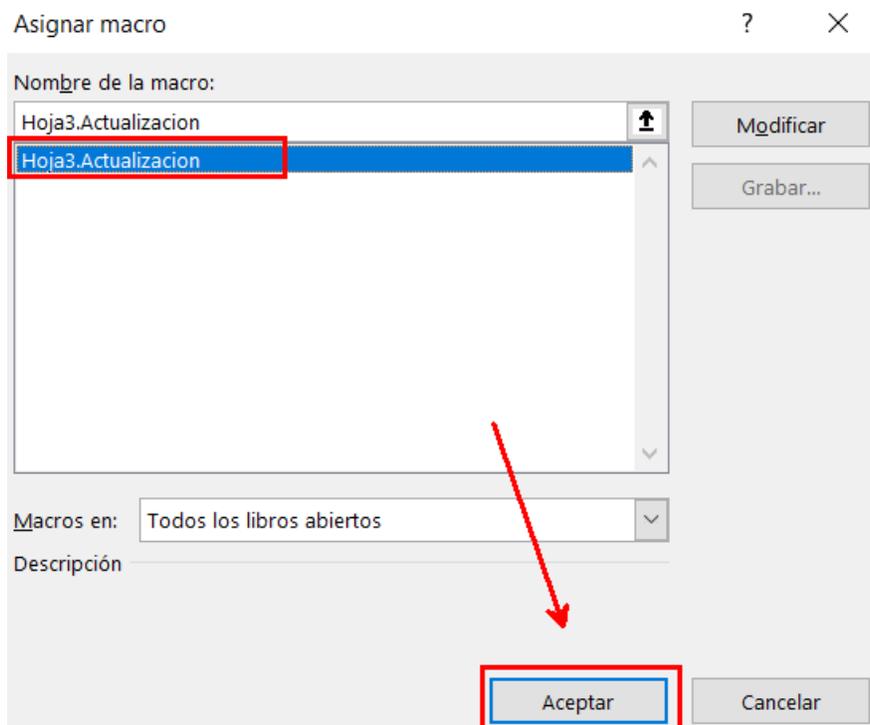


Figura 5.104 Asignar macro

Aceptar, damos clic en una celda y el botón esta listo para ser ejecutado.

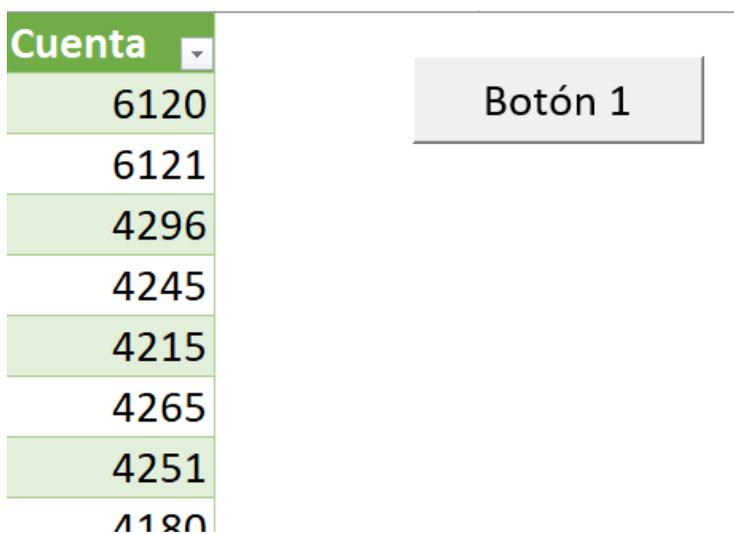


Figura 5.105 Botón con macro creado



ADVERTENCIA: Para guardar archivos con macros se debe guardar con una extensión especial, sino se pierde el código: Libro habilitado para macros



HACKS: Este truco no solo mejora la experiencia al usuario, también se puede combinar con otros procesos en Excel que nos ayuden a la productividad en el manejo de la información.

