

Generación de macros para la automatización de tareas en procesos empresariales

Breve descripción:

Este componente formativo brinda al aprendiz el conocimiento necesario para automatizar procesos en Excel usando macros y VBA, con foco en escenarios reales de analítica de datos y reporting. Asimismo, se busca desarrollar habilidades que le permita importar datos de múltiples fuentes, transformarlos para análisis, cargarlos en estructuras fiables (tablas/modelos) y automatizar todo el flujo de trabajo con un clic.

Tabla de contenido

| | |
|-------------------------------------|----|
| Introducción | 4 |
| 1. Transformar datos | 5 |
| 1.1. Tipos | 6 |
| 1.2. Importación de datos..... | 14 |
| 1.3. Carga..... | 17 |
| 1.4. Automatización..... | 20 |
| 2. Tarea..... | 22 |
| 2.1. Concepto | 23 |
| 2.2. Definición de requisitos | 26 |
| 2.3. Cálculos personalizados | 28 |
| 3. Macros | 31 |
| 3.1. Creación..... | 31 |
| 3.2. Ejecución | 35 |
| 3.3. Interfaces..... | 38 |
| Síntesis | 51 |
| Material complementario..... | 52 |
| Glosario..... | 53 |
| Referencias bibliográficas | 54 |



| | |
|----------------|----|
| Créditos | 55 |
|----------------|----|

Introducción

Excel se ha consolidado como una de las herramientas más versátiles en el ámbito empresarial, académico y personal, gracias a su capacidad para procesar y organizar grandes volúmenes de información. Sin embargo, el aprovechamiento pleno de sus funcionalidades exige ir más allá de las operaciones básicas, integrando procesos avanzados que permitan optimizar la gestión de datos y generar valor en la toma de decisiones.

En este sentido, las macros y el lenguaje VBA se presentan como aliados estratégicos para la automatización de tareas repetitivas y la creación de soluciones personalizadas. A través de estas herramientas, es posible reducir errores, ahorrar tiempo y fortalecer la consistencia en los procesos, lo que contribuye a mejorar la calidad del análisis de la información y a incrementar la productividad en distintos contextos.

Este componente formativo ofrece a los aprendices la oportunidad de adentrarse en el mundo de la automatización en Excel, combinando teoría y práctica para desarrollar competencias aplicables en escenarios reales. Con ello, no solo se busca que los estudiantes dominen técnicas avanzadas, sino también que adquieran una visión crítica y analítica capaz de identificar oportunidades de mejora en los flujos de trabajo empresariales.

1. Transformar datos

La **transformación de datos** es el conjunto de operaciones necesarias para hacer que los datos sean **utilizables**: consistentes, limpios, con tipos correctos y estructura apta para análisis y visualización. Este proceso convierte datos brutos en **información útil y accionable**. Entre los pasos fundamentales se incluyen:

Limpieza de datos

Eliminación de errores, inconsistencias y duplicados.

Normalización

Estandarización de formatos y unidades.

Conversión de tipos de datos

Asegura que cada variable tenga el formato adecuado (ej. fechas como fechas, números como números).

Estructuración de datos

Adaptación del formato para que sea compatible con herramientas de análisis y visualización.

Agregación de datos

Combinación de datos provenientes de múltiples fuentes

Creación de nuevas variables derivadas

Generación de variables a partir de otras existentes.

Codificación de variables categóricas

Transformación de categorías en códigos numéricos o etiquetas.

Imputación de valores faltantes

Sustitución o estimación de datos ausentes.

Estas operaciones no solo preparan los datos para su análisis inmediato, sino que también **mejoran su calidad y confiabilidad a largo plazo**. Una transformación efectiva es crucial para garantizar la **integridad de los análisis** y para facilitar la **extracción de insights valiosos** que informen la toma de decisiones basada en datos.

Transformaciones en Excel

En el dinámico mundo de los negocios y la analítica de datos, **Excel** se posiciona como una herramienta clave y predominante. Este software actúa como un **punto entre los datos masivos y las decisiones estratégicas**, gracias a sus capacidades de transformación y automatización. En Excel, las transformaciones pueden realizarse mediante:

- Métodos manuales (filtros, fórmulas, tablas dinámicas, etc.).
- Power Query, que permite aplicar transformaciones complejas de manera visual y reproducible.
- Macros o VBA, que automatizan procesos y optimizan flujos de trabajo, ahorrando tiempo valioso.

Vamos a sumergirnos en el fascinante universo de las **macros en Excel**, una habilidad poderosa que te permitirá llevar tu productividad al siguiente nivel.

1.1. Tipos

En **Excel**, la transformación de datos implica convertir información en bruto en un formato adecuado para su **análisis, visualización o presentación**. Este proceso puede adoptar diversas formas, entre ellas:

- **Conversión de formatos:** cambiar tipos de datos, como números a texto, fechas o moneda.
- **Limpieza:** eliminar espacios innecesarios, duplicados o caracteres no válidos.
- **Agrupación y resumen:** uso de tablas dinámicas para condensar y analizar información.
- **Normalización:** ajustar escalas y unidades para permitir comparaciones consistentes.

Estas transformaciones se pueden clasificar en dos grandes categorías:

Transformaciones estructurales

Modifican la organización o forma de los datos para hacerlos más analizables o visualizables.

Transformaciones de formato

Cambian el tipo o presentación de los datos sin alterar su estructura general.

Pivotear datos

El pivoteo convierte filas en columnas para reorganizar los datos y facilitar su análisis.

Tabla 1. Datos sin pivotear

| Vendedor | Trimestre | Ventas |
|--------------|-----------|-----------|
| Ana García | Q1 | \$ 15.000 |
| Ana García | Q2 | \$ 18.500 |
| Carlos López | Q1 | \$ 12.300 |
| Carlos López | Q2 | \$ 16.800 |

| Vendedor | Trimestre | Ventas |
|-------------|-----------|-----------|
| María Silva | Q1 | \$ 14.700 |
| María Silva | Q2 | \$ 19.200 |

En esta tabla, cada fila representa una venta realizada por un vendedor en un trimestre específico. A continuación, los mismos datos se presentan en formato horizontal.

Tabla 2. Datos pivoteados

| Vendedor | Q1 | Q2 |
|--------------|-----------|-----------|
| Ana García | \$ 15.000 | \$ 18.500 |
| Carlos López | \$ 12.300 | \$ 16.800 |
| María Silva | \$ 14.700 | \$ 19.200 |

¿Qué cambió en la transformación?

A continuación, se detallan las diferencias estructurales entre los datos sin pivotar y los pivoteados.

Antes del pivoteo

- 6 filas de datos (una por cada combinación vendedor - trimestre).
- 3 columnas: **vendedor, trimestre, ventas**.
- Formato **largo** o vertical.
- Cada venta ocupa una fila completa.

Después del pivoteo

- 3 filas de datos (una por vendedor).
- 3 columnas: **vendedor, Q1, Q2**.

- Formato **ancho** u horizontal.
- Los trimestres se convirtieron en columnas.
- Se facilita la comparación entre trimestres.

Ventajas del pivoteo

El formato pivotado ofrece beneficios clave para el análisis, visualización y presentación de datos.

- **Comparación visual más clara:** se puede ver rápidamente el rendimiento de cada vendedor en diferentes trimestres.
- **Reducción de filas:** los datos ocupan menos espacio y son más compactos.
- **Mejor para gráficos:** Excel puede generar gráficos comparativos con mayor facilidad.
- **Análisis horizontal:** permite aplicar cálculos como diferencias entre trimestres o totales por fila.

¿Cuándo usar cada formato?

La elección del formato dependerá del objetivo: entrada, análisis o presentación de los datos.

Sin pivotear

Ideal para la **entrada de datos, bases de datos o filtros complejos.**

Pivoteado

Adecuado para **reportes, presentaciones o análisis comparativo.**

Normalizar y desnormalizar

Este proceso busca estructurar y organizar la información de manera eficiente y consistente. La **normalización** consiste en dividir datos complejos en **tablas más simples**, eliminando redundancias. En contraste, la **desnormalización** une tablas para facilitar el análisis directo. Estas prácticas son especialmente importantes al trabajar con **bases de datos relacionales**.

Principios básicos de la normalización

A continuación, se presentan los fundamentos que guían una estructura de datos normalizada:

- **Una fila por registro:** cada fila debe representar una única entidad o evento.
- **Una columna por atributo:** cada columna debe contener un solo tipo de información.
- **Eliminar redundancia:** evitar duplicar información innecesariamente.
- **Consistencia en formatos:** usar formatos uniformes para fechas, números y texto.

Veamos un ejemplo concreto de normalización; se parte de una tabla con información duplicada y poco estructurada.

