



Universidad Central del Ecuador  
Facultad de Ciencias Administrativas  
**Modalidad a Distancia**

## **UNIDAD DIDÁCTICA**

**Asignatura: TIC APLICADAS**

**Nivel: TERCERO**

**Autor: Dr. Marco Quintanilla**

**Periodo: 2026 - 2026**

## Contenido

|                                                                   |          |
|-------------------------------------------------------------------|----------|
| <b>Introducción .....</b>                                         | <b>4</b> |
| <b>Objetivos de la Asignatura .....</b>                           | <b>5</b> |
| <b>Contribución de la Asignatura.....</b>                         | <b>6</b> |
| <b>Relación de la Asignatura con otras Materias .....</b>         | <b>7</b> |
| INTRODUCCIÓN.....                                                 | 12       |
| UNIDAD 1: SISTEMAS GESTORES DE BASES DE DATOS.....                | 14       |
| Access.....                                                       | 15       |
| Relaciones en Access.....                                         | 22       |
| Formularios .....                                                 | 25       |
| Consultas.....                                                    | 27       |
| Informes.....                                                     | 29       |
| Seguridad de la Base de Datos.....                                | 30       |
| UNIDAD 2: SOFTWARE PARA GESTIÓN DE PROYECTOS .....                | 30       |
| OpenProj .....                                                    | 32       |
| Creación de un proyecto.....                                      | 32       |
| Calendario laboral de un proyecto .....                           | 33       |
| Creación de Tareas y sus Dependencias .....                       | 34       |
| Gestión de recursos en el proyecto .....                          | 41       |
| <i>Listado de Recursos</i> .....                                  | 41       |
| Modificar datos de recursos .....                                 | 43       |
| Asignar un recurso. ....                                          | 43       |
| <i>El campo Unidades al Asignar recurso</i> .....                 | 45       |
| <i>Calendario de recursos</i> .....                               | 45       |
| Costos.....                                                       | 46       |
| <i>Determinado por Tasa</i> .....                                 | 46       |
| <i>Determinado por Uso</i> .....                                  | 47       |
| <i>Costo Fijo</i> .....                                           | 47       |
| <i>Tasas de costo en períodos diferentes</i> .....                | 47       |
| Vistas para planificación y monitoreo.....                        | 48       |
| UNIDAD 3: LA HOJA ELECTRÓNICA COMO HERRAMIENTA DE DESARROLLO..... | 51       |
| Conceptos básicos de programación .....                           | 53       |

|                                                               |    |
|---------------------------------------------------------------|----|
| Dato.....                                                     | 54 |
| Tipos de datos .....                                          | 55 |
| Operadores .....                                              | 56 |
| Diseño de formularios. Objetos y sus propiedades básicas..... | 57 |
| El Formulario y sus propiedades.....                          | 59 |
| Estructura secuencial de programación .....                   | 59 |
| Uso del control Label .....                                   | 2  |
| Uso del control TextBox .....                                 | 4  |
| Uso del control CommandButton .....                           | 6  |
| Comentarios en la programación .....                          | 8  |
| Programación de Entrada o lectura de datos .....              | 8  |
| <i>Función VAL</i> .....                                      | 9  |
| Programación del Proceso .....                                | 10 |
| Programación de la Salida o Mostrar Resultados .....          | 10 |
| <i>Función FORMAT</i> .....                                   | 11 |
| Botón Nuevo y Salir.....                                      | 13 |
| UNIDAD 4: APLICACIÓN DE ESTRUCTURAS CONDICIONALES .....       | 17 |
| Estructura de Control Selectivo Simple .....                  | 17 |
| Estructura de Control Selectivo Doble.....                    | 20 |
| Estructura de Control Selectivo Anidado.....                  | 23 |
| Interoperabilidad con la hoja electrónica.....                | 26 |
| Declaración General de Variables .....                        | 27 |
| Programación del botón GRABAR.....                            | 29 |
| Funciones Integradas en VBA .....                             | 32 |

## **Introducción**

La asignatura de TIC APLICADAS, se desarrollará a distancia y combinada con tutorías presenciales, en 6 meses.

La metodología de estudio se basa en la lectura del material y la realización de ejercicios y trabajos individuales, así como en la retroalimentación y autoevaluación del alumno y la asesoría del tutor, apoyados en los recursos de información.

## Objetivos de la Asignatura

La evaluación es integral; buscamos asegurar que el objetivo de aprendizaje de la asignatura sea alcanzado por cada uno de los alumnos, por lo que se incluyen diversas técnicas e instrumentos de evaluación como la aplicación de portafolios, resolución de ejercicios y el examen final.

La comunicación será a través de tutorías presenciales.

A continuación, algunas consideraciones que debes tomar en cuenta, para alcanzar los objetivos propuestos:

- ❖ La asignatura está compuesta por 3 Unidades, con sus respectivos temas, objetivos, actividades de aprendizaje y mesografía.
- ❖ Las tutorías presenciales se efectuarán de lunes a viernes y sábado, en las aulas del ISED, en el horario establecido.
- ❖ El material didáctico para el desarrollo de cada una de las actividades de aprendizaje, lo encontrarás en cada unidad estudiada.
- ❖ Fomentaremos la utilización de medios electrónicos para búsqueda, organización, procesamiento y presentación de información. Los recursos de información y medios de comunicación que utilizaremos en el semestre son:
  - ❖ Buscadores
  - ❖ Sitios de Interés
  - ❖ Bibliografía
  - ❖ Netgrafía
  - ❖ Google Docs
- ❖ Fomentar la investigación de temas en diferentes fuentes de información.
- ❖ Propiciar que el estudiante valore sus actividades, reflexiones y llegue a un proceso de auto evaluación continuo.

## **Contribución de la Asignatura**

La asignatura de TIC APLICADAS contribuye a la formación profesional al equipar a los futuros contadores con las herramientas y conocimientos necesarios para registrar transacciones, generar información financiera y apoyar la toma de decisiones estratégicas en las organizaciones. Al dominar estos sistemas, los profesionales pueden ofrecer un análisis ágil y preciso, facilitando el cumplimiento de obligaciones fiscales, la evaluación de la rentabilidad y la implementación de controles internos, habilidades esenciales para el éxito en el campo laboral y el desarrollo económico de las empresas.

## **Relación de la Asignatura con otras Materias**

La asignatura de TIC APLICADAS se vincula estrechamente con la Administración, la Economía, las Matemáticas, la Estadística y la Informática, así como con el Derecho (mercantil, laboral y tributario), ya que proporciona la información financiera necesaria para la toma de decisiones empresariales, el análisis económico, la gestión de recursos y el cumplimiento de obligaciones legales.

Será continua a lo largo del módulo, se aplicará la evaluación diagnóstica al inicio de cada módulo para determinar el nivel de conocimiento que tiene los alumnos mediante el Cuadro CQA, la formativa que será utilizada al finalizar cada unidad y cada módulo y la suma aplicada al finalizar el semestre, a fin de determinar el nivel que han alcanzado los estudiantes

|                                                    |    |
|----------------------------------------------------|----|
| INTRODUCCIÓN.....                                  | 12 |
| UNIDAD 1: SISTEMAS GESTORES DE BASES DE DATOS..... | 14 |
| Access.....                                        | 15 |
| Elementos de una base de datos .....               | 16 |
| Clave Principal.....                               | 17 |
| Creación de una tabla .....                        | 17 |
| Tipo de datos.....                                 | 18 |
| Propiedades de los campos .....                    | 20 |
| Relaciones en Access.....                          | 22 |
| Procedimiento para relacionar tablas.....          | 23 |
| Tipos de relaciones .....                          | 24 |
| Formularios .....                                  | 25 |
| Creación de formulario tipo menú.....              | 26 |
| Consultas.....                                     | 27 |
| Creación de una consulta.....                      | 28 |
| Informes.....                                      | 29 |
| Seguridad de la Base de Datos.....                 | 30 |
| Cifrar con contraseña .....                        | 30 |
| UNIDAD 2: SOFTWARE PARA GESTIÓN DE PROYECTOS ..... | 30 |
| OpenProj .....                                     | 32 |
| Creación de un proyecto.....                       | 32 |
| Calendario laboral de un proyecto .....            | 33 |
| Creación de Tareas y sus Dependencias .....        | 34 |
| Ingreso de tareas.....                             | 34 |
| Duración estimada y duración real. ....            | 36 |
| Hitos .....                                        | 36 |
| Vincular tareas .....                              | 37 |
| Dependencia de tareas .....                        | 38 |
| Tiempo de posposición y de adelanto. ....          | 39 |
| Dividir tareas .....                               | 40 |
| Tareas críticas .....                              | 40 |
| Gestión de recursos en el proyecto .....           | 41 |
| Tipos de recursos .....                            | 41 |
| <i>Listado de Recursos</i> .....                   | 41 |
| Modificar datos de recursos .....                  | 43 |
| Asignar un recurso. ....                           | 43 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| <i>El campo Unidades al Asignar recurso</i> .....                  | 45 |
| <i>Calendario de recursos</i> .....                                | 45 |
| Costos .....                                                       | 46 |
| <i>Determinado por Tasa</i> .....                                  | 46 |
| <i>Determinado por Uso</i> .....                                   | 47 |
| <i>Costo Fijo</i> .....                                            | 47 |
| <i>Tasas de costo en períodos diferentes</i> .....                 | 47 |
| Vistas para planificación y monitoreo .....                        | 48 |
| UNIDAD 3: LA HOJA ELECTRÓNICA COMO HERRAMIENTA DE DESARROLLO ..... | 51 |
| Conceptos básicos de programación .....                            | 53 |
| Dato .....                                                         | 54 |
| Tipos de datos .....                                               | 55 |
| Operadores .....                                                   | 56 |
| Diseño de formularios. Objetos y sus propiedades básicas .....     | 57 |
| El Formulario y sus propiedades .....                              | 59 |
| Estructura secuencial de programación .....                        | 59 |
| Uso del control Label .....                                        | 2  |
| Uso del control TextBox .....                                      | 4  |
| Uso del control CommandButton .....                                | 6  |
| Comentarios en la programación .....                               | 8  |
| Programación de Entrada o lectura de datos .....                   | 8  |
| <i>Función VAL</i> .....                                           | 9  |
| Programación del Proceso .....                                     | 10 |
| Programación de la Salida o Mostrar Resultados .....               | 10 |
| <i>Función FORMAT</i> .....                                        | 11 |
| Botón Nuevo y Salir .....                                          | 13 |
| UNIDAD 4: APLICACIÓN DE ESTRUCTURAS CONDICIONALES .....            | 17 |
| Estructura de Control Selectivo Simple .....                       | 17 |
| Estructura de Control Selectivo Doble .....                        | 20 |
| Estructura de Control Selectivo Anidado .....                      | 23 |
| Interoperabilidad con la hoja electrónica .....                    | 26 |
| Declaración General de Variables .....                             | 27 |
| Programación del botón GRABAR .....                                | 29 |
| Funciones Integradas en VBA .....                                  | 32 |

## Índice de figuras

|                                                           |    |
|-----------------------------------------------------------|----|
| Figura 1. Pantalla inicial de Access .....                | 9  |
| Figura 2. Elementos del entorno de Access .....           | 10 |
| Figura 3. Tabla, Campos, Registros .....                  | 11 |
| Figura 4. Creación de una tabla .....                     | 12 |
| Figura 5. Propiedades de los campos .....                 | 15 |
| Figura 6. Relaciones en Access .....                      | 16 |
| Figura 7. Cuadro de diálogo Mostrar tabla .....           | 16 |
| Figura 8. Ventana Relaciones .....                        | 16 |
| Figura 9. Ventana Modificar Relaciones .....              | 17 |
| Figura 10. Ejemplo de tablas relacionadas .....           | 19 |
| Figura 11. Opciones para crear formularios .....          | 19 |
| Figura 12. Grupo de controles .....                       | 20 |
| Figura 13. Asistente para botones de comando .....        | 21 |
| Figura 14. Ejemplo de consulta .....                      | 22 |
| Figura 15. Consulta con cálculo .....                     | 23 |
| Figura 16. Íconos para ejecutar consulta .....            | 23 |
| Figura 17. Abrir en modo exclusivo .....                  | 24 |
| Figura 18. Ventana Creación de un proyecto .....          | 27 |
| Figura 19. Diagrama de Gantt, tabla de entrada .....      | 27 |
| Figura 20. Calendario Laboral .....                       | 28 |
| Figura 21. Ejemplo de hito .....                          | 31 |
| Figura 22. Tarea Predecesora y Tarea Sucesora .....       | 31 |
| Figura 23. Vincular y desvincular tareas .....            | 32 |
| Figura 24. Tipos de Dependencia de tareas .....           | 32 |
| Figura 25. Cuadro de diálogo dependencia de tareas .....  | 33 |
| Figura 26. Posposición o adelanto .....                   | 34 |
| Figura 27. Ejemplo Dividir Tarea .....                    | 34 |
| Figura 28. Ejemplo de tarea crítica .....                 | 35 |
| Figura 29. Vista Recursos .....                           | 36 |
| Figura 30. Cuadro de Diálogo Información de Recurso ..... | 37 |
| Figura 31. Cuadro de diálogo Asignar recursos .....       | 38 |
| Figura 32. Tareas con recursos asignados .....            | 38 |
| Figura 33. Ejemplo de asignación de Recursos .....        | 39 |
| Figura 37. Cambiar Calendario Laboral de un recurso ..... | 40 |
| Figura 38. Opciones de calendario laboral .....           | 40 |
| Figura 34. Ejemplo Costo por Tasa y por Uso .....         | 41 |
| Figura 35. Asignación de costo fijo .....                 | 41 |
| Figura 36. Tasas de costos en períodos diferentes .....   | 42 |
| Figura 39. Opciones de Vistas .....                       | 42 |

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Figura 40 Activación de Ficha DESARROLLADOR .....                                  | 45 |
| Figura 41 Opciones de la Ficha Desarrollador .....                                 | 46 |
| Figura 42 Cómo Guardar una macro .....                                             | 46 |
| Figura 43 ¿Qué es un Algoritmo? .....                                              | 47 |
| Figura 44 Partes de la elaboración de un algoritmo .....                           | 48 |
| Figura 45 Tipos de variables .....                                                 | 50 |
| Figura 46 Operadores Aritméticos - Ejemplos .....                                  | 50 |
| Figura 47 Operadores Relacionales o de Comparación - Ejemplos .....                | 51 |
| Figura 48 Operadores Lógicos .....                                                 | 51 |
| Figura 49 Entorno de Desarrollo de Visual Basic .....                              | 52 |
| Figura 50 Insertar Formulario .....                                                | 53 |
| Figura 51 Ejemplo de estructura secuencial .....                                   | 54 |
| Figura 52 Formulario insertado .....                                               | 55 |
| Figura 53 Cuadro de Herramientas .....                                             | 56 |
| Figura 54 Cuadro de Propiedades .....                                              | 56 |
| Figura 55 Ejemplo de propiedades de una etiqueta .....                             | 57 |
| Figura 56 Cuadros de texto y sus propiedades en un formulario .....                | 58 |
| Figura 57 Formulario con etiquetas y cuadros de texto .....                        | 59 |
| Figura 58 Formulario con etiqueta para mostrar resultados .....                    | 59 |
| Figura 59 Formulario con Botón de Acción .....                                     | 60 |
| Figura 60 Ventana de código .....                                                  | 60 |
| Figura 61 Subrutina con comentarios .....                                          | 61 |
| Figura 62 Ejemplo de Código de Programación .....                                  | 65 |
| Figura 63 Ejemplo de Formulario en ejecución .....                                 | 65 |
| Figura 64 Ejemplo de Formulario con datos y resultado .....                        | 66 |
| Figura 65 Ejemplo de formulario con Botón NUEVO y SALIR .....                      | 66 |
| Figura 66 Código completo .....                                                    | 67 |
| Figura 67 Ejemplo de estructura secuencial .....                                   | 68 |
| Figura 68 Formulario Ejemplo 2 .....                                               | 68 |
| Figura 69 Formulario con datos y resultado Ejemplo 2 .....                         | 69 |
| Figura 70 Representación Gráfica - Estructura selectiva simple .....               | 70 |
| Figura 71 Ejemplo de estructura selectiva simple .....                             | 71 |
| Figura 72 Formulario Ejemplo Control Selectivo Simple .....                        | 71 |
| Figura 73 Formulario Ejemplo con datos y resultado Control Selectivo Simple .....  | 72 |
| Figura 74 Representación Gráfica - Estructura selectiva doble .....                | 73 |
| Figura 75 Ejemplo de estructura selectiva doble .....                              | 73 |
| Figura 76 Formulario Ejemplo Control Selectivo Doble .....                         | 74 |
| Figura 77 Formulario Ejemplo con datos y resultado Control Selectivo Doble .....   | 75 |
| Figura 78 Representación Gráfica - Estructura selectiva anidada o múltiple .....   | 75 |
| Figura 79 Ejemplo de estructura selectiva doble .....                              | 76 |
| Figura 80 Formulario Ejemplo Control Selectivo Anidado .....                       | 76 |
| Figura 81 Formulario Ejemplo con datos y resultado Control Selectivo Anidado ..... | 77 |
| Figura 82 Formulario con Botón GRABAR .....                                        | 78 |
| Figura 83 Sección de Declaración global de variables .....                         | 79 |
| Figura 84 Hoja electrónica donde se guardarán los datos .....                      | 80 |
| Figura 85 Código completo con programación del botón Grabar .....                  | 82 |
| Figura 86 Ejemplo de ejecución del botón GRABAR .....                              | 83 |

## INTRODUCCIÓN

La incorporación tecnológica en las actividades cotidianas ha transformado de manera significativa la forma en la que se lleva a cabo la contabilidad y la auditoría. Las Tecnologías de la Información y la Comunicación (TIC) han desempeñado un papel fundamental en este proceso, al ofrecer herramientas y soluciones que agilizan, automatizan y mejoran la precisión de los procesos contables y de auditoría.

A través del análisis y la utilización de programas informáticos se puede sistematizar, procesar y analizar la información que se genera en una organización, con el propósito de apoyar y sustentar las decisiones estratégicas gerenciales de manera oportuna y efectiva y obtener así ventajas competitivas. Las TIC aplicadas a la contabilidad y la auditoría apuntan a la integración de sistemas informáticos, software especializado y plataformas en línea para facilitar la gestión de la información organizacional, permitiendo a las empresas y profesionales de la contabilidad optimizar sus operaciones, reducir errores, mejorar la seguridad de los datos y cumplir con las regulaciones y estándares contables de manera más eficiente.

Tanto la contabilidad como la auditoría son parte de los ejes que sostienen la gestión eficiente en empresas públicas y privadas; por consiguiente, sus profesionales deben estar capacitados respecto al uso de herramientas tecnológicas y su aplicación en el entorno financiero, económico y social, de manera que, al formar parte de los equipos multidisciplinarios puedan agregar valor y aportar en el cumplimiento de las metas organizacionales sugiriendo mejoras a la calidad y a la productividad.

En este contexto los contenidos a tratar están dirigidos al aprendizaje de programas de aplicación que guardan relación con los propósitos mencionados; a través del conocimiento de herramientas tecnológicas para el manejo de bases de datos, la gestión de proyectos y la utilización de las funcionalidades de una hoja electrónica utilizada como herramienta de

desarrollo, mediante lo cual se podrán adquirir habilidades y destrezas necesarias para dar respuesta a las nuevas necesidades de información y de formación, adicionalmente se aprovecharán las ventajas que surgen al adoptar estas tecnologías como apoyo para la toma de decisiones informadas.

La tecnología con sus desafíos y tendencias constituye un soporte para el procesamiento de los datos convirtiéndolos en información valiosa. El uso de la tecnología en el ámbito contable propicia la confiabilidad e integridad de los datos y el acceso a información veraz en forma oportuna; con estas características se promueve a que el profesional contable que se capacite adquiriendo competencias digitales se desempeñará con mayor eficiencia y eficacia en cualquier entorno laboral.

## UNIDAD 1: SISTEMAS GESTORES DE BASES DE DATOS

La irrupción de las TIC en las actividades empresariales ha generado la necesidad de automatizar gran parte de sus procesos y considerar simultáneamente mecanismos de almacenamiento que permitan la manipulación de los datos guardados.

La organización, el tratamiento y el acceso de la información, se hace posible a través de los sistemas gestores de bases de datos (SGBD), los cuales permiten una gestión eficaz de la información, así como administrar los datos y ayudar en la toma de decisiones (Hueso, 2015).

En este sentido, las bases de datos se definen como una colección o un conjunto de datos, que se encuentran almacenados de forma organizada y sistemática, permitiendo su manipulación, con la ventaja de poder recuperarlos o consultarlos en forma eficiente a través de un SGBD.

Los SGBD son importantes en la gestión, almacenamiento y análisis de datos, facilitan la eficiencia en la gestión de registros contables y proporcionan a los auditores las herramientas necesarias para llevar a cabo auditorías, garantizan la integridad y la seguridad de los datos, lo que es esencial en estos aspectos donde la precisión y la confidencialidad son fundamentales.

Los sistemas para la gestión o administración de las bases de datos proporcionan integridad de los datos, confiabilidad con la información almacenada evitando redundancias, facilidad de trabajo en cuanto al manejo de los mismos, eliminación de inconsistencias, seguridades de acceso, mecanismos de respaldo y recuperación de la información, acceso concurrente, entre otros aspectos que generan rapidez y agilidad en el tratamiento de la información, aportando productividad en las empresas y elevando su competitividad.

Existen numerosos SGBD en el mercado, cada uno diseñado para satisfacer necesidades específicas, como por ejemplo MySQL, Oracle Database, PostgreSQL, SQLite, Access, entre otros. En el siguiente apartado se revisará conceptos acerca de Access, un SGBD de escritorio

desarrollado por Microsoft que es ampliamente utilizado por usuarios individuales y pequeñas empresas para gestionar datos locales

## Access

Esta aplicación se basa en el argumento de las bases de datos relacionales, las cuales están constituidas por una o más tablas que contienen información ordenada de manera organizada (Viescas, 2007). Access parte del conjunto de aplicaciones de Microsoft Office y se utiliza para crear y gestionar bases de datos relacionales en entornos de usuario único o pequeños grupos de trabajo.

