

---

# 500 MACROS

## COMPLETO MANUAL DE PROGRAMACIÓN VBA

Para versiones Excel xlsx (2007 en adelante)

---

### INDICE DE CONTENIDOS EN ESTA VERSIÓN DEMO

En esta versión se incluye el **Índice de Contenidos completo y el desarrollo de algunos de sus capítulos** a cuyos temas se puede acceder desde el mismo Índice.

### ***CAPÍTULOS DESARROLLADOS 1,2,3 Y 10***

El manual en su versión completa presenta **290 páginas, con más de una macro de ejemplo en cada tema.**

Con Imágenes, notas aclaratorias y vínculos a las distintas hojas.

Con ejercicios al final de cada capítulo. El manual se acompaña con el libro conteniendo la resolución de los ejercicios.

*por Elsa Matilde Meyer*

- 
- 1 INTRODUCCIÓN
  - 2 CONCEPTOS BÁSICOS
    - 2.1 ¿QUÉ ES UNA MACRO?
    - 2.2 ELEMENTOS DE UNA MACRO
  - 3 EL EDITOR DE MACROS
    - 3.1 ELEMENTOS DEL EDITOR
    - 3.2 CÓMO CREAR UNA MACRO
      - 3.2.1 *LA GRABADORA DE MACROS*
      - 3.2.2 *COPIAR O REDACTAR MACROS*
      - 3.2.3 *IMPORTAR MACROS*
    - 3.3 DÓNDE COLOCAR LAS MACROS?
  - 4 CÓMO EJECUTAR MACROS
    - 4.1 LA EJECUCIÓN SEGÚN EL TIPO DE MACROS
    - 4.2 CREAR GRUPO DE BOTONES PERSONALES EN CINTA DE OPCIONES
    - 4.3 PRIVADAS O PÚBLICAS
    - 4.4 LAS MACROS AUTO\_OPEN Y AUTO\_CLOSE
    - 4.5 ASIGNAR MACRO A UN BOTÓN (mediante código).
    - 4.6 EJERCICIOS
  - 5 EVENTOS, MÉTODOS Y PROCEDIMIENTOS
    - 5.1 LOS EVENTOS
      - 5.1.1 *EVENTOS DE LIBROS*
      - 5.1.2 *EVENTOS DE HOJAS*
      - 5.1.3 *EVENTOS DE OBJETOS*
      - 5.1.4 *EVENTOS DE LA APLICACIÓN*
    - 5.2 MÉTODOS Y PROPIEDADES
      - 5.2.1 *MÉTODOS DE LIBROS*
      - 5.2.2 *PROPIEDADES DE LIBROS*
      - 5.2.3 *MÉTODOS DE HOJAS*
      - 5.2.4 *PROPIEDADES DE HOJAS*

- 
- 5.2.5 *MÉTODOS DE RANGE O CELDAS*
  - 5.2.6 *PROPIEDADES DE RANGE O CELDAS*
  - 5.2.7 *MÉTODOS DE LA APLICACIÓN*
  - 5.2.8 *PROPIEDADES DE LA APLICACIÓN*
- 6 REFERENCIAS A OBJETOS EXCEL
- 6.1 HACIENDO REFERENCIA A LIBROS
  - 6.2 HACIENDO REFERENCIA A HOJAS
  - 6.3 HACIENDO REFERENCIA A RANGOS Y CELDAS
    - 6.3.1 *RANGE O CELLS*
    - 6.3.2 *SELECTION*
    - 6.3.3 *TARGET*
  - 6.4 NOMBRANDO OBJETOS
  - 6.5 EJERCICIOS
- 7 PROBANDO Y CONTROLANDO LA EJECUCIÓN DE MACROS
- 7.1 CÓMO PROBAR UNA MACRO
  - 7.2 CONTROLAR UNA MACRO
  - 7.3 CONOCER EL VALOR QUE TOMAN LAS VARIABLES
  - 7.4 CÓMO EVITAR QUE UNA INSTRUCCIÓN SE EJECUTE
  - 7.5 ACCEDER A LA AYUDA DESDE UNA LÍNEA DE CÓDIGO
  - 7.6 SALIR DE UNA RUTINA
  - 7.7 INTERRUPIR LA EJECUCIÓN DE UNA MACRO
  - 7.8 EJERCICIOS
- 8 SEGURIDAD EN EL PROYECTO
- 8.1 CÓMO PROTEGER UN PROYECTO
  - 8.2 EVITAR QUE LAS MACROS SE VEAN DESDE EL MENÚ
  - 8.3 HABILITAR O NO LAS MACROS
  - 8.4 IMPEDIR QUE SE DETENGA LA EJECUCIÓN DE MACROS
- 9 TRATAMIENTO DE VARIABLES
- 9.1 TIPO DE VARIABLES

- 
- 9.2 DURACIÓN DE LAS VARIABLES
  - 9.3 DETERMINAR EL TIPO DE VARIABLE
  - 9.4 LIMPIANDO VARIABLES
  - 9.5 FUNCIONES DE CONVERSIÓN DE TIPOS
  - 9.6 OTRAS FUNCIONES DE CONVERSIÓN
  - 9.7 EJERCICIOS
  - 10 TRABAJANDO CON CADENAS Y MATRICES
    - 10.1 OBTENER EL LARGO DE UNA CADENA
    - 10.2 EXTRAER PARTES DE UNA CADENA
    - 10.3 DETECTAR TEXTO DENTRO DE UNA CADENA
    - 10.4 ARMANDO CADENAS
    - 10.5 INTRODUCIR CARACTERES ESPECIALES
    - 10.6 CREANDO CADENAS DE LARGO FIJO
    - 10.7 OBTENER LA PARTE NUMERICA DE UNA CADENA
    - 10.8 OTRAS FUNCIONES PARA CADENAS
    - 10.9 MATRICES
      - 10.9.1 *OBTENER EL VALOR DE UN ELEMENTO CONOCIENDO SU NÚMERO O UBICACIÓN DENTRO DE LA MATRIZ.*
      - 10.9.2 *LLENAR UN CONTROL COMBOBOX CON LOS MESES DEL AÑO.*
      - 10.9.3 *DEFINIR LONGITUD DE LA MATRIZ*
    - 10.10 EJERCICIOS
  - 11 CONTROLANDO ERRORES
    - 11.1 ON ERROR RESUME NEXT
    - 11.2 ON ERROR GOTO *llamada*
    - 11.3 ON ERROR GOTO 0
    - 11.4 ERR
    - 11.5 INHABILITAR EVENTOS QUE PROVOQUEN ERROR
    - 11.6 EJERCICIOS
  - 12 BUCLES – OTROS COMANDOS ESPECIALES
    - 12.1 FOR EACH .... NEXT

- 
- 12.2 FOR ..... NEXT
  - 12.3 WHILE .... WEND
  - 12.4 DO WHILE .... LOOP
  - 12.5 DO UNTIL.... LOOP
  - 12.6 IF.... ELSEIF....ELSE... END IF
  - 12.7 USO DE SET
  - 12.8 WITH....END WITH
  - 12.9 USO DE SELECT CASE
  - 12.10 EJERCICIOS
- 13 CUADROS DE MENSAJE: MSGBOX e INPUTBOX
- 13.1 LOS CUADROS DE MENSAJE O MSGBOX
    - 13.1.1 *SINTAXIS Y EJEMPLOS DE MSGBOX*
    - 13.1.2 *MSGBOX COMO POPUP*
  - 13.2 LOS CUADROS DE DIÁLOGO O INPUTBOX
    - 13.2.1 *SINTAXIS Y EJEMPLOS DE INPUTBOX*
    - 13.2.2 *CONTROLAR QUE EL CAMPO NO QUEDE VACÍO*
    - 13.2.3 *ESTABLECER UN VALOR PREDETERMINADO EN UN INPUTBOX*
    - 13.2.4 *INGRESAR - SELECCIONAR UN RANGO CON INPUTBOX*
    - 13.2.5 *CONTROLAR QUE EL DATO INGRESADO SEA FECHA*
  - 13.3 EJERCICIOS:
- 14 TRABAJANDO CON CELDAS Y RANGOS
- 14.1 SELECCIÓN DE CELDAS O RANGOS
    - 14.1.1 *SELECCIÓN DE CELDAS A CIERTA DISTANCIA DE LA ACTIVA*
    - 14.1.2 *AMPLIAR UN RANGO SELECCIONADO*
    - 14.1.3 *SELECCIÓN DE RANGO UTILIZANDO VARIABLES*
    - 14.1.4 *SELECCIONAR LA **REGIÓN** DONDE SE ENCUENTRA LA CELDA ACTIVA*
    - 14.1.5 *IDENTIFICAR EL **RANGO** OCUPADO EN LA HOJA*
  - 14.2 ENCONTRAR FIN DE RANGO – ÚLTIMA FILA / ÚLTIMA COLUMNA
  - 14.3 DEVOLVER LA DIRECCIÓN O REFERENCIA DE UN RANGO O CELDA

- 
- 14.4 OBTENER EL NÚMERO DE CELDAS, FILAS O COLUMNAS DE UN RANGO
  - 14.5 COMBINAR – DESCOMBINAR CELDAS
  - 14.6 BORRAR O LIMPIAR CELDAS
  - 14.7 ELIMINAR CELDAS
  - 14.8 INSERTAR CELDAS O RANGOS
  - 14.9 ELIMINAR FILAS - COLUMNAS
  - 14.10 INSERTAR FILAS - COLUMNAS
  - 14.11 OCULTAR – MOSTRAR (FILAS y COLUMNAS)
  - 14.12 COMENTARIOS EN CELDAS
    - 14.12.1 INSERTAR – MODIFICAR - QUITAR COMENTARIOS*
    - 14.12.2 COPIAR COMENTARIOS DE OTRAS CELDAS*
  - 14.13 FORMATO A CELDAS Y RANGOS
  - 14.14 AUTOAJUSTAR COLUMNAS
  - 14.15 INSERTAR NOMBRES DE RANGO
  - 14.16 INSERTAR HIPERVÍNCULOS
  - 14.17 TRANSPONER FILAS Y COLUMNAS
  - 14.18 DETECTAR SI LA CELDA CONTIENE FÓRMULA
  - 14.19 REALIZAR BÚSQUEDAS CONTINUAS – Acciones a tomar
  - 14.20 BLOQUEAR CELDAS
  - 14.21 RESALTAR CELDA ACTIVA
  - 14.22 ORDENAR RANGOS
  - 14.23 ORDENAR UN RANGO VARIABLE
  - 14.24 FILTRAR RANGOS
  - 14.25 RELLENAR SERIES (FECHA/HORA)
  - 14.26 COPIAR UN MISMO RANGO EN VARIAS HOJAS
  - 14.27 EXPORTAR UNA COLUMNA O RANGO COMO TXT
  - 14.28 EJERCICIOS
- 15 TRABAJANDO CON HOJAS
    - 15.1 SELECCIONAR HOJAS

- 
- 15.1.1 *SELECCIONAR LA HOJA ANTERIOR O POSTERIOR*
  - 15.1.2 *SELECCIONAR TODAS LAS HOJAS DE UN LIBRO*
  - 15.2 OBTENER EL NOMBRE Y POSICIÓN DE LAS HOJAS
  - 15.3 CAMBIAR COLOR DE PESTAÑA O ETIQUETA
  - 15.4 MOVER UNA HOJA DE LUGAR
  - 15.5 PROTEGER – DESPROTEGER HOJAS
  - 15.6 VISTA PREVIA - IMPRESIÓN DE HOJAS
  - 15.7 INSERTAR HOJAS
  - 15.8 ELIMINAR HOJAS
    - 15.8.1 *ELIMINAR HOJAS MEDIANTE CÓDIGO*
    - 15.8.2 *CONTROLAR LA ELIMINACIÓN DE HOJAS*
  - 15.9 COPIAR HOJA
  - 15.10 COPIAR HOJA COMO LIBRO
  - 15.11 EXPORTAR HOJA COMO TXT
  - 15.12 OCULTAR – MOSTRAR HOJAS
  - 15.13 INSERTAR IMAGEN EN UNA HOJA
  - 15.14 ESTABLECER ÁREA VISIBLE DE UNA HOJA
  - 15.15 ELIMINAR TODOS LOS COMENTARIOS DE UNA HOJA
  - 15.16 CONOCER SI EL LIBRO CONTIENE TABLAS (CONTARLAS)
  - 15.17 EJERCICIOS
  - 16 TRABAJANDO CON DIRECTORIOS
    - 16.1 ESTABLECER LA RUTA DE ACCESO PREDETERMINADO
    - 16.2 CREANDO UN NUEVO DIRECTORIO
    - 16.3 CONFIRMAR SI UN DIRECTORIO EXISTE
    - 16.4 SABER SI UN ARCHIVO EXISTE EN UN DIRECTORIO
    - 16.5 ELIMINAR UN ARCHIVO DE UN DIRECTORIO
    - 16.6 OBTENER SOLO LAS SUBCARPETAS DE UN DIRECTORIO
    - 16.7 OBTENER TODOS LOS LIBROS DE UNA CARPETA
    - 16.8 OBTENER TODOS LOS ELEMENTOS DE UNA CARPETA (SUBCARPETAS Y ARCHIVOS)