Tabla 3. Datos sin transformar

| ID_Empleado | Nombre_Empleado | Departamento | Jefe_Departamento | Telefono_Jefe | Proyecto | Fecha_Inicio |
|-------------|-----------------|--------------|-------------------|---------------|-----------------|--------------|
| E001 | Juan Pérez | Marketing | Ana Rodríguez | 555-0101 | Campaña Digital | 15/01/2024 |
| E002 | María González | Marketing | Ana Rodríguez | 555-0101 | Rediseño Web | 1/02/2024 |

| ID_Empleado | Nombre_Empleado | Departamento | Jefe_Departamento | Telefono_Jefe | Proyecto | Fecha_Inicio |
|-------------|-----------------|--------------|-------------------|---------------|------------------|--------------|
| E003 | Carlos Silva | Ventas | Roberto Martín | 555-0102 | Expansión Norte | 20/01/2024 |
| E004 | Laura Torres | Ventas | Roberto Martín | 555-0102 | Clientes Premium | 1/03/2024 |
| E005 | Diego López | IT | Carmen Ruiz | 555-0103 | Migración Cloud | 1/02/2024 |

Esta tabla contiene **información redundante**, especialmente en los campos relacionados con el departamento y sus responsables. Ahora, observemos cómo queda tras la normalización. Se crean tres tablas separadas y relacionadas por identificadores únicos.

Tabla 4. Empleados (información básica individual)

| ID_Empleado | Nombre_Empleado | ID_Departamento |
|-------------|-----------------|-----------------|
| E001 | Juan Pérez | D01 |
| E002 | María González | D01 |
| E003 | Carlos Silva | D02 |
| E004 | Laura Torres | D02 |
| E005 | Diego López | D03 |

Tabla 5. Departamentos (información única por departamento)

| ID_Departamento | Nombre_Departamento | Jefe_Departamento | Telefono_Jefe |
|-----------------|---------------------|-------------------|---------------|
| D01 | Marketing | Ana Rodríguez | 555-0101 |
| D02 | Ventas | Roberto Martín | 555-0102 |
| D03 | IT | Carmen Ruiz | 555-0103 |

Tabla 6. Proyectos (relación entre empleados y proyectos)

| ID_Empleado | Proyecto | Fecha_Inicio |
|-------------|------------------|--------------|
| E001 | Campaña Digital | 15/01/2024 |
| E002 | Rediseño Web | 1/02/2024 |
| E003 | Expansión Norte | 20/01/2024 |
| E004 | Clientes Premium | 1/03/2024 |
| E005 | Migración Cloud | 1/02/2024 |

¿Cómo aplicar esto en Excel?

En Excel, los datos normalizados pueden gestionarse de varias maneras:

- Usar **múltiples hojas** para representar cada tabla.
- Relacionarlas con funciones como **BUSCARV** o **XLOOKUP**.
- Utilizar **Power Query** para crear relaciones y mantener consistencia.
- Crear **tablas dinámicas** para análisis a partir de datos vinculados.

¿Qué problemas existían antes de normalizar?

La tabla original presentaba varios inconvenientes:

Redundancia

La información del departamento se repite varias veces.

Inconsistencias

Si cambia el número del jefe, hay que actualizar múltiples registros.

Desperdicio de espacio

Datos duplicados ocupan más memoria.

Errores

Mayor posibilidad de equivocarse al modificar datos repetidos.

Beneficios de la normalización

Después del proceso, los datos mejoran significativamente en calidad y estructura:

- **Eliminación de redundancia:** cada dato aparece una sola vez.
- **Consistencia:** un único lugar para modificar información repetida.
- **Integridad:** relaciones claras entre entidades mediante identificadores.
- **Eficiencia:** menor uso de almacenamiento y mayor facilidad de mantenimiento.

Transponer

Esta transformación permite intercambiar filas por columnas y viceversa. En Excel, **transponer** significa convertir filas en columnas y columnas en filas. Es útil para reorganizar datos cuando el diseño original no facilita el análisis o la presentación.

Ejemplo de transposición

A continuación, se muestran los datos antes y después de aplicar la transformación.

Tabla 7. Datos originales (formato horizontal)

| Dato | Ene | Feb | Mar | Abr |
|--------|---------------|---------------|---------------|---------------|
| Ventas | \$ 100.000,00 | \$ 150.000,00 | \$ 120.000,00 | \$ 180.000,00 |
| Costos | \$ 60.000,00 | \$ 90.000,00 | \$ 70.000,00 | \$ 110.000,00 |

Tabla 8. Datos transpuestos (formato vertical)

| Mes | Ventas | Costos |
|-----|---------------|---------------|
| Ene | \$ 100.000,00 | \$ 60.000,00 |
| Feb | \$ 150.000,00 | \$ 90.000,00 |
| Mar | \$ 120.000,00 | \$ 70.000,00 |
| Abr | \$ 180.000,00 | \$ 110.000,00 |

¿Cómo transponer datos en Excel?

Existen tres métodos principales para realizar esta transformación:

Pegado especial

Copia los datos → clic derecho → Pegado especial → marcar "Transponer".

Función transponer

=TRANSPONER(A1:D3) en un rango del mismo tamaño. Requiere Ctrl+Shift+Enter.

Power Query

Usar obtener datos → permite transposición dinámica con actualización.

1.2. Importación de datos

Este proceso permite incorporar información desde diversas fuentes externas a Excel. La **importación de datos** consiste en traer información desde archivos, bases de datos o servicios web hacia una hoja de cálculo, para su posterior análisis. Excel admite múltiples formatos y fuentes.

¿Desde dónde se pueden importar datos?

A continuación, se clasifican las fuentes más comunes:

Archivos locales

CSV, TXT, XML, JSON, Excel (.xlsx/.xls), Access (.mdb/.accdb).

Bases de datos

SQL Server, MySQL, Oracle, PostgreSQL, SQLite.

Fuentes web

APIs REST, servicios web, páginas web (web scraping), archivos en la nube (OneDrive, SharePoint).

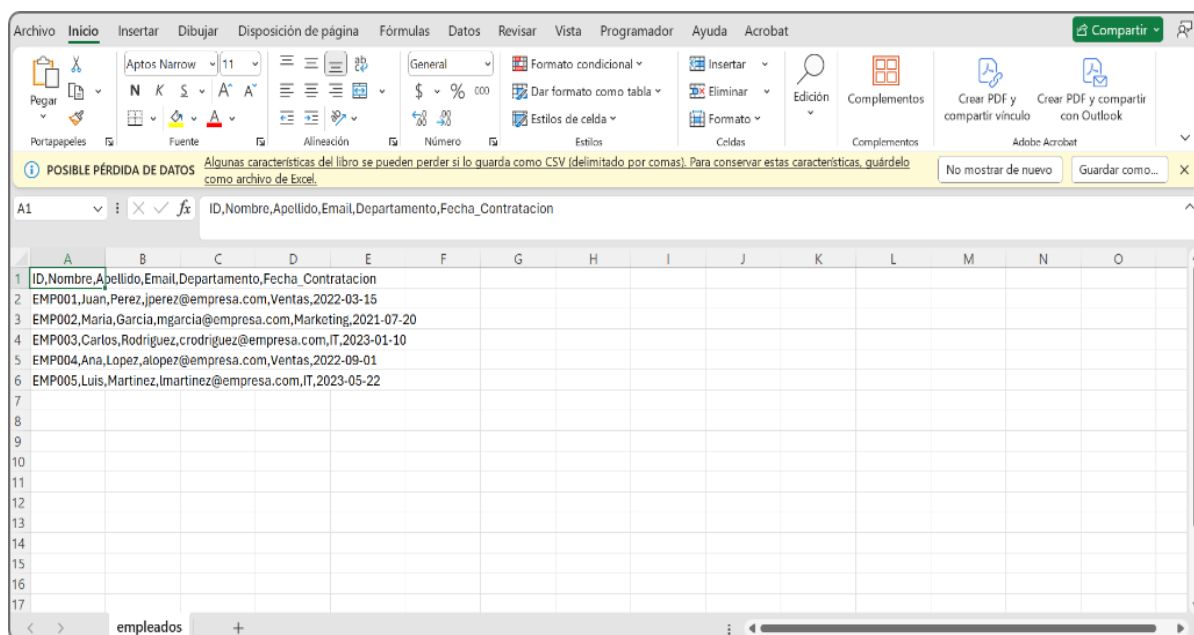
Importar un archivo CSV paso a paso

A continuación, se explica el procedimiento para importar archivos CSV a Excel. Los archivos **CSV (Comma-Separated Values)** son archivos de texto plano que almacenan datos en forma de tabla. Cada línea representa una fila, y los valores están separados por comas o punto y coma, dependiendo de la configuración regional. Este tipo de archivo tiene propiedades específicas que lo hacen ideal para el intercambio de datos:

- **Formato simple:** no contiene fórmulas, estilos ni estructuras complejas como los archivos de Excel (.xlsx).
- **Separadores:** los valores se separan por comas (,) o punto y coma (;), dependiendo de la configuración regional.
- **Portabilidad:** es ampliamente utilizado en plataformas de bases de datos y servicios web.

