

Excel con IA para Contadores. Nivel Avanzado

- Por: C.P. Alberto Monroy Salinas



COFIDE[®]
CAPACITACIÓN EMPRESARIAL

TEMA 1.

Validaciones

- 1. Validación de datos**
- 2. Cálculos automatizados de impuestos y retenciones con fórmulas inteligentes.**

Validación	¿Qué es?	¿Para qué sirve?	Cómo se aplica en Excel	Ejemplo práctico
Validación de datos	Herramienta que restringe o controla el tipo de información que se puede capturar en una celda.	Evita errores de captura, estandariza información y mejora el control interno.	Datos → Validación de datos (Lista, Número, Fecha, Longitud, Personalizada) + mensajes de entrada y error.	Lista desplegable para Tipo de operación: Ingreso/Egreso/Traslado; o validar que un RFC tenga 12–13 caracteres.
Cálculos automatizados de impuestos y retenciones con fórmulas inteligentes	Uso de fórmulas y reglas para calcular automáticamente impuestos/retenciones según condiciones (tasa, régimen, tipo de operación, moneda, etc.).	Reduce cálculos manuales, asegura consistencia y facilita auditoría/revisión.	Fórmulas con SI , SI.CONJUNTO , SUMAR.SI.CONJUNTO , REDONDEAR , BUSCARX/BUSCARV , ÍNDICE+COINCIDIR y tablas de tasas. Se complementa con formato condicional para alertas.	Calcular IVA 16% solo si el “Objeto impuesto” aplica: <code>=SI(Objeto="Sí";REDONDEAR(Base*0.16;2);0)</code> ; o retención ISR según tabla de tasas por tipo de servicio usando BUSCARX .

Cálculos automatizados de impuestos y retenciones con fórmulas inteligentes, se verán directo en Excel

TEMA 2.

Macros con IA

- 1. Grabador de macros de Excel**
- 2. Grabando mi primera macro**
- 3. Análisis de la macro**
- 4. Solicitar una macro a la IA**

Tema	Descripción	¿Para qué sirve?	Ejemplo práctico
Grabador de macros de Excel	Herramienta que registra las acciones realizadas por el usuario y las convierte automáticamente en código VBA.	Automatizar tareas repetitivas y aprender la estructura básica de VBA.	Grabar el formato de encabezados, ajustes de columnas o copiado de datos.
Grabando mi primera macro	Proceso guiado para crear una macro sin escribir código manualmente.	Crear automatizaciones simples de forma rápida.	Grabar una macro que aplique negritas, color y centrado a una tabla.
Código generado por el grabador	VBA automático que incluye selecciones, rangos fijos y formatos.	Servir como base para entender cómo Excel traduce acciones a código.	Uso de Range("A1").Select y Selection.Font.Bold = True.
Análisis de la macro	Revisión del código grabado para entender su funcionamiento y detectar mejoras.	Optimizar macros, hacerlas más rápidas y dinámicas.	Identificar rangos fijos que deben volverse dinámicos.
Optimización de macros	Ajustes al código para eliminar selecciones, usar variables y rangos dinámicos.	Mejorar desempeño y evitar errores al cambiar los datos.	Reemplazar Select por referencias directas a rangos.
Macros asistidas por IA	Uso de inteligencia artificial para generar o mejorar código VBA a partir de instrucciones en lenguaje natural.	Reducir tiempo de desarrollo y evitar errores de programación.	Pedir a la IA una macro para importar datos o validar información.
Solicitud correcta de una macro a la IA	Describir claramente el objetivo, reglas y resultados esperados.	Obtener macros funcionales y adaptadas a necesidades reales.	“Crear una macro que importe datos y los valide automáticamente”.
Uso combinado: grabador + IA	Integración del código grabado con mejoras propuestas por la IA.	Crear macros avanzadas partiendo de acciones simples.	Grabar una macro básica y pedir a la IA que la optimice.

MACROS CON IA (Excel VBA)

1) Grabador de macros de Excel

El **grabador de macros** es una herramienta que registra acciones que haces en Excel (clics, formatos, copiado, filtros, etc.) y las convierte en **código VBA**. Sirve para:

- Automatizar tareas repetitivas sin programar desde cero.
- Aprender VBA viendo el código que genera Excel.
- Crear una “base” que luego se mejora (limpieza, variables, rangos dinámicos).

Dónde está: pestaña **Desarrollador (Developer)** → **Grabar macro**

(Si no ves “Desarrollador”: Archivo → Opciones → Personalizar cinta → activar “Desarrollador”)

2) Grabando mi primera macro (paso a paso)

Ejemplo: dar formato a encabezados y ajustar columnas.