Al ingresar a Access, desde el botón Inicio de Windows, *Todos los Programas*, *Microsoft Office* y luego *Microsoft Access*, se direcciona a la pantalla inicial del programa. En este entorno se presenta una pantalla como la siguiente:

## Entorno inicial de Access

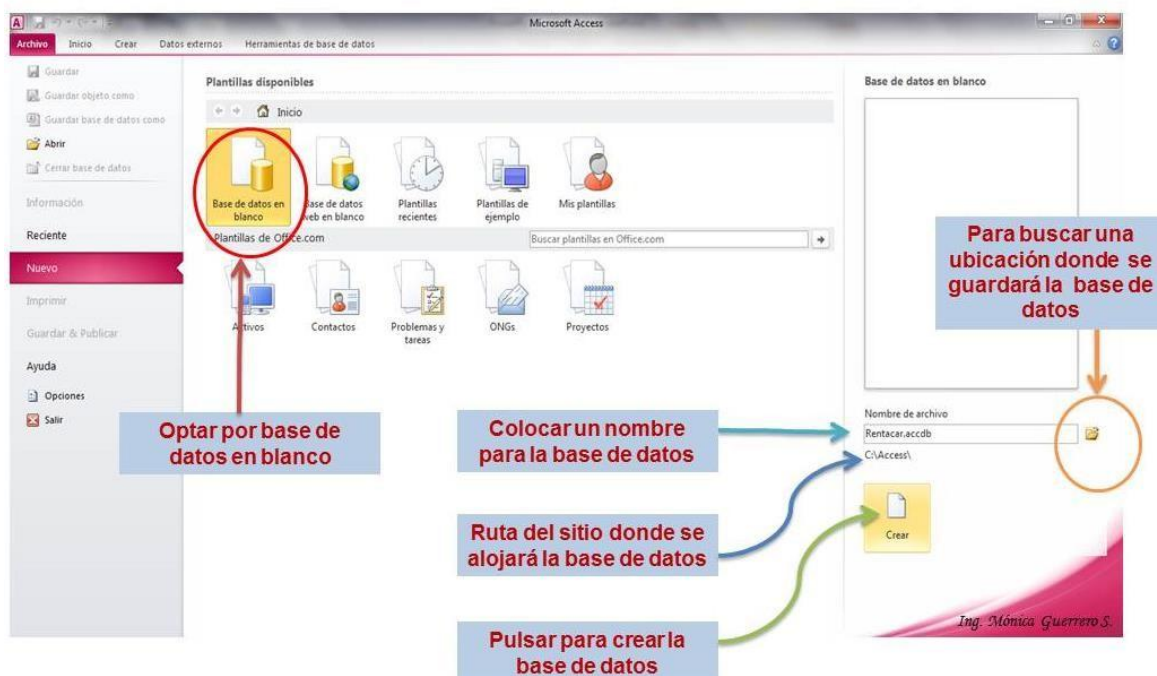


Figura 1. Pantalla inicial de Access

La imagen anterior expone la creación de una base de datos en blanco, también es factible empezar la creación a partir de una plantilla existente, o abrir una base de datos que se encuentre grabada en el equipo para proceder a modificar su diseño o su información.

Cuando se opta por crear una base de datos en blanco, aparece por defecto un primer y único objeto denominado *Tabla1*, la pantalla que se presenta con sus principales elementos es la siguiente:

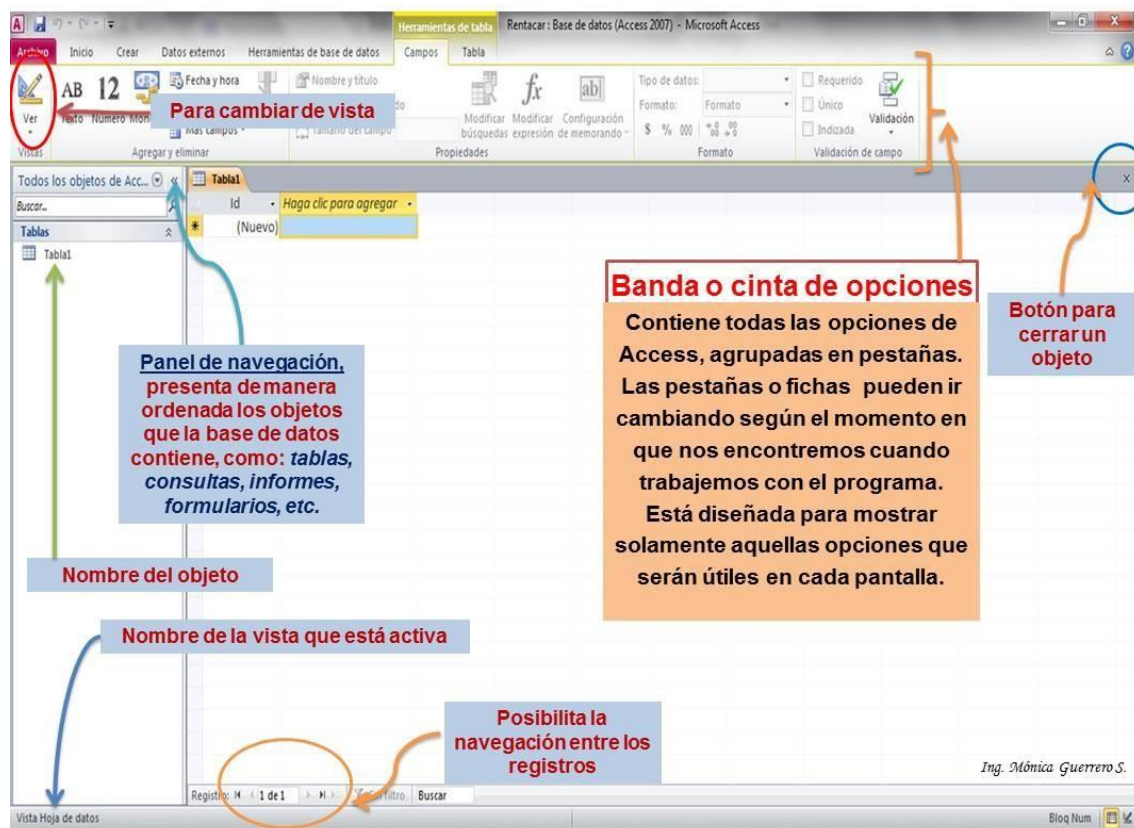


Figura 2. Elementos del entorno de Access

#### Elementos de una base de datos

Entre los objetos o elementos que conforman una base de datos, se encuentra la **TABLA**. La tabla llamada también entidad, es el elemento principal de la base de datos, ya que almacena todos los datos, y es a partir de la cual se crean o se generan los demás objetos. En toda tabla se debe guardar únicamente la información relacionada con un mismo aspecto, por ejemplo: Clientes, Artículos, Autos, etc.

Cada tabla está constituida por **CAMPOS** o atributos, que vienen a ser las características propias que posee la entidad y que la identifican. Así, por ejemplo:

- Para la tabla Clientes, los campos podrían ser: Nombre, Dirección, Teléfono, Cédula, Género, Fecha de Nacimiento, etc.
- Para la tabla Artículos, los campos serían: Código, Descripción, Precio, Stock, etc.,

Para la tabla Autos, los campos podrían ser: Placa, Marca, Color, Costo, etc. Por otra parte, existen en la tabla los REGISTROS o filas, que vienen a ser los datos propiamente dichos, es decir aquellos datos que se almacenan. Estos 3 elementos se grafican a continuación:

El diagrama muestra una tabla titulada 'AUTOS' dentro de un recuadro etiquetado como 'TABLA'. La tabla tiene tres columnas: 'PLACA', 'MARCA' y 'COLOR', que están etiquetadas como 'CAMPOS'. Hay tres filas de datos, etiquetadas como 'REGISTROS': una con 'XYZ1234', 'Mazda', 'Azul'; otra con 'ABC3210', 'Ford', 'Blanco'; y una tercera con 'RST6789', 'Chevrolet', 'Rojo'. Las flechas indican la relación entre las etiquetas y los elementos correspondientes en la tabla.

| TABLA   |           |        |
|---------|-----------|--------|
| AUTOS   |           |        |
| PLACA   | MARCA     | COLOR  |
| XYZ1234 | Mazda     | Azul   |
| ABC3210 | Ford      | Blanco |
| RST6789 | Chevrolet | Rojo   |

Figura 3. Tabla, Campos, Registros

Es importante destacar que en cada tabla debe existir un campo clave o una **CLAVE PRINCIPAL**, también denominada clave primaria.

#### Clave Principal

Su función será identificar a los registros en forma única, esto quiere decir que entre los registros este dato considerado como campo clave no se repetirá nunca.

La clave principal es útil para distinguir cualquier registro y nos facilita la relación entre las tablas. Por ejemplo, los siguientes datos serían únicos en una tabla: el número de cédula, el código de un artículo, la placa de un auto, el número de matrícula de un alumno, el número de historia clínica de un paciente, el número de factura, el número de pedido, entre otros.

#### Creación de una tabla

Para crear una tabla existen varias maneras, una de ellas es hacerlo en vista diseño, para lo cual se debe ir a la pestaña Crear, luego en el grupo Tablas, seleccionar Diseño de tabla.

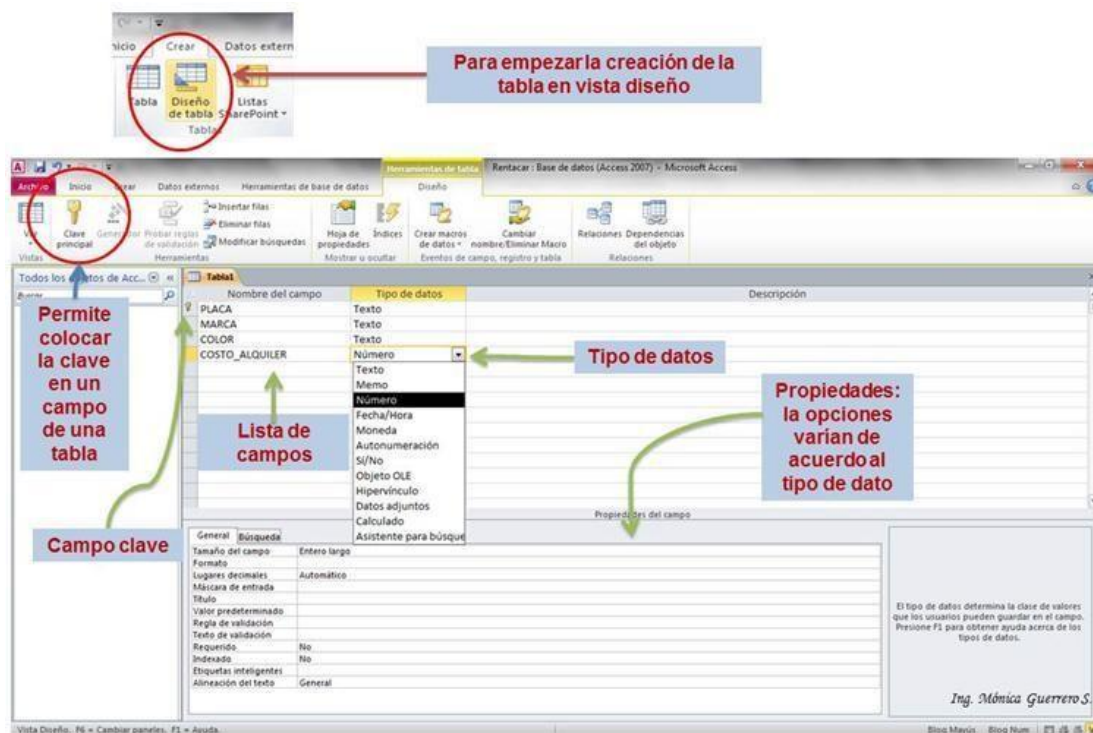


Figura 4. Creación de una tabla

En la columna *Nombre del campo*, se debe colocar un nombre que represente o guarde relación con el contenido del campo, se sugiere no colocarle símbolos especiales, excepto el guion bajo.

Igualmente hay que considerar que un nombre de campo no debe empezar con un dígito.

#### Tipo de datos

La columna *Tipo de datos* se refiere al contenido que cada campo podrá admitir, y al uso posterior que va a tener; por ejemplo:

- Texto: Puede contener hasta 255 caracteres de tipo alfanumérico, esto es: letras, símbolos especiales o datos numéricos, pero que éstos no vayan a ser utilizados en operaciones matemáticas. Por ejemplo: nombre, dirección, teléfono, marca, etc.; a partir de la versión 2013, este tipo de datos se llama Texto Corto.
- Memo, a partir de la versión 2013 este tipo de datos se llama Texto Largo: Se puede almacenar hasta 64.000 caracteres por cada campo, es ideal para campos que pueden contener grandes cantidades de información, por ejemplo: un diagnóstico médico,

observaciones sobre la entrega de un libro en biblioteca, el argumento de una película, etc.,

- Número: Para almacenar datos de tipo numérico, con los cuales sí se podrá realizar operaciones matemáticas. Por ejemplo: el número de empleados de una empresa, horas extras, sueldo, número de hijos, cargas familiares, costo de un artículo, etc.
- Fecha/Hora: Ideales para almacenar datos de fechas y horas en un formato especial (d/m/a h/m/s) que permite especificar el día, el mes, el año, las horas, los minutos y/o los segundos. Por ejemplo: fecha de nacimiento, fecha de caducidad, fecha de ingreso, hora de entrada.
- Moneda: Se utilizan cuando se quiere almacenar datos numéricos de tipo monetario, como por ejemplo salario, precio de compra, costo del producto, etc.
- Autonumeración: En este campo se puede almacenar un dato numérico que será generado de forma automática cada vez que se introduzca un nuevo registro, siempre comienza en 1 y se incrementa de manera secuencial, con la característica de que no se repite nunca, por lo cual puede ser aprovechado para considerarlo como un *campo clave*. Por cada tabla puede haber un solo campo de este tipo. Por ejemplo: código del empleado, números de factura, números de pedido, etc.
- Sí/No: Es un tipo de dato que admite sólo valores lógicos como Verdadero o Falso, al estar Activado o Desactivado. Por ejemplo: Experiencia (Sí/No), Aire Acondicionado (Sí/No), Garantía (Sí/No).
- Objeto OLE: Permite almacenar objetos de diferente naturaleza, como hojas de cálculo de Excel, dibujos, imágenes, sonido, etc. Por ejemplo: foto del empleado.
- Hipervínculo: Es ideal para almacenar el acceso directo hacia direcciones de sitios web o de correo electrónico. Por ejemplo: la dirección página web de la empresa, email, entre otros.

- Datos adjuntos: Es un tipo de dato similar a un objeto OLE, útil para adjuntar archivos como fotografías, imágenes, archivos de Office (de Word, de Excel, etc.).
- Calculado: Permite crear un campo que almacena el valor de un cálculo.

En la pestaña o ficha del objeto que se está creando, aparece el nombre de la tabla (como todavía no se ha asignado un nombre a la tabla, Access le asigna el nombre *Tabla1* por defecto). En la parte inferior se tiene dos pestañas (*General* y *Búsqueda*) para definir las propiedades del campo, es decir, características adicionales de la columna que estamos definiendo.

Se debe definir obligatoriamente un campo clave en cada una de las tablas. Para asignar una clave principal hay que pulsar clic sobre el nombre del campo que será clave principal. En la pestaña *Diseño de Herramientas de tabla*, seleccionar el botón *Clave principal*.

Luego de este proceso, a la izquierda del nombre del campo aparecerá una llave amarilla que indica que el campo es la clave principal de la tabla.

Cabe resaltar que un campo que sea definido como clave principal en una tabla, no puede contener valores nulos o en blanco. Si al llenar datos, se pretende dejar el campo en blanco o llenarlo con valores repetidos, el programa nos advierte con un mensaje.

#### *Propiedades de los campos*

Cada campo de una tabla dispone de una serie de características que facilitan el ingreso adicional de ciertas particularidades, sobre la forma de operar el campo. Estas propiedades, como se manifestó anteriormente, son especiales para cada tipo de dato y aparecen en la parte inferior de la Vista Diseño cuando se tiene un campo seleccionado. Por ejemplo, se muestra en la siguiente figura, las propiedades de un campo tipo texto.

| General                | Búsqueda      |
|------------------------|---------------|
| Tamaño del campo       | 20            |
| Formato                |               |
| Máscara de entrada     |               |
| Título                 |               |
| Valor predeterminado   |               |
| Regla de validación    |               |
| Texto de validación    |               |
| Requerido              | No            |
| Permitir longitud cero | Sí            |
| Indexado               | No            |
| Compresión Unicode     | Sí            |
| Modo IME               | Sin Controles |
| Modo de oraciones IME  | Nada          |
| Etiquetas inteligentes |               |

Figura 5. Propiedades de los campos

En la pestaña *General* se pueden ingresar las características generales de un campo, a su vez en la pestaña *Búsqueda* se puede definir una lista de valores válidos para el campo, es decir se puede definir una lista con los elementos que puede contener el campo, cuyo origen puede ser una lista de valores o una tabla o consulta.

Una de las propiedades de la pestaña *General* es: *Tamaño del campo*, esta propiedad permite delimitar el número de caracteres que puede contener un campo tipo Texto.

En un campo tipo Autonumérico el tamaño del campo es *Entero largo*

Para un campo tipo Número, las opciones para el *Tamaño del campo* son:

- *Byte* es equivalente a un caracter, almacena valores enteros entre 0 y 255.
- *Entero* para valores enteros entre -32.768 y 32.767.
- *Entero largo* para valores enteros entre -2.147.483.648 y 2.147.483.647.
- *Simple* para la introducción de valores entre -3,402823E38 y -1,401298E-45 para valores negativos, y entre 1,401298E-45 y 3,402823E38 para valores positivos.
- *Doble* para valores entre -1,79769313486231E308 y -4,94065645841247E-324

*Id. de réplica* se utiliza para claves autonuméricas en bases réplicas.

- *Decimal* para almacenar valores comprendidos entre  $-10^{38-1}$  y  $10^{38-1}$  (si estamos en una base de datos.adp) y números entre  $-10^{28-1}$  y  $10^{28-1}$  (si estamos en una base de datos.accdb).

## Relaciones en Access

Después de crear las tablas hay que relacionar las tablas. Para lo cual, se deben tener cerradas todas las tablas y luego pulsar en la opción *Relaciones* que está dentro de la pestaña *Herramientas de bases de datos*.

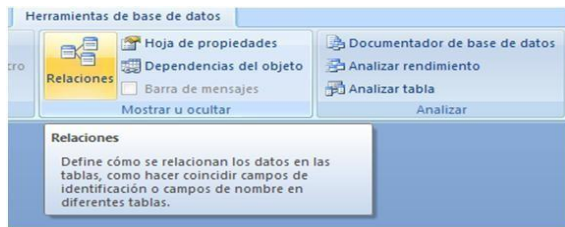


Figura 6. Relaciones en Access

Luego de escoger esta opción, aparece el cuadro de diálogo *Mostrar tabla*, en éste se deben agregar a la ventana Relaciones, las tablas que se van a relacionar.

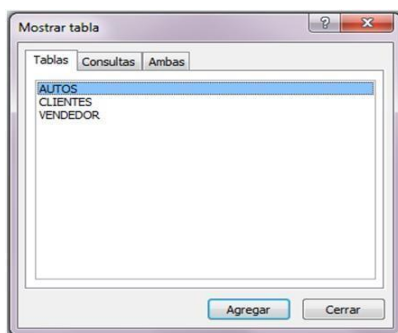


Figura 7. Cuadro de diálogo Mostrar tabla

Cuando se añaden las tablas, la ventana relaciones tiene el siguiente aspecto:

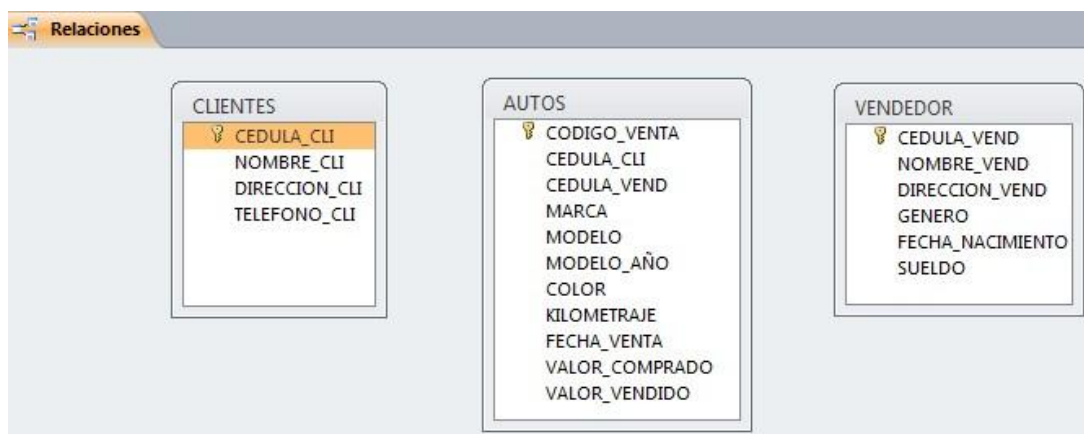


Figura 8. Ventana Relaciones

#### *Procedimiento para relacionar tablas*

Para empezar a relacionar las tablas que se han añadido, se procede de la siguiente manera:

**Arrastrar** el campo que se va a relacionar, desde una tabla y colocarlo sobre el campo con el que se quiere relacionar en la otra tabla. Es imprescindible tomar en cuenta que los campos a relacionar tengan el mismo tipo de dato y el mismo tamaño de campo, no necesariamente deben coincidir los nombres de los campos.

En este caso arrastrar el campo CEDULA\_CLI de la tabla CLIENTES sobre el campo CEDULA\_CLI de la tabla AUTOS, luego de lo cual aparecerá la siguiente ventana:

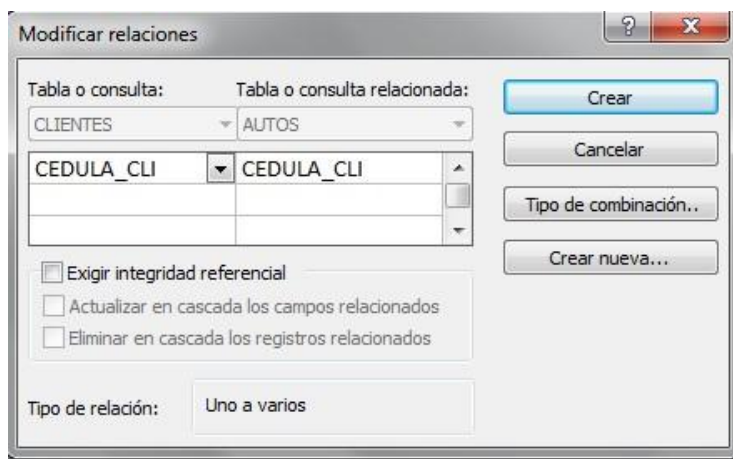


*Figura 9. Ventana Modificar Relaciones*

En primer lugar, aparecen los nombres de las dos tablas relacionadas (CLIENTES y AUTOS), debajo de ellas el nombre de los campos que se está relacionando CEDULA\_CLI y CEDULA\_CLI.

Luego están las opciones:

- Exigir integridad referencial: se activa esta opción para que los datos que se guarden sean consistentes y haya sincronización entre ellos. Por ejemplo, en este caso la cedula del cliente deberá ser ingresada en primera instancia en la tabla CLIENTES, para que luego se pueda registrar una transacción en la tabla AUTOS.