- 
- 16.9 ELIMINAR UN DIRECTORIO DEL DISCO
  - 16.10 ESTABLECER NUEVA UNIDAD DE DISCO
  - 16.11 ELEGIR DIRECTORIO PARA COMPLETAR ALGÚN PROCESO
  - 16.12 EJERCICIOS
- 17 TRABAJANDO CON LIBROS
- 17.1 ABRIR LIBROS (OPEN)
    - 17.1.1 *ABRIR UN LIBRO. ABRIR LIBRO CON CLAVE*
    - 17.1.2 *ABRIR LIBRO BUSCANDO LA CARPETA QUE LO CONTIENE*
    - 17.1.3 *EJECUTAR MACRO AL ABRIR UN LIBRO*
    - 17.1.4 *INHABILITAR LA OPCIÓN DE ACTUALIZAR VÍNCULOS AL ABRIR OTROS LIBROS*
    - 17.1.5 *IMPEDIR QUE SE ABRAN OTROS LIBROS MIENTRAS ESTÉ NUESTRO LIBRO ABIERTO*
    - 17.1.6 *CONOCER FECHA DE CREACIÓN DE UN LIBRO*
    - 17.1.7 *SEPARAR NOMBRE DE LIBRO DE SU RUTA.*
    - 17.1.8 *NO ABRIR EL LIBRO SI ESTÁ EN MODO LECTURA.*
  - 17.2 ACTIVAR O SELECCIONAR UN SEGUNDO LIBRO
  - 17.3 OBTENER LA RUTA DE UN LIBRO ABIERTO
  - 17.4 MINIMIZAR UN LIBRO (MOSTRAR UN USERFORM)
  - 17.5 OBTENER EL NOMBRE DE UN LIBRO ABIERTO
  - 17.6 MODIFICAR LA FORMA DE CÁLCULO DEL LIBRO O DE UNA HOJA
  - 17.7 ELIMINAR VINCULOS EXTERNOS
  - 17.8 OBTENER PROPIEDAD FECHA Y HORA DE LIBROS GUARDADOS
  - 17.9 OBTENER TAMAÑO DE ARCHIVOS
  - 17.10 COPIAR UN LIBRO SIN ABRIRLO
  - 17.11 CAMBIAR EL NOMBRE DE UN LIBRO SIN ABRIRLO
  - 17.12 ELIMINAR NOMBRES DE RANGOS EN LIBRO
  - 17.13 GUARDANDO LIBROS
    - 17.13.1 *GUARDAR EL LIBRO ACTIVO*
    - 17.13.2 *GUARDAR UN LIBRO CON CLAVE*
    - 17.13.3 *GUARDAR UN LIBRO CON OTRO NOMBRE – FORMATO O RUTA*

- 
- 17.13.4 *GUARDAR UN LIBRO COMO COPIA*
  - 17.13.5 *GUARDAR UNA HOJA COMO LIBRO*
  - 17.13.6 *GUARDAR LIBRO SIN MÓDULOS*
  - 17.13.7 *GUARDAR LIBRO SIN CÓDIGO O MACROS*
  - 17.13.8 *CONTROLAR ERRORES ANTES DE GUARDAR UN LIBRO*
  - 17.13.9 *IMPEDIR QUE UN LIBRO SE GUARDE*
  - 17.14 CERRANDO LIBROS
    - 17.14.1 *CERRAR LIBRO – CERRAR TODOS LOS LIBROS EN USO*
    - 17.14.2 *CERRAR UN LIBRO SIN GUARDAR LOS CAMBIOS*
    - 17.14.3 *CERRAR UN LIBRO GUARDANDO LOS CAMBIOS SIN CONSULTAR*
  - 17.15 EJERCICIOS
  - 18 CONTROLANDO TECLAS Y MENSAJES DE ALERTA
    - 18.1 NO MOSTRAR AVISOS DE ALERTA
    - 18.2 NO MOSTRAR LA EJECUCIÓN DE LA MACRO O EL MOVIMIENTO DE HOJAS
    - 18.3 NO MOSTRAR EL CURSOR O LA CELDA ACTIVA
    - 18.4 PERMANECER EN LA CELDA ACTIVA LUEGO DEL ENTER
    - 18.5 EJECUTAR UNA MACRO CON DOBLE CLICK
    - 18.6 CONTROLAR LAS TECLAS ENTER Y FLECHAS ABAJO\_ARRIBA
    - 18.7 CONTROLAR OTRAS TECLAS MEDIANTE CÓDIGO
    - 18.8 CONTROLAR LA TECLA ESC- CANCELAR A CRITERIO
    - 18.9 IMPEDIR QUE SE CANCELE UNA MACRO CON ESC
    - 18.10 ASIGNAR UNA MACRO A UNA TECLA ESPECIAL
    - 18.11 EJECUTAR UNA MACRO AL PRESIONAR ENTER
    - 18.12 EJERCICIOS
  - 19 TRABAJANDO CON FILTROS
    - 19.1 FILTRAR UN RANGO
    - 19.2 FILTRAR UN RANGO POR COLUMNA VARIABLE – CRITERIO VARIABLE
    - 19.3 FILTRAR RANGO POR MÁS DE UNA COLUMNA
    - 19.4 RECONOCER SI UNA TABLA TIENE APLICADOS LOS FILTROS

- 
- 19.5 COPIAR LAS CELDAS FILTRADAS A OTRO DESTINO
  - 19.6 FILTRO AVANZADO – USO DE VARIABLES PARA CRITERIO Y DESTINO
  - 19.7 FILTRO AVANZADO - CRITERIOS CON VALORES DECIMALES
  - 19.8 AUTOFILTROS CON CAMPOS FECHA
  - 19.9 OBTENER REFERENCIAS DE CELDAS FILTRADAS
    - 19.9.1 SELECCIONAR LA PRIMER FILA DE UN FILTRO
    - 19.9.2 *GUARDAR LA REFERENCIA DE LA PRIMER CELDA DE UN FILTRO (NO CRITERIO)*
    - 19.9.3 *INDICAR O SELECCIONAR LA ÚLTIMA FILA DE UN FILTRO*
    - 19.9.4 *OBTENER TODAS LAS FILAS FILTRADAS DE UN RANGO*
  - 19.10 OBTENER EL TOTAL DE CELDAS FILTRADAS
  - 19.11 UBICAR EN RANGO FILTRADO LA CELDA QUE CUMPLA CON CRITERIO
  - 19.12 SUMAR UNA COLUMNA DE DATOS EN UN RANGO FILTRADO
  - 19.13 FILTRAR POR COLORES O ICONOS
  - 19.14 EJERCICIOS
  - 20 IMPRESIÓN
    - 20.1 ACERCA DE LA IMPRESIÓN
    - 20.2 ESTABLECER ÁREA DE IMPRESIÓN
    - 20.3 IMPEDIR LA IMPRESIÓN
    - 20.4 ENVIAR INFORMACIÓN AL ENCABEZADO Y PIÉ
    - 20.5 INSERTAR IMAGEN EN ENCABEZADOS O PIE DE PÁGINA
    - 20.6 CONFIGURAR LA PÁGINA
    - 20.7 LLAMAR AL CUADRO DE OPCIONES PARA IMPRIMIR
    - 20.8 IMPRIMIR LISTA DE HOJAS
    - 20.9 EJERCICIOS
  - 21 TRABAJANDO CON GRÁFICOS
    - 21.1 SELECCIONAR GRÁFICOS IGNORANDO SUS NOMBRES
    - 21.2 CORTAR UN GRÁFICO DE UNA HOJA Y PEGARLO EN OTRA
    - 21.3 ASIGNAR LOS DATOS DE ORIGEN DE UN GRÁFICO
    - 21.4 CREAR UNA HOJA PARA UBICAR UN GRÁFICO

- 
- 21.5 MODIFICAR DIMENSIONES DEL GRÁFICO
  - 21.6 ELIMINAR UNO O TODOS LOS GRAFICOS DE UNA HOJA
  - 21.7 COLOREAR SERIES POR MACRO
  - 21.8 AJUSTAR LA UBICACIÓN DE UN GRÁFICO
  - 21.9 ACTUALIZAR TÍTULOS DE GRÁFICOS SEGÚN VALOR DE CELDA
  - 21.10 EJERCICIOS
- 22 TRABAJANDO CON FÓRMULAS y FUNCIONES
- 22.1 NOTACIÓN PARA FÓRMULAS
  - 22.2 INTRODUCIR FÓRMULAS EN CELDAS
  - 22.3 ARMAR FÓRMULA CON DATOS VARIABLES
  - 22.4 COLOCAR RESULTADO DE FÓRMULA EN CELDAS
  - 22.5 EL OPERADOR MOD
- 23 FUNCIONES PERSONALES - COMPLEMENTOS
- 23.1 CÓMO CREAR UN COMPLEMENTO
  - 23.2 CREAR FUNCIONES PERSONALES
  - 23.3 SINTAXIS DE FUNCIONES
  - 23.4 CÓMO LLAMAR A UNA FUNCIÓN PERSONAL
  - 23.5 EJEMPLO: FUNCION ENCONTRAR ULTIMO DATO
  - 23.6 EJERCICIOS
- 24 TRABAJANDO CON EL ENTORNO EXCEL
- 24.1 COMENTARIOS SOBRE EL ASPECTO GENERAL
    - 24.1.1 *OCULTAR – MOSTRAR: PESTAÑAS DE LAS HOJAS*
    - 24.1.2 *OCULTAR – MOSTRAR: BARRAS DE DESPLAZAMIENTO*
    - 24.1.3 *OCULTAR – MOSTRAR: ENCABEZADOS DE FILAS Y COLUMNAS*
    - 24.1.4 *OCULTAR – MOSTRAR: PANTALLA COMPLETA*
    - 24.1.5 *OCULTAR – MOSTRAR: CINTA DE OPCIONES (RIBBON)*
  - 24.2 IMPEDIR EL USO DEL CLIC DERECHO
  - 24.3 MOSTRAR CAJAS DE DIÁLOGO
  - 24.4 MOSTRAR MENSAJES EN LA BARRA DE ESTADO

- 
- 24.5 OCULTAR BOTONES DE MINIMIZAR Y MAXIMIZAR
  - 24.6 INHABILITAR EL BOTÓN CERRAR DEL LIBRO
  - 24.7 EJECUTAR MACROS DE OTROS LIBROS
  - 24.8 EXPORTAR – IMPORTAR MODULOS
  - 24.9 SUSTITUIR UN TEXTO EN TODO EL CÓDIGO DEL PROYECTO
  - 24.10 EJERCICIOS
- 25 TRABAJANDO CON OTRAS APLICACIONES
- 25.1 TRABAJANDO CON WORD
    - 25.1.1 *COPIAR CELDAS EXCEL A UN DOCUMENTO WORD*
    - 25.1.2 *IMPRIMIR UN DOCUMENTO DE WORD DESDE EXCEL*
    - 25.1.3 *ABRIR UN WORD DESDE EXCEL Y GUARDARLO CON NOMBRE VALOR DE CELDA*
  - 25.2 TRABAJANDO CON OUTLOOK
    - 25.2.1 *IMPORTAR DATOS DE OUTLOOK*
    - 25.2.2 *ENVÍO DE CORREOS.*
  - 25.3 TRABAJANDO CON POWERPOINT
    - 25.3.1 *CREAR UN PPT CON DATOS DE CELDAS EXCEL*
    - 25.3.2 *AGREGAR DATOS A UN PPT YA CREADO*
    - 25.3.3 *COPIAR TABLA DE DATOS + GRÁFICO*
    - 25.3.4 *AJUSTAR UBICACIÓN DE LOS OBJETOS PEGADOS EN LA DIAPOSITIVA*
    - 25.3.5 *COPIAR DE EXCEL A UNA DIAPOSITIVA DETERMINADA*
  - 25.4 TRABAJANDO CON ACCESS
    - 25.4.1 *IMPORTAR DATOS DE ACCESS*
    - 25.4.2 *CAMBIAR EL NOMBRE A UNA CONEXIÓN EXISTENTE*
    - 25.4.3 *AGREGAR OTRAS CONSULTAS A LA CONEXIÓN EXISTENTE*
    - 25.4.4 *ELIMINAR CONEXIONES*
    - 25.4.5 *ACTUALIZAR CONEXIONES*
  - 25.5 ACTIVANDO OTRAS APLICACIONES
    - 25.5.1 *LLAMANDO AL BLOC DE NOTAS*
    - 25.5.2 *LLAMANDO A LA CALCULADORA*