Ve a **Desarrollador** → **Grabar macro**

Nombre: FormatoEncabezados

Guardar macro en: **Este libro**

(Opcional) Tecla rápida y descripción

Clic en **Aceptar**

Ahora realiza acciones (por ejemplo):

- Selecciona **A1:D1**
- Poner **negritas**
- Aplicar **relleno**
- Centrar texto
- Autoajustar columnas

Ve a **Desarrollador** → **Detener grabación**

3) Análisis de la macro (qué revisar y cómo mejorarla)

Para ver el código:

• **Desarrollador → Macros → seleccionar → Editar**

(o bien ALT + F11 para abrir el editor VBA)

En el código grabado normalmente verás cosas como:

- Range("A1:D1").Select (mucho "Select", no es eficiente)
- Referencias absolutas a celdas (no es dinámico)
- Formatos repetidos

Qué debes analizar:

- **Qué rangos usa** (¿son fijos o deberían adaptarse?)
- **Qué objetos selecciona** (ideal: evitar Select/Activate)
- **Qué hace exactamente** (formato, copiar, filtrar, etc.)
- **En qué hoja trabaja** (ActiveSheet vs hoja específica)
- **Qué pasa si cambian los datos** (última fila, columnas nuevas)

Mejoras típicas:

- Quitar Select y trabajar directo:
 - En vez de seleccionar, aplicar formato directamente al rango.
- Hacer rangos dinámicos:
 - Usar última fila/columna para que la macro funcione siempre.
- Agregar control de errores:
 - Validar si existe hoja, si hay datos, etc.
- Agregar comentarios para documentar.

4) Solicitar una macro a la IA (cómo pedirla bien)

Aquí es donde “Macros con IA” te ahorra muchísimo tiempo: tú describes la tarea y la IA te devuelve el VBA listo (y lo ajustamos con tus reglas).

Estructura recomendada para pedir una macro:

- 1.**Objetivo:** qué quieres lograr
- 2.**Hoja(s):** nombre de hoja y dónde inicia la tabla
- 3.**Rangos:** columnas, encabezados, fila de inicio
- 4.**Reglas:** filtros, condiciones, validaciones
- 5.**Salida:** dónde debe escribir resultados (otra hoja, misma hoja, tabla)
- 6.**Restricciones:** velocidad, no usar Select, late binding, etc.

Ejemplo de solicitud bien hecha:

•“Quiero una macro que importe XML de pagos (1.0 y 2.0) a la hoja CFDI_Pagos, colocando encabezados en la fila 7, que calcule el mes/año por fecha de pago, y que registre la ruta del archivo. Debe funcionar con rangos dinámicos y sin Select.”

Buenas prácticas al pedir macros:

- Pide que sea **compatible** (late binding si aplica, sin referencias).
- Pide que use Application.ScreenUpdating=False, EnableEvents=False.
- Pide que escriba en bloque (arrays) si son muchos registros.
- Pide validaciones y mensajes de error claros.

TEMA 3.

Automatización con macros

- 1. Definición de variables**
- 2. Guardar datos en variables**
- 3. Ejecución de macros mediante el valor de una celda**
- 4. Calculadora de ISR de salarios con macros**

Tema	Descripción	¿Para qué se utiliza?	Ejemplo práctico
Automatización con macros	Uso de código VBA para ejecutar tareas de forma automática en Excel.	Reducir trabajo manual, ahorrar tiempo y estandarizar procesos.	Automatizar el cálculo de nómina, impuestos o generación de reportes.
Definición de variables	Declaración de espacios en memoria para almacenar datos temporales durante la ejecución de una macro.	Manipular valores sin depender directamente de las celdas.	Dim sueldo As Double para guardar el salario del trabajador.
Tipos de variables	Existen distintos tipos según el dato que se almacene.	Evitar errores y mejorar el rendimiento del código.	String (texto), Double (importes), Integer (números enteros), Date (fechas).
Guardar datos en variables	Asignar a una variable el valor contenido en una celda o calculado en la macro.	Trabajar con datos sin modificar directamente la hoja.	sueldo = Range("B2").Value
Uso de variables en cálculos	Emplear variables para realizar operaciones matemáticas o lógicas.	Hacer cálculos más claros y reutilizables.	isr = sueldo * 0.15
Ejecución de macros mediante el valor de una celda	Ejecutar diferentes macros según el contenido de una celda.	Controlar procesos con listas desplegables o indicadores.	Si la celda dice "Calcular", ejecutar una macro específica.
Condiciones con celdas	Uso de estructuras If para evaluar valores en celdas.	Automatizar decisiones dentro del archivo.	Si el régimen es "Salarios", ejecutar cálculo de ISR.
Calculadora de ISR de salarios con macros	Macro que calcula automáticamente el ISR a partir del sueldo del trabajador.	Automatizar el cálculo conforme a reglas definidas.	Capturar sueldo, calcular ISR y mostrar neto a pagar.
Ventajas de macros en cálculos fiscales	Integran lógica, validación y cálculo en un solo proceso.	Reducir errores humanos y asegurar consistencia.	Uso en plantillas de nómina o simuladores fiscales.