Después de importar el archivo CSV, el registro de empleados debe visualizarse como se presenta en la figura.

Figura 1. Registro de empleados



Excel genera una advertencia cuando se intenta abrir un archivo de este formato con un doble clic. La forma correcta es hacerlo desde la opción **datos** del menú contextual. Las instrucciones son: inicio → libro en blanco → datos → desde un archivo → desde texto/CSV. En el siguiente video se detalla el paso a paso:

Video 1. Importar archivos en Excel formato separado csv



[Enlace de reproducción del video](#)

Síntesis del video: importar archivos en Excel formato separado csv

En este video, se presenta un enfoque práctico y eficaz para importar datos desde diversas fuentes, enfocándose en archivos con formato Excel y CSV. El experto, en análisis de datos, explica las diferencias entre los formatos de archivo y cómo utilizar correctamente las herramientas de Excel para evitar errores comunes durante la importación. Su metodología se centra en la previsualización de datos y la organización eficiente de la información, asegurando que los datos sean accesibles y utilizables. Este proceso es fundamental para cualquier profesional que trabaje con grandes volúmenes de datos y busque maximizar la efectividad en su análisis.

1.3. Carga

Este paso consiste en incorporar los datos transformados en una hoja de cálculo para su análisis final. La **carga de datos** es la etapa en la que la información,

previamente importada y transformada, se inserta en Excel para su visualización y uso práctico. Para ilustrar este proceso, se utiliza un ejemplo con un **dataset en formato JSON**.

¿Qué es un archivo JSON?

Es un formato ligero y estructurado para el intercambio de datos. **JSON (javascript object notation)** es un formato de texto plano ampliamente utilizado en aplicaciones web y servicios API. Aunque tiene su origen en JavaScript, es **independiente del lenguaje** y está diseñado para ser legible tanto por humanos como por máquinas. Su estructura se basa en **pares clave-valor**, muy similar a objetos en JavaScript o diccionarios en otros lenguajes. A continuación, se presenta un fragmento de datos con distintos tipos de información:

```
{
  "nombre": "María García",
  "edad": 28,
  "activo": true,
  "hobbies": ["lectura", "natación", "fotografía"],
  "direccion": {
    "calle": "Av. Principal 123",
    "ciudad": "Madrid",
    "codigo_postal": "28001"
  },
  "telefono": null
}
```

En color rojo se han señalado las claves de este registro de ejemplo, mientras que en color verde se identifican los valores asignados a cada una de ellas.

En el siguiente video se explica cómo cargar un archivo en formato JSON en Excel. La información corresponde a empleados de una empresa e incluye datos anidados como departamentos, arreglos de habilidades y diferentes tipos de datos (números, fechas, booleanos y texto).

El dataset contiene 15 registros. Para importarlo en Excel se deben seguir los siguientes pasos:

- a) Datos > obtener datos > desde archivo > desde JSON.
- b) Seleccionar el archivo JSON.
- c) Expandir la tabla "vehículos".
- d) Cargar en la hoja de cálculo.

Puntos clave del proceso:

- Usar la función Obtener datos en la pestaña Datos.
- Expandir correctamente los datos anidados.
- Configurar los tipos de datos adecuados.
- Gestionar arrays y estructuras complejas.

En el siguiente video se explica paso a paso cómo importar un archivo en formato JSON a Excel, transformando su estructura de claves y valores en una tabla organizada que permite trabajar con los datos de manera más clara y práctica.

Video 2. Importar archivos en Excel formato JSON



[Enlace de reproducción del video](#)

Síntesis del video: importar archivos en Excel formato JSON

En este video, un experto en análisis de datos, ofrece una guía completa sobre la importación de archivos JSON a Microsoft Excel. La presentación está estructurada para facilitar la comprensión a los usuarios familiarizados con los formatos CSV, destacando las diferencias entre estos tipos de archivo. El experto explica la estructura de pares clave-valor de los archivos JSON y muestra el proceso con Excel, ilustrando la transformación de datos de su formato original a una tabla utilizable. Con ejemplos prácticos y una metodología clara, esta guía proporciona a los profesionales las habilidades esenciales para la manipulación y el análisis de datos.

1.4. Automatización

Este es el puente hacia las macros. Aunque, como se ha visto en los numerales previos, Power Query automatiza la importación de datos en diferentes formatos, hay

muchas otras tareas que necesitan automatización: **aplicar formatos condicionales complejos, crear gráficos, generar archivos PDF, mover datos entre hojas**, entre otras. La **automatización con macros** toma el control donde otras herramientas se detienen, permitiendo **orquestrar un flujo de trabajo completo con un solo comando**.

Una analogía útil

La automatización se puede presentar como **un iceberg**: lo visible son tareas simples, pero debajo hay una gran cantidad de procesos que pueden automatizarse para ganar eficiencia.

Beneficios de la automatización

Entre las ventajas que se pueden obtener, se encuentran:

Reducción de errores

Minimiza errores manuales en tareas repetitivas.

Ahorro de tiempo

Disminuye el tiempo dedicado a procesos rutinarios.

Consistencia en procesos

Asegura que las tareas se ejecuten de forma uniforme.

Mejora en la calidad de datos

Aumenta la precisión y confiabilidad de la información.

2. Tarea

No se trata solo de crear macros, sino de resolver un problema de negocio real.

Una **tarea** no es simplemente “hacer una macro”. La tarea es el **problema de negocio** que se busca resolver mediante una solución automatizada. Por ejemplo: “generar el reporte de ventas diario”.

El paso más crítico: definir los requisitos

Esto implica traducir el problema en un lenguaje técnico y estructurado. Antes de comenzar a automatizar, es fundamental **comprender con precisión qué se necesita**. Para ello, se utiliza el modelo **IPO (Input - Process - Output)**, que permite descomponer una tarea en tres componentes esenciales.

Modelo IPO (Input - Process - Output)

Este modelo ayuda a estructurar cada tarea de manera clara y lógica.

Input (Entrada)

¿Con qué datos se empieza? ¿En qué formato están? ¿Dónde se encuentran?

Process (Proceso)

¿Qué pasos lógicos y secuenciales se deben aplicar para transformar la entrada en salida?

Ejemplo:

- a) Limpiar datos.
- b) Calcular comisión.
- c) Crear tabla resume.

Output (Salida)

¿Cuál es el resultado final esperado? ¿Una tabla en una nueva hoja? ¿Un gráfico?
¿Un PDF enviado por correo?

2.1. Concepto

Una tarea es una unidad de trabajo sistemática, repetible y medible. En el contexto de Excel, una **tarea** se define como una unidad de trabajo que **transforma entradas en salidas** siguiendo reglas de negocio claramente establecidas. Esta transformación ocurre mediante un conjunto de pasos que, bajo condiciones conocidas, produce un resultado esperado sobre datos o estructuras de Excel.

¿Qué caracteriza a una tarea en Excel?

Debe cumplir ciertos criterios técnicos y de negocio.

- **Repetible:** puede ejecutarse múltiples veces sin intervención manual significativa.
- **Medible y verificable:** permite evaluar su eficacia, precisión y confiabilidad.
- **Adherencia a reglas de negocio:** sigue condiciones previamente definidas.
- **Automatizable:** puede implementarse mediante macros o VBA para mejorar la eficiencia.
- **Trazable:** genera evidencias de su ejecución (logs, resultados intermedios, versiones).

Propiedades clave de una tarea bien definida

Estas propiedades aseguran su confiabilidad y consistencia operativa.

Repetible / Idempotente

Puede ejecutarse varias veces sin alterar el resultado, salvo que se indique lo contrario.

Determinista

Las mismas entradas siempre producen las mismas salidas (excepto tareas aleatorizadas).

Observabilidad

Registro claro de entradas, filas procesadas, tiempos y mensajes de error.

Trazabilidad

Control de versiones de la tarea y sus reglas de negocio asociadas.

Tipos comunes de tarea en Excel

Las tareas pueden clasificarse según su propósito principal.

Análisis

Análisis estadístico, proyecciones financieras, evaluación de tendencias.

Procesamiento

Limpieza de datos, transformación de formatos, consolidación, generación de reportes.

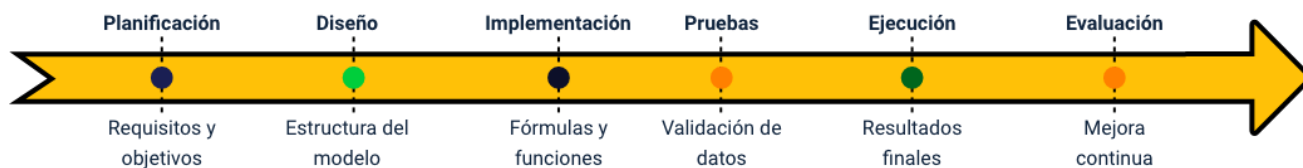
Automatización

Macros, formularios, actualizaciones automáticas, validaciones.