- 
- 25.5.3 *ACTIVAR SONIDOS*
  - 25.5.4 *ACCEDER AL PORTAPAPELES*
  - 25.6 TRABAJANDO CON PDF
    - 25.6.1 *EXPORTAR COMO ARCHIVO PDF*
    - 25.6.2 *LLAMAR A UN PDF*
  - 25.7 EJERCICIOS
  - 26 CORREOS DESDE EXCEL – TRABAJANDO CON OUTLOOK Y OTROS
    - 26.1 *ENVIAR UN MAIL (SOLO TEXTO) DESDE EXCEL*
    - 26.2 *ENVIAR UN MAIL con varias filas de texto en el cuerpo del mensaje*
    - 26.3 *ENVIAR UN MAIL CON ARCHIVO ADJUNTO*
    - 26.4 *ENVIAR CORREO A LISTA DE DESTINATARIOS*
    - 26.5 *ENVIAR UNA HOJA EXCEL POR MAIL*
    - 26.6 *ENVIAR UN RANGO DE CELDAS COMO LIBRO ADJUNTO*
    - 26.7 *ENVÍO DE VARIAS HOJAS COMO LIBRO ADJUNTO*
    - 26.8 *ENVIO DEL LIBRO ACTIVO*
    - 26.9 *ENVIO DE UN LIBRO DISTINTO AL ACTIVO*
    - 26.10 *ENVIAR RANGO DE HOJA COMO CUERPO DEL MENSAJE*
  - 27 TRABAJANDO CON TABLAS DINÁMICAS
    - 27.1 *COMENTARIOS SOBRE TABLAS DINÁMICAS*
    - 27.2 *EVITAR LA ACTUALIZACIÓN AUTOMÁTICA DE TD*
    - 27.3 *CREAR UNA TABLA DINÁMICA CON DATOS VARIABLES*
    - 27.4 *CONVERTIR UN CAMPO EN UN CAMPO DE FILAS o columnas*
    - 27.5 *ELIMINAR TODAS LAS TABLAS DINAMICAS DE UNA HOJA*
    - 27.6 *USO DE VARIABLES EN TD*
    - 27.7 *AGREGAR FILTRO A TABLA DINÁMICA*
    - 27.8 *QUITAR FILTROS A UNA TABLA DINÁMICA*
    - 27.9 *ACTUALIZAR TABLA AL CAMBIO DE UNA CELDA*
    - 27.10 *EJERCICIOS*
  - 28 USO DE TEMPORIZADOR

- 
- 28.1 CARACTERÍSTICAS DEL MÉTODO ONTIME
  - 28.2 GUARDAR EL LIBRO CADA MINUTO
  - 28.3 MOSTRAR UN USERFORM DURANTE UN TIEMPO Y OCULTARLO
  - 28.4 LLLAMAR A UN USERFORM POR UNOS SEGUNDOS
  - 28.5 EJERCICIOS
- 29 PRINCIPALES MENSAJES DE ERROR – SU SIGNIFICADO Y CÓMO RESOLVERLOS
- 29.1 NO SE PUEDE CARGAR UN OBJETO.....
  - 29.2 NO SE ENCUENTRA LA BIBLIOTECA
  - 29.3 NO SE HA DEFINIDO LA VARIABLE.
  - 29.4 EL ACCESO AL PROYECTO NO ES DE CONFIANZA
  - 29.5 NO SE HA DEFINIDO SUB O FUNCIÓN
  - 29.6 SUBÍNDICE FUERA DE INTERVALO
  - 29.7 EL OBJETO NO ADMITE ESTA PROPIEDAD O MÉTODO
  - 29.8 NO SE ENCONTRÓ EL MÉTODO O EL DATO MIEMBRO
  - 29.9 DESBORDAMIENTO
  - 29.10 NO COINCIDEN LOS TIPOS
  - 29.11 FALTA DE PERFIL AL ENVIAR CORREO
- 30 TRABAJANDO CON OFFICE DE 64 BITS
- 30.1 COMENTARIOS SOBRE LOS DIFERENTES ENTORNOS
  - 30.2 Escritura de código que funciona en ambas versiones de 32 bits y 64 bits de Office
  - 30.3 Escritura de código que funciona en Microsoft Office 2010 (32 bits o 64 bits) y versiones anteriores de Office
  - 30.4 Resumen de actualizaciones de idioma de VBA7
- 31 RESOLUCIÓN DE EJERCICIOS:
- 31.1 EJERCICIOS CAPÍTULO 6
  - 31.2 EJERCICIOS CAPÍTULO 7
  - 31.3 EJERCICIOS CAPÍTULO 9
  - 31.4 EJERCICIOS CAPÍTULO 10
  - 31.5 EJERCICIOS CAPÍTULO 11
  - 31.6 EJERCICIOS CAPÍTULO 12

- 
- 31.7 EJERCICIOS CAPÍTULO 13
  - 31.8 EJERCICIOS CAPÍTULO 14
  - 31.9 EJERCICIOS CAPÍTULO 15
  - 31.10 EJERCICIOS CAPÍTULO 16
  - 31.11 EJERCICIOS CAPÍTULO 17
  - 31.12 EJERCICIOS CAPÍTULO 18
  - 31.13 EJERCICIOS CAPÍTULO 19
  - 31.14 EJERCICIOS CAPÍTULO 20
  - 31.15 EJERCICIOS CAPÍTULO 21
  - 31.16 EJERCICIOS CAPÍTULO 23
  - 31.17 EJERCICIOS CAPÍTULO 24
  - 31.18 EJERCICIOS CAPÍTULO 25
  - 31.19 EJERCICIOS CAPÍTULO 27
  - 31.20 EJERCICIOS CAPÍTULO 28

## 1 INTRODUCCIÓN

La intención de este Manual es **guiar** al usuario de Excel a potenciar las planillas de cálculo con programación, y **prepararlo** para que, al final del estudio de este manual, sea un experto en **Programación en Excel**. Los que nunca han trabajado con algún lenguaje de programación verán que **muy fácilmente** podrán 'personalizar' sus libros **adaptando o creando rutinas** con código **VBA** (Visual Basic para Aplicaciones)

El manual está dirigido a:

**1-** los que **nunca han programado**: Todos los temas se han ido desarrollando paso a paso, con ejemplos, imágenes, tips o recordatorios.

**2-** aquellos que solo han **copiado rutinas** desconociendo el significado de cada instrucción: Cada línea se explica en español y se presenta gran cantidad de ejemplos con las mismas instrucciones para comprender acabadamente su uso.

**3-** a los que **recorren foros** en busca de una rutina que luego, sin muchas aclaraciones, no les es posible adaptar a sus libros: Los ejemplos se presentan con imágenes, tips y recordatorios *o aclaraciones* para poder adaptar las rutinas a otras situaciones.

**4-** los que **ya han programado en VBA**: Nuevos estilos en la programación, más el agregado de la nueva programación para versiones 2007, 2010 Y 2013.

Con ejercicios al final de cada capítulo. Y su resolución al final del manual.

**NOTA:** Las rutinas contenidas en este manual, fueron desarrolladas y probadas en las siguientes versiones: Windows XP con Office 2007, Windows7 con Office 2010 Y 2013.

**En cada tema se hace mención a las diferencias entre versiones si las hubiere.**

**Además un capítulo con todos los cambios que deben introducirse en versiones de 64 bits.**

**IMPORTANTE:** es imposible colocar ejemplos para las miles de situaciones que se pueden presentar a la hora de desarrollar un libro o aplicación en Excel. Cada macro es un ejemplo que luego se puede adaptar a otras situaciones. Incluso en muchos temas se dejaron varios ejemplos.

Se recomienda utilizar el BUSCADOR del PDF para encontrar otros ejemplos del uso de alguna expresión o función. Por ejemplo para comprender el uso de 'EntireRow' se encontraron alrededor de 12 ejemplos, para el uso de SET en alguna macro de búsqueda se obtendrán alrededor de 46 entradas

### ¿Cómo aprender a programar en Excel?

Como todo estudio, requiere del alumno un **seguimiento ordenado del manual**.

Comenzaremos con los conceptos básicos para obtener un acabado conocimiento del significado de los términos utilizados en VBA. A los que ya saben programar un pequeño repaso siempre es bienvenido.

Además, los **ejercicios** incluidos en cada capítulo serán una ayuda para decidir si continuar con el capítulo siguiente o repasar algunos conceptos antes de avanzar.

Se adjunta además un libro de Ejemplos-Ejercicios para práctica del usuario.

---

## 2 CONCEPTOS BÁSICOS

### 2.1 ¿QUÉ ES UNA MACRO?

Al hablar de Excel, no podemos dejar de mencionar las '**macros**'.

Una **macro** o '**rutina**' es un conjunto de instrucciones en lenguaje de programación, que en el caso de Excel se conoce como **VBA** (Visual Basic for Applications).

Estas instrucciones nos permiten ejecutar ciertas tareas que no se pueden realizar con las herramientas de la hoja. También nos permiten automatizar tareas que sí podemos hacerlas manualmente pero por ser repetitivas nos valemos de un código o macro.

#### *Ejemplos de algunas situaciones programables:*

- al abrir un libro, se incremente un contador.
- al abrir un libro se oculten hojas o se muestre algún formulario
- al ingresar datos en una columna, se complete el resto del registro con datos de otra tabla
- copiar datos entre distintas hojas o libros.
- al cerrar un libro, guardarlo automáticamente con otro nombre o destino.

Y así como estos ejemplos hay infinidad de tareas que pueden ser programadas.

### 2.2 ELEMENTOS DE UNA MACRO

Para crear estas rutinas o 'macros' básicamente se necesitan los siguientes **elementos**:

- Un espacio
- Un lenguaje de programación
- Una acción
- Ocasionalmente un formulario

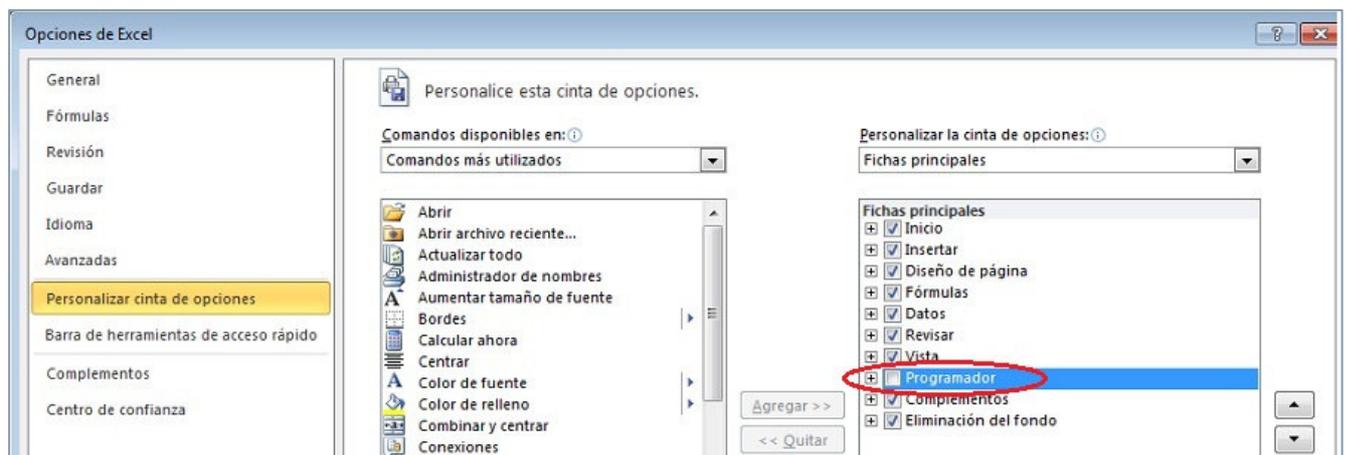
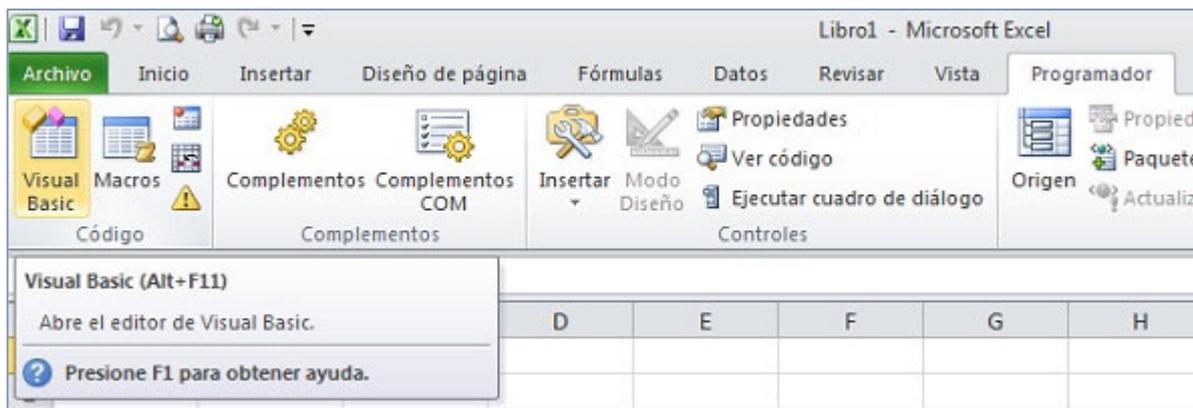
1- un **espacio** donde escribir las instrucciones o rutinas que harán esas tareas: este espacio lo encontramos dentro del **Editor de Macros** (lo veremos ampliamente en el capítulo siguiente).

Para acceder a él presionar juntas las teclas **Alt y F11**.

También se accede desde la **ficha Programador (o Desarrollador)**, botón **Visual Basic** del grupo *Código*

**Nota:** Si la ficha 'Programador' (o Desarrollador) no aparece en la cinta de opciones, presionar el **botón Archivo (Botón Office), Opciones de Excel**. En la ventana abierta, seleccionar según la imagen.

En Excel 2007 optar por '**Más frecuentes**' y tildar la opción: '**Mostrar ficha Programador en cinta de opciones**'.