- Actualizar en cascada los campos relacionados: se activa con la finalidad de que se actualicen automáticamente en otras tablas los cambios que se realicen en la que se considere como tabla principal
- Eliminar en cascada los registros relacionados: se activa para que se eliminen automáticamente en otras tablas los registros que se borren en la tabla principal

Se recomienda activar cada uno de los recuadros ☒

#### *Tipos de relaciones*

En la parte inferior aparece el *Tipo de relación* que se va a crear, la cual depende de las características de los campos que intervienen en la relación, existiendo los siguientes tres tipos de relaciones:

- *Uno a varios* (1- $\infty$ ), es la relación más común, un registro es único en una tabla y en la otra tabla puede estar un indeterminado número de veces
- *Uno a Uno* (1-1), solo puede un registro coincidente en cada una de las tablas que se relacionen
- *Varios a Varios* ( $\infty$ - $\infty$ ), este tipo de relación existe, pero no tiene que darse. Por lo que se debe considerar una tabla adicional para romper la relación y que esta pase a ser de cualquiera de los dos tipos anteriores

Interpretando estos conceptos con el ejemplo que se plantea en el grafico anterior, se puede decir que *un* CLIENTE puede comprar *varios* AUTOS.

Hay que tomar en cuenta que, si el tipo de relación sale como *Indeterminado*, se debe *Cancelar* y volver a realizar la relación, ya que esto quiere decir que no es correcta la relación.

Una vez realizado el proceso y verificado el tipo de relación se debe pulsar el botón *Crear*.

Se repite el proceso con la otra tabla y finalmente las relaciones, en este caso, quedarían como se muestra a continuación:

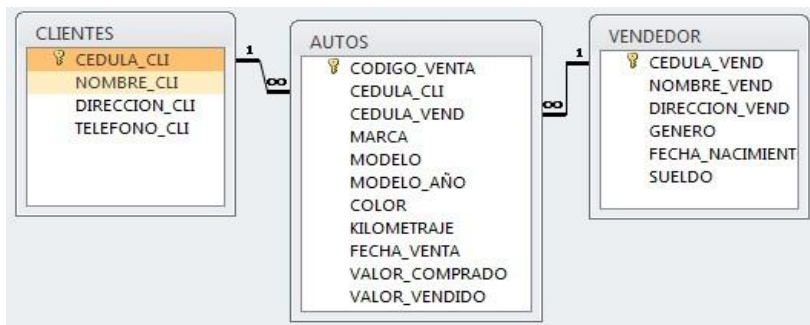


Figura 10. Ejemplo de tablas relacionadas

Para modificar una relación, hacer clic con el botón derecho sobre la línea de la relación a cambiar para elegir la opción *Modificar relación* y realizar los cambios pertinentes. Si lo que se desea es eliminar una relación, repetir el proceso anterior y seleccionar la opción *Eliminar*. No olvidar que se debe grabar las relaciones realizadas, pulsando clic en el ícono del disco que se encuentra en la barra de herramientas de inicio rápido.

Con las tablas creadas y relacionadas correctamente, se puede empezar a ingresar los datos. Este proceso se puede realizar dando doble clic en cada una de las tablas para abrirlas en *Vista Hoja de Datos*, o a su vez, se pueden crear formularios para el ingreso de información.

### Formularios

Los formularios sirven para poder ingresar, modificar y observar la información de los registros que conforman una tabla de una forma mucho más ágil. Para crear un formulario tenemos las siguientes opciones que se puede acceder a todas ellas desde la pestaña Crear:

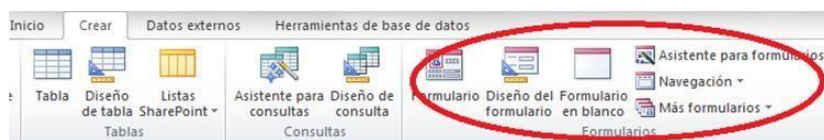


Figura 11. Opciones para crear formularios

Cada una de las opciones tiene una finalidad:

- **Formulario:** consiste en crear automáticamente un nuevo formulario que contiene todos los datos del objeto que esté seleccionado en el Panel de Navegación.

- Diseño del formulario: abre un formulario en blanco en la Vista Diseño para personalizarlo con los elementos que requiera incorporar. Esta opción es ideal para crear un formulario principal, que con botones de acción permita ser manejado como un menú.
- Formulario en blanco: crea un formulario sin contenido para poder arrastrar en él los objetos del Panel de Navegación.
- Asistente para formularios: utiliza un asistente para guiar paso a paso, el proceso de creación de un formulario.
- Navegación: crea un formulario para navegación, el cual permitirá navegar entre los objetos que contenga. Se dispone de diseños que se pueden personalizar.
- Más formularios despliega un menú con otros tipos de formularios disponibles, como tabla dinámica, cuadro modal, hoja de datos o el formulario dividido.

#### *Creación de formulario tipo menú*

Un formulario tipo menú tiene permite un manejo fácil de los objetos (formularios, consultas, informes) que se encuentran en el panel de navegación. El proceso a seguir es el siguiente:

- En la pestaña Crear, optar por Diseño del Formulario
- En la nueva pestaña que se crea, utilizar los objetos que se encuentran en el grupo

#### Controles

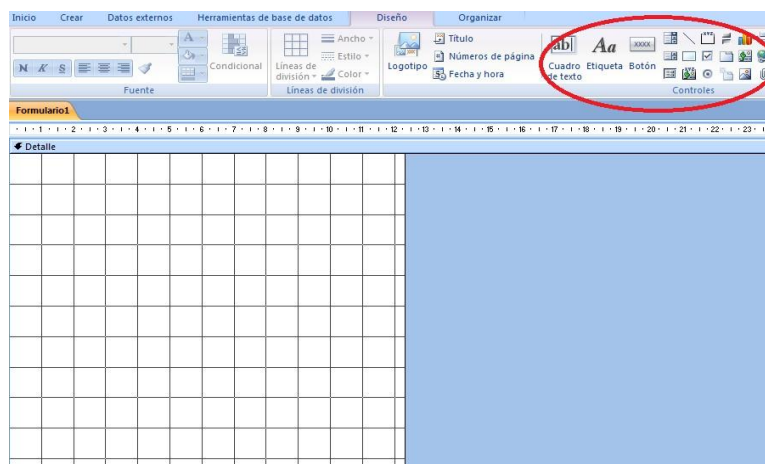


*Figura 12. Grupo de controles*

- Los principales controles a utilizar son: *Botón de acción* y *Etiqueta*

- El *botón de acción* ejecuta alguna acción al pulsar sobre él, se debe dar clic en botón de acción y dibujar en la cuadrícula el elemento a manera de un cuadro.
- Inmediatamente se va a desplegar el cuadro *Asistente para botones de comando*:

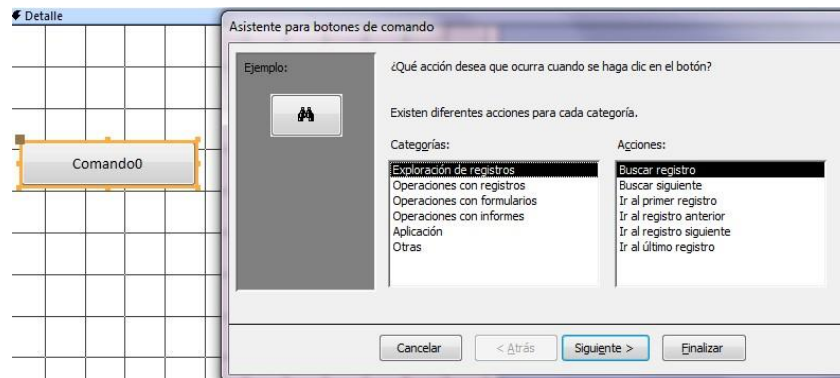


Figura 13. Asistente para botones de comando

- Hay que seguir los pasos del asistente, en el primero se especifica que quiere que haga el botón cuando se pulse clic sobre él. Seleccionar la categoría y aparecerá la acción. Por ejemplo, si se quiere abrir un formulario, escoger en *Categorías* la opción Operaciones con formularios y en *Acciones* escoger la opción Abrir formulario, luego pulsar *Siguiente*
- Después se debe escoger el formulario que se va a abrir.
- Luego introducir el texto o seleccionar imagen que se desee mostrar en la cara del botón.
- Es opcional, en el último paso colocar un nombre único al botón el cual servirá únicamente como referencia.
- Luego pulsar en Finalizar para salir del asistente.

En todos los casos se tiene la opción de manejar un formato de presentación personalizado.

- Para ver los efectos se debe cambiar la vista a la opción *Vista Formulario*.
- Cualquier cambio puede ser implementado en la *Vista Diseño*
- Finalmente se debe grabar el formulario realizado.

#### Consultas

Las consultas sirven básicamente para introducir cálculos o colocar criterios para filtrar cierto tipo de información que quiera ser visualizado. Existen varios tipos de consultas como:

**Consultas de selección:** Muestran los datos de una tabla que cumplen con criterios específicos.

Los datos que arroje la consulta pueden ser modificados

- **Consultas de acción:** Son consultas que realizan modificaciones en los datos grabados en las tablas.

#### *Creación de una consulta*

Para crear una consulta, ir a la pestaña *Crear* y en el grupo denominado *Otros*, escoger:

*Asistente para Consultas o Diseño de consulta.*

En el primer caso un asistente guiará el proceso de creación. En el segundo caso hay que especificar las tablas y los criterios para que se genere la consulta en la base de datos, lo primero es incluir las tablas que estarán involucradas en la consulta.

En la fila denominada *Campos*, colocar el campo que se quiere mostrar, automáticamente se establece la fila *Tabla*.

En la fila *Orden*, escoger un criterio de ordenamiento de los datos, que puede ser ascendente o descendente según el caso.

En la fila *Mostrar*, se debe activar si el campo va a ser visible en la ejecución de la consulta

En la fila de *Criterios*, introducir una condición que cumpla con los criterios para filtrar a los registros que se encuentran almacenados para ser mostrados en la consulta, ejemplo:

| Campo:     | NOMBRE VEND                         | DIRECCION VEND                      | GENERO                              |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Tabla:     | VENDEDOR                            | VENDEDOR                            | VENDEDOR                            |
| Orden:     |                                     |                                     |                                     |
| Mostrar:   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: |                                     |                                     | M                                   |
| o:         |                                     |                                     |                                     |

*Figura 14. Ejemplo de consulta*

**Cálculos en consultas:** es posible incluir ciertos cálculos con los campos de una tabla, para el efecto se debe escribir una expresión en la fila denominada campo de una nueva columna

tomando en cuenta la siguiente sintaxis: **Nombre del campo: expresión**



Figura 15. Consulta con cálculo

En la imagen anterior se muestra una consulta que calcula cual será el incremento del 10% al sueldo de los vendedores. La fórmula colocada es: Incremento: [SUELDO]\*10/100 Para grabar una consulta pulsar sobre el ícono del disco en la barra de inicio rápido, y para ejecutar la consulta pulsar en cualquiera de los íconos que se muestra a continuación:

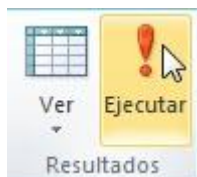


Figura 16. Íconos para ejecutar consulta

Para modificar el diseño de una consulta, se debe seleccionarla y luego abrirla en vista diseño pulsando en:



## Informes

Sirven para visualizar datos de una tabla o de una consulta en pantalla e imprimirlos. Para crear un informe se puede utilizar las opciones del grupo *Informes*, en la pestaña *Crear*, que son las siguientes:

- Informe: consiste en crear automáticamente un nuevo informe con los datos de la tabla o consulta seleccionada en el Panel de Navegación.
- Diseño de informe: abre un informe en blanco en la vista diseño para colocar los objetos y controles de forma personalizada

- Informe en blanco: abre un informe en blanco para incluir contenido según se requiera. ·

Asistente para informes: utiliza un asistente que va guiando paso por paso en la creación del informe. Esta opción es la más utilizada.

## Seguridad de la Base de Datos

### *Cifrar con contraseña*

Para proteger la información guardada en la base de datos se puede cifrarla con una contraseña, con esto se solicitará la contraseña cuando se intente abrir la base de datos. Se debe tener abierta la base de datos en modo exclusivo para realizar el proceso, para lo cual al abrir la base de datos se debe escoger la opción Abrir en modo exclusivo, como se muestra en la siguiente figura:

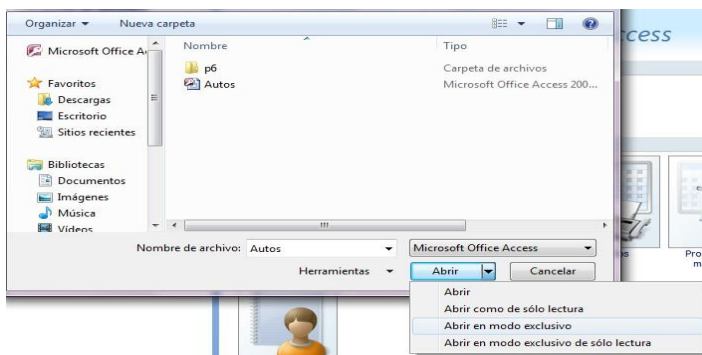


*Figura 17. Abrir en modo exclusivo*

Una vez abierta la base de datos en modo exclusivo, ir a *Archivo > Información > Cifrar con contraseña*. En el cuadro de diálogo que aparece se debe introducir la contraseña y confirmarla nuevamente.

Para quitar la contraseña, hay que primero abrir la base en modo exclusivo y luego pulsar Descifrar en *Archivo > Información*.

## UNIDAD 2: SOFTWARE PARA GESTIÓN DE PROYECTOS

Los mercados globalizados exigen cada vez más la utilización de la tecnología. El uso de las TIC para la presentación de proyectos académicos y empresariales involucra el uso de

aplicativos o programas, cuyo procesamiento de datos generen indicadores para la toma de decisiones empresariales. La implementación de innovaciones tecnológicas permite desarrollar nuevas formas de producción y generar nuevas demandas incrementando la rentabilidad, en este sentido las TIC son un vehículo necesario para generar competitividad.

La gestión de proyectos desempeña un papel crítico en el campo de la contabilidad y la auditoría al garantizar mediante el uso de herramientas la planificación, la organización y la ejecución eficiente de tareas y proyectos relacionados con la gestión financiera y la fiscalización. Una eficiente gestión de proyectos mejora la calidad y la precisión de los resultados y asegura que se cumplan los requisitos reglamentarios y las expectativas del cliente. La gestión de proyectos es fundamental para garantizar que los proyectos se completen dentro del plazo, el presupuesto y los estándares de calidad definidos. Se basa en principios de planificación, organización, dirección y control para lograr los objetivos del proyecto de manera eficiente y efectiva. Cabe indicar que un proyecto puede definirse como un esfuerzo temporal diseñado para lograr un resultado único. Cada proyecto tiene un objetivo claro, un plazo definido y recursos asignados (Bataller, 2016).

La gestión de proyectos a través de un software es una práctica común y efectiva que permite una mayor organización, colaboración y seguimiento de las tareas y recursos involucrados. Al emplear un aplicativo en la gestión de proyectos, no solo se mejora la eficiencia, sino que también proporciona una mayor transparencia y colaboración en el equipo. Además, facilita la generación de informes y la toma de decisiones basadas en datos consistentes.

Existen muchos programas tanto propietario como de código abierto para la gestión de proyectos, que pueden adaptarse a las necesidades de una organización, como son: Trello, Asana, Microsoft Project, Jira, Basecamp, entre otras.

Entre el software libre para la gestión de proyectos podría mencionarse a Project Libre (Bucki, 2019) y a OpenProj, los cuales tienen características y opciones similares en su manejo.

En el siguiente apartado se incluyen los aspectos más importantes a considerar para el manejo del OpenProj.

#### OpenProj

Es una plataforma de gestión de proyectos de código abierto que ofrece una amplia gama de opciones para planificar, colaborar y realizar seguimiento a proyectos, aunque al ser de carácter libre podría tener ciertas limitaciones; sin embargo, esta aplicación proporciona herramientas y funciones que ayudan a los equipos de un proyecto a definir tareas, asignar recursos, programar actividades y realizar un seguimiento del progreso o avance de un proyecto.

OpenProj es un software libre para administrar proyectos basado en tecnologías web, lo que quiere decir que se puede instalar en un servidor o equipo propio o también utilizar una versión alojada en la nube (Blokdyk, 2018), proporcionando flexibilidad y control sobre la gestión de proyectos.

Como la mayoría de los softwares libres, OpenProj está respaldado por una comunidad de desarrolladores y usuarios, lo que significa que constantemente se están realizando mejoras y actualizaciones en su plataforma.

#### Creación de un proyecto

Al iniciar y seleccionar la opción *Crea Proyecto*, se genera un archivo con extensión POD. Se deben proporcionar detalles como el nombre del proyecto y el nombre del administrador de este. Respecto a la temporalización, se puede partir la programación de actividades a partir de una fecha de comienzo o de fin del proyecto. Por defecto, el programa empieza la programación del proyecto con la opción planificación adelantada activada, pero se puede programar a partir de una fecha de fin, desactivando esta opción, en cuyo caso se trataría de una programación de actividades en retrospectiva.

En la siguiente figura se muestra lo indicado:

Figura 18. Ventana Creación de un proyecto

Al pulsar OK, se ingresa a la vista predeterminada que es el *Diagrama de Gantt*, que por defecto se muestra en la Tabla denominada de entrada, como se muestra en la figura:

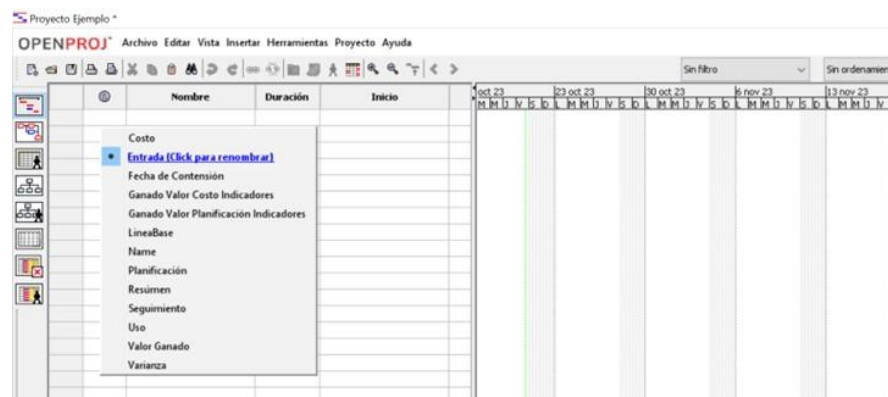


Figura 19. Diagrama de Gantt, tabla de entrada

Cabe indicar que, al pulsar clic derecho sobre el vértice superior izquierdo de cualquier de cualquiera de las Vistas que se encuentran a la izquierda de la pantalla, se accede a otras tablas que tienen como propósito facilitar el ingreso y la organización de la información en un proyecto.

#### Calendario laboral de un proyecto

En el ícono *Cambiar tiempo de trabajo* , se tienen las opciones para ingresar el horario laboral y definir los días de excepción dentro de la jornada laboral.

Esta aplicación posee tres calendarios base como plantillas que pueden ser modificadas según las necesidades del proyecto como son: un calendario estándar, un calendario para labores de 24 horas, y otro para jornadas nocturnas.

En la siguiente figura se muestran los ajustes disponibles que se pueden aplicar para definir el tiempo de trabajo dentro de un proyecto, se muestra un calendario estándar que tiene de forma predeterminada, un día laborable de 8 horas de lunes a viernes; sin embargo, esto es modificable según la necesidad que se tenga:

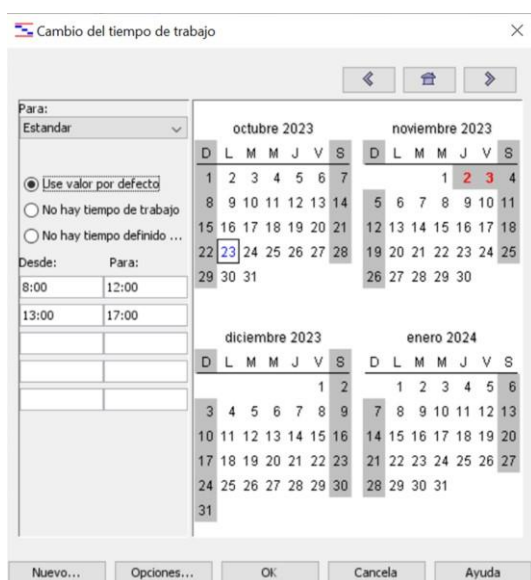


Figura 20. Calendario Laboral

### Creación de Tareas y sus Dependencias

Luego de determinar la fecha de inicio o fin de un proyecto y haber definido el calendario laboral, se prosigue con el ingreso de las tareas o actividades que conformarán el proyecto.

En este punto se deben definir las tareas de manera clara y específica y establecer relaciones de dependencia que tendrán entre ellas o su vinculación (Blokdyk, 2018, esto aporta a visualizar la secuencia de actividades del proyecto.

### Ingreso de tareas

Como se mencionó, al ingresar se puede apreciar la vista predeterminada que es el Diagrama de Gantt, el cual tiene dos componentes: la tabla (izquierda) y el calendario (derecha) con su escala temporal.

En la tabla se ingresan las tareas o actividades, completando las celdas *Nombre de tarea* y *Duración*. Por ejemplo, la duración de las tareas puede ser en días (*d*) o en horas (*h*), pudiendo utilizar decimales, por ejemplo 1,5 d. Si se requiere que una tarea sea realizada en períodos continuos no laborales o días transcurridos, se escribe *ed*.

Al ingresar una duración de tarea se puede observar que en la parte del *calendario* la duración se representa por medio de una barra y comienza el día en que se ha definido como fecha de comienzo del proyecto (observar en la escala temporal).

Con estos botones disponibles en la barra de herramientas  , es posible realizar un esquema de proyecto o crear una jerarquía de tareas que desglosa el trabajo en componentes más pequeños. Esto se denomina Estructura de Desglose del Trabajo (EDT) y muestra cómo se divide el proyecto en tareas de resumen y subtareas.

El botón de la izquierda denominado *Sin Sangría o No indentado* elimina la relevancia de una tarea a nivel de otra; y el botón de la derecha es el de *Sangría o Indentar*, al pulsarlo automáticamente se aplica una sangría en la tarea o tareas seleccionadas, haciendo que la tarea anterior se convierta en una tarea de resumen o tarea contenedora de otras tareas (tarea de nivel superior).