Ciclo de vida de una tarea

Toda tarea pasa por un conjunto de etapas que garantizan su correcta ejecución.

Figura 2. Ciclo de vida de una tarea



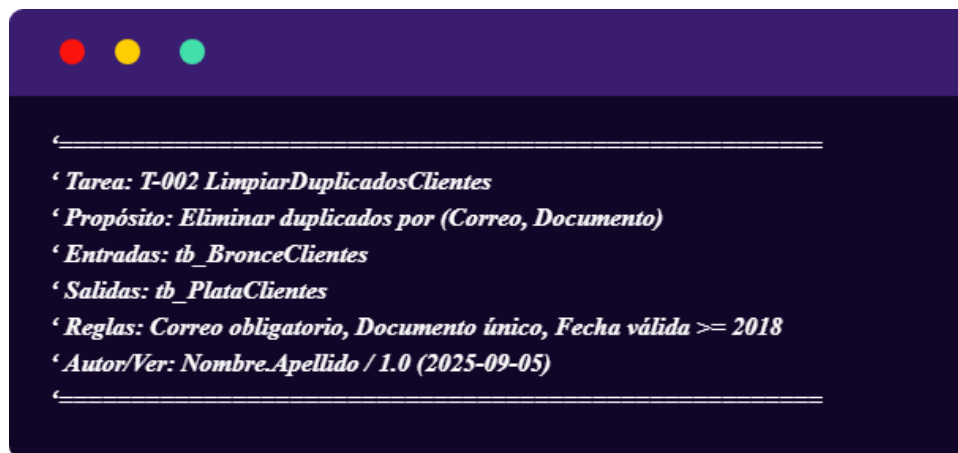
Buenas prácticas para el diseño de tareas

Estas recomendaciones ayudan a construir tareas limpias y sostenibles.

- Una responsabilidad por tarea. Si realiza muchas funciones distintas, debe dividirse en subtareas.
- Usar datos inmutables.
- Separar los parámetros del código.
- Proporcionar mensajes útiles: errores con contexto (hoja/columna/celda) y sugerencias de solución.

Ejemplo de encabezado estándar para una tarea en VBA

Este tipo de encabezado documenta los aspectos esenciales de la tarea programada.



2.2. Definición de requisitos

Consiste en convertir una necesidad en una especificación técnica clara y verificable. La **definición de requisitos** es el proceso sistemático de **identificar, documentar y especificar** todas las condiciones, restricciones y expectativas que una tarea en Excel debe cumplir para ser considerada exitosa. Su objetivo principal es traducir las **necesidades del usuario** en **especificaciones comprobables** que orienten el desarrollo en Excel/VBA.

Estructura mínima de requisitos

A continuación, se describen los componentes esenciales que deben incluirse al definir una tarea.

RF (Requisitos funcionales)

Qué debe hacer la tarea.

RNF (Requisitos no funcionales)

Requisitos de rendimiento, seguridad, usabilidad, mantenibilidad.

Datos y diccionario

Nombres de tablas o columnas, tipos de datos, dominios permitidos.

Pre/Postcondiciones y reglas

Condiciones antes y después de ejecutar la tarea, y validaciones necesarias.

Criterios de aceptación

Condiciones que deben cumplirse para que la tarea sea considerada correcta (con casos de prueba).

RACI/RASCI y riesgos (opcional)

Roles y responsabilidades en la tarea, así como riesgos potenciales.

Plantilla de especificación corta de requisitos

Este es un ejemplo de encabezado estándar para una tarea en VBA.

[ID_Tarea] T-00X | [Nombre] Preparar reporte mensual de ventas

*RF1. Importar CSV desde Parametros!B2 a hoja DatosBrutos y convertir a tb_DatosBrutos.
RF2. Limpiar espacios y caracteres no imprimibles, quitar duplicados por (Factura, Cliente).
RF3. Derivar campos: Año, Mes, ImporteNeto, IVA(19%), ImporteTotal.
RF4. Cargar salida en tb_VentasLimpia (hoja Plata) con tipos correctos.
RF5. Actualizar tabla dinámica "td_ResumenMensual" y guardar log.*

*RNF1. Tiempo total ≤ 5 s para $N \leq 50k$ filas.
RNF2. Sin intervención del usuario salvo botón "Actualizar todo".
RNF3. Código comentado y con manejo de errores.*

Datos:

tb_DatosBrutos(Fecha[date], Cliente[text], Factura[text], Importe[number]).

Validaciones:

Fecha \geq 2018-01-01

Importe \geq 0

Cliente no vacío

Criterios de aceptación:

CA1. Tras ejecutar, td_ResumenMensual refleja totales por Mes.

CA2. LOG registra fecha, filas procesadas y "OK" sin errores.

CA3. Si ruta CSV es inválida, se informa error y LOG registra "ERROR".

2.3. Cálculos personalizados

Se aplican cuando las funciones estándar de Excel no son suficientes. Los **cálculos personalizados** son operaciones matemáticas, lógicas o de manipulación de datos diseñadas para resolver necesidades específicas que **no pueden resolverse completamente con funciones nativas de Excel**.

Clasificación por nivel de complejidad

Dependiendo del tipo de cálculo, se pueden distinguir tres niveles:

Nivel 1

Combinación de funciones nativas de Excel.

Nivel 2

Funciones personalizadas creadas con VBA.

Nivel 3

Integración con servicios externos para realizar el cálculo.

Etapas para realizar un cálculo personalizado

A continuación, se describen los pasos habituales según la complejidad del cálculo.

Análisis de requerimientos

Comprensión profunda del problema.

- Definición precisa.
- Identificación de variables de entrada.
- Especificación de restricciones.
- Casos límite y validaciones.

Diseño del algoritmo

Construcción lógica de la solución.

- Pseudocódigo del proceso.

- Diagramas de flujo.
- Optimización de performance.
- Manejo de errores.

Implementación

Codificación y puesta en marcha.

- Selección de la tecnología apropiada.
- Codificación siguiendo mejores prácticas.
- Documentación inline del código.
- Pruebas unitarias.

Validación y pruebas

Verificación del funcionamiento y resultados.

- Casos de prueba representativos.
- Validación con datos reales.
- Pruebas de estrés y volumen.
- Verificación de precisión.

3. Macros

Permiten automatizar procesos repetitivos en Excel mediante código. Aunque Excel ofrece potentes funciones nativas (tablas, validación, Power Query, entre otros), la **automatización con macros** aporta **velocidad, consistencia y reducción de errores**, elementos críticos para el **Business Intelligence (BI)** y la toma de decisiones. Las **macros** son secuencias de **scripts y funciones** que automatizan tareas repetitivas en Excel. Estas instrucciones están escritas en el lenguaje **VBA (Visual Basic for Applications)**.

3.1. Creación

La creación de macros en Excel puede realizarse de dos maneras principales. Cada una ofrece diferentes niveles de control, flexibilidad y complejidad según las necesidades del usuario. Existen dos formas principales de crear macros en Excel:

- **Grabadora de macros.** Funciona como una cámara que registra cada clic y acción del teclado. Es ideal como punto de partida para procesos simples y repetitivos.
- **Editor VBA (VBE).** Permite escribir y editar código de forma manual. Ofrece control total sobre la lógica, lo que permite desarrollar soluciones más potentes y flexibles.

Grabadora de macros

Este método permite generar código automáticamente sin necesidad de conocimientos de programación. Es una excelente herramienta para usuarios que desean automatizar tareas repetitivas de forma sencilla.

Ventajas

- No requiere conocimientos de programación.
- Captura acciones exactas del usuario.
- Ideal para procesos simples y lineales.
- Generación automática de código.

Limitaciones

- El código generado no está optimizado.
- Dificulta el mantenimiento.
- No maneja lógica condicional compleja.
- Utiliza referencias absolutas por defecto.

Proceso paso a paso

- Ir a Desarrollador grabar macro.
- Asignar nombre y ubicación.
- Realizar las acciones a automatizar.
- Detener la grabación.
- Revisar y optimizar el código generado.

Editor VBA (Visual Basic for Applications)

El uso del editor VBA permite crear instrucciones personalizadas desde cero. Es ideal para usuarios avanzados que requieren soluciones flexibles, robustas y escalables. Las ventajas son:

- Control total sobre la lógica.
- Código más limpio, optimizado y eficiente.
- Capacidad para manejar excepciones y condiciones.

- Reutilización y modularidad del código.