Una vez dentro del **Editor de macros**, escribiremos nuestras instrucciones en distintos espacios: insertando módulos, insertando formularios personales llamados Userforms, en los objetos Hoja o en el objeto Libro. Ya veremos en capítulos siguientes cuándo elegimos uno u otro espacio.

2- un **lenguaje de programación**. En Excel utilizamos **VBA** (Visual Basic para Aplicaciones). Es un lenguaje con términos en inglés que aprenderemos a reconocer y comprender a lo largo de este manual. **Importante:** una buena costumbre es 'aclamar' lo que se programa en cada línea. Los comentarios se inician con una comilla, por ej:

'seleccionamos la primer celda del rango  
Range("A5").Select

3- una **acción** que hará que las instrucciones programadas se ejecuten. A esto llamamos '**Eventos**'. Son los disparadores de una macro. Los más habituales son: abrir o cerrar un libro, entrar o salir de una hoja, cambios en celdas, selección de celdas, antes de imprimir o guardar, al hacer 'clic' en algún botón, al presionar un atajo de teclado, y otros más.

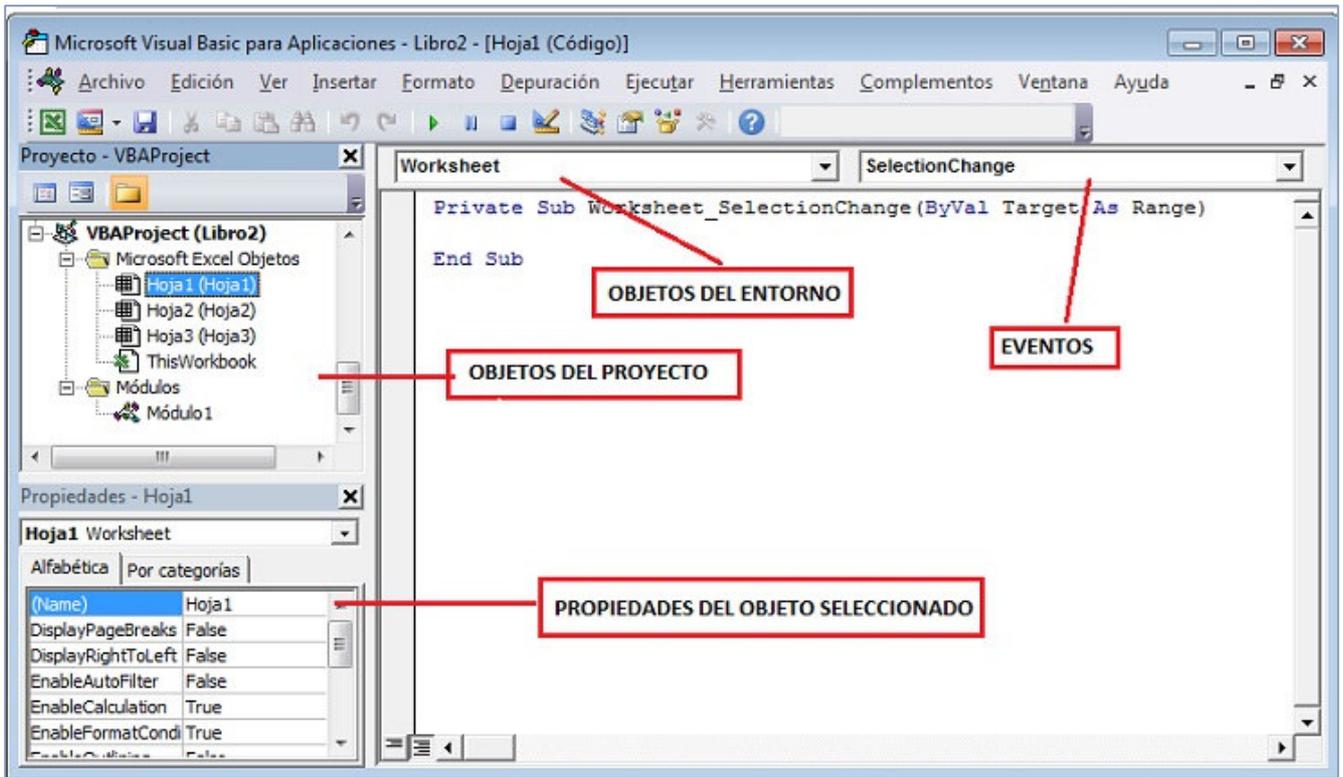
4- Ocasionalmente un **formulario** donde trabajar para presentar información de un modo diferente o para registrar datos que luego serán volcados en las hojas. Son los llamados **Userforms**.

### 3 EL EDITOR DE MACROS

Con el atajo de teclado **Alt+F11** tenemos acceso a una nueva ventana, diferente a las hojas habituales de Excel.  
Aquí en este espacio, escribiremos nuestro código, macros o rutinas, programando nuestro libro o aplicación para que se ejecuten las tareas necesarias.

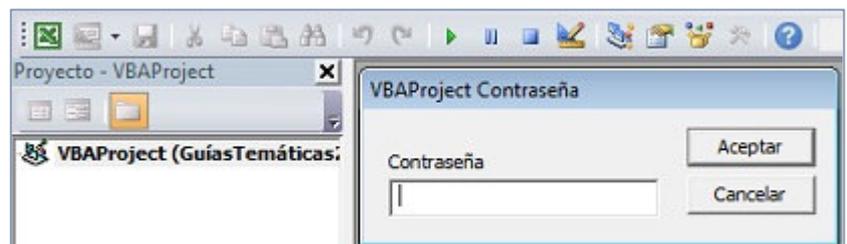
Para regresar a la hoja presionar juntas las teclas Alt y Q

Veamos qué encontramos en la ventana del Editor:



Si al entrar al Editor solo vemos el nombre del proyecto haremos doble clic sobre él para que se nos muestren los objetos del mismo.

Si el Editor ha sido *protegido con clave*, se nos pedirá su ingreso para poder ver y trabajar en los distintos espacios, ya sea módulos, hojas o libro.



#### 3.1 ELEMENTOS DEL EDITOR

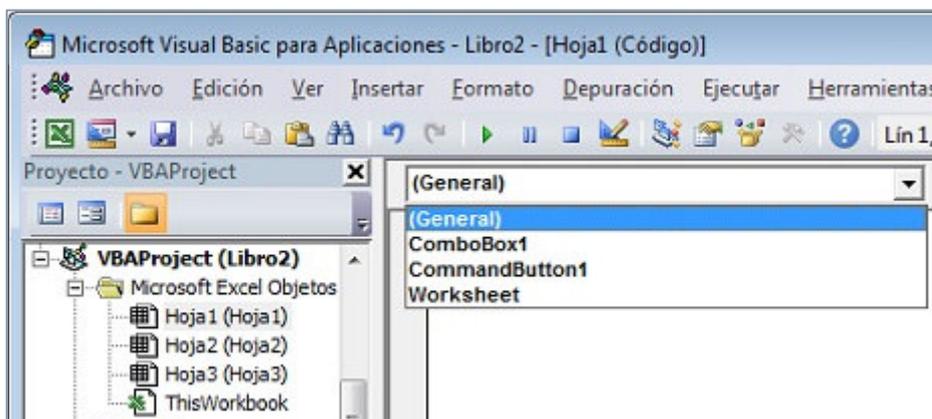
En la imagen anterior podemos observar los siguientes elementos en la ventana del Editor:

**-Lista de Proyectos:** En el panel de la izquierda. Cada libro es un proyecto y contiene los objetos Hojas, Libro (ThisWorkbook), Formularios (opcional) y Módulos (opcional).

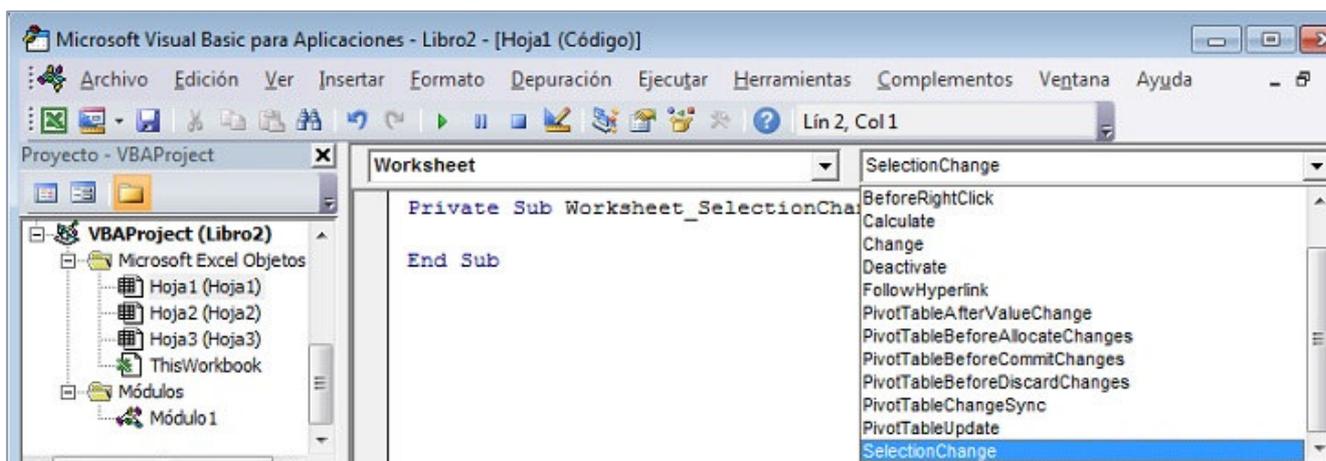
-**Propiedades del objeto seleccionado:** en el mismo panel a la izquierda. Como el objeto seleccionado es la Hoja1 allí aparece la lista de sus propiedades (ya las veremos más adelante)

-**Objetos del entorno:** este desplegable nos permite seleccionar cada objeto que tiene el objeto seleccionado, en este caso la Hoja1, siendo la hoja el objeto **Worksheet**.

Si hubiese *controles ActiveX* dibujados en la hoja aparecerán en la lista también.



-**Eventos:** son las acciones que se programarán por cada objeto. En la imagen siguiente se está tratando de programar el evento `Selection_Change`, es decir una rutina para cuando se seleccione alguna celda de la hoja



La ventana del Editor además cuenta con su propia barra de herramientas que iremos viendo a lo largo de este manual.

### 3.2 CÓMO CREAR UNA MACRO

Hay 3 maneras de *crear o agregar una macro a un libro*:

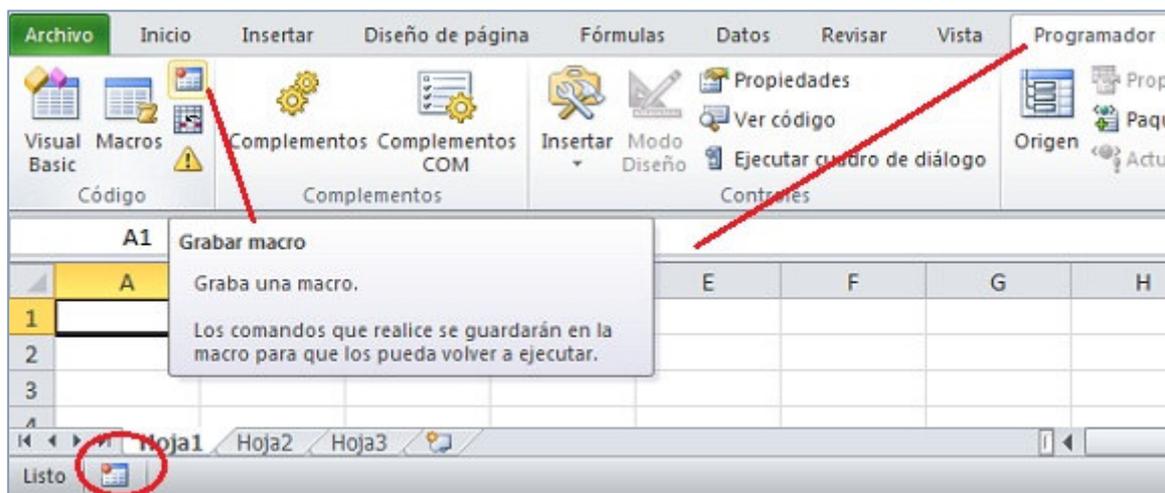
a) *Utilizando la Grabadora de Macros.*

- b) *Escribiendo (o copiando) el código en el Editor.*
- c) *Importando un módulo con una o más rutinas.*

### 3.2.1 LA GRABADORA DE MACROS

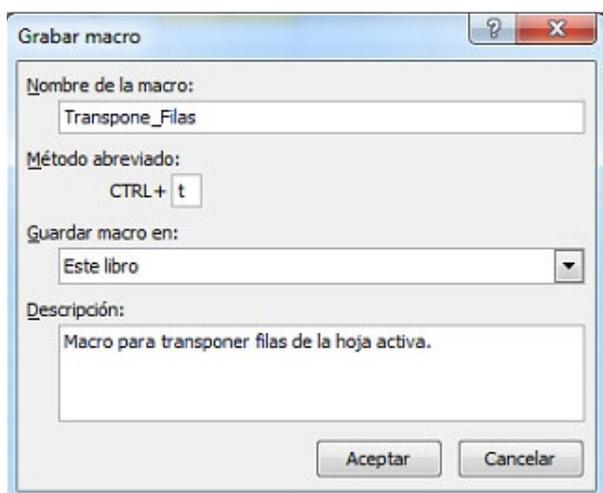
La **grabadora de macros** es la herramienta que nos permite obtener rápidamente el código de las acciones que vayamos realizando sobre la hoja o el libro.