TIPOS DE VARIABLES EN VBA (EXCEL)

Tipo de variable	¿Qué almacena?	Ejemplo de declaración	Uso práctico
Integer	Números enteros pequeños (–32,768 a 32,767).	Dim cantidad As Integer	Contar registros o filas.
Long	Números enteros grandes.	Dim ultimaFila As Long	Identificar la última fila con datos.
Single	Números con decimales (precisión simple).	Dim tasa As Single	Porcentajes simples.
Double	Números con decimales (mayor precisión).	Dim importe As Double	Cálculos de impuestos y montos.
Currency	Valores monetarios con alta precisión.	Dim sueldo As Currency	Importes financieros y nómina.
String	Texto y cadenas de caracteres.	Dim rfc As String	Guardar nombres, RFC, conceptos.
Boolean	Valores lógicos: Verdadero o Falso.	Dim aplicaSR As Boolean	Evaluar condiciones.
Date	Fechas y horas.	Dim fechaPago As Date	Manejo de fechas de pago o emisión.
Variant	Puede almacenar cualquier tipo de dato.	Dim dato As Variant	Datos variables o desconocidos.
Object	Referencia a objetos de Excel.	Dim ws As Worksheet	Manipular hojas, libros, rangos.
Range	Celdas o rangos específicos.	Dim celda As Range	Leer o escribir valores en celdas.
Array	Conjunto de valores en una sola variable.	Dim totales(1 To 12) As Double	Almacenar datos por mes.

EJEMPLOS: GUARDAR VALORES EN VARIABLES (VBA)

1 Guardar el valor de una celda en una variable

Código

```
Dim sueldo As Double  
sueldo = Range("B2").Value
```

```
Dim nombre As String  
nombre = Cells(2, 1).Value
```

Qué hace

Guarda el valor de la celda B2 en la variable **sueldo**.

Guarda el texto de la celda A2 en la variable **nombre**.

2 Guardar valores numéricos directos

Código

```
Dim tasaISR As Double  
tasaISR = 0.16
```

```
Dim edad As Integer  
edad = 35
```

Qué hace

Guarda un valor fijo (16%) en la variable.

Guarda un número entero.

3 Guardar el resultado de un cálculo

Código

Dim total As Doubletotal = Range("B2").Value * 0.16

Dim neto As Currencyneto = sueldo - isr

Qué hace

Guarda el resultado del cálculo en **total**.

Guarda el sueldo neto calculado.

4 Guardar texto concatenado

Código

Dim mensaje As Stringmensaje = "Sueldo neto: \$" & neto

Dim folio As Stringfolio = "FOL-" & Range("A2").Value

Qué hace

Guarda un texto combinado con una variable.

Crea y guarda un folio automático.

5 Guardar fechas en variables

Código

```
Dim fechaPago As Date  
fechaPago = Date  
fechaPago = Range("C2").Value
```

Qué hace

Guarda la fecha actual.
Guarda una fecha capturada en la hoja.

6 Guardar valores según una condición (IF)

Código

```
Dim aplicaSR As Boolean  
aplicaSR = (Range("D2").Value = "Sí")
```

Qué hace

Guarda VERDADERO o FALSO según el contenido de la celda.

7 Guardar valores usando Select Case

```
Dim tasa As Double
Select Case Range("E2").Value
Case "General"
    tasa = 0.16
Case "Reducido"
    tasa = 0.08
Case Else
    tasa = 0
End select
```

Casos prácticos directamente en Excel



**POR SU
ATENCIÓN
¡GRACIAS!**

COFIVE®
CAPACITACIÓN EMPRESARIAL

CONTÁCTANOS



PÁGINA WEB

www.cofide.mx



TELÉFONO

01 (55) 46 30 46 46



DIRECCIÓN

Av. Río Churubusco 594 Int. 203,
Col. Del Carmen Coyoacán, 04100
CDMX

SIGUE NUESTRAS REDES SOCIALES



COFIDE



Cofide SC



Cofide SC



@cofide.mx