Normalmente, las macros siguen una estructura común compuesta por distintos elementos esenciales:

- Subrutinas (Sub) y funciones (Function).
- Variables y constantes.
- Estructuras de control, como If-Then, Select Case, bucles For, Do While.
- Objetos y propiedades de Excel.
- Manejo de errores (On Error).

Se recomienda seguir buenas prácticas al momento de crear macros, tanto para mantener la claridad del código como para facilitar su mantenimiento y reutilización:

Convenciones de nomenclatura

- Usar nombres descriptivos y significativos.
- Aplicar camelCase o snake_case.
- Utilizar prefijos para identificar objetos.
- Documentar propósito y parámetros.

Estructura del código

- Declarar variables al inicio.
- Agrupar operaciones relacionadas.
- Comentar la lógica compleja.
- Implementar manejo de errores.
- Optimizar para rendimiento.

En el siguiente video se explica cómo habilitar la opción de desarrollador y, posteriormente, cómo crear y ejecutar una macro:

Video 3. Creación de MACROS



[Enlace de reproducción del video](#)

Síntesis del video: creación de MACROS

En este video, un experto en Excel y programación con Visual Basic para Aplicaciones (VBA), guía a los profesionales en el proceso de creación de macros, destacando la importancia de automatizar tareas repetitivas. El experto explica cómo habilitar la pestaña de desarrollador en Excel, activar la seguridad de macros y acceder al editor de VBA. A través de un ejemplo práctico, muestra cómo insertar un módulo, escribir un código sencillo y ejecutar la macro, enfatizando la utilidad de las macros para mejorar la eficiencia en el uso de Excel.

3.2. Ejecución

Las macros se pueden ejecutar de **tres formas principales: manual, automática y programada**. A continuación, se explica cómo realizar cada tipo de ejecución:

Ejecución manual

Esta es la forma más común y directa de ejecutar una macro. Es ideal cuando el usuario desea tener control total sobre el momento de activación.

La ejecución manual se puede realizar mediante:

- El menú **Desarrollador**.
- El atajo de teclado **Alt + F8** (abre la ventana de macros).
- El **Editor VBA**, presionando **F5**.
- **Botones personalizados** en la cinta de opciones.

En el siguiente video se explica cómo grabar una macro en Excel, utilizando una tabla de ejemplo para aplicar formatos y automatizar el proceso, de modo que pueda repetirse fácilmente con un solo comando.

Video 4. Grabación de MACROS



[Enlace de reproducción del video](#)

Síntesis del video: grabación de MACROS

En este video instruccional se proporciona una guía práctica sobre cómo grabar y ejecutar una macro en Excel, facilitando la automatización de tareas repetitivas. El experto, presenta un enfoque sistemático para aplicar formatos a tablas de datos, aprovechando las funcionalidades de Visual Basic for Applications (VBA). A lo largo del video, se destacan métodos específicos para la creación de macros, incluyendo la asignación de atajos y la aplicación de formatos de celda adecuados.

Este enfoque permite a los profesionales optimizar su flujo de trabajo en Excel, aumentando su eficiencia en la gestión de datos.

Ejecución automática

Este tipo de ejecución se activa como respuesta a eventos dentro del libro o la hoja de Excel, sin intervención directa del usuario.

La ejecución automática se puede activar mediante:

- Eventos de libro, como Open, Close, Save.
- Eventos de hoja, como Change, Activate, Calculate.
- Temporizadores usando la función OnTime.
- Combinaciones de teclas personalizadas, asignadas a macros específicas

Ejecución programada

Esta modalidad permite ejecutar macros de forma planificada, incluso sin abrir Excel directamente, lo que es útil para tareas rutinarias o de mantenimiento.

La ejecución programada puede realizarse a través de:

- El **Scheduler del sistema operativo** (como el programador de tareas de Windows).
- **Power Automate**, para flujos automatizados.
- **Aplicaciones de terceros** que permiten lanzar macros.
- **Scripts de inicio** configurados para activarse al encender el sistema o abrir archivos específicos.

3.3. Interfaces

Las interfaces en macros de Excel corresponden a los elementos visuales y a los métodos de interacción que permiten a los usuarios relacionarse con ellas de manera intuitiva y dinámica.

Tipos de interfaces para macros en Excel

Uno de los tipos más utilizados de interfaces personalizadas en VBA son los UserForms (formularios de usuario). Los UserForms son **ventanas personalizadas** que se pueden crear en el Editor de VBA para solicitar o mostrar información al usuario. Dentro de estos formularios es posible incluir diversos **controles** interactivos, tales como:

- Cuadros de texto para entrada de datos.
- Botones de comando.
- Listas desplegables.
- Casillas de verificación.
- Etiquetas informativas.

A continuación, se presenta una tabla con las etiquetas necesarias que se utilizarán como ejercicio práctico. El objetivo es crear un UserForm utilizando los datos del archivo empleados_ejemplo.csv, agregando las siguientes etiquetas en las posiciones indicadas:

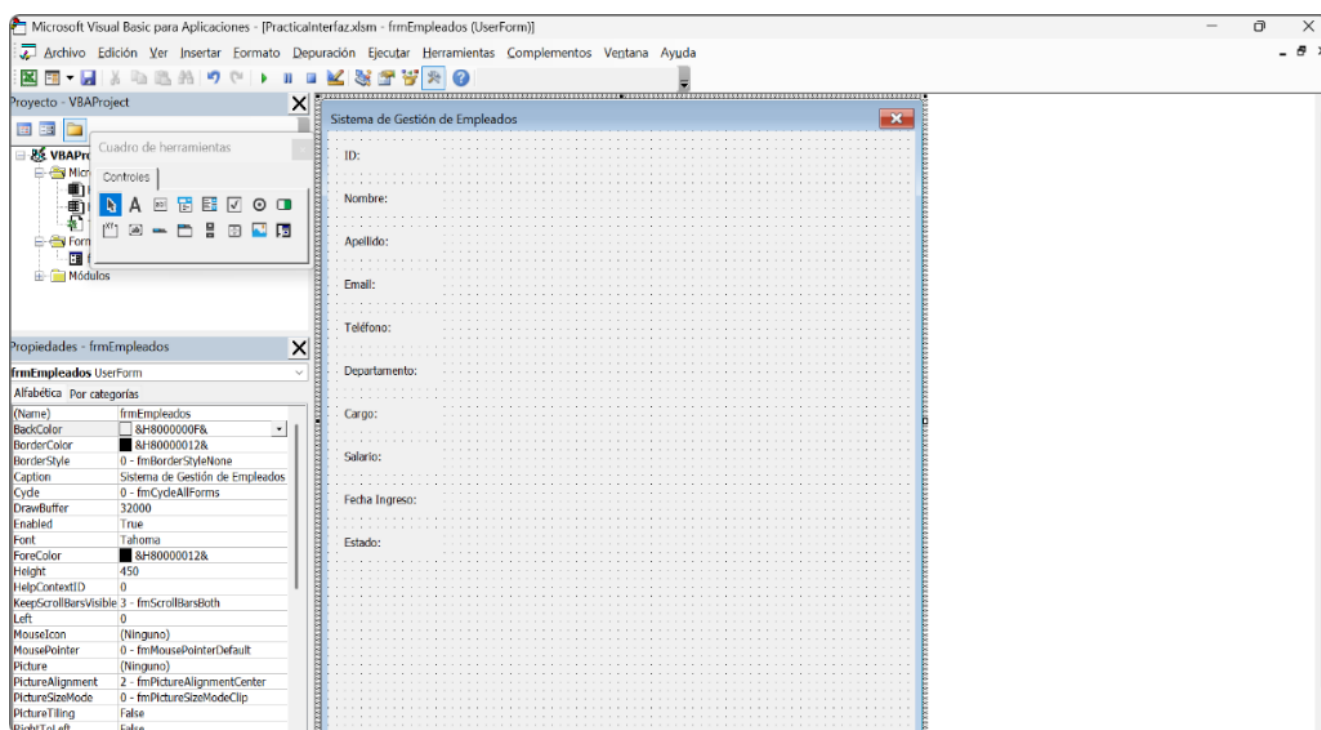
Tabla 9. Etiquetas del UserForm

| Control | Posición (aprox.) | Caption |
|---------|-------------------|---------|
| Label1 | Top: 12, Left: 12 | ID: |
| Label2 | Top: 42, Left: 12 | Nombre: |

| Control | Posición (aprox.) | Caption |
|---------|--------------------|----------------|
| Label3 | Top: 72, Left: 12 | Apellido: |
| Label4 | Top: 102, Left: 12 | Email: |
| Label5 | Top: 132, Left: 12 | Teléfono: |
| Label6 | Top: 162, Left: 12 | Departamento: |
| Label7 | Top: 192, Left: 12 | Cargo: |
| Label8 | Top: 222, Left: 12 | Salario: |
| Label9 | Top: 252, Left: 12 | Fecha Ingreso: |
| Label10 | Top: 282, Left: 12 | Estado: |

Después de crear las etiquetas de la tabla 9, el UserForm debe presentarse como se presenta en la siguiente figura.