Para utilizarla se puede presionar el botón que se encuentra al pie, en la **barra de Estado**, o accediendo a la **ficha Programador, grupo Código**



El mismo botón se utiliza tanto para activar la grabadora como para detenerla.

Al activar la grabadora se nos presenta un cuadro de diálogo para introducir los siguientes datos:



**Nombre de macro:** utilizar nombres que nos ayuden a recordar qué tarea realiza la macro.

**Método abreviado:** *opcional.* Ingresar una tecla o mayúscula tecla (\* Ver tema **Atajo de Teclado**).

**Guardar macro en:** Si optamos por *Este libro* será de ejecución en el libro activo, si optamos por *Libro Personal* será de ejecución en todos los libros Excel.

**Descripción:** es recomendable ingresar algún detalle de lo que realiza la macro.

A continuación ejecutaremos las acciones necesarias para que la grabadora las registre. Una vez finalizadas *detendremos la grabadora*. Ingresamos al Editor (con Alt y F11) y allí veremos que se ha *insertado un módulo* donde quedaron registradas las instrucciones de lo que hemos hecho.

### 3.2.2 COPIAR O REDACTAR MACROS

Podemos redactar una rutina directamente en el espacio correspondiente, ya sea un módulo, en alguna hoja o dentro del formulario.

Este tema lo veremos en detalle en el capítulo siguiente.

### 3.2.3 IMPORTAR MACROS

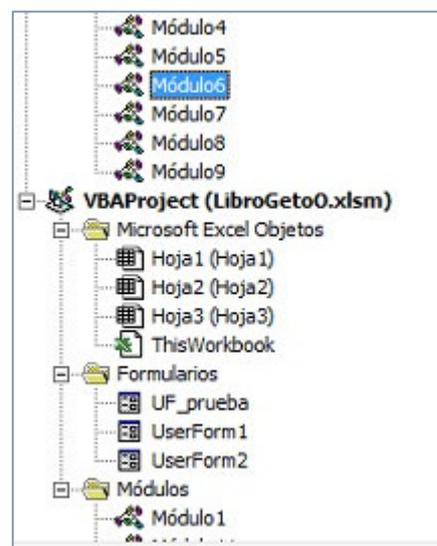
En ocasiones ya tenemos alguna función, rutina o formulario del tipo Userform en otro libro y queremos agregarla a nuestro libro actual.

Para ello tendremos que **'exportar'** el módulo o formulario seleccionándolo y buscando la opción desde el **menú Archivo**. El archivo guardado tendrá la extensión **.bas** si se trata de un módulo (por ej: Módulo1.bas) o **.frm** si se trata de un Userform.

Luego, desde nuestro libro actual y desde el mismo **menú Archivo** optar por **ImportarArchivo**, buscar la ruta y el archivo y Aceptar.

**Nota:** si el libro ya cuenta con un módulo de igual nombre, el agregado se llamará agregando un dígito más al nombre. Por ej: si ya cuenta con un objeto llamado **Módulo1** el importado pasará a llamarse **Módulo11**.

En cambio no podrán importarse Userforms si nuestro libro ya cuenta con un objeto del mismo nombre en el grupo Formularios. Habrá que cambiarle previamente el nombre.



Un modo rápido de pasar módulos de un libro a otro es tenerlos abiertos y desde el Editor, tomar y arrastrar el objeto soltándolo en el proyecto del libro actual. En la imagen se pasará el **Módulo6** al proyecto que se encuentra más abajo.

## 3.3 DÓNDE COLOCAR LAS MACROS?

Para que una rutina sea efectiva, habrá que colocarla en el espacio correcto.

Uno de los motivos más frecuentes del fracaso al tratar de ejecutar una macro, es que no se la copió en el lugar adecuado dentro del Editor.

Básicamente las reglas son las siguientes dependiendo de lo que deba ejecutarse:

- ✚ Si las instrucciones se deben ejecutar, por ejemplo, al *abrir o cerrar un libro*, se colocan en el **objeto ThisWorkbook** (o EsteLibro según la versión). Aquí entran también las rutinas que se ejecutan antes de imprimir o antes de guardar, entre otras.
- ✚ Si las instrucciones deben ejecutarse al *entrar o salir de una hoja* o al *seleccionar o modificar una celda*, se colocan en el **objeto Hoja**. Seleccionar con doble clic la **hoja** donde se va a aplicar la rutina y copiarla en la ventana de código.  
Aquí entran también las rutinas que se ejecutan al clic derecho sobre una celda, al doble clic o al seguir un hipervínculo, entre otras.

- 
- Si queremos ejecutar una rutina *desde un botón* -en una hoja o en una barra personal- o con un *atajo de teclado*, se colocará en un **módulo** que previamente se habrá insertado desde el menú **Insertar** (de la barra de herramientas del Editor)

**Nota:** *las macros de botones o controles dibujados con la barra ActiveX se colocan en la hoja donde se los ha dibujado.*

Un módulo puede contener más de un procedimiento o subrutina

- Las rutinas que se ejecutarán al clic en objetos, como imágenes, autoformas o gráficos se colocan en un módulo que previamente se habrá insertado desde el menú **Insertar** (de la barra de herramientas del Editor)
- Las instrucciones para los controles de un formulario o **Userforms** se colocan en la sección Código del mismo formulario.

---

## 4 CÓMO EJECUTAR MACROS

Ya hemos mencionado que se necesita de una '**acción**' o **Evento** que dispare o ejecute una macro guardada en algún espacio del Editor. Veremos a continuación los distintos tipos de rutinas que puede contener nuestro libro.

### 4.1 LA EJECUCIÓN SEGÚN EL TIPO DE MACROS

### 4.2 CREAR GRUPO DE BOTONES PERSONALES EN CINTA DE OPCIONES

### 4.3 PRIVADAS O PÚBLICAS

### 4.4 LAS MACROS AUTO\_OPEN Y AUTO\_CLOSE

### 4.5 ASIGNAR MACRO A UN BOTÓN (MEDIANTE CÓDIGO).

### 4.6 EJERCICIOS

---

## 5 EVENTOS, MÉTODOS Y PROCEDIMIENTOS

Cuando mencionamos en el capítulo 2 los elementos de una macro, hacíamos referencia a un espacio donde colocar el código (el Editor de macros) y a una acción-evento (que dispare la ejecución de las mismas).

### 5.1 LOS EVENTOS

---

#### 5.1.1 EVENTOS DE LIBROS

---

#### 5.1.2 EVENTOS DE HOJAS

---

#### 5.1.3 EVENTOS DE OBJETOS

---

#### 5.1.4 EVENTOS DE LA APLICACIÓN

### 5.2 MÉTODOS Y PROPIEDADES

---

#### 5.2.1 MÉTODOS DE LIBROS

---

#### 5.2.2 PROPIEDADES DE LIBROS

---

#### 5.2.3 MÉTODOS DE HOJAS

---

#### 5.2.4 PROPIEDADES DE HOJAS

---

#### 5.2.5 MÉTODOS DE RANGE O CELDAS

---

#### 5.2.6 PROPIEDADES DE RANGE O CELDAS

---

#### 5.2.7 MÉTODOS DE LA APLICACIÓN

---

#### 5.2.8 PROPIEDADES DE LA APLICACIÓN

---

## 6 REFERENCIAS A OBJETOS EXCEL

Antes de avanzar con la ejecución de macros o rutinas, debemos saber cómo hacer referencia a los objetos de Excel en código VBA.

A lo largo del manual veremos que es posible utilizar distintas sintaxis para hacer referencia a objetos como hojas, rangos o celdas.

### 6.1 HACIENDO REFERENCIA A LIBROS

### 6.2 HACIENDO REFERENCIA A HOJAS

### 6.3 HACIENDO REFERENCIA A RANGOS Y CELDAS

---

#### 6.3.1 *RANGE O CELLS*

---

#### 6.3.2 *SELECTION*

---

#### 6.3.3 *TARGET*

### 6.4 NOMBRANDO OBJETOS

### 6.5 EJERCICIOS

---

## 7 PROBANDO Y CONTROLANDO LA EJECUCIÓN DE MACROS

7.1 CÓMO PROBAR UNA MACRO

7.2 CONTROLAR UNA MACRO

7.3 CONOCER EL VALOR QUE TOMAN LAS VARIABLES

7.4 CÓMO EVITAR QUE UNA INSTRUCCIÓN SE EJECUTE

7.5 ACCEDER A LA AYUDA DESDE UNA LÍNEA DE CÓDIGO

7.6 SALIR DE UNA RUTINA

7.7 INTERRUPIR LA EJECUCIÓN DE UNA MACRO

7.8 EJERCICIOS

---

## 8 SEGURIDAD EN EL PROYECTO

### 8.1 CÓMO PROTEGER UN PROYECTO

### 8.2 EVITAR QUE LAS MACROS SE VEAN DESDE EL MENÚ

### 8.3 HABILITAR O NO LAS MACROS

### 8.4 IMPEDIR QUE SE DETENGA LA EJECUCIÓN DE MACROS

---

## 9 TRATAMIENTO DE VARIABLES

### 9.1 TIPO DE VARIABLES

### 9.2 DURACIÓN DE LAS VARIABLES

### 9.3 DETERMINAR EL TIPO DE VARIABLE

### 9.4 LIMPIANDO VARIABLES

### 9.5 FUNCIONES DE CONVERSIÓN DE TIPOS

### 9.6 OTRAS FUNCIONES DE CONVERSIÓN

### 9.7 EJERCICIOS

## 10 TRABAJANDO CON CADENAS Y MATRICES

Llamamos *'cadena'* al texto contenido en alguna celda o variable. Podemos obtener solo parte de este contenido o modificarlo, utilizando algunas funciones especiales para ello.

### 10.1 OBTENER EL LARGO DE UNA CADENA

Para saber la cantidad de caracteres que ocupa una cadena o su *'largo'* utilizamos la función **LEN**

**Ejemplo1:**

```
Sub cantCaracteres()  
Dim largo as integer  
largo= Len(ActiveCell)  
End Sub
```

**Ejemplo 2:** devolver en una celda el largo del contenido de otra. Esto reemplazaría a la función LARGO.

```
Sub largoCadena()  
Dim canti as integer  
canti = Len([A5])  
'devolvemos en una celda el total de caracteres  
[B5] = canti  
End Sub
```

**Ejemplo 3:** devolver en un mensaje el total de caracteres de un rango que previamente se habrá seleccionado.

```
Sub CuentaCaract()  
Dim celda As Range  
Dim I As Integer  
'se recorre cada celda del rango seleccionado  
For Each celda In Selection  
    I = I + Len(celda.Value)  
Next  
MsgBox "Se encontraron " & I & "caracteres y espacios en la selección"  
End Sub
```

**Nota:** el tema **For Each...Next** se halla explicado en el cap 12: **Bucles-Comandos especiales.**

### 10.2 EXTRAER PARTES DE UNA CADENA

Para **extraer partes de una cadena**, ya sea el contenido de una celda, una variable o el resultado de un **InputBox**, contamos con las siguientes funciones:

**LEFT:** devuelve el contenido de una cadena comenzando desde la izquierda a partir de la posición indicada en el segundo argumento.

**RIGHT:** devuelve lo que se encuentra a la derecha de la cadena

**MID:** devuelve lo que se encuentra a partir de cierta posición y del largo establecido

### Ejemplos:

Si la celda A1 contiene el valor 'ABC567DEF' entonces:

```
Left([A1], 3) 'devuelve 'ABC'  
Right([A1].Value, 2) 'devuelve 'EF'  
Mid(Range("A1"), 3) 'devuelve 'C567DEF'
```

**Nota:** Las 3 sintaxis son válidas para hacer referencia a la celda A1.

```
Mid([A1].Value, 4, 2) 'devuelve "56"
```

En este último caso el valor devuelto queda como un texto (string). Entonces para convertirlo a número utilizaremos además la función **Val**:

```
Val(Mid([A1].Value, 4, 2) )
```

## 10.3 DETECTAR TEXTO DENTRO DE UNA CADENA

Podemos utilizar la función **INSTR** conjuntamente con **MID** para obtener una cadena a partir de cierto caracter.

**Ejemplo 1:** Se trata de obtener el apellido sabiendo que se ubica después del caracter 'espacio'

```
Sub variables()  
Dim esp As Integer  
Dim apellido As String  
Dim cadena As String  
cadena = "Juan Perez"  
'se obtiene la ubicación del espacio  
esp = Instr("Juan Perez", " ")  
'se obtiene la cadena a partir de la posición del espacio  
apellido = Mid(cadena, esp)  
'se muestra el resultado  
MsgBox apellido  
End Sub
```

**Ejemplo 2:** también se utiliza **INSTR** para determinar si un texto o cierto carácter se encuentra en una celda. En este ejemplo se desea saber si la celda A1 contiene la palabra "JUAN". Si el valor devuelto es **> 0** significará que el texto **ha sido encontrado** en la cadena.

```
Sub buscaTexto()  
Dim texto1 As String  
'primero convertimos a mayúsculas para realizar la comparación  
texto1 = UCase(Range("A1").Value)  
'si la ubicación de la palabra "JUAN" es mayor a 0 ha sido encontrado  
If Instr(texto1, "JUAN") > 0 Then  
    'instrucciones si el texto fue encontrado  
    MsgBox "Encontrado"
```

```
Else
    'instrucciones si el texto NO fue encontrado
MsgBox "No Encontrado"
End If
End Sub
```

**Nota:** como estamos convirtiendo el contenido de la celda a mayúsculas con la función UCASE, se indicará como 'encontrado' tanto si está en mayúsc o minúsculas.