### **Tarea de Resumen**

Una tarea de resumen es aquella que agrupa y contiene un conjunto de tareas relacionadas o subordinadas (subtareas), totalizando la duración y el costo de las tareas que subordina o incluye. Este tipo de tarea no implica necesariamente la realización de un trabajo, más bien sirve para organizar y resumir tareas más detalladas que deben llevarse a cabo para alcanzar un objetivo específico del proyecto, son de ayuda para simplificar la visualización y la gestión de proyectos complejos, al agrupar tareas relacionadas en una estructura jerárquica. En la lista de tareas, las tareas de resumen aparecen en negrita, y su barra en el diagrama de Gantt es diferente a la de una tarea normal o subtaska en este caso.

#### *Duración estimada y duración real.*

Cuando se crea una tarea, el programa le asigna a ésta una duración estimada de 1 día, y se identifica con un signo de interrogación, así: 1 day?.

Cuando se modifica ese valor el signo de interrogación desaparece, haciendo que esa duración sea ya real. La duración estimada de una tarea es la cantidad de tiempo que se prevé tome completar esa tarea en el plan del proyecto, este tipo de duración se establece generalmente durante la fase de planeación del proyecto.

Mientras que la duración real de una tarea es la cantidad de tiempo que realmente tomó completar esa tarea una vez que se ha ejecutado. Se determina después de que la tarea se haya completado y se podría comparar con la duración estimada.

La diferencia entre estos dos tipos de duración radica en que la duración estimada se establece antes de que una tarea comience, es una estimación inicial; mientras que la duración real se determinaría después de que la tarea se completa y representa el tiempo real que se necesitó para llevar a cabo la tarea. La comparación entre estas dos duraciones permite evaluar el rendimiento del proyecto. Si la duración real es significativamente diferente de la duración estimada, puede indicar la necesidad de ajustar el plan del proyecto, asignar más recursos o realizar cambios en la gestión para cumplir con los plazos.

#### *Hitos*

Cuando se necesita destacar algo importante o marcar un punto de referencia en algún lugar de nuestro proyecto se puede ingresar un hito. Para ello debemos ingresar una tarea con una duración de cero días, esto hará que la tarea se muestre automáticamente como hito, cuyo ícono distintivo es un rombo de color negro.

Un hito constituye un evento o punto de referencia significativo que marca un logro importante en un proyecto. Se podría marcar un hito, por ejemplo en momentos cruciales que

marcan el final o el inicio de una fase importante, la consecución de un objetivo clave o la finalización de una tarea crítica, entre otros.

Los hitos son útiles para realizar un seguimiento del progreso del proyecto y asegurarse de que se están cumpliendo los plazos y objetivos, y además para establecer puntos de control y comunicar de manera efectiva el avance a los miembros del equipo y a las partes interesadas.

| Nombre        | Duración | 16 oct 23           |
|---------------|----------|---------------------|
| Fase completa | 0 days   | V S D L M M J V S D |
|               |          | ◆ 19/10             |

Figura 21. Ejemplo de hito

#### Vincular tareas

En un proyecto las tareas no siempre comienzan y se ejecutan todas juntas, por lo cual se programa la vinculación entre las tareas del proyecto, que es cómo se relaciona o va a depender una tarea de otra.

Este proceso genera los conceptos de tarea predecesora y tarea sucesora, que se detallan enseguida:

- Toda tarea tiene una tarea que se ejecuta **antes** que ella (excepto la tarea inicial).
- La tarea que se ejecuta antes que la tarea actual se llama **TAREA PREDECESORA**.
- Toda tarea tiene una tarea que se ejecuta **después** que ella (excepto la tarea final).
- La tarea que se ejecuta después de la tarea actual se llama **TAREA SUCESORA**.

Figura 22. Tarea Predecesora y Tarea Sucesora

Para vincular las tareas se utilizan los siguientes botones de la barra de herramientas:

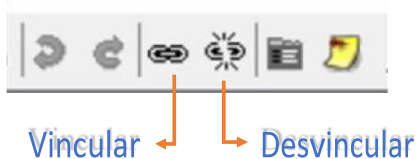


Figura 23. Vincular y desvincular tareas

### Dependencia de tareas

La dependencia de tareas se refiere a la relación entre dos o más tareas y determina el orden en que deben ejecutarse o completarse. Estas relaciones establecen cómo una tarea afecta a otra en términos de secuencia y tiempo. Las dependencias que pueden establecerse a las tareas de un proyecto garantizan que este avance de manera organizada, y son las que se grafican y describen en la siguiente figura:

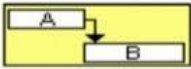
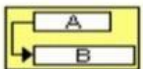
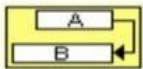
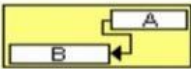
| Diagrama de Dependencia                                                             | Descripción                                                                        | Ejemplo                                                                                                                                      |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|    | <b>Fin a Comienzo (FC)</b><br>El final de una tarea marca el principio de la otra. | Actualizar una lista de distribución antes de imprimir las nuevas etiquetas postales.                                                        |
|    | <b>Comienzo a Comienzo (CC)</b><br>Las dos tareas comienzan a la vez.              | La electricidad y la fontanería de un edificio pueden instalarse al mismo tiempo.                                                            |
|    | <b>Fin a Fin (FF)</b><br>Las dos tareas terminan a la vez.                         | La copia de seguridad de la información de una computadora a reemplazar debe estar lista cuando esté instalada la nueva computadora.         |
|  | <b>Comienzo a Fin (CF)</b><br>El principio de una tarea marca el final de la otra. | La ejecución de informes en el antiguo Sistema de contabilidad no se acabará hasta que comiencen a ejecutarse informes con el nuevo sistema. |

Figura 24. Tipos de Dependencia de tareas

Las dependencias de tareas pueden clasificarse en varios tipos, como dependencias de inicio a inicio (una tarea comienza cuando otra comienza CC o SS), de inicio a final (una tarea comienza cuando otra termina CF o SF), de final a inicio (una tarea comienza cuando otra termina FC o FS) y de final a final (una tarea termina cuando otra termina FF). La elección del tipo de dependencia depende de la relación que se necesite establecer entre las tareas, y es visible en la columna *Predecesoras* de la vista Diagrama de Gantt.

Para cambiar la dependencia, se debe pulsar sobre la flecha del vínculo entre dos tareas y seleccionar el tipo de dependencia:

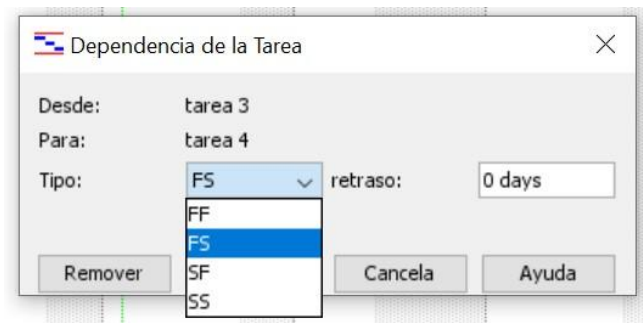


Figura 25. Cuadro de diálogo dependencia de tareas

*Tiempo de posposición y de adelanto.*

Estos dos términos se refieren al ajuste de las dependencias entre tareas en el cronograma de un proyecto, se utilizan para modificar el tiempo entre el final de una tarea predecesora y el inicio de una tarea sucesora. Cuando se necesite tener un tiempo entre dos tareas vinculadas se puede establecer un tiempo de posposición (retraso) o adelanto según la necesidad. Esto hará que la tarea sucesora se demore o adelante en el tiempo que haya establecido según se requiera. Por ejemplo: Si se tiene una tarea *Pintar pared* y le sigue la tarea *Colgar cuadro*, no se podría hacer la segunda tarea de manera inmediata ya que habría que esperar el secado de la pintura.

Como establecer el tiempo de posposición o adelanto:

1. Ir al Diagrama de Gantt y hacer doble clic justo sobre la línea que une las tareas.  
(También se puede realizar al mostrar la *Información de la Tarea*, en la ficha *Predecesoras*).
2. En el cuadro de diálogo que aparece introducir el tiempo, si es de posposición el valor debe ser positivo y si es de adelanto el valor será negativo.

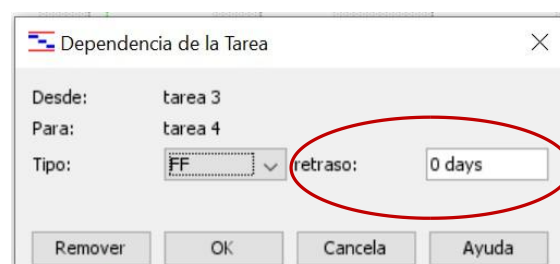


Figura 26. Posposición o adelanto

### Dividir tareas

Hay tareas que por alguna circunstancia no podrán realizarse de manera normal o continua, sino que pueden interrumpirse en su ejecución. Significa que la tarea tiene una duración que se ejecutará en diferentes etapas. Para dividir una tarea:

1. En la barra de la tarea, pulsar clic derecho y seleccionar *Dividir Tarea*
2. Observar que el cursor toma una forma diferente, arrastrar hasta la fecha en que se quiere reanudar la tarea. La tarea quedará similar a esta forma.

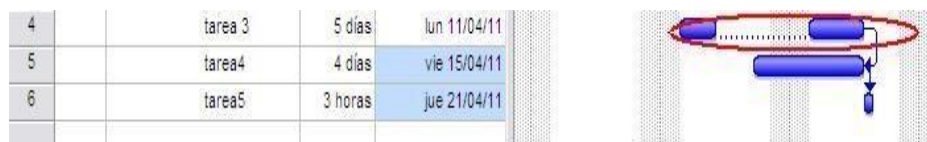


Figura 27. Ejemplo Dividir Tarea

### Tareas críticas

Una tarea crítica es aquella que si se retrasara o variaría su duración, se afectaría a la duración del proyecto, es decir haría que todo el proyecto se retrase; por lo tanto, las tareas críticas son las que tienen un impacto directo en la duración del proyecto, siendo elemental que estas tareas se gestionen cuidadosamente y se completen a tiempo.

En el Diagrama de Gantt estas tareas se distinguen de las otras porque la barra que representa a su duración suele tener un color distintivo diferente (generalmente es rojo), con el propósito de que los gestores de proyectos puedan identificarlas a simple vista.

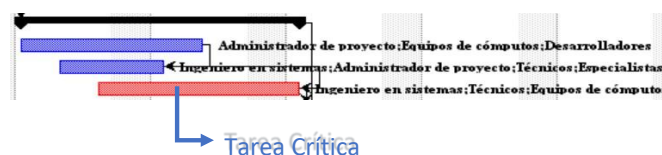


Figura 28. Ejemplo de tarea crítica

A medida que el proyecto avanza y estas tareas se van completando dejan de ser críticas. Las tareas críticas requieren un seguimiento constante para garantizar que se cumplan en el plazo establecido. El conjunto de tareas críticas forma la denominada ruta o camino crítico.

### Gestión de recursos en el proyecto

Los recursos de un proyecto son aquellos elementos necesario que se requieren para ejecutar o llevar a cabo las tareas de un proyecto. Los recursos pueden ser de tipo *Trabajo* o *Material*, y se asignan a las tareas para poder cumplir con la actividad programada; por esta razón, la gestión de recursos es importante ya que garantiza la ejecución eficiente y efectiva de un proyecto. Una asignación adecuada de los recursos a las actividades del proyecto garantiza que las actividades se completen dentro del plazo y el presupuesto previsto.

### *Tipos de recursos*

Recursos de trabajo: son los recursos humanos y de equipamiento que realizan trabajo para completar una tarea. Consumen tiempo (horas o días) para llevar a cabo la tarea. Por ejemplo: personas, alquiler maquinarias, etc.

Recursos materiales: son aquellos elementos consumibles que se adquieren y se utilizan para llevar a cabo una tarea. Por ejemplo: suministros de oficina, materiales de construcción, materia prima, etc.

Cuando se gestiona o administra un proyecto se debe considerar la disponibilidad de los recursos, las restricciones de tiempo, las limitaciones de costos y las necesidades específicas de cada tarea para optimizar la utilización de los recursos, ya que su uso eficiente es valioso para el éxito del proyecto y la entrega de los resultados esperados.

La asignación, seguimiento y gestión de los recursos materiales son tareas importantes en la planificación y ejecución de proyecto. Los recursos se enlistan en la vista *Recursos* y se asignan en el *Diagrama de Gantt*.

### Listado de Recursos



Mediante el ícono que se muestra se accede a la vista *Recursos*, que es donde se ingresan los recursos que se usarán en el proyecto.

[illegible]

### Modificar datos de recursos

Para modificar los datos de un recurso se debe situar en la celda correspondiente y realizar su modificación. También se puede hacer este proceso desde el cuadro de diálogo *Información del recurso*, haciendo doble clic sobre el recurso para acceder a la ventana:

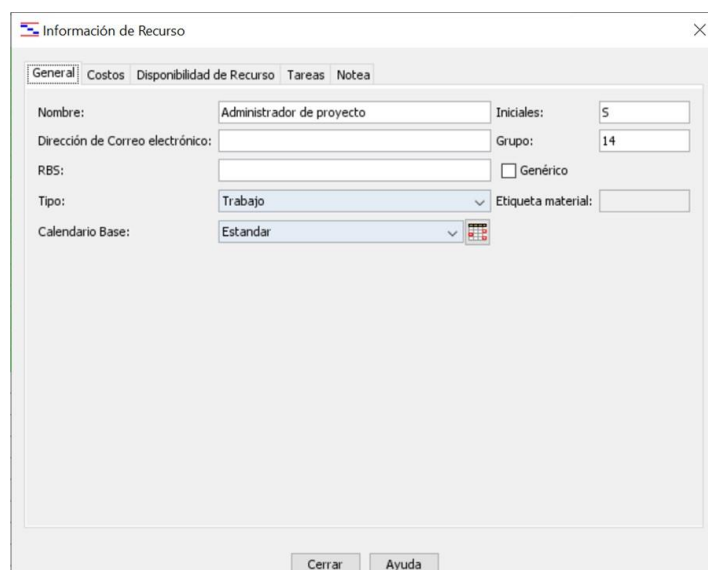


*Figura 30. Cuadro de Diálogo Información de Recurso*

### Asignar un recurso.

Una vez definidos los recursos en la Vista *Recursos* hay que asignarlos a las tareas. Esto consiste en identificar qué recursos son necesarios para llevar a cabo las tareas del proyecto.

Para lo cual en la vista Diagrama de Gantt, se debe:

1. Seleccionar la tarea a la cual se le asignará el/los recursos
2. En la barra de herramientas, seleccionar el siguiente ícono: .
3. Aparecerá el siguiente cuadro, donde se selecciona el/los recursos.

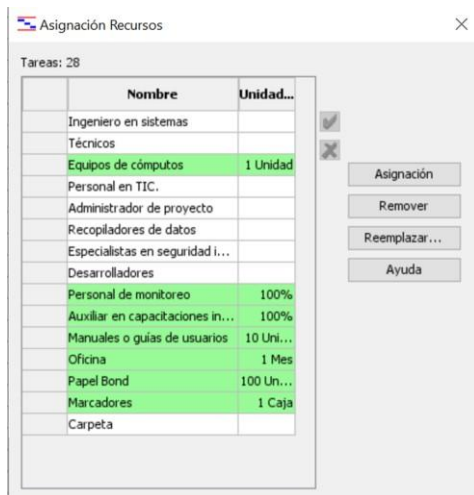


Figura 31. Cuadro de diálogo Asignar recursos

Se pulsa en el recurso deseado y se le asigna mediante el botón *Asignación*, pudiendo asignar cualquier cantidad de recursos materiales y trabajo que sean necesarios para la tarea.

Sobreasignación de Recursos: Si en la hoja de recursos algunos de estos se encuentran en color rojo, es porque están sobreasignados. Hay que tener en cuenta que la asignación de recursos debe programarse de manera que no haya conflictos de disponibilidad. Sin embargo, de presentarse este inconveniente, se debe corregir esta sobreasignación por cualquier mecanismo que se pueda aplicarse de acuerdo al proyecto, como por ejemplo: aumentando su capacidad máxima, redistribuyendo el trabajo del recurso o estableciendo restricciones de tareas en el proyecto.

Cabe resaltar que la asignación de los recursos de tipo material no influyen en la duración de las tareas, pero sí afectan al presupuesto del proyecto.

En la siguiente imagen se observan las tareas en la vista *Diagrama de Gantt* cuando se les ha asignado recursos:

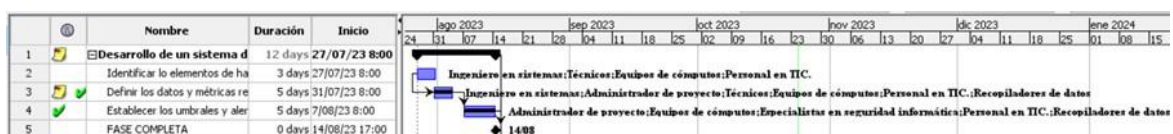


Figura 32. Tareas con recursos asignados

### El campo Unidades al Asignar recurso.

Cuando se asignan recursos se puede, desde la columna *Unidades* del cuadro de diálogo *Asignar recurso*, definir cuanto tiempo (en porcentaje) o cuantas unidades (para los recursos de tipo material) se asignarán a la tarea. Por ejemplo, en la imagen se muestra que, para la tarea 28, se ha asignado: 3 unidades de marcadores, 50 hojas de papel bond, los Técnicos laborarán al 50% en la tarea (medio tiempo), y el Ingeniero en Sistemas al 100% de su capacidad máxima en esa tarea (este último porcentaje es el predeterminado).

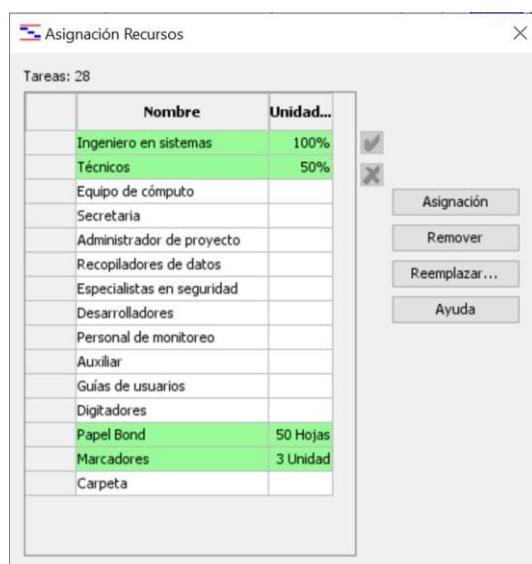


*Figura 33. Ejemplo de asignación de Recursos*

### Calendario de recursos.

De la misma forma que se definió un calendario laboral para el proyecto, se puede fijar un calendario para un recurso en particular, de manera que se puedan determinar y programar las fechas y horarios en los que un recurso estará disponible para trabajar en el proyecto. El calendario de recursos permite puntualizar el tiempo que estará disponible un recurso para trabajar en el proyecto; esto incluye días laborables, horas de trabajo por día y cualquier excepción, como días festivos o períodos de vacaciones en los que el recurso no estará disponible. Al precisar la jornada de trabajo para los recursos, se garantiza que se tiene la información concreta sobre cuando estos estarán disponibles para las tareas se programen. Para establecer un calendario para un recurso, seguir los siguientes pasos:

1. En la vista Recursos, pulsar doble clic sobre el recurso al cuál se va a modificar el calendario. Se abrirá el cuadro de diálogo *Información del recurso*.
2. En la ficha *General*, pulsar sobre *Cambiar tiempo de trabajo*.

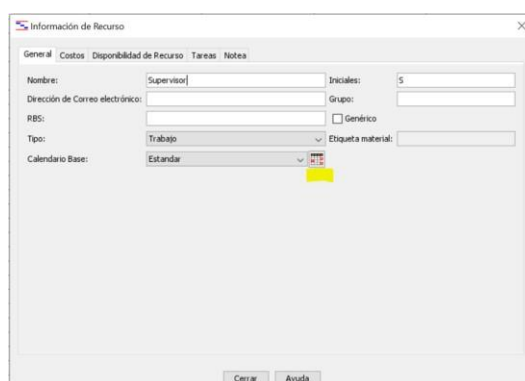


Figura 34. Cambiar Calendario Laboral de un recurso

3. Se abrirá la ventana para modificar el calendario del recurso, de acuerdo con la necesidad. Por ejemplo en la siguiente figura se ha determinado que el recurso “Administrador” no labora los martes de cada semana.

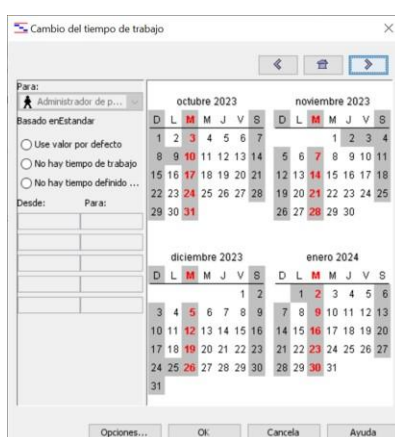


Figura 35. Opciones de calendario laboral

## Costos

Los costos son un factor muy importante en la gestión de proyectos, ya que ayudan a garantizar que las actividades programadas con sus recursos se ejecuten de manera eficiente y cumplan con las restricciones presupuestarias. Se manejan los siguientes costos:

### Determinado por Tasa

Son las tasas de pago. Para los recursos de trabajo se tiene la *Tasa Estándar* y la *Tasa*

*Sobretiempo* o de horas extras. Ambas se refieren al costo que tiene el recurso por una unidad

de tiempo. Para el caso de recursos materiales se define la *Tasa Estándar* de acuerdo con la unidad que se haya establecido en la etiqueta de material.