Figura 3. Sistema de gestión de empleados



En el siguiente video se explica cómo iniciar la creación de un sistema de gestión de empleados en Excel, utilizando formularios y macros para estructurar la interfaz y permitir la inserción automatizada de datos.

Video 5. Creación de MACRO para un sistema de gestión de empleados



[Enlace de reproducción del video](#)

Síntesis del video: creación de MACRO para un sistema de gestión de empleados

En este video, se presenta un tutorial sobre la creación de un sistema de gestión de empleados utilizando formularios y macros en Excel. El experto, en programación y gestión de datos, guía a los profesionales a través del proceso de importación de datos desde un archivo CSV y la creación de un formulario de usuario en Visual Studio. A lo largo del vídeo, se subrayan las metodologías de validación de datos y diseño de interfaces, proporcionando un enfoque práctico para la

automatización de la gestión de empleados. Este enfoque es esencial para optimizar la eficiencia operativa en la gestión de recursos humanos.

Además de los UserForms, Excel permite insertar controles directamente sobre la hoja de cálculo.

Esto resulta útil para usuarios que prefieren trabajar con interfaces integradas dentro del libro. Los controles disponibles en las hojas de cálculo incluyen:

- Botones de comando.
- Cuadros combinados.
- Controles de formulario.
- Controles ActiveX.

A continuación, se debe complementar el UserForm con los controles restantes según las configuraciones establecidas:

Tabla 10. Configuración de TextBoxes

| Control | Name | Posición | Ancho |
|---------|-------------|---------------------|------------|
| TextBox | txtID | Top: 12, Left: 100 | Width: 80 |
| TextBox | txtNombre | Top: 42, Left: 100 | Width: 150 |
| TextBox | txtApellido | Top: 72, Left: 100 | Width: 150 |
| TextBox | txtEmail | Top: 102, Left: 100 | Width: 200 |
| TextBox | txtTelefono | Top: 132, Left: 100 | Width: 120 |
| TextBox | txtSalario | Top: 222, Left: 100 | Width: 100 |

Para seleccionar valores predefinidos como departamento o cargo, se usarán listas desplegables (ComboBoxes).

Tabla 11. Configuración de ComboBoxes

| Control | Name | Posición |
|----------|-----------------|---------------------|
| ComboBox | cboDepartamento | Top: 162, Left: 100 |
| ComboBox | cboCargo | Top: 192, Left: 100 |

La fecha de ingreso será capturada inicialmente mediante un cuadro de texto. Más adelante puede reemplazarse por un control de calendario si se desea mayor validación.

Tabla 12. Configuración de Fecha de Ingreso (TextBox temporal)

| Control | Name | Posición |
|---------|-----------------|---------------------|
| TextBox | txtFechaIngreso | Top: 252, Left: 100 |

Para indicar el estado del empleado (activo o inactivo), se empleará un grupo de botones de opción (OptionButtons), contenidos dentro de un marco (Frame).

Frame (contenedor de los botones)

- Name: fraEstado.
- Caption: Estado.
- Posición: Top: 282, Left: 100.
- Tamaño: Width: 150, Height: 50.

Dentro del Frame

OptionButton1:

- Name: optActivo.
- Caption: Activo.
- Posición: Top: 15, Left: 10.

OptionButton2:

- Name: optInactivo.
- Caption: Inactivo.
- Posición: Top: 15, Left: 80.

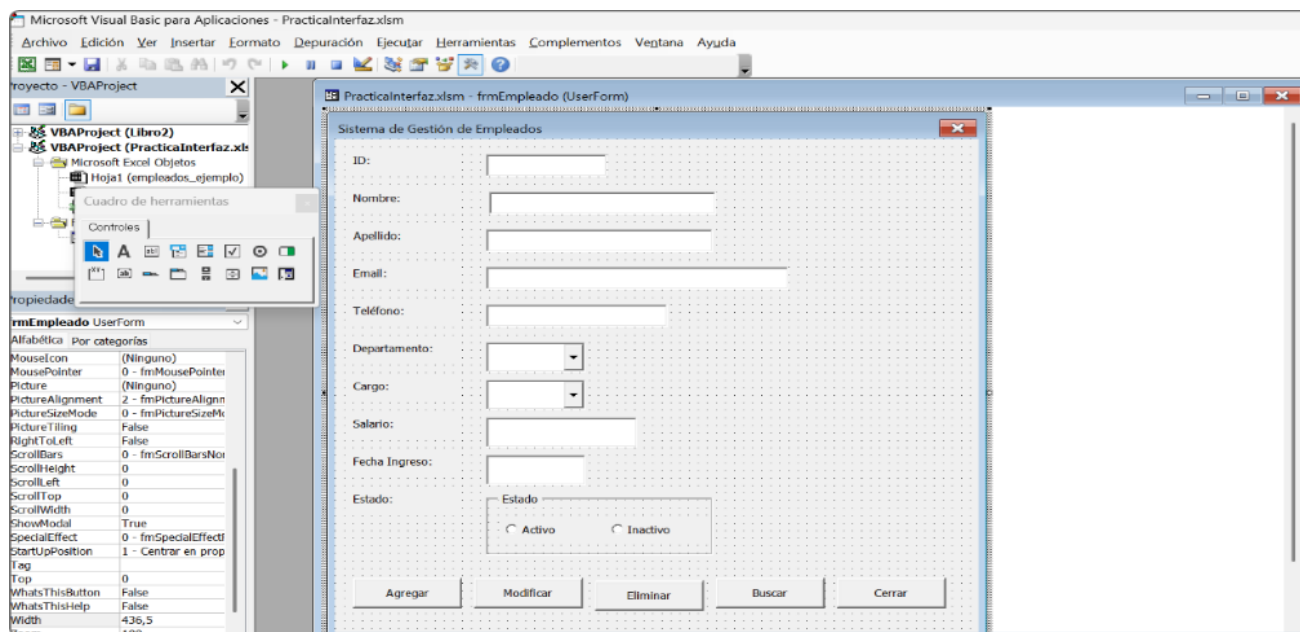
Finalmente, se deben agregar los botones de acción en la parte inferior del formulario. Estos permitirán ejecutar las funciones básicas del sistema: agregar, modificar, eliminar, buscar y cerrar.

Tabla 13. Configuración de CommandButtons

| Button | Name | Caption | Posición |
|---------------|--------------|-----------|---------------------|
| CommandButton | btnAgregar | Agregar | Top: 350, Left: 12 |
| CommandButton | btnModificar | Modificar | Top: 350, Left: 92 |
| CommandButton | btnEliminar | Eliminar | Top: 350, Left: 172 |
| CommandButton | btnBuscar | Buscar | Top: 350, Left: 252 |
| CommandButton | btnCerrar | Cerrar | Top: 350, Left: 332 |

Ingresa y gestione la información de los empleados de manera rápida y organizada:

Figura 4. UserForm completo del sistema de gestión de empleados



Después de haber agregado y configurado todos los elementos visuales en el UserForm, es necesario **cargar los datos en los ComboBoxes** para que el formulario esté completamente funcional al momento de iniciarse.

A continuación, se presenta el código que debe agregarse en el evento `UserForm_Initialize`: Para ello, se debe hacer doble clic sobre el `UserForm` en el Editor de VBA para abrir su módulo de código y copiar el siguiente bloque:

```
Private Sub UserForm_Initialize()  
    ' Configurar ComboBox Departamento  
    With cboDepartamento  
        .AddItem "Recursos Humanos"  
        .AddItem "Finanzas"  
        .AddItem "Marketing"  
        .AddItem "Ventas"  
        .AddItem "IT"  
        .AddItem "Operaciones"  
    End With  
  
    ' Configurar ComboBox Cargo  
    With cboCargo  
        .AddItem "Gerente"  
        .AddItem "Supervisor"  
        .AddItem "Analista"  
        .AddItem "Asistente"  
        .AddItem "Especialista"  
        .AddItem "Coordinador"  
    End With  
  
    ' Estado por defecto  
    optActivo.Value = True  
End Sub
```

Este procedimiento asegura que los valores de **Departamento**, **Cargo** y **Estado** estén disponibles y configurados por defecto al momento de abrir el formulario.

Prueba del formulario en Excel

Una vez finalizado el desarrollo del formulario, es necesario probar su funcionamiento. Para ello, se deben realizar los siguientes pasos:

- Guardar el proyecto (**Ctrl + S**).
- Cerrar el **Editor de VBA**.
- En Excel, presionar **Alt + F8**.
- Si no aparece ninguna macro, regresar al Editor de VBA.
- Presionar **F5** mientras tienes seleccionado el UserForm para probarlo.