## 10.4 ARMANDO CADENAS

Para **armar cadenas**, ya sea en una variable, una celda o un mensaje, utilizamos el caracter '&'

### Ejemplos:

1- Si B10 contiene el valor 'ABC', la instrucción:

```
Range("B10").value = Range("B10").value & "-" 'devuelve: 'ABC-'
```

2- Si la variable miValor contiene la cadena: 'Enero', la siguiente rutina devolverá, siguiendo el ejemplo anterior el resultado: Enero/ABC-, tal como lo mostrará el cuadro de mensaje:

```
miValor = "Enero"
miValor = miValor & "/" & Range("B10")
MsgBox miValor
```

3- Ahora mostramos en un cuadro de mensaje el contenido de la variable miValor a la que concatenaremos la fecha actual:

```
MsgBox miValor & Date() 'devolverá: 'Enero/ABC-dd/mm/aaaa o en el formato de fecha del sistema'
```

**Nota:** La rutina completa para observar los 3 ejemplos nos queda así:

```
Sub ejemplo_Cadena( )
    'partimos de un valor inicial
    Range("B10")= "ABC"
    'le agregamos el guión
    Range("B10").value = Range("B10").value & "-" 'devuelve: 'ABC-'
    miValor = "Enero"
    'agregamos a la cadena de la variable 'miValor' el contenido de la celda B10
    miValor = miValor & "/" & Range("B10")
    MsgBox miValor
    'agregamos a la variable la fecha del día
    MsgBox miValor & Date()
End Sub
```

---

4- Establecer la ruta de un archivo:

```
nbreLibro = Range("A1")
```

```
Ruta = "C:\Users\Usuario\Documents\" & nbreLibro
```

**Nota:** Siguiendo estos ejemplos se pueden armar todo tipo de cadenas.

## 10.5 INTRODUCIR CARACTERES ESPECIALES

La función **CHR()** permite introducir caracteres especiales en cuadros de mensajes o al introducir valores en celdas.

Ya hemos visto un ejemplo de salto de línea en un MsgBox en el anterior:

**Ejemplo 1:** **CHR(10)** o **CHR(13)** para introducir texto en diferentes líneas

```
MsgBox "Bienvenidos " & Chr(10) & "al curso VBA"
```

**Ejemplo 2:** para introducir un espacio de tabulación se utiliza **Chr(9)**

```
MsgBox "Este libro se llama " & Chr(13) & Chr(9) & ActiveWorkbook.Name
```

**Ejemplo 3:** para escribir en 2 líneas dentro de una celda (como presionando Alt+Enter), utilizamos **Chr(10)**

```
ActiveCell.Value = "Código de" & Chr(10) & "Proveedor"
```

**Ejemplo 4:** para introducir comillas dobles en un texto, se usa **Chr(34)**

```
MsgBox "El nombre del capítulo es " & _  
Chr(34) & "Trabajar con Cadenas" & Chr(34)
```

**Ejemplo 5:** la siguiente rutina devuelve en una columna de la Hoja 1, la lista de símbolos que se pueden obtener a partir de **Chr(33)**. Los caracteres del 1 al 32 son caracteres 'no imprimibles' Esta lista corresponde a los caracteres ASCII

### **Sub listarCaracteres()**

```
Dim l As Integer
```

```
'colocamos título a la columna A y B
```

```
Worksheets(1).Cells(1, 1).Value = "Chr #"
```

```
Worksheets(1).Cells(1, 2).Value = "Símbolo"
```

```
'recorremos el rango de caracteres Ascii
```

```
For l = 33 To 255
```

```
Worksheets(1).Cells(l - 30, 1).Value = l
```

```
Worksheets(1).Cells(l - 30, 2).Value = Chr(l)
```

---

Next  
End Sub

## 10.6 CREANDO CADENAS DE LARGO FIJO

Para crear **cadenas de largo fijo** como **asteriscos** u otro caracter podemos utilizar instrucciones con la función **STRING** como en los siguientes ejemplos:

**Ejemplo 1:** `miAster = String(10, "*")` ' devolverá en la variable 10 asteriscos

**Ejemplo 2:** `Range("B2") = String(5,35)` ' devolverá 5 numerales # . 35 corresponde al caracter ASCII

**Ejemplo 3:** `Range("C5") = Range("B5") & String(2, "*")`  
'coloca en C5 el contenido de B5 con 2 asteriscos a continuación

## 10.7 OBTENER LA PARTE NUMERICA DE UNA CADENA

Podemos utilizar la función **VAL** para obtener la parte numérica de una cadena.

**Ejemplo1:**

```
valor1 = "5401 - ABC"  
valor2 = Val(valor1) 'devolverá 5401
```

**Ejemplo2:** si la celda A3 = \$120, para obtener el valor sin el signo, haremos:

```
Sub quitaSigno()  
valor1 = Range("A3")  
If Left(valor1, 1) = "$" Then  
valor2 = Val(Mid(valor1, 2)) 'tomamos el valor de la variable a partir de la posición 2  
Else  
valor2 = Val(valor1)  
End If  
'devolvemos el nuevo valor en celda B3  
Range("B3") = valor2  
End Sub
```

**Nota:** utilizar **CDBl** en lugar de **Val** en casos de aplicaciones internacionales, donde exista la posibilidad de utilizar diferentes separadores decimales.

## 10.8 OTRAS FUNCIONES PARA CADENAS

**TRIM:** devuelve una cadena sin espacios. Solo se mantendrán los espacios entre cada palabra de la cadena.

---

**Ejemplo:** `Trim(" Hola gente ")` 'devuelve: "Hola gente"

Otras variantes son: **LTrim** (sin espacios a la izquierda) y **RTrim** (sin espacios a la derecha)

**Ejemplos:**

`LTrim(" Hola gente ")` 'devuelve: "Hola gente "

`RTrim(" Hola gente ")` 'devuelve: "Hola gente"

**UCASE:** convierte en mayúsculas el contenido de una celda o variable.

**Ejemplo:** `texto1 = UCASE(Range("A1").Value)`

**LCASE:** convierte a minúsculas el contenido de una celda o variable.

## 10.9 MATRICES

Una matriz es un conjunto de elementos que tienen el mismo tipo de datos y que están ordenados secuencialmente. Cada elemento de una matriz posee un número de índice único que lo identifica. Los cambios efectuados a un elemento de una matriz no afectan a los demás elementos.

El uso de matrices es amplio. Aquí veremos algunos ejemplos de casos más frecuentes.

### 10.9.1 OBTENER EL VALOR DE UN ELEMENTO CONOCIENDO SU NÚMERO O UBICACIÓN DENTRO DE LA MATRIZ.

Por ejemplo, si tenemos en una celda un *número de mes* y queremos obtener su *nombre*, podemos utilizar una matriz como en este ejemplo:

**Ejemplo:**

```
Sub Obtener_Mes()  
Dim Cadena()  
Cadena = Array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio")  
Nro = [B5] ' la celda B5 contiene el número de mes  
Nbre = Cadena(Nro)  
MsgBox Nbre  
End Sub
```

### 10.9.2 LLENAR UN CONTROL COMBOBOX CON LOS MESES DEL AÑO.

**Ejemplo:**

```
Sub alimentaCombobox()  
Dim mesi()  
mesi = Array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio")
```

```

For i = 0 To 5
  ComboBox1.AddItem mesi(i)
Next
End Sub

```

**Nota:** Los elementos de una matriz se numeran a partir de 0, por tal razón en el 1er ejemplo se obtendrá como resultado el mes de Mayo y en el 2do ejemplo el bucle va de 0 a 5 para incluir los 6 meses.

### 10.9.3 DEFINIR LONGITUD DE LA MATRIZ

Se puede declarar al inicio la cantidad de elementos que tendrá una matriz.

Por ejemplo, si consideramos que la matriz empieza en 0 la siguiente instrucción nos mostrará los 12 meses del año:

**Dim meses (11)**

Para modificar el límite inferior de una matriz se indicará en su definición. Por ejemplo en la siguiente instrucción se indica que la matriz podrá tomar valores que van de 1 a 5 inclusive.

**Dim gastos(1 To 5)**

#### REDEFINIR

Otro uso interesante de una matriz o *Array*, es el de guardar o acumular datos en lugar de tener una variable para cada uno.

Por ejemplo, si necesitamos sumar los valores diarios para luego mostrar un resumen de una tabla como la de la imagen, nos resultará de utilidad contar con una matriz.

	A	B	C	D	E
1	DIA	IMPORTE			
2	1	10			
3	2	12			
4	3	14			
5	4	16			
6	1	18		4	38
7	2	20		5	56
8	4	22			
9	5	24			
10	3	26			
11	2	28			
12	1	30			
13	5	32			

Para desarrollar esta rutina necesitaremos conocer el uso de bucles , que se desarrolla en el cap. 12.  
 Por tal motivo se deja como ejercicio en ese capítulo.

## 10.10 EJERCICIOS

---

Desarrollar una rutina que cumpla con estas solicitudes:

1-Extraer en una variable el mes de la celda A5 cuyo texto es: *Resumen del mes: Enero*

2-Convertir a mayúsculas el valor obtenido en el pto 1

3-Devolver en otra variable el largo de la cadena obtenida en el pto 1

4-Definir las variables según su tipo (Dim.....)

5-Si la celda B3 contiene este texto: ATENCIÓN, convertirla a este texto: atención \*\*\*

\*\* **ver** [solución](#)

---

## 11 CONTROLANDO ERRORES

Ya hemos visto cómo crear una macro, dónde colocarlas, cómo se hace referencia a los objetos Excel (libros, hojas, celdas y objetos en general), algunos eventos, métodos y propiedades.

Ahora ya estamos casi listos para empezar a programar.

### 11.1 ON ERROR RESUME NEXT

### 11.2 ON ERROR GOTO *LLAMADA*

### 11.3 ON ERROR GOTO 0

### 11.4 ERR

### 11.5 INHABILITAR EVENTOS QUE PROVOQUEN ERROR

### 11.6 EJERCICIOS

---

## 12 BUCLES – OTROS COMANDOS ESPECIALES

En Visual Basic para Excel, contamos con algunas **instrucciones especiales**, que nos permiten realizar acciones generalmente **repetitivas**, llamadas **bucles**.

**Por ejemplo:** recorrer un rango hasta llegar a una celda vacía, ejecutar instrucciones hasta llegar a un valor, ejecutar acciones en cada hoja del libro y tantas más.

En este capítulo se dará una idea de los distintos tipos de Bucles. Luego encontraremos ejemplos a lo largo de los distintos capítulos. También el **Manual de Bucles** es un compilado con más de 100 macros dedicadas a este tema.

### 12.1 FOR EACH .... NEXT

### 12.2 FOR ..... NEXT

### 12.3 WHILE .... WEND

### 12.4 DO WHILE .... LOOP

### 12.5 DO UNTIL.... LOOP

### 12.6 IF.... ELSEIF....ELSE... END IF

### 12.7 USO DE SET

### 12.8 WITH....END WITH

### 12.9 USO DE SELECT CASE

### 12.10 EJERCICIOS

---

## 13 CUADROS DE MENSAJE: MSGBOX E INPUTBOX

Las funciones **InputBox** y **MsgBox** muestran mensajes en cuadros de diálogo esperando que el usuario presione algún botón de los allí presentados o escriba algún texto en el caso de un InputBox.

### 13.1 LOS CUADROS DE MENSAJE O MSGBOX

---

#### 13.1.1 SINTAXIS Y EJEMPLOS DE MSGBOX

---

#### 13.1.2 MSGBOX COMO POPUP

### 13.2 LOS CUADROS DE DIÁLOGO O INPUTBOX

---

#### 13.2.1 SINTAXIS Y EJEMPLOS DE INPUTBOX

---

#### 13.2.2 CONTROLAR QUE EL CAMPO NO QUEDE VACÍO

---

#### 13.2.3 ESTABLECER UN VALOR PREDETERMINADO EN UN INPUTBOX

---

#### 13.2.4 INGRESAR - SELECCIONAR UN RANGO CON INPUTBOX

---

#### 13.2.5 CONTROLAR QUE EL DATO INGRESADO SEA FECHA

### 13.3 EJERCICIOS:

---

## 14 TRABAJANDO CON CELDAS Y RANGOS

Un rango es un conjunto de celdas. En general las instrucciones utilizadas en celdas son aptas también para un conjunto de ellas, es decir para un rango. Los métodos y propiedades puede ser aplicados tanto en una sola celda como en la hoja completa.

En el capítulo 5 tenemos la lista completa de Eventos, Métodos y Propiedades, veremos aquí la aplicación de algunos de ellos.