#### Determinado por Uso

Se determina un valor como el *Costo por Uso*, el cual se sumará al presupuesto del proyecto cada vez que se utilice o se asigne el recurso, sea material o de trabajo.

|   |  | Nombre                | RBS | Tipo     | Dirección ... | Etiqueta... | Iniciales | Grupo | Unidades Max | Tasa Estándar | Tasa sobretiempo | Costo Por Uso |
|---|--|-----------------------|-----|----------|---------------|-------------|-----------|-------|--------------|---------------|------------------|---------------|
| 1 |  | Ingeniero en sistemas |     | Trabajo  |               |             | I         |       | 1.500%       | 25,19 €/hora  | 5,00 €/hora      | 0,00 €        |
| 2 |  | Técnicos              |     | Trabajo  |               |             | T         |       | 1.600%       | 10,96 €/hora  | 0,00 €/hora      | 0,00 €        |
| 3 |  | Equipo de cómputo     |     | Material |               | Unidad      | E         |       |              | 500,00 €      |                  | 50,00 €       |

Figura 36. Ejemplo Costo por Tasa y por Uso

#### Costo Fijo

Está enfocado a las tareas que, sin importar la duración o el trabajo, se les establece un costo adicional. Para ingresar un costo fijo, realizar lo siguiente:

1. Cambiarse a la tabla *Costos* dando clic derecho sobre la esquina superior izquierda del *Diagrama de Gantt*.
2. Ingresar el valor correspondiente en la celda de la columna *Costo Fijo* que se relaciona a la tarea en la que se quiere ingresar un costo fijo.

|   | Nombre                       | Costo     | Costo Actual | Costo Remanente | Costo Fijo |
|---|------------------------------|-----------|--------------|-----------------|------------|
| 1 | Desarrollo de un sistema     | 4403,54 € | 3246,26 €    | 1157,28 €       | 0,00 €     |
| 2 | Identificar los elementos de | 1157,28 € | 0,00 €       | 1157,28 €       | 0,00 €     |
| 3 | Definir los datos y métricas | 2097,25 € | 1597,25 €    | 500,00 €        | 500,00 €   |
| 4 | Establecer los umbrales y a  | 1649,01 € | 1649,01 €    | 0,00 €          | 0,00 €     |

Figura 37. Asignación de costo fijo

#### Tasas de costo en períodos

#### diferentes.

Se refiere a la variación de los costos o tarifas estándar de los recursos a lo largo del tiempo en el que el proyecto se ejecuta. Es común que los costos de los recursos no sean constantes y puedan cambiar en diferentes períodos, influyendo en la gestión del presupuesto.

Para incluir esta variación de costos, tanto en recursos de trabajo como en materiales, en la

*Vista Recursos* se pulsa doble clic en un recurso para acceder al cuadro de diálogo

*Información del Recurso*. En la sección o pestaña de *Costos*, y en las diferentes tablas se determina la fecha efectiva en la que registrará cada variación de costo. Esta variación puede aplicarse tanto para la *Tasa Estándar*, para *Tasa Sobretiempo* o para *Costo por Uso*, como se muestra en el siguiente ejemplo:

| Fecha efectiva | Tasa Estándar | Tasa sobretiempo | Costo Por Uso |
|----------------|---------------|------------------|---------------|
| 01/01/79 0:00  | \$800.00/mes  | \$0.00/hora      | \$0.00        |
| 01/10/79 0:00  | \$850.00/mes  | \$0.00/hora      | \$0.00        |

Figura 38. Tasas de costos en períodos diferentes

## Vistas para planificación y monitoreo

Esta herramienta de gestión de proyectos ofrece diversas vistas para planificar, monitorear y controlar un proyecto, la elección de una vista depende de la necesidades y de cómo se quiera visualizar y gestionar un proyecto, teniendo la flexibilidad de personalizar y cambiar entre estas vistas para optimizar la gestión. A continuación se muestran las vistas disponibles:



Figura 39. Opciones de Vistas

- Vista GANTT, como se ha visto anteriormente se refiere al Diagrama de Gantt, es la vista predeterminada al ingresar a la aplicación. Esta es básicamente una herramienta gráfica utilizada en la gestión de proyectos para planificar, programar y hacer un seguimiento de las tareas y actividades a lo largo del tiempo. Este tipo de diagrama muestra las tareas del proyecto en una línea de tiempo, lo que facilita la visualización de la secuencia de actividades, las dependencias entre tareas y el progreso o avance del proyecto, contiene una jerarquía de tareas a la izquierda y un diagrama de tiempo en la derecha, expresando la duración de las tareas mediante barras.
- Vista RED, conocida también como Diagrama PERT, muestra las dependencias entre tareas usando un gráfico por nodos, se utiliza para representar y analizar la secuencia y duración de las tareas así como las dependencias entre ellas, ayuda a comprender la estructura del proyecto y, a identificar las áreas críticas en las que se debe centrar la atención para garantizar que el proyecto se complete en el tiempo planificado.
- Vista RECURSOS, igualmente como se indicó, esta vista a través de una hoja permite visualizar y gestionar los recursos asignados a un proyecto, contiene un listado de recursos de trabajo y materiales disponibles para ser usados en el proyecto.
- Vista WBS, muestra una jerarquía de tareas del proyecto. Esta vista, Estructura de Desglose del Trabajo (WBS, por sus siglas en inglés), permite visualizar y gestionar la graduación de tareas y actividades de un proyecto de manera estructurada.
- Vista RBS, La vista de Estructura de Responsabilidad del Proyecto (RBS, por sus siglas en inglés) es una funcionalidad que permite visualizar y gestionar la jerarquía de responsabilidades y roles dentro de un proyecto. A través de esta vista, es posible identificar quiénes son los responsables de las diferentes partes del proyecto y cómo se distribuyen las responsabilidades en la organización.

- Vista INFORMES, consolida información sobre el proyecto, como fecha de inicio y terminación, duración, información sobre trabajo y costos especificando el avance y lo restante, equivalente a las estadísticas de un proyecto.
- Vista DETALLE USO TAREAS, también denominada Vista de Uso de Tareas, proporciona información detallada sobre cómo se están utilizando los recursos de trabajo y materiales en las tareas del proyecto. Es de ayuda para verificar que recursos se han asignado eficazmente a las tareas y para visualizar si la carga de trabajo se ha distribuido de manera equitativa.
- Vista DETALLE USO RECURSOS, se observa mediante esta vista la asignación y gestión de los recursos de un proyecto, en que tareas o actividades están involucrados cada uno de los recursos que se han considerado para la ejecución del proyecto.

## UNIDAD 3: LA HOJA ELECTRÓNICA COMO HERRAMIENTA DE DESARROLLO

La hoja electrónica de cálculo es de utilidad en todas las áreas de una organización principalmente para el almacenamiento, procesamiento y análisis de datos. En este apartado se tratará una de las ventajas del programa Excel que posee para facilitar la elaboración de macros (automatización de un proceso) y el acceso a su programación, así como a crear formularios y programarlos a través de controles.

Las macros en Excel se refieren a secuencias de comandos o instrucciones programadas que automatizan tareas repetitivas en una hoja de cálculo. Bajo esta premisa se detallan a continuación conceptos básicos que permitirán el desarrollo de aplicaciones para automatizar varios casos propuestos mediante formularios creados en el entorno de desarrollo de macros, los cuales permiten crear una interfaz o entorno amigable para la entrada de datos y salida de información.

En el programa Excel, las diferentes opciones de la pestaña o ficha *Desarrollador* (llamada también *Programador*) se utilizan para la creación de macros y la configuración de controles dentro de un formulario (Díaz, 2015). Para activar esta ficha se deben seguir los siguientes pasos:

Archivo → Opciones → Personalizar cinta de opciones → Desarrollador (activar)

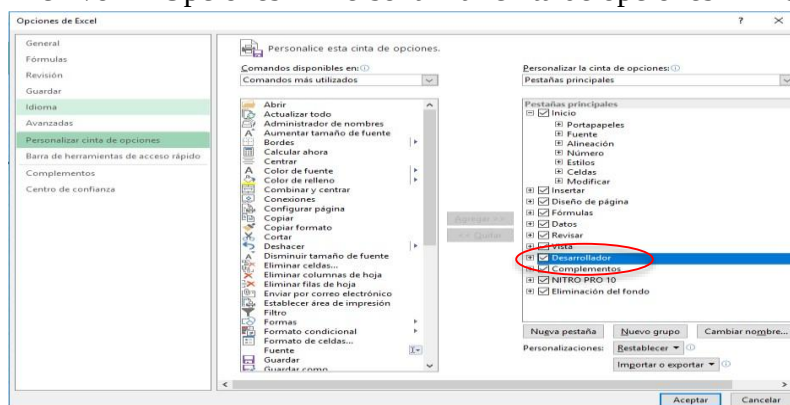


Figura 40 Activación de Ficha DESARROLLADOR

En la ficha *Desarrollador*, se encuentran las opciones para trabajar en la elaboración de macros, las mismas que se muestran en la figura que sigue:

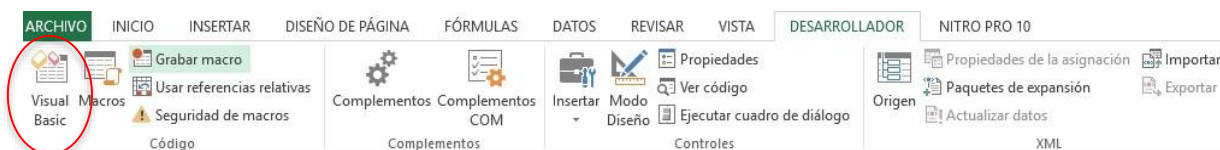


Figura 41 Opciones de la Ficha Desarrollador

Mediante la opción *Visual Basic* disponible en el grupo *Código*, se accede al *Editor de Visual Basic*, el cual permite escribir código en el lenguaje de programación **Visual Basic for Applications (VBA)**, traducido como lenguaje visual para aplicaciones (Camaño,2019), esta es una herramienta que permite a los usuarios automatizar tareas, a través de la creación de macros o secuencias de comandos.

**IMPORTANTE:** Cuando se grabe el archivo de Excel que contiene una macro o un formulario, se elige la opción **GUARDAR COMO Libro de Excel habilitado para macros (\*.xlsm)**, tal como consta en el espacio resaltado de la siguiente imagen:

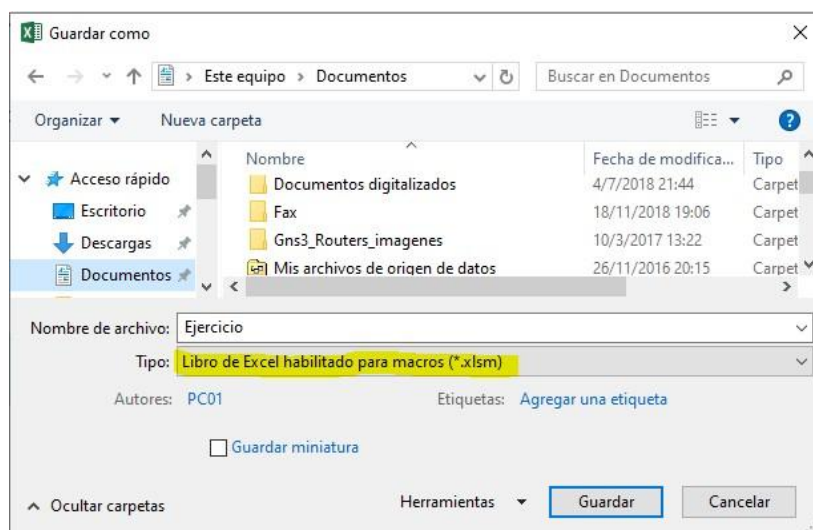


Figura 42 Cómo Guardar una macro

Cuando se ha guardado el archivo, en el ícono del documento se muestra un signo de admiración, así:



## Conceptos básicos de programación

Un lenguaje de programación es utilizado para escribir un programa o un conjunto de instrucciones. Las operaciones que conducen a expresar un algoritmo en forma de programa se llama programación.

Para realizar un programa se debe tener en primer lugar un algoritmo adecuado. El algoritmo luego se ha de expresar en un formato al que se denomina programa.

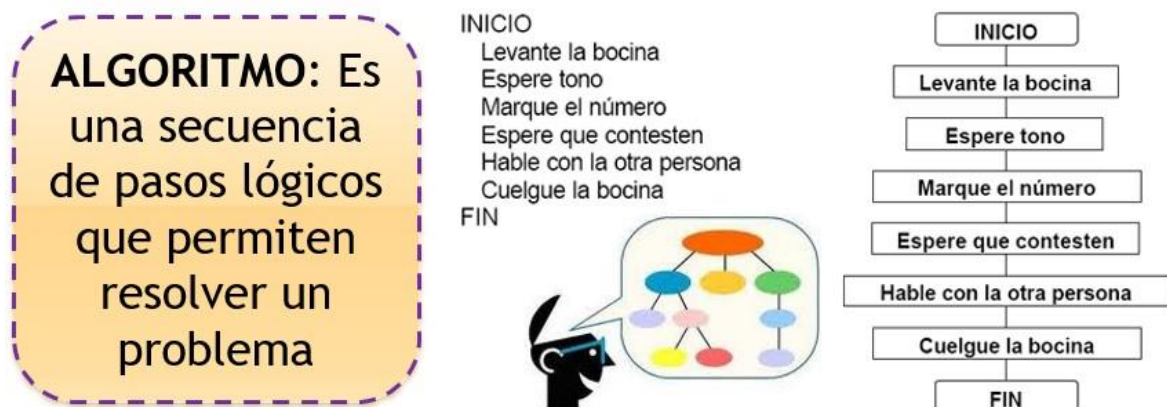


Figura 43 ¿Qué es un Algoritmo?

Un algoritmo debe tener las siguientes características:

- Tener principio.
- Ser simples, claros, precisos, exactos.
- Tener un orden lógico.
- Tener fin.

En la elaboración de un algoritmo es importante describir tres partes:

- Entrada: los datos que se necesitan conocer para resolver un problema
- Proceso: el conjunto de acciones que se realizará, para obtener un resultado
- Salida: la información obtenida o los resultados

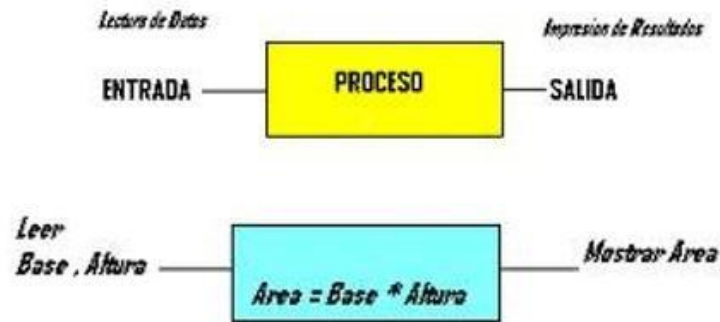


Figura 44 Partes de la elaboración de un algoritmo

La figura anterior muestra como ejemplo el cálculo del área de un rectángulo:

Entrada: Se necesita saber o conocer la Base y la Altura

Proceso: Para calcular el área se debe multiplicar la base por la altura Salida:

El resultado es el Área calculada.

#### Dato

En la elaboración de algoritmos se requerirá la definición de datos. Un dato es la representación de un hecho, de un evento o de un elemento. Por ejemplo: un empleado puede ser representado por los siguientes datos: nombre, cédula de identidad, cargo, edad, género, etc.

Puede decirse que los datos son todos aquellos objetos que se van a procesar. Los datos pueden ser de diferentes tipo, por ejemplo: Entero, Real, Carácter, Lógico, entre otros.

Los datos pueden ser:

- CONSTANTES: Son datos que no cambian durante la ejecución de un programa.  
Ejemplo: 12% (IVA), 5,00 (Valor Hora), 450,00 (Sueldo Básico).
- VARIABLES: Son datos que cambian durante la ejecución de un programa. Las variables se utilizan para representar un dato. Ejemplo: Nombre, Sueldo, Teléfono, etc.

Una variable guarda un dato para luego poder usarlo en la ejecución de un programa.

Para asignar el nombre de una variable se debe considerar las siguientes sugerencias:

- Su nombre NO debe contener espacios en blanco, ni ningún símbolo especial (;,\$'!"

\* / %), excepto el guion bajo: \_

- Su nombre NO puede tener un dígito como primer caracter.

□ Se recomienda que el nombre que se le asigne a una variable guarde relación con su contenido, no se pueden usar palabras reservadas de Excel o de VBA.

□ La longitud del nombre de una variable no puede exceder los 250 caracteres.

Ejemplos: Sueldo, Cargo, Valor\_Total. En la ejecución de la macro tomarían valores como por ejemplo: *Sueldo = 1200; Cargo= “Asistente”; Valor\_Total=900*

#### Tipos de datos

Byte: Se utiliza para guardar números positivos enteros desde 0 a 255.

Boolean: Permite almacenar un valor lógico: Verdadero (*True*) o Falso (*False*)

Currency: Para cálculos en formato moneda, guarda desde -922.334.203.685.477,5808 hasta 922.337.203.685.477,5807

Single: Sirve para almacenar números fraccionarios periódicos desde -3,4028235E+38 a 1,401298E-45 para números negativos y 1,401298E-45 a 3,4028235E+38 para números positivos.

Double: Este tipo sirve para almacenar números fraccionarios con mucha más capacidad, desde -1,79769313486231570E+308 a -4,94065645841246544E-324 para los valores negativos y desde 4,94065645841246544E-324 a 1,79769313486231570E+308 para los valores positivos.

Date: Ideal para guardar fechas y horas como números de serie, lo cual facilita aplicar funciones de cálculo de fecha y hora.

Integer: Sirve para almacenar números entre un rango desde -32.768 hasta 32.767.

Long: Para guardar números enteros desde -2.147.483.648 hasta 2.147.483.647.

String: Permite guardar cadenas de caracteres o de texto; un carácter es cualquier letra, número o símbolo.

Variant: Es un tipo de variable especial que puede almacenar cualquier tipo de dato (numéricos, textos o fechas). Si una variable no se declara como uno de los tipos mencionados anteriormente, se supone que es de tipo *Variant*.

Los principales tipos de variables descritos se resumen en la siguiente figura:

| Tipo     | Descripción                                      | Rango                                                                                        |
|----------|--------------------------------------------------|----------------------------------------------------------------------------------------------|
| Boolean  | Binario                                          | True o False                                                                                 |
| Byte     | Entero Corto                                     | 0 a 255                                                                                      |
| Integer  | Entero (2 bytes)                                 | -32768 a 32767                                                                               |
| Long     | Entero Largo (4 bytes)                           | -2147483648 a 2147483647                                                                     |
| Single   | Real simple precisión (4 bytes)                  | -3.40E+38 a 3.40E+38                                                                         |
| Double   | Real doble precisión (8 bytes)                   | -1.79D+308 a 1.79D+308                                                                       |
| Currency | Número con punto decimal fijo (8 bytes)          | -9.22E+14 a 9.22E+14                                                                         |
| String   | String                                           | 0 a 65500 caracteres                                                                         |
| Date     | Fecha (8 bytes)                                  | 1 de enero de 100 a 31 de diciembre de 9999. Indica también la hora desde 0:00:00 a 23:59:59 |
| Variant  | Fecha/hora. Números enteros, reales, caracteres. |                                                                                              |

Figura

#### 45 Tipos de variables

### Operadores

Los operadores que se pueden incluir en la elaboración de un programa son, entre otros:

Operadores Aritméticos: Sirven para realizar operaciones aritméticas entre datos constantes o variables. El resultado de una operación aritmética es siempre un número.

| Operadores Aritméticos |                  |              |           |
|------------------------|------------------|--------------|-----------|
| Operador Aritmético    | Operación        | Ejemplo      | Resultado |
| ** ó ^                 | Potencia         | 3**2         | 9         |
| *                      | Multiplicación   | 5.25*3       | 15.75     |
| /                      | División         | 17/4         | 4.25      |
| +                      | Suma             | 58+79        | 137       |
| -                      | Resta            | 25.96 -11.25 | 14.71     |
| Mod                    | Módulo (residuo) | 21 mod 2     | 1         |

Figura 46 Operadores Aritméticos - Ejemplos

Operadores Relacionales o de Comparación: Permiten comparar entre dos operandos o valores.

El resultado de una expresión con operadores relacionales es VERDADERO o FALSO.

| Operadores Relacionales |                   |                 |           |
|-------------------------|-------------------|-----------------|-----------|
| Operador                | Operación         | Ejemplo         | Resultado |
| =                       | Igual que         | "lalo" = "lola" | Falso     |
| < >                     | Diferente a       | 'x' <> 'y'      | Verdadero |
| <                       | Menor que         | 14 < 25         | Verdadero |
| >                       | Mayor que         | 35 > 50         | Falso     |
| < =                     | Menor o igual que | 5 <= 6          | Verdadero |
| > =                     | Mayor o igual que | 7 >= 7          | Verdadero |

Figura 47 Operadores Relacionales o de Comparación - Ejemplos

Operadores Lógicos: Son operadores que permiten formular condiciones complejas a partir de condiciones simples.

|           |                                                                                |
|-----------|--------------------------------------------------------------------------------|
| <b>Y</b>  | Devuelve VERDADERO, si <b>todos</b> los valores que se comparan son verdaderos |
| <b>O</b>  | Devuelve VERDADERO, si <b>uno</b> de los valores que se comparan es verdadero  |
| <b>NO</b> | Da como resultado un valor opuesto al que tenga el operando                    |

Figura 48 Operadores Lógicos

## Diseño de formularios. Objetos y sus propiedades básicas

Al acceder mediante la opción *Visual Basic* disponible en el grupo *Código* de la pestaña *Desarrollador*, o mediante la combinación de teclas **ALT+F11**, se muestra el entorno para el diseño de formularios, en este escenario que es diferente a la hoja de cálculo normal, se pueden distinguir los siguientes elementos:

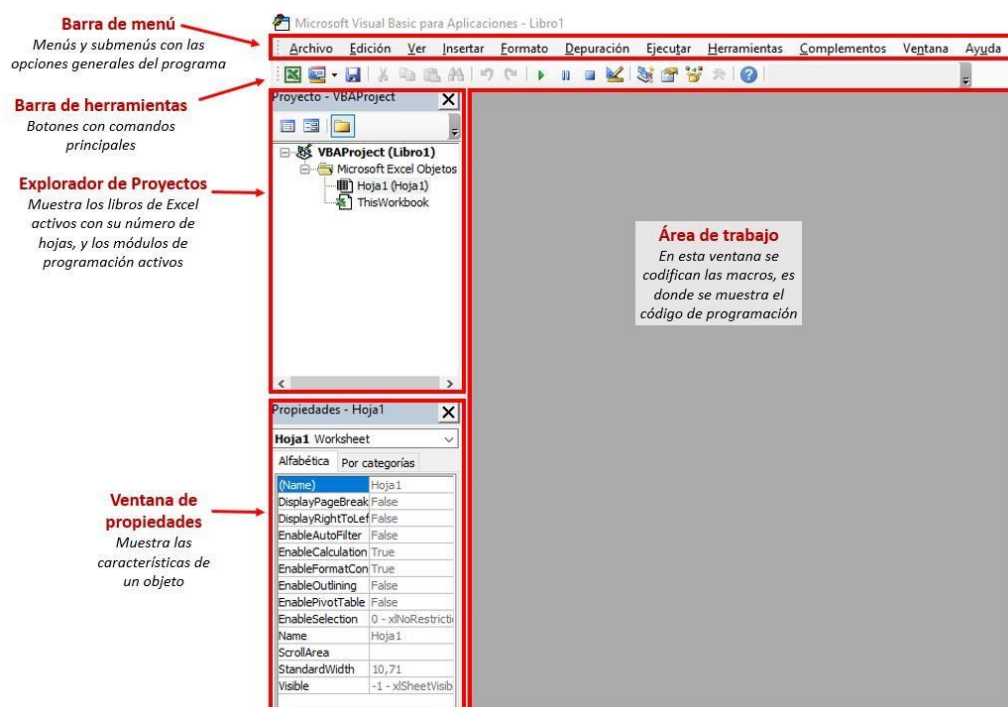


Figura 49 Entorno de Desarrollo de Visual Basic

Tanto el explorador de proyectos como la ventana de propiedades también pueden ser activadas desde la opción **VER** disponible en la *Barra de menú*.