Crear una macro para abrir el formulario

Terminado todo el desarrollo del formulario, se debe crear la macro que permitirá abrirlo. En el Editor de VBA, seguir estos pasos:

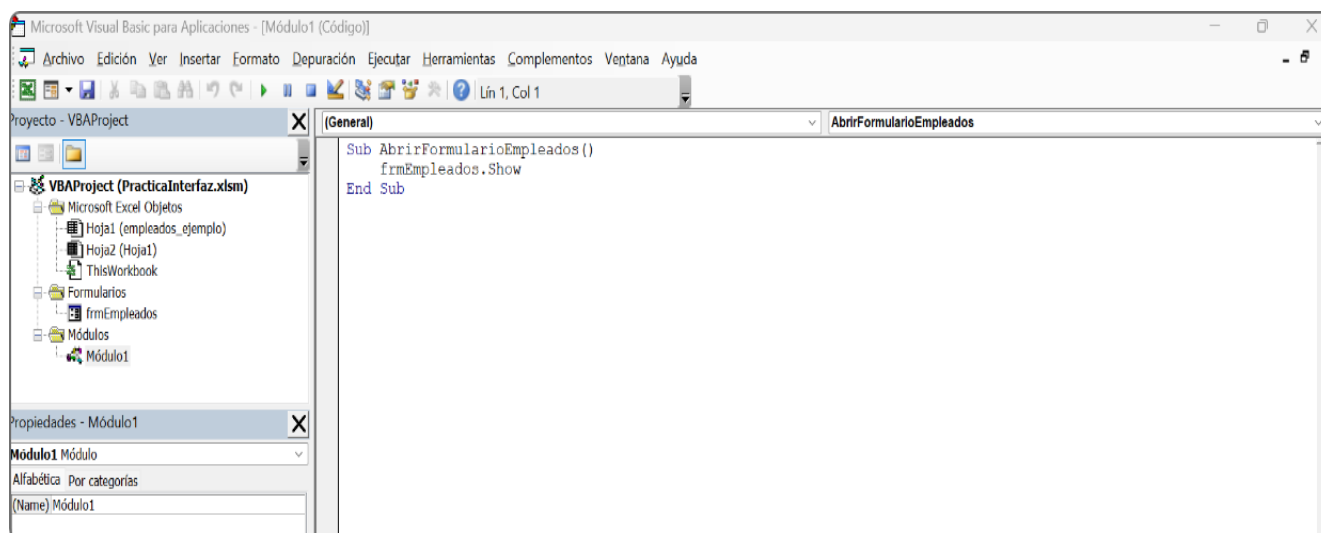
- a) Insertar un módulo.
- b) Agregar el siguiente código:



```
Sub MostrarFormulario()  
    UserForm1.Show  
End Sub
```

En el Editor de VBA, se inserta un módulo donde se escribe la macro que permite abrir el formulario **frmEmpleados** mediante el método `UserForm.Show`.

Figura 5. Macro para abrir el formulario de empleados en Excel VBA



Ejecución de la macro

En el siguiente video se explica detalladamente cómo ejecutar la macro, junto con el formulario que automatiza la captura de los datos.

Video 6. Complemento MACRO gestión de empleados



[Enlace de reproducción del video](#)

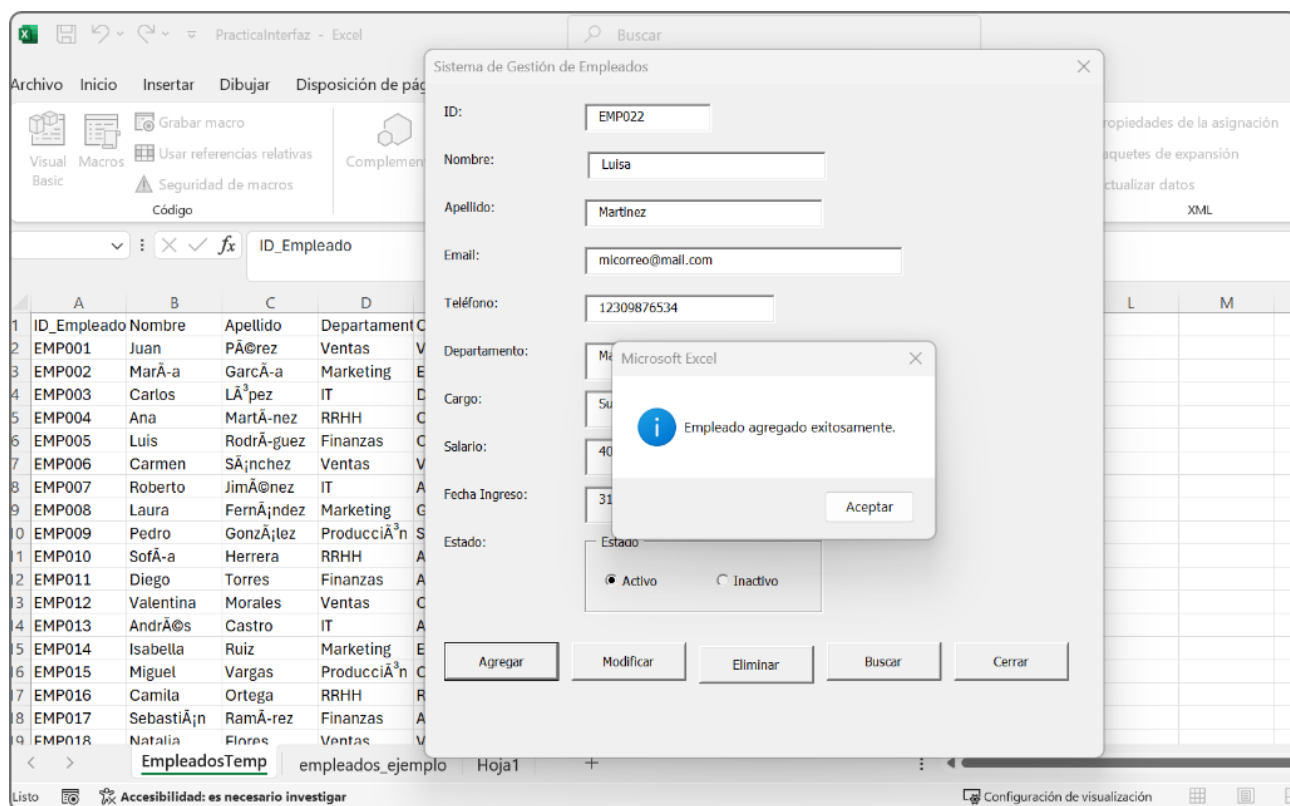
Síntesis del video: complemento MACRO gestión de empleados

En este video, el experto en programación y especialista en lenguaje de programación Visual Basic, guía a los profesionales a través del desarrollo de una macro destinada a la gestión de empleados en una empresa. Con una trayectoria destacada en la creación de interfaces funcionales y una sólida comprensión de la programación VBA, el experto demuestra cómo construir un formulario que integre diversos elementos esenciales, como cuadros de texto y botones de comando, para optimizar la entrada y el manejo de datos. A lo largo de la presentación, se enfatiza la importancia de un diseño intuitivo y la funcionalidad del formulario, asegurando que los usuarios puedan interactuar fácilmente con el sistema. Este enfoque práctico y estructurado permite a los profesionales adquirir habilidades aplicables en el manejo de datos empresariales.

Confirmación del registro agregado

a vez creado el formulario y grabada la macro, se puede ejecutar esta última para automatizar el ingreso de datos a un archivo en formato **.csv**. Si el procedimiento es correcto, se mostrará un mensaje confirmando el registro agregado.

Figura 6. Confirmación de registro de empleado en el sistema de gestión



Cuadros de diálogo integrados en VBA

VBA permite interactuar con el usuario mediante funciones simples como:

InputBox

Solicita información al usuario.

MsgBox

Muestra mensajes o confirmaciones.

Estas funciones son útiles para formularios básicos o mensajes rápidos.

Ventajas de usar interfaces en macros

Las interfaces gráficas dentro de macros de Excel ofrecen una serie de beneficios tanto para usuarios avanzados como para quienes no conocen programación.

Facilidad de uso

Los usuarios no necesitan conocer VBA para ejecutar macros complejas.

Entrada de datos controlada

Permite validar y dar formato a la información antes de procesarla.

Experiencia profesional

Una interfaz bien diseñada mejora la presentación de tus herramientas.