### 14.1 SELECCIÓN DE CELDAS O RANGOS

---

#### 14.1.1 SELECCIÓN DE CELDAS A CIERTA DISTANCIA DE LA ACTIVA

---

#### 14.1.2 AMPLIAR UN RANGO SELECCIONADO

---

#### 14.1.3 SELECCIÓN DE RANGO UTILIZANDO VARIABLES

---

#### 14.1.4 SELECCIONAR LA **REGIÓN** DONDE SE ENCUENTRA LA CELDA ACTIVA

---

#### 14.1.5 IDENTIFICAR EL **RANGO** OCUPADO EN LA HOJA

### 14.2 ENCONTRAR FIN DE RANGO – ÚLTIMA FILA / ÚLTIMA COLUMNA

### 14.3 DEVOLVER LA DIRECCIÓN O REFERENCIA DE UN RANGO O CELDA

### 14.4 OBTENER EL NÚMERO DE CELDAS, FILAS O COLUMNAS DE UN RANGO

### 14.5 COMBINAR – DESCOMBINAR CELDAS

### 14.6 BORRAR O LIMPIAR CELDAS

### 14.7 ELIMINAR CELDAS

---

14.8 INSERTAR CELDAS O RANGOS

14.9 ELIMINAR FILAS - COLUMNAS

14.10 INSERTAR FILAS - COLUMNAS

14.11 OCULTAR – MOSTRAR (FILAS Y COLUMNAS)

14.12 COMENTARIOS EN CELDAS

---

14.12.1 *INSERTAR – MODIFICAR - QUITAR COMENTARIOS*

---

14.12.2 *COPIAR COMENTARIOS DE OTRAS CELDAS*

14.13 FORMATO A CELDAS Y RANGOS

14.14 AUTOAJUSTAR COLUMNAS

14.15 INSERTAR NOMBRES DE RANGO

14.16 INSERTAR HIPERVÍNCULOS

14.17 TRANSPONER FILAS Y COLUMNAS

14.18 DETECTAR SI LA CELDA CONTIENE FÓRMULA

14.19 REALIZAR BÚSQUEDAS CONTINUAS – ACCIONES A TOMAR

14.20 BLOQUEAR CELDAS

---

14.21 RESALTAR CELDA ACTIVA

14.22 ORDENAR RANGOS

14.23 ORDENAR UN RANGO VARIABLE

14.24 FILTRAR RANGOS

14.25 RELLENAR SERIES (FECHA/HORA)

14.26 COPIAR UN MISMO RANGO EN VARIAS HOJAS

14.27 EXPORTAR UNA COLUMNA O RANGO COMO TXT

14.28 EJERCICIOS

---

## 15 TRABAJANDO CON HOJAS

Las instrucciones para el **manejo de Hojas**, pueden ser incluidas en cualquier punto de nuestras rutinas, ya sea que las tengamos en módulos o en ciertos eventos como por ejemplo en el evento **Open** del libro.

**Ejemplo:**

```
Private Sub Workbook_Open()  
'selecciona la hoja2 al abrir el libro  
Sheets(2).Select  
End Sub
```

### 15.1 SELECCIONAR HOJAS

---

#### 15.1.1 SELECCIONAR LA HOJA ANTERIOR O POSTERIOR

---

#### 15.1.2 SELECCIONAR TODAS LAS HOJAS DE UN LIBRO

### 15.2 OBTENER EL NOMBRE Y POSICIÓN DE LAS HOJAS

### 15.3 CAMBIAR COLOR DE PESTAÑA O ETIQUETA

### 15.4 MOVER UNA HOJA DE LUGAR

### 15.5 PROTEGER – DESPROTEGER HOJAS

### 15.6 VISTA PREVIA - IMPRESIÓN DE HOJAS

### 15.7 INSERTAR HOJAS

### 15.8 ELIMINAR HOJAS

---

#### 15.8.1 ELIMINAR HOJAS MEDIANTE CÓDIGO

---

#### 15.8.2 CONTROLAR LA ELIMINACIÓN DE HOJAS

---

15.9 COPIAR HOJA

15.10 COPIAR HOJA COMO LIBRO

15.11 EXPORTAR HOJA COMO TXT

15.12 OCULTAR – MOSTRAR HOJAS

15.13 INSERTAR IMAGEN EN UNA HOJA

15.14 ESTABLECER ÁREA VISIBLE DE UNA HOJA

15.15 ELIMINAR TODOS LOS COMENTARIOS DE UNA HOJA

15.16 CONOCER SI EL LIBRO CONTIENE TABLAS (CONTARLAS)

15.17 EJERCICIOS

---

## 16 TRABAJANDO CON DIRECTORIOS

En este capítulo se intenta dar una idea de las funciones que se utilizan para tratar con estos objetos. Utilizando el buscador del PDF se podrá encontrar otros ejemplos del uso de cada instrucción o función incluidas en este capítulo: [CurDir](#), [MkDir](#), [Dir](#), [Kill](#), [Rmdir](#), [ChDrive](#)

16.1 ESTABLECER LA RUTA DE ACCESO PREDETERMINADO

16.2 CREANDO UN NUEVO DIRECTORIO

16.3 CONFIRMAR SI UN DIRECTORIO EXISTE

16.4 SABER SI UN ARCHIVO EXISTE EN UN DIRECTORIO

16.5 ELIMINAR UN ARCHIVO DE UN DIRECTORIO

16.6 OBTENER SOLO LAS SUBCARPETAS DE UN DIRECTORIO

16.7 OBTENER TODOS LOS LIBROS DE UNA CARPETA

16.8 OBTENER TODOS LOS ELEMENTOS DE UNA CARPETA (SUBCARPETAS Y ARCHIVOS)

16.9 ELIMINAR UN DIRECTORIO DEL DISCO

16.10 ESTABLECER NUEVA UNIDAD DE DISCO

16.11 ELEGIR DIRECTORIO PARA COMPLETAR ALGÚN PROCESO

16.12 EJERCICIOS

---

## 17 TRABAJANDO CON LIBROS

Las principales acciones para con libros son: abrirlos, trabajarlos, guardarlos y cerrarlos. Ya veremos en cada tema cuáles son las opciones a la hora de realizar alguna de estas tareas.

### 17.1 ABRIR LIBROS (OPEN)

---

*17.1.1 ABRIR UN LIBRO. ABRIR LIBRO CON CLAVE*

---

*17.1.2 ABRIR LIBRO BUSCANDO LA CARPETA QUE LO CONTIENE*

---

*17.1.3 EJECUTAR MACRO AL ABRIR UN LIBRO*

---

*17.1.4 INHABILITAR LA OPCIÓN DE ACTUALIZAR VÍNCULOS AL ABRIR OTROS LIBROS*

---

*17.1.5 IMPEDIR QUE SE ABRAN OTROS LIBROS MIENTRAS ESTÉ NUESTRO LIBRO ABIERTO*

---

*17.1.6 CONOCER FECHA DE CREACIÓN DE UN LIBRO*

---

*17.1.7 SEPARAR NOMBRE DE LIBRO DE SU RUTA.*

---

*17.1.8 NO ABRIR EL LIBRO SI ESTÁ EN MODO LECTURA.*

### 17.2 ACTIVAR O SELECCIONAR UN SEGUNDO LIBRO

### 17.3 OBTENER LA RUTA DE UN LIBRO ABIERTO

### 17.4 MINIMIZAR UN LIBRO (MOSTRAR UN USERFORM)

### 17.5 OBTENER EL NOMBRE DE UN LIBRO ABIERTO

### 17.6 MODIFICAR LA FORMA DE CÁLCULO DEL LIBRO O DE UNA HOJA

---

17.7 ELIMINAR VINCULOS EXTERNOS

17.8 OBTENER PROPIEDAD FECHA Y HORA DE LIBROS GUARDADOS

17.9 OBTENER TAMAÑO DE ARCHIVOS

17.10 COPIAR UN LIBRO SIN ABRIRLO

17.11 CAMBIAR EL NOMBRE DE UN LIBRO SIN ABRIRLO

17.12 ELIMINAR NOMBRES DE RANGOS EN LIBRO

17.13 GUARDANDO LIBROS

---

17.13.1 *GUARDAR EL LIBRO ACTIVO*

---

17.13.2 *GUARDAR UN LIBRO CON CLAVE*

---

17.13.3 *GUARDAR UN LIBRO CON OTRO NOMBRE – FORMATO O RUTA*

---

17.13.4 *GUARDAR UN LIBRO COMO COPIA*

---

17.13.5 *GUARDAR UNA HOJA COMO LIBRO*

---

17.13.6 *GUARDAR LIBRO SIN MÓDULOS*

---

17.13.7 *GUARDAR LIBRO SIN CÓDIGO O MACROS*

---

17.13.8 *CONTROLAR ERRORES ANTES DE GUARDAR UN LIBRO*

---

17.13.9 *IMPEDIR QUE UN LIBRO SE GUARDE*

17.14 CERRANDO LIBROS

17.14.1 *CERRAR LIBRO – CERRAR TODOS LOS LIBROS EN USO*

17.14.2 *CERRAR UN LIBRO SIN GUARDAR LOS CAMBIOS*

17.14.3 *CERRAR UN LIBRO GUARDANDO LOS CAMBIOS SIN CONSULTAR*

17.15 EJERCICIOS

---

## 18 CONTROLANDO TECLAS Y MENSAJES DE ALERTA

Ya hemos visto que un buen programa necesita controlar los posibles errores que pudieran presentarse durante su ejecución

También es importante controlar los posibles avisos que nos envía Excel, como por ejemplo que hacer con los vínculos en la apertura de un libro, o el aviso de confirmación al eliminar alguna hoja y tantos otros..

Las **instrucciones de control** pueden ser colocadas en cualquier rutina, antes de realizar una acción como Guardar, Eliminar hoja, o Abrir libros. Excel tomará el valor predeterminado del cuadro de mensaje.

Otro tema que trataremos en este capítulo es el método **SendKeys** y cómo controlar ciertas teclas como ESC, Enter, flechas, clic derecho y algunas más.

Y para empezar algunas instrucciones para evitar ciertas acciones, como el movimiento de hojas o la selección de celdas activas

### 18.1 NO MOSTRAR AVISOS DE ALERTA

### 18.2 NO MOSTRAR LA EJECUCIÓN DE LA MACRO O EL MOVIMIENTO DE HOJAS

### 18.3 NO MOSTRAR EL CURSOR O LA CELDA ACTIVA

### 18.4 PERMANECER EN LA CELDA ACTIVA LUEGO DEL ENTER

### 18.5 EJECUTAR UNA MACRO CON DOBLE CLICK

### 18.6 CONTROLAR LAS TECLAS ENTER Y FLECHAS ABAJO\_ARRIBA

### 18.7 CONTROLAR OTRAS TECLAS MEDIANTE CÓDIGO

### 18.8 CONTROLAR LA TECLA ESC- CANCELAR A CRITERIO

### 18.9 IMPEDIR QUE SE CANCELE UNA MACRO CON ESC

### 18.10 ASIGNAR UNA MACRO A UNA TECLA ESPECIAL

---

18.11 EJECUTAR UNA MACRO AL PRESIONAR ENTER

18.12 EJERCICIOS

---

## 19 TRABAJANDO CON FILTROS

Dentro de las herramientas más importantes o de mayor uso en Excel son las de Orden y Filtro.

Las nuevas versiones de Excel presentan mejoras en estas 2 herramientas y desde aquí veremos cómo realizarlas mediante código.

### 19.1 FILTRAR UN RANGO

### 19.2 FILTRAR UN RANGO POR COLUMNA VARIABLE – CRITERIO VARIABLE

### 19.3 FILTRAR RANGO POR MÁS DE UNA COLUMNA

### 19.4 RECONOCER SI UNA TABLA TIENE APLICADOS LOS FILTROS

### 19.5 COPIAR LAS CELDAS FILTRADAS A OTRO DESTINO

### 19.6 FILTRO AVANZADO – USO DE VARIABLES PARA CRITERIO Y DESTINO

### 19.7 FILTRO AVANZADO - CRITERIOS CON VALORES DECIMALES

### 19.8 AUTOFILTROS CON CAMPOS FECHA

### 19.9 OBTENER REFERENCIAS DE CELDAS FILTRADAS

#### 19.9.1 SELECCIONAR LA PRIMER FILA DE UN FILTRO

#### 19.9.2 GUARDAR LA REFERENCIA DE LA PRIMER CELDA DE UN FILTRO (NO CRITERIO)

#### 19.9.3 INDICAR O SELECCIONAR LA ÚLTIMA FILA DE UN FILTRO

---

19.9.4 *OBTENER TODAS LAS FILAS FILTRADAS DE UN RANGO*

19.10 OBTENER EL TOTAL DE CELDAS FILTRADAS

19.11 UBICAR EN RANGO FILTRADO LA CELDA QUE CUMPLA CON CRITERIO

19.12 SUMAR UNA COLUMNA DE DATOS EN UN RANGO FILTRADO

19.13 FILTRAR POR COLORES O ICONOS

19.14 EJERCICIOS

---

## 20 IMPRESIÓN

### 20.1 ACERCA DE LA IMPRESIÓN

### 20.2 ESTABLECER ÁREA DE IMPRESIÓN

### 20.3 IMPEDIR LA IMPRESIÓN

### 20.4 ENVIAR INFORMACIÓN AL ENCABEZADO Y PIÉ

### 20.5 INSERTAR IMAGEN EN ENCABEZADOS O PIE DE PÁGINA

### 20.6 CONFIGURAR LA PÁGINA

### 20.7 LLAMAR AL CUADRO DE OPCIONES PARA IMPRIMIR

### 20.8 IMPRIMIR LISTA DE HOJAS

### 20.9 EJERCICIOS

---

## 21 TRABAJANDO CON GRÁFICOS

Para crear rutinas con gráficos siempre recomiendo utilizar la **grabadora de macros**.