El entorno de desarrollo de *Visual Basic* dispone de muchas herramientas que facilitan la programación de una macro o la automatización de un proceso.

Lo que en la figura anterior se muestra, se detalla a continuación:

La barra de menú y la de herramientas: contienen las opciones disponibles para trabajar con el editor VBA.

El explorador de proyectos: permite visualizar los proyectos y módulos abiertos; en la parte superior se encuentran los botones que permiten alternar entre la vista objeto y la vista código.

La ventana de propiedades: muestra las propiedades o características que poseen cualquiera de los objetos que se encuentren establecidas en el área de trabajo.

Con relación a la tipografía, cuando se trabaja con el editor de VBA, las instrucciones aparecen en color **azul**; cuando se hace referencia a objetos métodos y propiedades aparecerán en color **negro**, y los comentarios se muestran en color **verde**.

## El Formulario y sus propiedades

Un formulario es una ventana que puede contener objetos o controles, y también su respectivo código. Para insertar en formulario se utiliza la opción *Insertar* y luego *UserForm*:

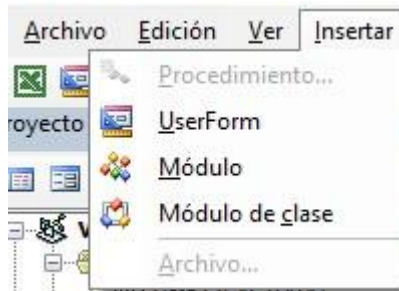


Figura 50 Insertar Formulario

Las propiedades o características de los objetos permiten ir configurando su aspecto o su comportamiento. Por ejemplo, al pulsar sobre el formulario se accede a las propiedades de este que son visibles desde la ventana *Propiedades*, como las siguientes:

*Name*: Nombre del formulario, útil para su uso en el código de programación.

*BackColor*: Color de fondo que tendrá el formulario.

*Caption*: Es el texto que aparece en el título del formulario

*Font*: Para establecer el tipo de Fuente, Estilo de Fuente y Tamaño de letra en el formulario

*ForeColor*: Color del texto del formulario.

## Estructura secuencial de programación

Como se mencionó, la elaboración de un algoritmo debe describir tres partes:



- Entrada. - Son los datos que van a procesarse
- Proceso. - Las operaciones o cálculos que se harán con los datos para obtener un resultado
- Salida. - La información o los resultados que se obtienen luego del proceso.

Una estructura secuencial de programación se caracteriza porque una acción, instrucción o actividad se ejecuta siempre detrás de otra, no hay condiciones de por medio. El orden de ejecución será aquel en el que se hayan colocado las instrucciones dentro de un programa. En una estructura secuencial de programación siempre se encuentra definida la entrada, el proceso y la salida.

A fin de facilitar el aprendizaje del tema en estudio, en los ejercicios que se exponen a continuación, se realizará primero el algoritmo (identificando Entrada, Proceso y Salida) con su representación gráfica mediante un diagrama de flujo, y luego el diseño del formulario con la codificación mediante el uso del lenguaje VBA.

### **Ejemplo 1:**

*Determinar el sueldo semanal de un empleado, conociendo el número de horas trabajadas y valor de cada hora.*

- **Entrada.** - Los datos a procesarse se identificarán por medio de variables, en este caso NUMERO\_HORAS y VALOR\_HORA

- **Proceso.** - Para determinar el sueldo semanal se multiplican los dos valores, este cálculo se le asignaría a una variable:

$SUELDO\_SEMANAL = NUMERO\_HORAS * VALOR\_HORA$

- **Salida.** - El resultado luego del proceso es el SUELDO\_SEMANAL

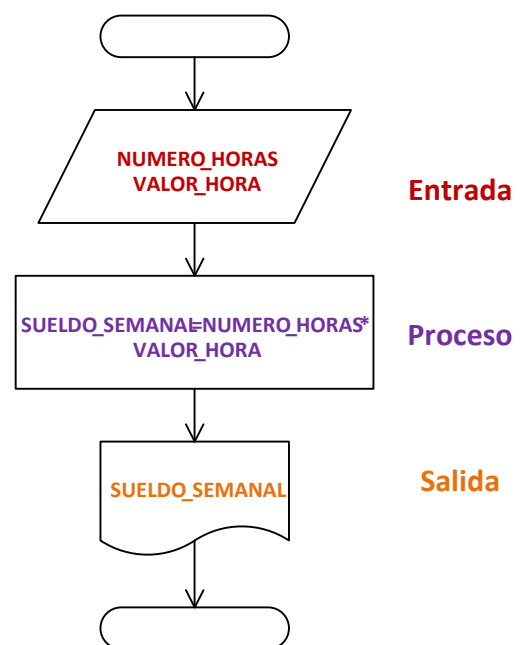


Figura 51 Ejemplo de estructura secuencial

A continuación, se desarrolla este ejercicio mediante un Formulario en una hoja electrónica:

Se recomienda ir realizando este primer ejercicio que está detallado paso a paso  
Para simplificar la solución de este problema, es necesario precisar y analizar lo siguiente:

**Entrada:** Se necesita conocer el número de horas trabajadas y valor de cada hora

**Proceso:** Para calcular el Sueldo se debe multiplicar el número de horas semanales trabajadas por el valor de cada hora

**Salida:** El resultado es el Sueldo semanal calculado

Luego, para ir al entorno de programación *Visual Basic* de *Excel*, se debe activar la ficha

*Desarrollador* (si aún no se ha activado antes), e ingresar mediante la opción *Visual Basic*.

El siguiente paso será insertar un formulario, para lo cual se pulsa *Insertar* y luego

*User Form*. El entorno es el siguiente:

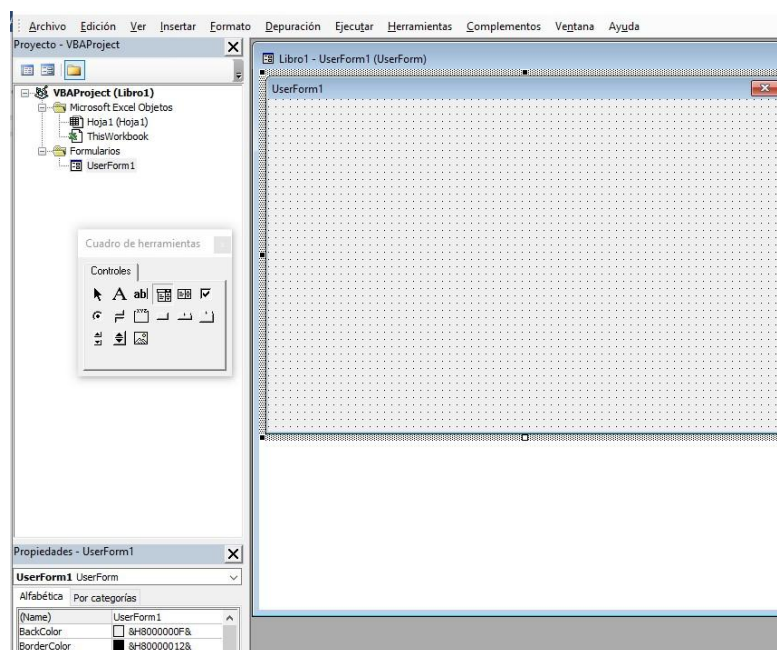


Figura 52 Formulario insertado

### Uso del control Label

En el cuadro de herramientas que aparece en un recuadro, se dispone de una serie de controles, por ejemplo: para colocar los títulos o mensajes en el formulario, se utiliza el control **ETIQUETA** (o **Label**) que se encuentra disponible en el cuadro de herramientas en el ícono  :



Figura 53 Cuadro de Herramientas

Se selecciona el control y se lo dibuja en el formulario. Una vez realizado esta acción, hay que configurar el control mediante las propiedades. Al pulsar sobre la etiqueta dibujada en el formulario, se puede configurar la misma en el cuadro de propiedades:

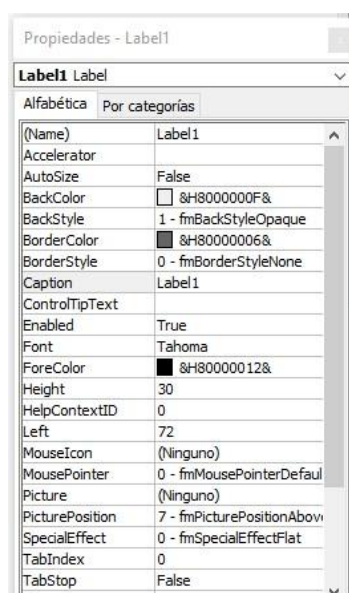


Figura 54 Cuadro de

*Propiedades* Las principales opciones que configurar son:

**Name:** se recomienda nombrar a los controles en función del tipo de control, aplicando un prefijo así: **lbl** para etiquetas, **btn** para botones de comando (*cmd* utilizan también algunos autores), **txt** para cuadros de texto, etc. En el caso de la etiqueta sería `lblTitulo`.

El nombre de un control es una sola palabra, NO puede contener espacios; es muy importante definir el nombre de un control, ya que dentro de la programación solo se puede hacer mención al control a través del nombre dado.

**Caption:** Es el título que va a contener el control, es lo que se va a mostrar en el formulario.

Ejemplo: Sueldo Semanal.

Existen otras propiedades, muchas de ellas relacionadas con el formato, como son las siguientes:

**Font:** Permite seleccionar la Fuente, Estilo de Fuente y Tamaño. Se accede desde el botón  que está en la misma propiedad, en el extremo derecho.

**ForeColor:** Se utiliza para definir un color de fuente, se puede escoger entre las alternativas que se presentan en la opción.

**TextAlign:** Sirve para especificar una alineación del texto en el control, que puede ser a la izquierda, al centro o a la derecha.

Luego de aplicar las propiedades básicas antes mencionadas, el cuadro de propiedades de la etiqueta insertada quedaría con este aspecto:

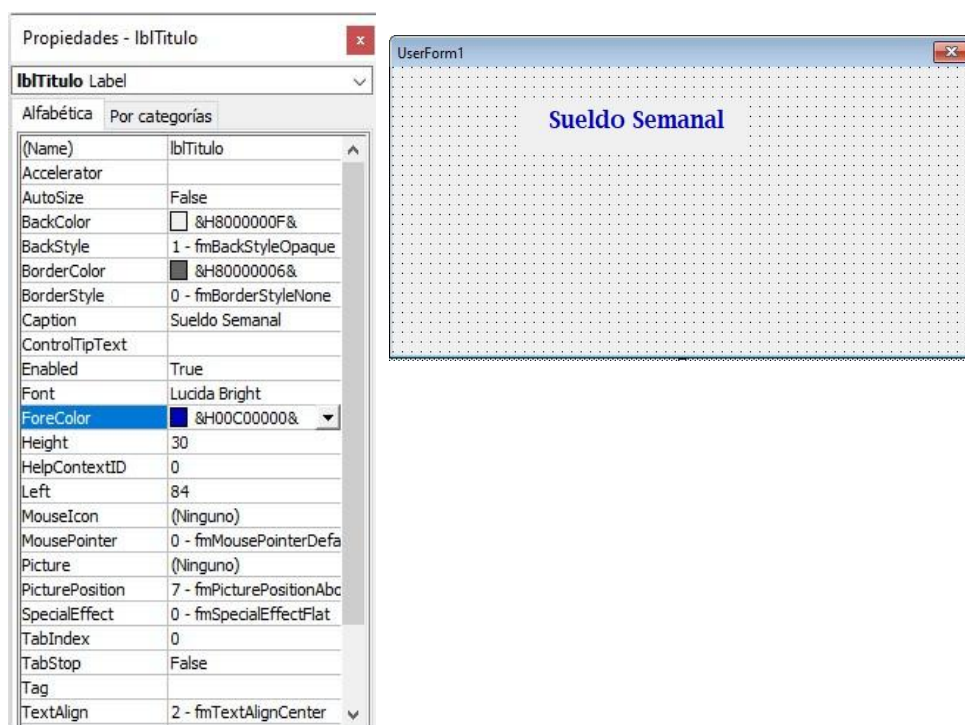


Figura 55 Ejemplo de propiedades de una etiqueta

#### Uso del control TextBox

Luego, como se estableció que como dato de *Entrada* se necesita conocer el número de horas trabajadas y valor de cada hora, se debe insertar en el formulario dos controles denominados **CUADRO DE TEXTO (o TextBox)**, para lo cual se debe seleccionar del cuadro de herramienta el ícono  y dibujarlo en el formulario.

De manera similar a lo anterior, en la ventana de propiedades de ese control se le coloca en la propiedad **Name**: txtNumeroHoras.

Seguidamente hay que insertar un segundo cuadro de texto y colocar en la propiedad **Name**: txtValorHora. A continuación, se muestran las propiedades para cada uno de los cuadros de texto insertados:

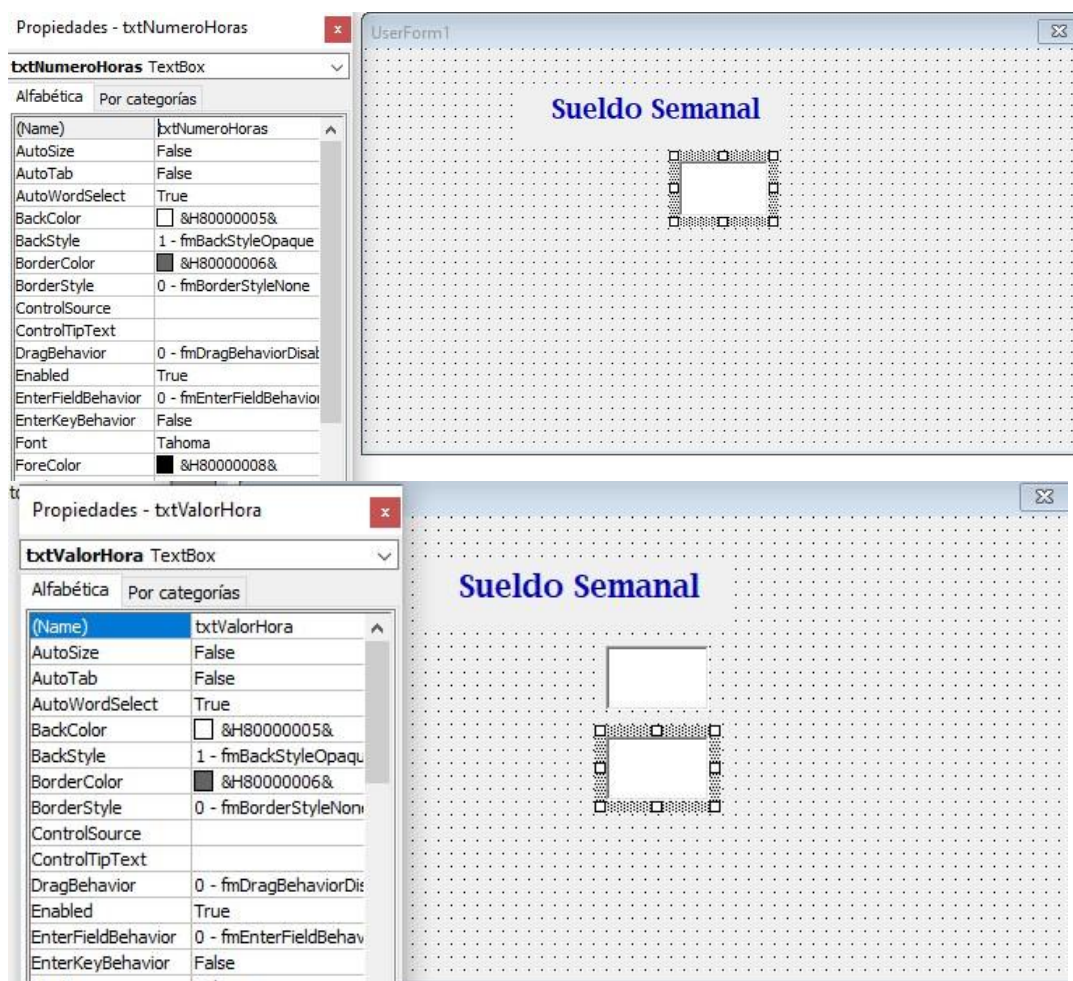


Figura 56 Cuadros de texto y sus propiedades en un formulario

Para complementar el aspecto de los formularios, se insertarán dos etiquetas que permitirán rotular los cuadros de texto ingresados; en cada etiqueta se pueden aplicar las propiedades antes descritas para mejorar la presentación. El formulario quedaría así:

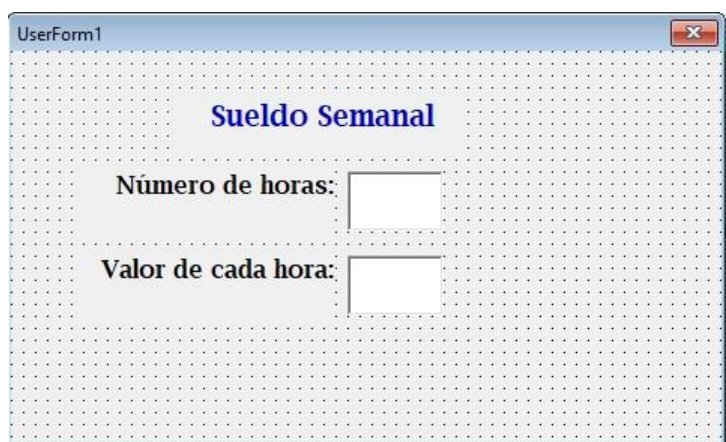


Figura 57 Formulario con etiquetas y cuadros de texto

Adicionalmente se colocará una etiqueta adicional que servirá para mostrar el resultado, a la cual se han establecido las siguientes propiedades:

**Name:** lblResultado

**BackColor:** &H00C0FFFF& (es un color de fondo seleccionado de la paleta de colores)

**Caption:** Vacío

**Font:** Arial Rounded MT, Negrita,

Tamaño 12 El formulario se presenta así:

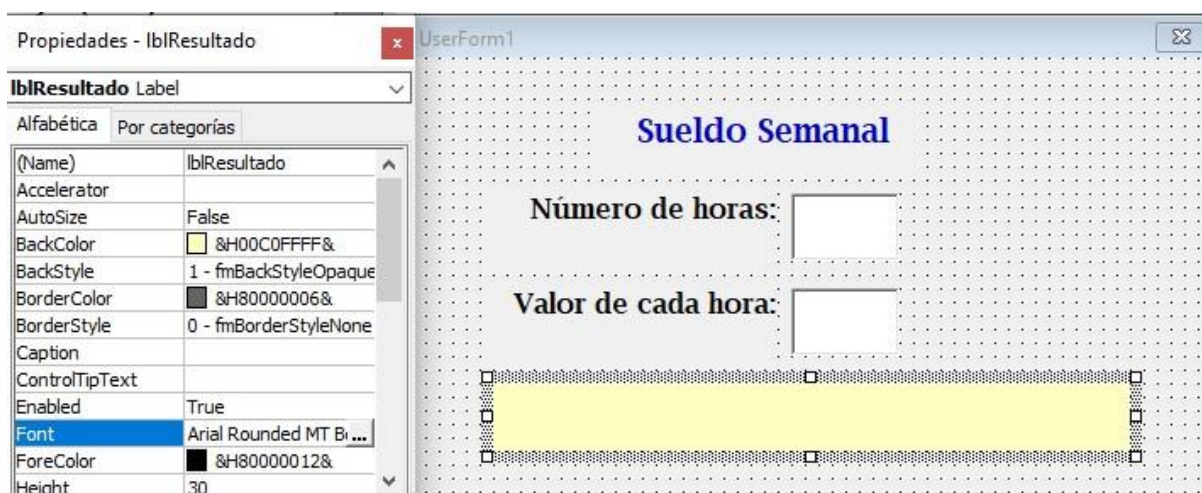


Figura 58 Formulario con etiqueta para mostrar resultados

Uso del control CommandButton

Lo siguiente será incluir un **BOTÓN DE COMANDO** (o *CommandButton*) del cuadro de

herramientas, el cual permitirá ejecutar una acción, en este caso será efectuar el cálculo.

Al botón de comando se le configurarán las siguientes propiedades básicas:

**Name:** btnCalcular

**Caption:** CALCULAR

**Font:** Tahoma, Negrita, Tamaño 10

El aspecto del formulario con las propiedades del botón de comando quedaría de la siguiente manera:

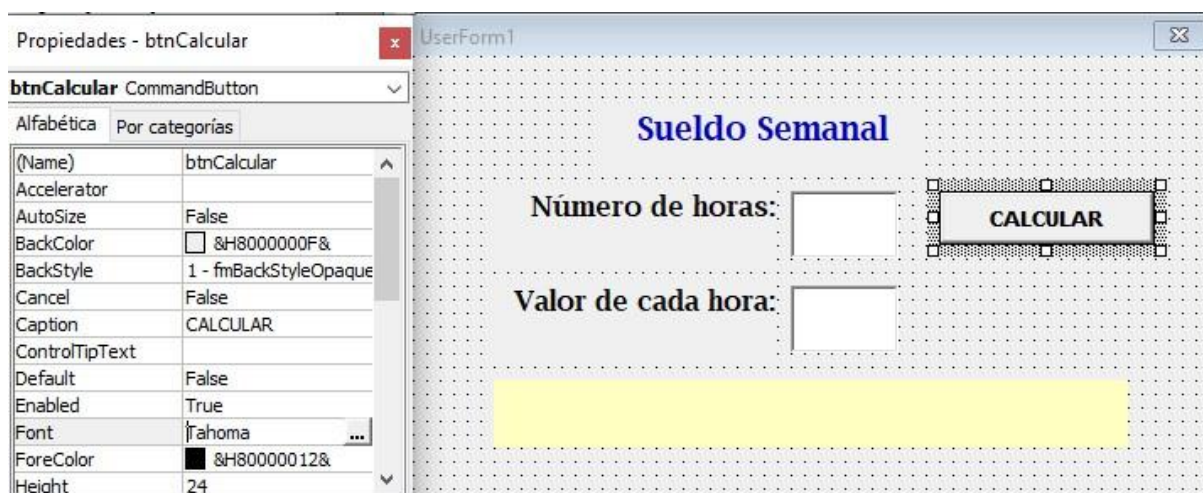


Figura 59 Formulario con Botón de Acción

Para realizar la acción del botón o **programar** el botón CALCULAR, se pulsa doble clic en el botón, luego de lo cual se abre la ventana de código:

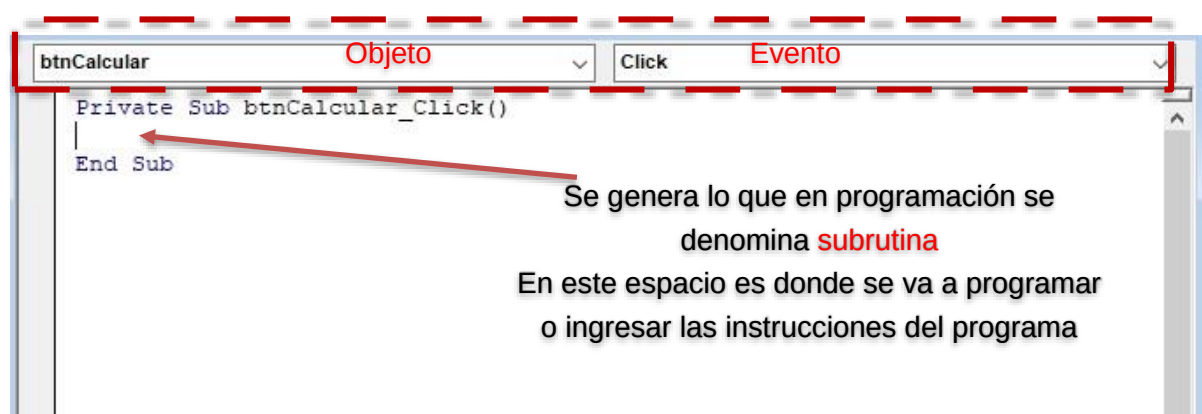


Figura 60 Ventana de código

Siempre, al pulsar doble clic sobre cada control insertado en el formulario, se accede a la ventana de código donde se programan los eventos que ocurrirán al ejecutar alguna acción sobre el control.