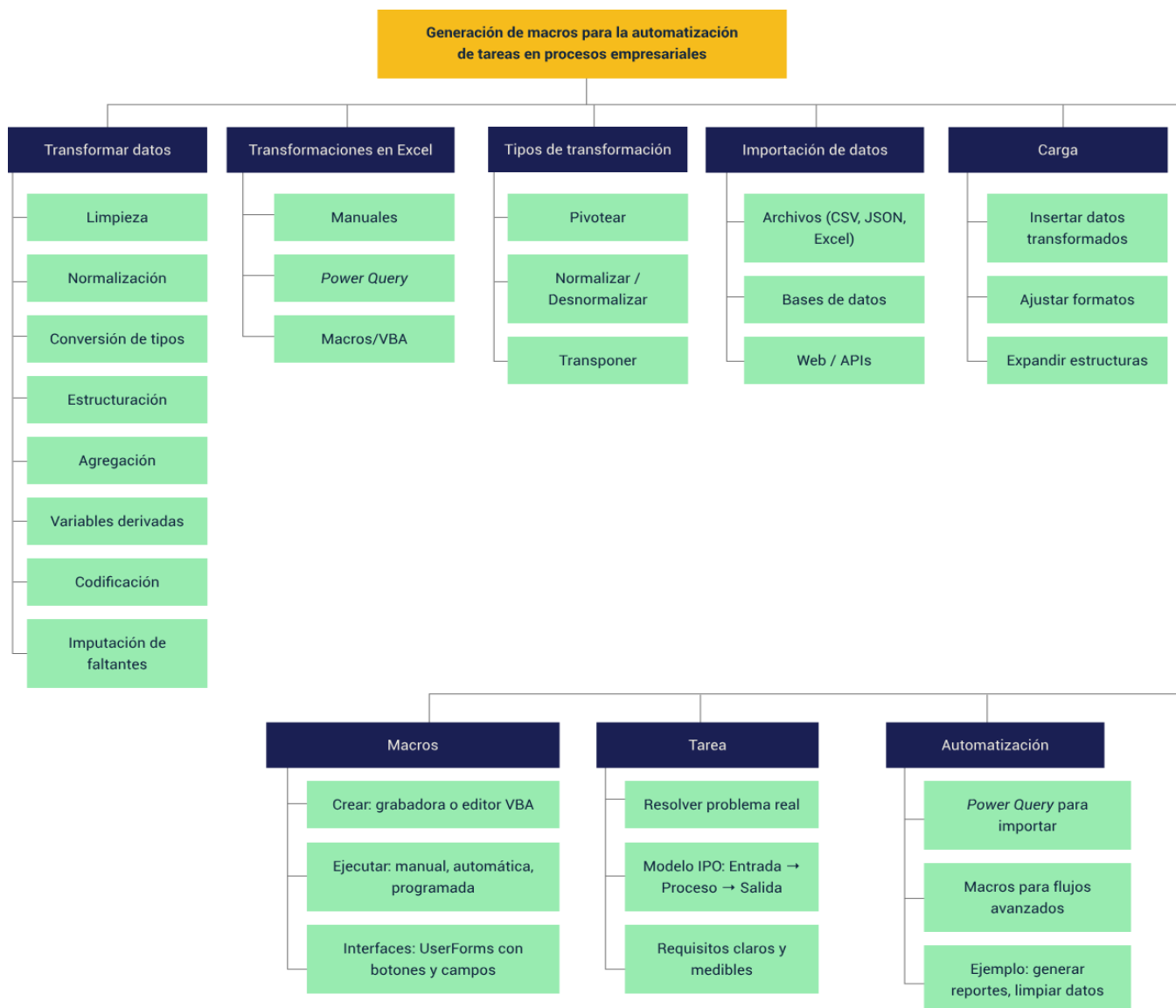
Reducción de errores

Al guiar al usuario por medio de formularios, se minimizan errores de ingreso.

Estas interfaces son especialmente útiles cuando se desarrollan herramientas de Excel que serán utilizadas por personas **no familiarizadas con macros o programación**, ya que **facilitan la interacción y reducen errores operativos**.

Síntesis

Este componente formativo proporciona una base para el dominio de las macros en Excel, abarcando desde los conceptos fundamentales de transformación de datos hasta la implementación y mantenimiento de soluciones automatizadas avanzadas.



Material complementario

| Tema | Referencia APA del material | Tipo | Enlace |
|--------|--|---------------|---|
| Macros | A2 Capacitación: Excel. (2020). Aprende macros en Excel en menos de 11 minutos [Video]. YouTube. | Video | https://www.youtube.com/watch?v=PvebpkrMG4o |
| Macros | SmartPro Academy. (2022). Aprende a crear macros en Excel desde cero [Video]. YouTube. | Video | https://www.youtube.com/watch?v=jjoeK6rzQDA |
| Macros | A2 Capacitación: Excel. (2020). Macros que necesitas en tu trabajo [Video]. YouTube. | Video | https://www.youtube.com/watch?v=D-zMyoEfiR8 |
| Macros | Deivis Eduard Ramírez Martínez (2025). Dataset empleados. | Archivo Excel | https://ecored-sena.github.io/CF2_EXCEL_AVANZADO_GESTION ESTRATEGICA INFORMACION/downloads/empleados.csv |
| Macros | Deivis Eduard Ramírez Martínez (2025). Dataset empleados_ejemplo. | Archivo Excel | https://ecored-sena.github.io/CF2_EXCEL_AVANZADO_GESTION ESTRATEGICA INFORMACION/downloads/empleados_ejemplo.csv |
| Macros | Deivis Eduard Ramírez Martínez (2025). Dataset Practica interfaz. | Archivo Excel | https://ecored-sena.github.io/CF2_EXCEL_AVANZADO_GESTION ESTRATEGICA INFORMACION/downloads/PracticaInterfaz.xlsm |

Glosario

Automatización: uso de macros o herramientas como Power Query para ejecutar tareas repetitivas sin intervención manual.

Carga de datos: inserción de la información transformada en la hoja de cálculo para su análisis final.

Importación de datos: incorporación de información desde diversas fuentes externas a Excel para su posterior análisis.

Limpieza de datos: eliminación de errores, inconsistencias y duplicados.

Macro: secuencia de instrucciones en VBA que permite automatizar tareas repetitivas dentro de Excel.

Normalización: estandarización de formatos y unidades que mejora la consistencia y elimina redundancias.

Pivotear datos: reorganización que convierte filas en columnas para facilitar el análisis comparativo.

Tarea: unidad de trabajo sistemática, repetible y medible que transforma entradas en salidas siguiendo reglas de negocio.

Transformar datos: conjunto de operaciones necesarias para hacer que los datos sean utilizables: consistentes, limpios, con tipos correctos y estructura apta para análisis y visualización.

UserForm: formulario personalizado creado en VBA que permite interacción con el usuario mediante controles visuales.

Referencias bibliográficas

Alexander, M., Kusleika, R., & Walkenbach, J. (2018). Excel 2019 bible. John Wiley & Sons.

Alexander, M., & Kusleika, D. (2019). Excel 2019 power programming with VBA. John Wiley & Sons.

Bernal-García, J. J., Dolores, S. M. M. M., & Bernal-Soto, P. (2011). Representación automática de funciones en Excel y su aplicación docente. Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA, 12(1), 141-157.

González, F. J. P., & Llamas, M. D. C. G. (2017). Análisis y representación gráfica de funciones matemáticas con Excel. Anales de ASEPUMA, (25), 1.

Winston, W. (2016). Microsoft Excel data analysis and business modeling. Microsoft press.

Winston, W. (2021). Microsoft Excel data analysis and business modeling (Office 2021 and Microsoft 365). Microsoft Press.

Créditos

| Nombre | Cargo | Centro de Formación y Regional |
|-------------------------------------|---|---|
| Milady Tatiana Villamil Castellanos | Líder del ecosistema | Dirección General |
| Olga Constanza Bermúdez Jaimes | Responsable de línea de producción | Dirección General |
| Deivis Eduard Ramírez Martínez | Experto temático | Centro de Comercio y Servicios – Regional Tolima |
| Paola Alexandra Moya | Evaluadora instruccional | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Blanca Flor Tinoco Torres | Diseñador de contenidos digitales | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Robinson Javier Ordoñez Barreiro | Desarrollador full stack | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Carlos Eduardo Garavito Parada | Animador y productor audiovisual | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| José Eduardo Solano Rivero | Animador y productor audiovisual | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Andrés Felipe Guevara Ariza | Locución | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Aixa Natalia Sendoya Fernández | Validador de recursos educativos digitales | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Jaime Hernán Tejada Llano | Validador de recursos educativos digitales | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |
| Raúl Mosquera Serrano | Evaluador para contenidos inclusivos y accesibles | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |

| Nombre | Cargo | Centro de Formación y Regional |
|----------------------------|---|---|
| Daniel Ricardo Mutis Gómez | Evaluador para contenidos inclusivos y accesibles | Centro Agroempresarial y Desarrollo Pecuario - Regional Huila |