Cada usuario tendrá un estilo de gráfico, con sus formatos, colores, rangos y series que no es tan fácil dejar ejemplos de macros que cubran todas las situaciones.

En este capítulo veremos algunas instrucciones clave para que luego cada uno pueda adaptarlas a las que le devuelva la grabadora de macros.

### 21.1 SELECCIONAR GRÁFICOS IGNORANDO SUS NOMBRES

### 21.2 CORTAR UN GRÁFICO DE UNA HOJA Y PEGARLO EN OTRA

### 21.3 ASIGNAR LOS DATOS DE ORIGEN DE UN GRÁFICO

### 21.4 CREAR UNA HOJA PARA UBICAR UN GRÁFICO

### 21.5 MODIFICAR DIMENSIONES DEL GRÁFICO

### 21.6 ELIMINAR UNO O TODOS LOS GRÁFICOS DE UNA HOJA

### 21.7 COLOREAR SERIES POR MACRO

### 21.8 AJUSTAR LA UBICACIÓN DE UN GRÁFICO

### 21.9 ACTUALIZAR TÍTULOS DE GRÁFICOS SEGÚN VALOR DE CELDA

### 21.10 EJERCICIOS

---

## 22 TRABAJANDO CON FÓRMULAS Y FUNCIONES

### 22.1 NOTACIÓN PARA FÓRMULAS

### 22.2 INTRODUCIR FÓRMULAS EN CELDAS

### 22.3 ARMAR FÓRMULA CON DATOS VARIABLES

### 22.4 COLOCAR RESULTADO DE FÓRMULA EN CELDAS

### 22.5 EL OPERADOR MOD

---

## 23 FUNCIONES PERSONALES - COMPLEMENTOS

Un **Complemento (Add-In)** es un archivo de extensión **.xlam** que permite agregar capacidades al entorno de programación VBA.

Estos archivos contienen **rutinas VBA**, generalmente **funciones**, y una vez instalados pueden ser utilizadas desde cualquier libro.

Primero veamos cómo **crear** un complemento y a continuación veremos cómo se **instalan**, tanto los nuestros como otros que podamos descargar .

### 23.1 CÓMO CREAR UN COMPLEMENTO

### 23.2 CREAR FUNCIONES PERSONALES

### 23.3 SINTAXIS DE FUNCIONES

### 23.4 CÓMO LLAMAR A UNA FUNCIÓN PERSONAL

### 23.5 EJEMPLO: FUNCION ENCONTRAR ULTIMO DATO

### 23.6 EJERCICIOS

---

## 24 TRABAJANDO CON EL ENTORNO EXCEL

### 24.1 COMENTARIOS SOBRE EL ASPECTO GENERAL

En la mayoría de las rutinas que se encuentran en este capítulo la idea es quitar algunos elementos al abrir el libro y volverlas a colocar antes de su cierre, para que desde otros libros no nos encontremos que nos falta tal o cual barra o elemento.

Entonces estas instrucciones se colocarán en el evento OPEN del libro con valores **False** y también en el evento BEFORECLOSE con valores en **True**.

Ambos eventos corresponden al objeto **THISWORKBOOK** (o Este libro según la versión de Excel )

---

#### 24.1.1 OCULTAR – MOSTRAR: PESTAÑAS DE LAS HOJAS

---

#### 24.1.2 OCULTAR – MOSTRAR: BARRAS DE DESPLAZAMIENTO

---

#### 24.1.3 OCULTAR – MOSTRAR: ENCABEZADOS DE FILAS Y COLUMNAS

---

#### 24.1.4 OCULTAR – MOSTRAR: PANTALLA COMPLETA

---

#### 24.1.5 OCULTAR – MOSTRAR: CINTA DE OPCIONES (RIBBON)

### 24.2 IMPEDIR EL USO DEL CLIC DERECHO

### 24.3 MOSTRAR CAJAS DE DIÁLOGO

### 24.4 MOSTRAR MENSAJES EN LA BARRA DE ESTADO

### 24.5 OCULTAR BOTONES DE MINIMIZAR Y MAXIMIZAR

### 24.6 INHABILITAR EL BOTÓN CERRAR DEL LIBRO

### 24.7 EJECUTAR MACROS DE OTROS LIBROS

---

24.8 EXPORTAR – IMPORTAR MODULOS

24.9 SUSTITUIR UN TEXTO EN TODO EL CÓDIGO DEL PROYECTO

24.10 EJERCICIOS

---

## 25 TRABAJANDO CON OTRAS APLICACIONES

En este capítulo desarrollaremos rutinas de ejemplo que relacionan a Excel con otras aplicaciones, como Word, Outlook, PowerPoint, Access y otras aplicaciones.

Se trata principalmente de dejar en claro cuáles son las referencias y cómo activar cada aplicación. Luego será tarea del usuario definir qué datos lleva y trae de cada una de ellas.

### 25.1 TRABAJANDO CON WORD

---

#### 25.1.1 COPIAR CELDAS EXCEL A UN DOCUMENTO WORD

---

#### 25.1.2 IMPRIMIR UN DOCUMENTO DE WORD DESDE EXCEL

---

#### 25.1.3 ABRIR UN WORD DESDE EXCEL Y GUARDARLO CON NOMBRE VALOR DE CELDA

### 25.2 TRABAJANDO CON OUTLOOK

---

#### 25.2.1 IMPORTAR DATOS DE OUTLOOK

---

#### 25.2.2 ENVÍO DE CORREOS.

### 25.3 TRABAJANDO CON POWERPOINT

---

#### 25.3.1 CREAR UN PPT CON DATOS DE CELDAS EXCEL

---

#### 25.3.2 AGREGAR DATOS A UN PPT YA CREADO

---

#### 25.3.3 COPIAR TABLA DE DATOS + GRÁFICO

---

#### 25.3.4 AJUSTAR UBICACIÓN DE LOS OBJETOS PEGADOS EN LA DIAPOSITIVA

---

25.3.5 *COPIAR DE EXCEL A UNA DIAPOSITIVA DETERMINADA*

25.4 **TRABAJANDO CON ACCESS**

25.4.1 *IMPORTAR DATOS DE ACCESS*

25.4.2 *CAMBIAR EL NOMBRE A UNA CONEXIÓN EXISTENTE*

25.4.3 *AGREGAR OTRAS CONSULTAS A LA CONEXIÓN EXISTENTE*

25.4.4 *ELIMINAR CONEXIONES*

25.4.5 *ACTUALIZAR CONEXIONES*

25.5 **ACTIVANDO OTRAS APLICACIONES**

25.5.1 *LLAMANDO AL BLOC DE NOTAS*

25.5.2 *LLAMANDO A LA CALCULADORA*

25.5.3 *ACTIVAR SONIDOS*

25.5.4 *ACCEDER AL PORTAPAPELES*

25.6 **TRABAJANDO CON PDF**

25.6.1 *EXPORTAR COMO ARCHIVO PDF*

25.6.2 *LLAMAR A UN PDF*

---

## 25.7 EJERCICIOS

---

## 26 CORREOS DESDE EXCEL – OUTLOOK, GMAIL Y OTROS

**NOTA:** Para las macros de este capítulo trabajamos con 2 hojas modelo conteniendo un formulario tipo en cada una (se adjunta al manual): Formato y CuotaFija. Y en otra hoja (Listas) se encuentra una tabla con registros de personas con o sin mail. Y otra hoja (Listas) de base de clientes con una columna conteniendo su e-mail.

Algunos ejemplos fueron desarrollados para envíos desde la opción ‘Enviar Correo’ del menú de Excel y otros para ser enviados a través de otros correos como Outlook o Gmail.

### 26.1 ENVIAR UN CORREO (SOLO TEXTO) DESDE EXCEL

### 26.2 ENVIAR UN CORREO CON VARIAS FILAS DE TEXTO EN EL CUERPO DEL MENSAJE

### 26.3 ENVIAR UN CORREO CON ARCHIVO ADJUNTO

### 26.4 ENVIAR CORREO A LISTA DE DESTINATARIOS

### 26.5 ENVIAR UNA HOJA EXCEL POR CORREO

### 26.6 ENVIAR UN RANGO DE CELDAS COMO LIBRO ADJUNTO

### 26.7 ENVÍO DE VARIAS HOJAS COMO LIBRO ADJUNTO

### 26.8 ENVIO DEL LIBRO ACTIVO

### 26.9 ENVIO DE UN LIBRO DISTINTO AL ACTIVO

### 26.10 ENVIAR RANGO DE HOJA COMO CUERPO DEL MENSAJE

### 26.11 ENVIAR CORREO GMAIL

---

## 27 TRABAJANDO CON TABLAS DINÁMICAS

27.1 COMENTARIOS SOBRE TABLAS DINÁMICAS

27.2 EVITAR LA ACTUALIZACIÓN AUTOMÁTICA DE TD

27.3 CREAR UNA TABLA DINÁMICA CON DATOS VARIABLES

27.4 CONVERTIR UN CAMPO EN UN CAMPO DE FILAS O COLUMNAS

27.5 ELIMINAR TODAS LAS TABLAS DINAMICAS DE UNA HOJA

27.6 USO DE VARIABLES EN TD

27.7 AGREGAR FILTRO A TABLA DINÁMICA

27.8 QUITAR FILTROS A UNA TABLA DINÁMICA

27.9 ACTUALIZAR TABLA AL CAMBIO DE UNA CELDA

27.10 EJERCICIOS

---

## 28 USO DE TEMPORIZADOR

### 28.1 CARACTERÍSTICAS DEL MÉTODO ONTIME

### 28.2 GUARDAR EL LIBRO CADA MINUTO

### 28.3 MOSTRAR UN USERFORM DURANTE UN TIEMPO Y OCULTARLO

### 28.4 LLLAMAR A UN USERFORM POR UNOS SEGUNDOS

### 28.5 EJERCICIOS

---

## 29 PRINCIPALES MENSAJES DE ERROR – SU SIGNIFICADO Y CÓMO RESOLVERLOS

En este capítulo trataremos de reconocer (y resolver) los mensajes más frecuentes que pueden presentarse ejecutando una aplicación con macros.

29.1 NO SE PUEDE CARGAR UN OBJETO.....

29.2 NO SE ENCUENTRA LA BIBLIOTECA

29.3 NO SE HA DEFINIDO LA VARIABLE.

29.4 EL ACCESO AL PROYECTO NO ES DE CONFIANZA

29.5 NO SE HA DEFINIDO SUB O FUNCIÓN

29.6 SUBÍNDICE FUERA DE INTERVALO

29.7 EL OBJETO NO ADMITE ESTA PROPIEDAD O MÉTODO

29.8 NO SE ENCONTRÓ EL MÉTODO O EL DATO MIEMBRO

29.9 DESBORDAMIENTO

29.10 NO COINCIDEN LOS TIPOS

29.11 FALTA DE PERFIL AL ENVIAR CORREO

---

## 30 TRABAJANDO CON OFFICE DE 64 BITS

Fuente consultada: <https://msdn.microsoft.com/es-es/library/office/gg264421.aspx#sectionSection0>

### 30.1 COMENTARIOS SOBRE LOS DIFERENTES ENTORNOS

### 30.2 ESCRITURA DE CÓDIGO QUE FUNCIONA EN AMBAS VERSIONES DE 32 BITS Y 64 BITS DE OFFICE

### 30.3 ESCRITURA DE CÓDIGO QUE FUNCIONA EN MICROSOFT OFFICE 2010 (32 BITS O 64 BITS) Y VERSIONES ANTERIORES DE OFFICE

### 30.4 RESUMEN DE ACTUALIZACIONES DE IDIOMA DE VBA7

---

## 31 RESOLUCIÓN DE EJERCICIOS:

31.1 EJERCICIOS CAPÍTULO 6

31.2 EJERCICIOS CAPÍTULO 7

31.3 EJERCICIOS CAPÍTULO 9

31.4 EJERCICIOS CAPÍTULO 10

31.5 EJERCICIOS CAPÍTULO 11

31.6 EJERCICIOS CAPÍTULO 12

31.7 EJERCICIOS CAPÍTULO 13

31.8 EJERCICIOS CAPÍTULO 14

31.9 EJERCICIOS CAPÍTULO 15

31.10 EJERCICIOS CAPÍTULO 16

31.11 EJERCICIOS CAPÍTULO 17

31.12 EJERCICIOS CAPÍTULO 18

31.13 EJERCICIOS CAPÍTULO 19

31.14 EJERCICIOS CAPÍTULO 20

---

31.15 EJERCICIOS CAPÍTULO 21

31.16 EJERCICIOS CAPÍTULO 23

31.17 EJERCICIOS CAPÍTULO 24

31.18 EJERCICIOS CAPÍTULO 25

31.19 EJERCICIOS CAPÍTULO 27

31.20 EJERCICIOS CAPÍTULO 28

---

----- **FIN DEL MANUAL** -----