En la figura anterior se puede distinguir el **objeto** Botón de Comando llamado *btnCalcular* y el **evento** CLICK; lo que se genera se denomina **SUBROUTINA** o **PROCEDIMIENTO** que

quiere decir que lo que se programe en el espacio permitirá resolver una tarea específica. El código o las letras en azul son palabras reservadas del lenguaje de programación.

Es importante indicar que en cada línea se programa una instrucción.

Para alternar la vista entre la ventana de código y la de objeto, se pulsa en los botones



que se encuentran en la parte superior del explorador de proyectos (*Ver Figura Entorno de Desarrollo de Visual Basic*).

#### Comentarios en la programación

En el código se pueden insertar **COMENTARIOS**, lo cual básicamente son

explicaciones o anotaciones de la secuencia de instrucciones y no afectan en la

ejecución del programa. Para insertar un comentario se coloca al inicio de esta una

comilla simple, y el texto de la línea aparecerá en color verde. Por ejemplo:

```

Private Sub btnCalcular_Click()
    'Entrada o lectura de datos

    'Proceso o cálculos

    'Salida o impresión de resultados
    |
End Sub

```

*Figura 61 Subrutina con comentarios*

Los comentarios son importantes ya que permiten documentar con aclaraciones y explicar el código de programación, siendo de gran utilidad para revisiones futuras.

Para facilitar la solución de este ejercicio, se ha colocado comentarios en el código, los cuales permitirán ir desarrollando en cada espacio las partes que se deben describir en todo algoritmo, como se mencionó anteriormente e ir explicando el proceso.

#### Programación de Entrada o lectura de datos

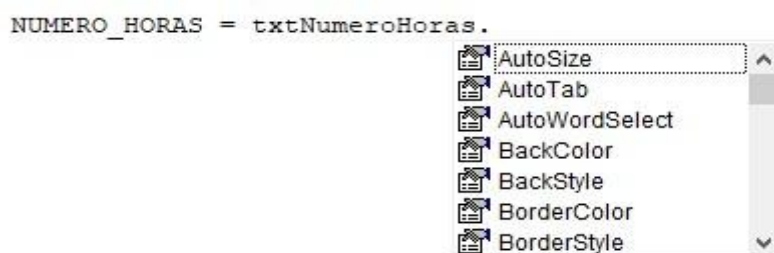
Con relación a la entrada o lectura de datos, se procede de la siguiente manera:

- Recordemos que los dos cuadros de texto en el formulario se denominan:

*txtNumeroHoras* y *txtValorHora*, estos controles contienen los datos que el usuario digita cuando se ejecuta el programa. Por lo tanto, hacemos uso de 2 variables cualquiera para asignar estos datos, por ejemplo:

- En la variable `NUMERO_HORAS` asignamos el texto que contiene el control *txtNumeroHoras*; y en la variable `VALOR_HORA` asignamos el dato o texto de *txtValorHora*.

Para esta acción al terminar de escribir el nombre del control se digita un punto (.) e inmediatamente aparecen una lista de propiedades y de métodos, de las mismas se escoge en este caso, la propiedad *TEXT* (que hace referencia o corresponde al texto que se encuentra almacenado en el control), tal como se muestra a continuación:



Es decir que para ambos casos queda:

```
NUMERO_HORAS = txtNumeroHoras.Text
VALOR_HORA = txtValorHora.Text
```

### Función VAL

Es necesario que, en cada una de las lecturas de datos, se haga uso de la función **VAL**, la cual permite convertir el texto ingresado en un valor numérico, quedando así:

```
Private Sub btnCalcular_Click()
'Entrada o lectura de datos
NUMERO_HORAS = Val(txtNumeroHoras.Text)
VALOR_HORA = Val(txtValorHora.Text)

End Sub
```

### Programación del Proceso

Respecto a lo que corresponde a proceso o cálculos, se procede de la siguiente manera:

- En una nueva variable a la que denominaremos SUELDO\_SEMANAL, se le asigna el resultado de multiplicar los valores de las variables NUMERO\_HORAS y VALOR\_HORA, quedando de la siguiente manera:

```
SUELDO_SEMANAL = NUMERO_HORAS * VALOR_HORA
```

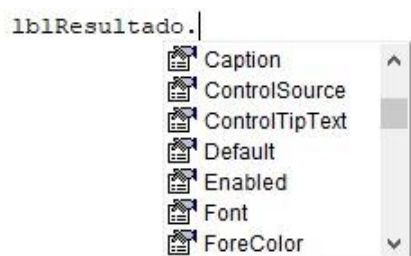
Esta fórmula permite realizar el cálculo que se analizó en la parte inicial de este ejercicio, donde se estableció que para calcular el Sueldo semanal que percibirá el empleado, se debe obtener el producto del número de horas semanales trabajadas con el valor de pago de cada hora.

### Programación de la Salida o Mostrar Resultados

Para la salida o impresión de resultados, se procede de la siguiente manera:

- En la etiqueta que se insertó en el formulario, se mostrará la respuesta. A esta etiqueta se la denominó *lblResultado* (Ver Figura Formulario con etiqueta para mostrar resultados). De igual forma que el control Cuadro de Texto, una Etiqueta también tiene una serie de propiedades y métodos, en este caso se utilizará la propiedad

**CAPTION:**



La propiedad **Caption** es una de las más importantes de un control *label*, ya que permite establecer el texto que se mostrará en el control.

Por lo tanto, quedaría:

```
lblResultado.Caption = SUELDO_SEMANAL
```

Si se quiere “mejorar” la presentación del resultado en el formulario se puede añadir un texto adicional en la etiqueta, para el efecto se utiliza el símbolo **&** que permite **concatenar** o unir textos (que deben ir entre comillas siempre) y variables, quedando el código de la siguiente manera:

```
lblResultado.Caption = "El sueldo semanal a recibir es: " & SUELDO_SEMANAL
```

### Función FORMAT

Por otra parte, se puede utilizar la Función FORMAT, para darle formato a la información a

mostrar. Los parámetros de esta función son: `Format (Expresión, formato)`

Para el caso del ejercicio que se está realizando sería:

`Format (SUELDO_SEMANAL, “0.00”)`

Nótese que en este caso se está determinando que se presente el sueldo semanal con dos decimales, al haber colocado entre comillas “0.00”. Ya con el uso de FORMAT, la línea de código para mostrar resultados sería la siguiente:

```
'Salida o impresión de resultados
lblResultado.Caption = "El sueldo semanal a recibir es: " & Format(SUELDO_SEMANAL, "0.00")
```

Finalmente, todo el código completo de esta subrutina o procedimiento quedaría así:

```
Private Sub btnCalcular_Click()
'Entrada o lectura de datos
NUMERO_HORAS = Val(txtNumeroHoras.Text)
VALOR_HORA = Val(txtValorHora.Text)

'Proceso o cálculos
SUELDO_SEMANAL = NUMERO_HORAS * VALOR_HORA

'Salida o impresión de resultados
lblResultado.Caption = "El sueldo semanal a recibir es: " & Format(SUELDO_SEMANAL, "0.00")
End Sub
```

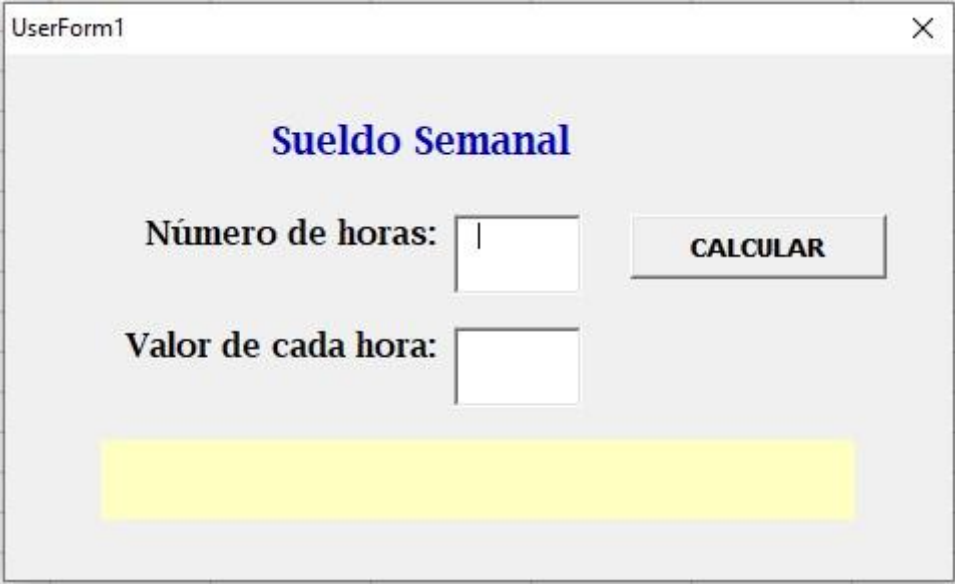
*Figura 62 Ejemplo de Código de Programación*

Para ejecutar el formulario realizado se utilizan los botones  disponibles en la barra de herramientas Estándar, que en su orden son: Ejecutar, Pausar y

Reestablecer. Las opciones también se encuentran disponibles en las opciones del menú EJECUTAR.

La ejecución de un formulario también es posible presionando la tecla **F5**.

El formulario en ejecución se presentaría de la siguiente manera:



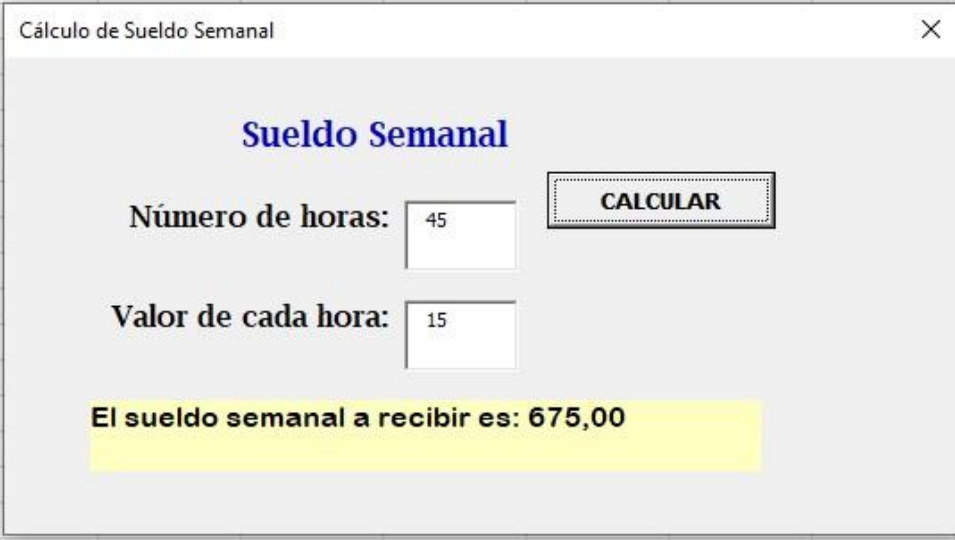
The screenshot shows a Windows form window titled "UserForm1". Inside the form, the title "Sueldo Semanal" is displayed in blue. Below the title, there are two input fields. The first is labeled "Número de horas:" and contains a single vertical bar cursor. To its right is a button labeled "CALCULAR". Below this, the second input field is labeled "Valor de cada hora:" and is currently empty. At the bottom of the form, there is a yellow rectangular area.

*Figura 63 Ejemplo de Formulario en ejecución*

Se puede cambiar el título que se presenta en la barra del formulario, en la propiedad

*Caption* del formulario: *Caption*: Cálculo de Sueldo Semanal

Ya con datos ingresados y pulsando en calcular, se muestra así:



The screenshot shows the same form window, but now the title bar reads "Cálculo de Sueldo Semanal". The input fields contain the values "45" and "15" respectively. The "CALCULAR" button is now highlighted with a dashed border. At the bottom, the yellow rectangular area displays the text "El sueldo semanal a recibir es: 675,00".

*Figura 64 Ejemplo de Formulario con datos y resultado*

### Botón Nuevo y Salir

Para complementar el formulario, se le incluirá dos botones de comando adicionales:

uno que se denominará **NUEVO** para “limpiar o borrar” los cuadros de texto y poder ingresar un nuevo dato a calcular; y otro que sea para **SALIR** o abandonar la ejecución del programa. Con el mismo formato del botón CALCULAR (Ver *Figura Formulario con Botón de Acción*), a los botones creados se le pondrá como nombre en la propiedad *Name*: btnNuevo al botón NUEVO, y *Name*: btnSalir al botón SALIR. El formulario debe quedar así:

Figura 65 Ejemplo de formulario con Botón NUEVO y SALIR

La programación del botón NUEVO quedaría:

```
Private Sub btnNuevo_Click()
    txtNumeroHoras = ""
    txtValorHora = ""
    lblResultado = ""
End Sub
```

Se asignan "" (2 comillas seguidas) a cada control para borrar su contenido.

Mientras que la programación del botón SALIR quedaría así:

```
Private Sub btnSalir_Click()
End
End Sub
```

La instrucción END finaliza la ejecución de la aplicación, también podría ser *Unload*

**Me. El código completo de este ejercicio sería el siguiente, incluyendo los comentarios respectivos:**

```
Private Sub btnCalcular_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN CALCULAR
'Entrada o lectura de datos
NUMERO_HORAS = Val(txtNumeroHoras.Text) 'se extrae el valor numérico
VALOR_HORA = Val(txtValorHora.Text)

'Proceso o cálculos
'se multiplican los dos datos para obtener el resultado
SUELDO_SEMANAL = NUMERO_HORAS * VALOR_HORA

'Salida o impresión de resultados
lblResultado.Caption = "El sueldo semanal a recibir es: " & Format(SUELDO_SEMANAL, "0.00")
' el uso de FORMAT permite presentar resultados con dos decimales por ejemplo
End Sub

Private Sub btnNuevo_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN NUEVO
'Se asigna "" (2 comillas seguidas) a cada control para borrar su contenido
txtNumeroHoras = ""
txtValorHora = ""
lblResultado = ""
End Sub

Private Sub btnSalir_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN SALIR
End 'finaliza la ejecución
End Sub
```

*Figura 66 Código completo*

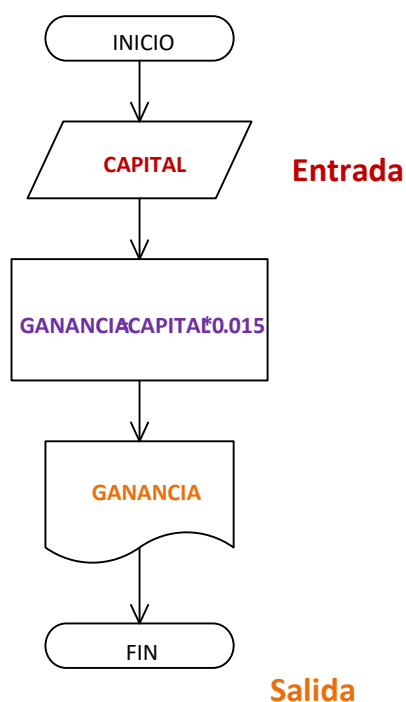
Se puede apreciar que cada procedimiento o subrutina se separa de otra con una línea horizontal

### **Ejemplo 2:**

Una persona desea invertir su capital en un banco y desea saber cuánto dinero ganaría después de un mes, si el banco paga a razón de 1,5% mensual.

- **Entrada.** - El dato que va a procesarse, lo identificamos por medio de una variable cualquiera, en este caso será la variable CAPITAL.

- **Proceso.** - El cálculo sería determinar el 1,5% del Capital, este cálculo se le asignaría a una variable:



$$\text{GANANCIA} = \text{CAPITAL} * 0,015$$

- **Salida.** - El resultado luego del proceso es la GANANCIA

Figura 67 Ejemplo de estructura secuencial

Para la solución de este ejercicio, en el entorno de VBA, diseñe el siguiente formulario:

El formulario, titulado "Ejercicio 2", contiene el título "Inversión y Ganancia". Debajo del título, hay un campo de entrada etiquetado "Capital a invertir:". Debajo de este campo, hay una barra decorativa de color naranja. En la parte inferior, hay tres botones: "CALCULAR", "NUEVO" y "SALIR".

Figura 68 Formulario Ejemplo 2

El código de programación completo quedaría:

```

Private Sub btnCalcular_Click()
'Entrada o lectura de datos
CAPITAL = Val(txtCapital.Text)

'Proceso o cálculos
ganancia = CAPITAL * 0.015

'Salida o impresión de resultados
lblResultado.Caption = "La ganancia al 1.5% mensual es: " & Format(ganancia, "0.00")

End Sub

Private Sub btnNuevo_Click()
'2 comillas seguidas) a cada control para borrar su contenido
txtCapital = ""
lblResultado = ""

End Sub

Private Sub btnSalir_Click()

End

End Sub

```

El formulario en ejecución y con datos quedaría:

Figura 69 Formulario con datos y resultado Ejemplo 2

En el ejercicio se hizo uso de la función **VAL** para obtener el valor numérico del Capital que se ingresa; y se utilizó la función **FORMAT** para mostrar los resultados con dos decimales.

## UNIDAD 4: APLICACIÓN DE ESTRUCTURAS CONDICIONALES

Las macros de Excel son herramientas eficaces que permiten automatizar tareas repetitivas y agilizar procesos en hojas de cálculo, en este apartado se realizarán aplicaciones empleando estructuras de control selectivo, también conocidas como estructuras de control de decisión o estructuras condicionales, que son un componente fundamental en la programación, ya que mediante ellas es posible que en un programa se tomen decisiones basadas en condiciones y se ejecuten diferentes conjuntos de instrucciones según si se cumple o no una condición. El resultado de la comprobación lógica o condición siempre es VERDADERO o FALSO.

El uso de estas estructuras permite evaluar una condición, y en función del resultado realizar una acción. Existen varios tipos de estas estructuras como: Simple, Doble y Anidado.

### Estructura de Control Selectivo Simple

Es la que, dentro de proceso, al evaluar una condición se ejecuta un conjunto de instrucciones cuando el resultado es VERDADERO, y si es FALSO no realiza ninguna acción.

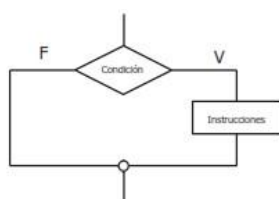


Figura 70 Representación Gráfica - Estructura selectiva

*simple* En VBA se utiliza la instrucción **IF THEN** cuyo formato es el siguiente:

**If** comprobación\_lógica **then**

*Código que se ejecuta cuando la instrucción es verdadera*

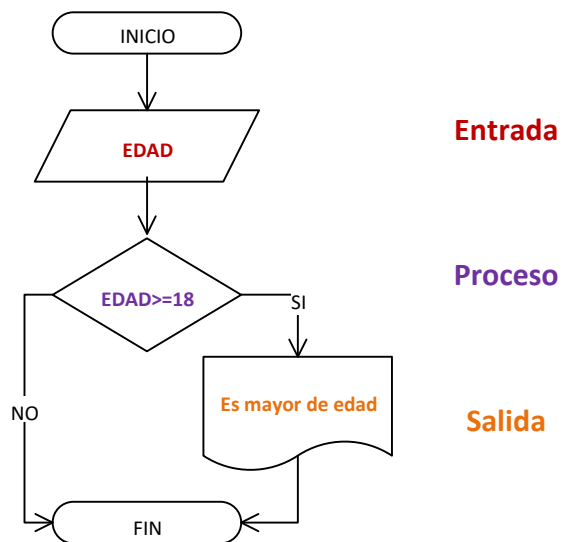
**End If**

En la sintaxis, la comprobación lógica se refiere a establecer una condición, en donde se hace uso de los operadores relacionales o de comparación indicados anteriormente.

En este tipo de estructura de control selectivo simple, sólo existe la opción de establecer las acciones que se ejecutarán en caso de que la condición establecida, dé como resultado VERDADERO. Se recomienda aplicar una sangría dentro del código para un mejor control y visualización de la programación.

Por ejemplo:

Determinar si una persona es mayor de edad.



información requerida

- **Entrada.** - El dato que va a procesarse es la variable EDAD.
- **Proceso.** - Mediante una condición se determina si la EDAD es mayor o igual a 18
- **Salida.** - El resultado luego de la condición es un mensaje con la

Figura 71 Ejemplo de estructura selectiva simple

Para la solución de este ejercicio, en el entorno de VBA, se diseñará el siguiente formulario:

Figura 72 Formulario Ejemplo Control Selectivo Simple

El código de programación completo quedaría:

```
Private Sub btnEjecutar_Click()
    'Entrada o lectura de datos
    EDAD = txtEdad.Text
    'Proceso
    If EDAD >= 18 Then
        'Salida o impresión de resultados
        lblResultado.Caption = "Es mayor de edad "
    End If
End Sub

Private Sub btnNuevo_Click()
    txtEdad = ""
    lblResultado = ""
End Sub

Private Sub btnSalir_Click()
    End
End Sub
```

El formulario en ejecución y con datos de ejemplo quedaría de la siguiente manera:

Figura 73 Formulario Ejemplo con datos y resultado Control Selectivo Simple

**Ejercicio Propuesto:** Con base en los contenidos revisados sobre estructura condicional simple, desarrollar el siguiente ejercicio:

Un local de comida rápida ofrece hamburguesas sencillas, dobles y triples, las cuales tienen un costo de \$2.00, \$3.50 y \$5.00 respectivamente. Si la compra es en efectivo, se aplica un descuento del 3% en la compra. Suponiendo que los clientes adquieren sólo un tipo de hamburguesa, determinar cuánto debe pagar una persona por N hamburguesas que compre.

### Estructura de Control Selectivo Doble

Una estructura de control selectivo doble es la que evalúa una condición, y cuando es VERDADERO ejecuta una acción, y si es FALSO ejecuta otra acción.

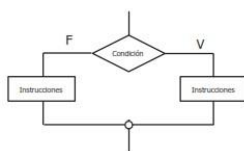


Figura 74 Representación Gráfica - Estructura selectiva doble

En VBA se utiliza la instrucción *IF THEN ELSE*, cuyo formato es el siguiente:

**If** comprobación\_lógica **Then**

*Código que se ejecuta cuando la instrucción es verdadera*

**Else**

*Código que se ejecuta cuando la instrucción es falsa*

**End If**

Por ejemplo: Determinar si al final del semestre un estudiante Aprueba o Reprueba el semestre.

El estudiante Aprueba si tiene como nota final 28 puntos o más, caso contrario Reprueba.

- **Entrada.** - El dato que va a procesarse es la variable **NOTA\_FINAL**.
- **Proceso.** - Mediante una condición se determina si la **NOTA\_FINAL** es mayor o igual a 28
- **Salida.** - El resultado luego de la condición es un mensaje con la información requerida

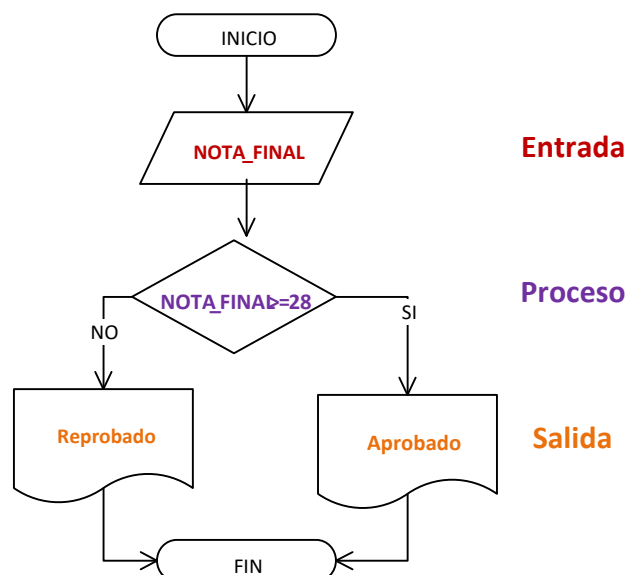


Figura 75 Ejemplo de estructura selectiva doble

Para la solución de este ejercicio, en el entorno de VBA, se diseñará el siguiente formulario:

The screenshot shows a VBA form window titled 'Ejemplo: Control Selectivo Doble'. The form has a title bar with a close button. The main area has a title 'Aprobados y Reprobados' in blue. Below the title, there is a label 'Ingrese la Nota Final:' followed by a text box. To the right of the text box are three buttons: 'EJECUTAR', 'NUEVO', and 'SALIR'.

Figura 76 Formulario Ejemplo Control Selectivo Doble

El código de programación completo quedaría así:

```

Private Sub btnEjecutar_Click()
    'Entrada o lectura de datos
    NOTA_FINAL = txtNotaFinal.Text
    'Proceso
    If NOTA_FINAL >= 28 Then
        'Salida o impresión de resultados
        'en caso de ser VERDADERO
        lblResultado.Caption = "APROBADO"
    Else
        'Salida o impresión de resultados
        'en caso de ser FALSO
        lblResultado.Caption = "REPROBADO"
    End If
End Sub

Private Sub btnNuevo_Click()
    txtNotaFinal = ""
    lblResultado = ""
End Sub

Private Sub btnSalir_Click()
End
End Sub

```

El formulario en ejecución y con datos de ejemplo quedaría de la siguiente manera para uno u otro caso:

The figure shows two instances of a Windows form titled "Ejemplo: Control Selectivo Doble". Both forms have a title bar with a close button. The main area of the form has a title "Aprobados y Reprobados" in blue italicized font. Below the title, there is a label "Ingrese la Nota Final:" followed by a text box. To the right of the text box is a button labeled "EJECUTAR". Below the text box is a label showing the result: "APROBADO" in the top screenshot and "REPROBADO" in the bottom screenshot. To the right of the result label are two buttons: "NUEVO" and "SALIR".

Figura 77 Formulario Ejemplo con datos y resultado Control Selectivo Doble

**Ejercicio Propuesto:** Con base en los contenidos revisados sobre estructura condicional doble, desarrollar el siguiente ejercicio:

En una camisería se desea saber el total a pagar por la compra de camisas. Si se compran tres o más camisas, el propietario aplica un descuento del 25% sobre el subtotal de la compra, en caso contrario se aplica el 10% de descuento. Como datos de entrada se tiene el número de camisas y el precio unitario. Mostrar como resultado con dos decimales: subtotal, descuento, IVA y valor total a pagar.

### Estructura de Control Selectivo Anidado

Una estructura selectiva anidada, también conocida como múltiple, es aquella que contiene una decisión dentro de otra, es decir que las condiciones se van “anidando”; en las condiciones suelen emplearse operadores lógicos.

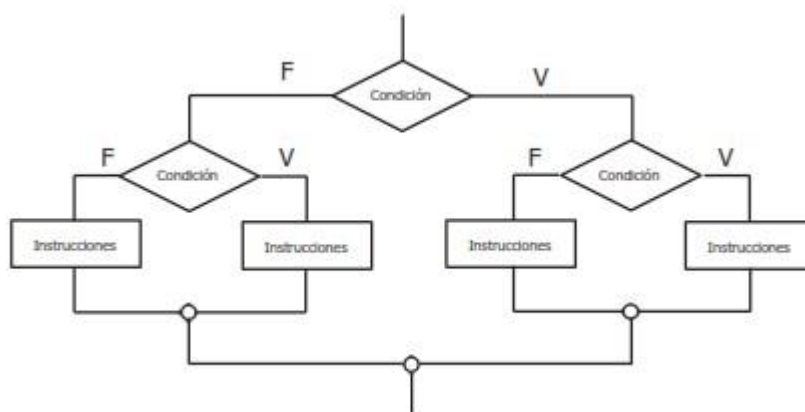


*Figura 78 Representación Gráfica - Estructura selectiva anidada o múltiple*

Por Ejemplo:

Se tienen tres números, determinar cuál es el mayor

- **Entrada.** - Se necesitan conocer tres números: A, B y C.
- **Proceso.** - Mediante condiciones anidadas se determina el valor mayor
- **Salida.** - El resultado luego de la condición, será mostrar el número mayor.

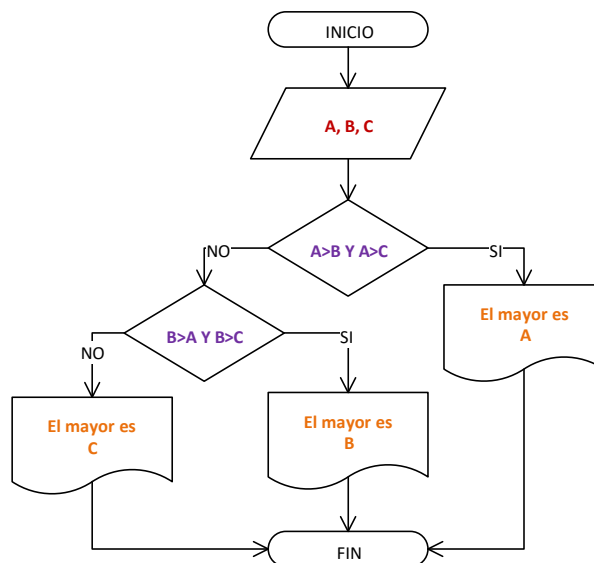


Figura 79 Ejemplo de estructura selectiva doble

Para la solución de este ejercicio, en el entorno de VBA, se diseñará el siguiente formulario:

Figura 80 Formulario Ejemplo Control Selectivo

Anidado El código de programación

completo sería así:

```

Private Sub btnEjecutar_Click()
'Entrada o lectura de datos
A = txtA.Text
B = txtB.Text
C = txtC.Text
'Proceso
If A > B And A > C Then
'Salida o impresión de resultados
'en caso de ser VERDADERO
lblResultado.Caption = "El número mayor es: " & A
Else
'en caso de ser FALSO otra condición (Si anidado)
If B > A And B > C Then
lblResultado.Caption = "El número mayor es: " & B
Else
lblResultado.Caption = "El número mayor es: " & C
End If
End If
End Sub
Private Sub btnNuevo_Click()
txtA = ""
txtB = ""
txtC = ""
lblResultado = ""
End Sub
Private Sub btnSalir_Click()
End
End Sub

```

El formulario en ejecución y con datos de ejemplo quedaría de la siguiente manera:

Figura 81 Formulario Ejemplo con datos y resultado Control Selectivo Anidado

Ejercicio Propuesto Con base en los contenidos revisados sobre estructura condicional anidada, desarrollar el siguiente ejercicio:

En un supermercado se efectúa una promoción que consiste en un descuento sobre el valor de la compra total, según un número que el cliente saque de un ánfora a pagar en la caja (considerar que solo existen los números del 1 al 4).

Las condiciones para este ejercicio son:

- Si el número es el 4 el cliente no tiene descuento, mostrar siga participando.
- Si el número es el 3 se hará un 10% descuento sobre el monto.
- Si el número es el 2 se hará un 25% de descuento.
- Y si el número es el 1, el descuento será del 100% del monto a pagar.

Mostrar el valor del descuento y la cantidad final que el cliente deberá pagar.

### Interoperabilidad con la hoja electrónica

Los datos que se procesan en un formulario pueden ser almacenados en una hoja electrónica de cálculo para su posterior uso, existiendo de esta manera una relación entre la hoja electrónica y el código de programación.

Para guardar los datos desde un formulario a una hoja electrónica, se deberá incluir un botón de comando (*CommandButton*), cuya acción sea grabar todos los datos que se encuentran en el formulario.

Para demostrar paso a paso el proceso de grabación de datos, se incluye el botón GRABAR en el formulario del Ejemplo 1 y se ha adicionado el nombre del empleado, a cuyos controles se les ha aplicado las propiedades ya conocidas:

Figura 82 Formulario con Botón GRABAR

Para proceder con el proceso de grabación hay que tomar en consideración estos aspectos:

#### Declaración General de Variables

La declaración de una variable permite especificar su nombre y su tipo antes de ser utilizada (Gómez, 2022), es una práctica recomendada declarar explícitamente todas las variables antes de usarlas, es aconsejable realizar la declaración ya que ayuda a reducir la posibilidad de errores y hace que el código sea más entendible. Cuando se utilizan variables dentro de una subrutina o procedimiento, no es tan necesaria su declaración ya que estas variables son locales; es decir solo se pueden utilizar y hacer referencia a ellas dentro del procedimiento más no fuera de él. Sin embargo, la declaración de variables sí es **obligatoria**, cuando se requieren ser utilizadas o llamadas desde cualquier subrutina dentro del programa sin perder su contenido. Por esta razón, las variables que se utilizan en el código de la programación deben declararse en forma global o general, cuando se van a grabar los datos en una hoja electrónica. Para forzar a la declaración de variables podría incluir en la sección general o superior la instrucción *Option Explicit*. Cuando una variable es declarada global, mantiene su valor cuando es utilizada en cualquier sección o subrutina de un programa. Para que una variable sea global debe ser declarada en la sección GENERAL del código de un programa, la cual

se encuentra ubicada siempre en la parte superior del mismo, como se observa en la siguiente figura:

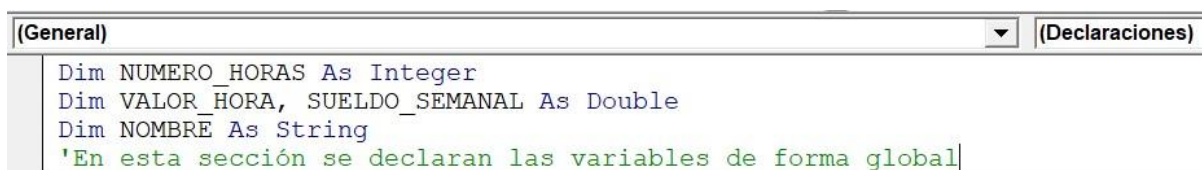


Figura 83 Sección de Declaración global de variables

La instrucción **DIM** permite declarar las variables y tiene la siguiente sintaxis:

#### Dim variable As Tipo

Cabe recordar que, las variables guardan valores y el tipo asignado a una variable dependerá de qué tipo de dato se pretende guardar, por ejemplo: valores enteros, decimales, textos, entre otros. Se puede declarar más de una variable del mismo tipo, separándoles con una coma. Para el caso del ejercicio detallado en el Ejemplo 1, la declaración global de variables quedaría de la siguiente manera:



Ahora bien, se sugiere que antes de realizar la codificación del botón GRABAR, se coloque un nombre a la hoja electrónica donde se guardarán los datos, y se dé un formato a la hoja preparando el encabezado de las columnas con los nombres de los datos a grabar e inclusive estableciendo formatos de número a las celdas, por ejemplo:

|    | A                   | B               | C                | D                |
|----|---------------------|-----------------|------------------|------------------|
| 1  | SUELDO SEMANAL      |                 |                  |                  |
| 2  | NOMBRE DEL EMPLEADO | NUMERO DE HORAS | VALOR DE LA HORA | SUELDO A RECIBIR |
| 3  |                     |                 |                  |                  |
| 4  |                     |                 |                  |                  |
| 5  |                     |                 |                  |                  |
| 6  |                     |                 |                  |                  |
| 7  |                     |                 |                  |                  |
| 8  |                     |                 |                  |                  |
| 9  |                     |                 |                  |                  |
| 10 |                     |                 |                  |                  |
| 11 |                     |                 |                  |                  |
| 12 |                     |                 |                  |                  |
| 13 |                     |                 |                  |                  |
| 14 |                     |                 |                  |                  |
| 15 |                     |                 |                  |                  |
| 16 |                     |                 |                  |                  |
| 17 |                     |                 |                  |                  |
| 18 |                     |                 |                  |                  |
| 19 |                     |                 |                  |                  |

Encabezado

Nombre de la hoja electrónica

SUELDOS

Figura 84 Hoja electrónica donde se guardarán los datos

### Programación del botón GRABAR

En la programación del botón GRABAR, se incluyen las siguientes instrucciones:

```
Private Sub btnGrabar_Click()
'Instrucción para seleccionar la hoja en la que se guardarán los datos
Sheets("SUELDOS").Select

'Instrucción que permite posicionarse en la última fila vacía de la hoja
K = ActiveSheet.UsedRange.Rows.Count + 1

'Para Grabar los datos
Cells(K, 1) = NOMBRE
Cells(K, 2) = NUMERO_HORAS
Cells(K, 3) = VALOR_HORA
Cells(K, 4) = SUELDO_SEMANAL
```

Explicación de las instrucciones utilizadas con sus respectivos formatos:

- ***Sheets("Nombre de la hoja de cálculo").Select.***- Permite seleccionar la hoja electrónica donde se van a guardar los datos del formulario. Para el caso de este ejercicio, la hoja se llama SUELDOS, y se coloca entre comillas, así: `Sheets("SUELDOS").Select`
- ***ActiveSheet.UsedRange.Rows.Count.***- Permite ubicar la última fila utilizada, este valor se asigna a una variable, y se le suma 1, para obtener el valor de la siguiente fila que es la que realmente se encuentra vacía, y es donde se debería

grabar. Si por ejemplo se le asigna a la variable K, quedaría de la siguiente manera:

- $K = \text{ActiveSheet.UsedRange.Rows.Count} + 1$
- **Cells(Fila, Columna).** - Permite seleccionar o ubicar una celda, indicando la Fila y la Columna, y se asigna la variable a grabar. En el caso del ejercicio para asignar los cuatro datos, se coloca:

- $\text{Cells}(K, 1) = \text{NOMBRE}$  7 En la última fila vacía, columna 1, se asigna la variable NOMBRE
- $\text{Cells}(K, 2) = \text{NUMERO\_HORAS}$  7 En la última fila vacía, columna 2, se asigna la variable NUMERO\_HORAS
- $\text{Cells}(K, 3) = \text{VALOR\_HORA}$  7 En la última fila vacía, columna 3, se asigna la variable VALOR\_HORA
- $\text{Cells}(K, 4) = \text{SUELDO\_SEMANAL}$  7 En la última fila vacía, columna 4, se asigna la variable SUELDO\_SEMANAL

El código completo, incluyendo la lectura de datos del nombre del empleado en el botón CALCULAR, queda de la siguiente manera:

| (General)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | (Declaraciones) |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <pre> Dim NUMERO_HORAS As Integer Dim VALOR_HORA, SUELDO_SEMANAL As Double Dim NOMBRE As String 'En esta sección se declaran las variables de forma global Private Sub btnCalcular_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN CALCULAR 'Entrada o lectura de datos NOMBRE = txtNombre.Text NUMERO_HORAS = Val(txtNumeroHoras.Text) 'se extrae el valor numérico VALOR_HORA = Val(txtValorHora.Text)  'Proceso o cálculos 'se multiplican los dos datos para obtener el resultado SUELDO_SEMANAL = NUMERO_HORAS * VALOR_HORA  'Salida o impresión de resultados lblResultado.Caption = "El sueldo semanal a recibir es: " &amp; Format(SUELDO_SEMANAL, "0.00") ' el uso de FORMAT permite presentar resultados con dos decimales por ejemplo End Sub  Private Sub btnNuevo_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN NUEVO 'Se asigna "" (2 comillas seguidas) a cada control para borrar su contenido txtNumeroHoras = "" txtValorHora = "" lblResultado = "" End Sub  Private Sub btnSalir_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN SALIR End 'finaliza la ejecución End Sub  Private Sub btnGrabar_Click() 'PROCEDIMIENTO O SUBROUTINA DEL BOTÓN GRABAR 'Instrucción para seleccionar la hoja en la que se guardarán los datos Sheets("SUELDOS").Select  'Instrucción que permite posicionarse en la última fila vacía de la hoja K = ActiveSheet.UsedRange.Rows.Count + 1  'Para Grabar los datos Cells(K, 1) = NOMBRE Cells(K, 2) = NUMERO_HORAS Cells(K, 3) = VALOR_HORA Cells(K, 4) = SUELDO_SEMANAL End Sub </pre> |                 |

*Figura 85 Código completo con programación del botón Grabar*

Como se puede observar en la figura anterior del código de programación, aparte de la sección de declaración general de variables, existen cuatro procedimientos o subrutinas que corresponden a cada uno de los botones de comando, y que a su vez se encuentran dispuestos estos procedimientos en orden alfabético.

La ejecución de este código, ya incluida la acción del botón grabar, da como resultado lo siguiente:

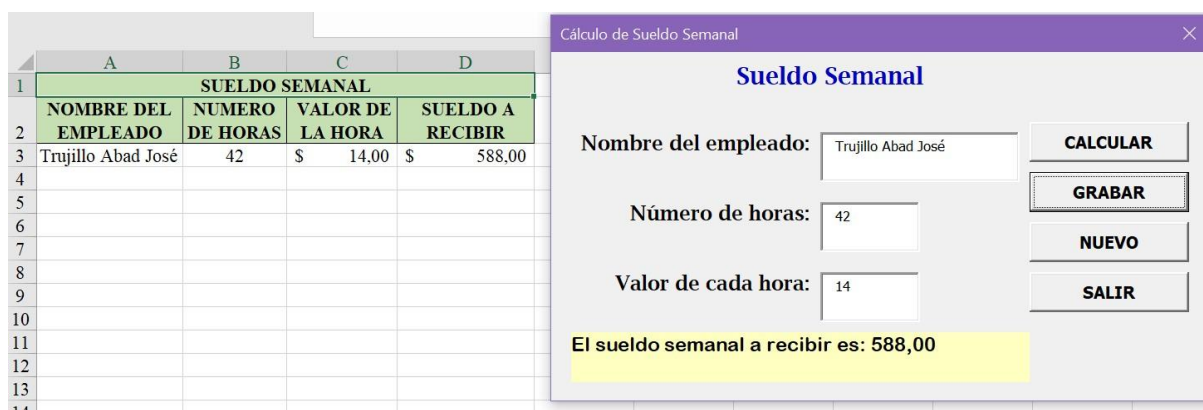


Figura 86 Ejemplo de ejecución del botón GRABAR

## Funciones Integradas en VBA

Entre las funciones integradas que opcionalmente se pueden utilizar dentro de la programación, se tiene a las siguientes:

**InputDialog:** Permite mediante un cuadro de diálogo solicitar un dato al usuario mientras se ejecuta el programa:

**MgsBox:** Muestra un cuadro de diálogo con información al usuario, permitiendo obtener una respuesta simple del mismo.

**Funciones de conversión de tipo:** Sirven para utilizar un tipo de datos de forma diferente al tipo de dato predeterminado. Entre las principales se encuentran las siguientes:

- **Cbool:** Convierte un número en un valor boolean.
- **Cbyte:** Convierte un número en byte.
- **Ccur:** Convierte un número en moneda.
- **Cdate:** Convierte un número en fecha.
- **Cdbl:** Convierte un número a double.
- **Cdec:** Convierte un número a Decimal.
- **Cint:** Convierte un número a Integer.
- **CLng:** Convierte un número a un entero largo.

- **Csng:** Convierte un número a Single.
- **CStr:** Convierte una variable de cualquier tipo a una de tipo String.
- **Cvar:** Convierte cualquier variable a una de tipo Variant.
- **Val:** Convierte una cadena de texto en un número.

- Blokdyk, G. (2018). *OpenProject The Ultimate Step-By-Step Guide*. Australia: 5STARCooks.
- Bucki, L.A. (2019). *ProjectLibre Practice Project: The Step-by-Step Sequence for Success*. 1x1 Media.
- Camaño, A. (2019). *Creación de Macros en Excel. Programación en Visual Basic para Aplicaciones (VBA)*. España: Editorial Ra-Ma.
- Díaz, S. J. M. (2015). *Aplicaciones informáticas de hojas de cálculo: Uf0321. Excel 2013*. Tutor Formación.
- Gómez, J.A (2022). *Aprender Visual Basic para Aplicaciones en Excel con 100 ejercicios prácticos*. Marcombo.
- Hueso, I. L. (2015). *Administración de sistemas gestores de bases de datos*. España: Editorial Ra-Ma.
- Viescas, J. L. (2007). *Soluciones de bases de datos con Access*. España: Editorial McGraw Hill