

INFORMÁTICA 2



Contenido

BLOQUE 1 ALGORITMOS.....	3
1.1 Definición de algoritmos	3
1.2 Características de los algoritmos.....	5
1.3 Metodología para la solución de un problema.	7
1.4 Diagramas.....	15
1.5 Simbología para elaborar diagramas de flujo	19
1.6 El pseudocódigo: características	23
1.7 Componentes de un pseudocódigo	23
1.8 Estructuras de control o algorítmicas	24
BLOQUE 2 HOJA DE CÁLCULO	27
2.1 Definición de la hoja de cálculo	27
2.2 Reconoce la aplicación de las hojas de cálculo y las ventajas de su uso.	31
2.3 Elabora hojas de cálculo	33
2.4 Elabora y edita gráficos con las características solicitadas.....	47
BLOQUE 3 BASE DE DATOS.....	50
3.1 Definición de base de datos.	50
3.2 Características de la base de datos.	50
3.3 Elementos básicos de Access.....	51
3.4 Elementos de la tabla de datos	53
3.5 Diseño de base de datos	53
3.6 Tipos de consulta.....	55
3.7 La finalidad de la base de datos.....	55
3.8 Aplica las bases de datos para manejar y administrar información.....	56
3.9 Tipos de datos en Access 2003.	59
3.10 Creación base de datos en relación con los requerimientos solicitados	63
BLOQUE 4 SOFTWARE EDUCATIVO.....	69
4.1 Definición de software educativo.....	69
4.2 Tipos de software educativo.....	70
4.3 Universidad virtual.....	73

BLOQUE 1 ALGORITMOS

1.1 Definición de algoritmos

Algoritmo (instrucciones de lavado)

- 1. Añada una medida de detergente**
Siga las instrucciones del fabricante del detergente. El uso de muy poco detergente es una causa común de problemas de lavandería.
- 2. Añada la carga adecuada**
Para obtener mejores resultados, no sobrecargue la lavadora. No envuelva las prendas alrededor del agitador.
- 3. Seleccione el tamaño de la carga y la temperatura**
Las prendas deben moverse libremente en la lavadora.
- 4. Seleccione el ciclo y el tiempo de lavado**
Reduzca el tiempo de lavado para cargas pequeñas y ropa ligeramente sucia.
- 5. Ponga a funcionar su lavadora**
Oprima el botón *Inicio/Pausa*. Retire la carga cuando termine el ciclo.

Ilustración A

Ilustración B

Ingredients

- ▼ 1 paquete de galletas "Marías"
- ▼ 2 cucharadas de jugo de limón
- ▼ 1/2 cucharadita de raspadura de limón
- ▼ 90 g de margarina derretida
- ▼ 1/2 paquete de gelatina de limón
- ▼ 1 cucharada de fécula de maíz diluida en 1/4 de taza de agua fría
- ▼ 2 huevos enteros
- ▼ 1 paquete de queso crema
- ▼ 1 taza de leche caliente
- ▼ 1 molde para pay

Algoritmo (modo de preparación)

- 1. Moler** en la licuadora el paquete de galletas.
- 2. Mezclar** las galletas con la margarina.
- 3. Untar** la mezcla en un molde para pay.
- 4. Disolver** la gelatina con la leche hirviendo, y agregar la fécula de maíz.
- 5. Licuar** esta mezcla con el resto de los ingredientes.
- 6. Calentar** en una cacerola, a fuego lento y sin dejar de mover, hasta que hierva.
- 7. Vaciar** sobre la base de galletas y refrigerar hasta que cuaje.

Cheese Cake de limón

¿Sabes que es un **algoritmo**?

Veamos algunas definiciones.

Un conjunto de pasos que nos permite obtener la solución a un problema.

Una serie de pasos organizados que describe el proceso que se debe seguir para dar solución a un problema específico.

Una secuencia finita de instrucciones realizables, no ambiguas, cuya ejecución conduce a la resolución de un problema.

¿En cuál de estas tres definiciones incluirías en los casos anteriores?

En el caso de la lavadora, ¿Dónde se halla el algoritmo?

¿Y en el caso de la receta del pay de queso?

Cómo pudiste darte cuenta, en la vida cotidiana empleamos algoritmos en multitud de ocasiones para resolver diversos problemas, como por ejemplo: para poner una

lavadora (conjunto de instrucciones pegadas en la tapa de la maquina), para hacer recetas de cocina (pasos de las recetas), para hacer música (las partituras), para hacer trucos de magia (pasos para hacer el truco) o incluso para construir un aeroplano a escala (expresados en las instrucciones).

La palabra algoritmo se deriva del nombre latinizado del gran matemático árabe Mohamed Ihn Al Kosv Rmzmi (781-850), quien entre los años 811 y 825 escribió su obra *Quitab Al Jarb; Al Mugabala* un tratado sobre manipulación de números y ecuaciones en el que se recogía el sistema de numeración india y el concepto del cero, junto con Euclides, el gran matemático griego (siglo IV a.c.) que inventó un método para encontrar el máximo común divisor de dos números, se les considera los padres de la algoritmia. Fue Fibonacci quien tradujo la obra al latín (*Liber Algorismi de numero indorum*) iniciando con la palabra Dixit algoritmi. Así del nombre de esa obra deriva del concepto de algoritmo.

El algoritmo de Euclides consiste en varias divisiones sucesivas a partir de dos cifras dadas; para la primera se toma como dividendo el número mayor y como divisor al menor.

Si tenemos las cifras $a= 945$ y $b= 651$, entonces:

$945/651= 1$ y quedo como restante 294

Una vez que se obtiene el resultado, el divisor se convierte en dividendo y la cifra restante en divisor, es decir:

$651/294$; así sucesivamente repetimos el proceso hasta obtener la cifra restante nula, como se muestra a continuación:

$651 / 294 = 2$ y como restante 63

$294 / 63 = 4$ y restante 42

$63 / 42 = 1$ y como restante 21

$42 / 21 = 2$ y como restante 0

Entonces, el máximo común divisor (mcd) de 945 y 651 es 21 porque fue el último restante no nulo.

Por lo tanto, esto permite detallar el siguiente algoritmo:

Datos de entrada a y b (valores absolutos)

Mientras $b = 0$ repetir las tres instrucciones siguientes:

r restante de a entre **b**

a **b**; convertir el valor de b en el nuevo valor de **a**

b **r**; convertir el valor de r en el nuevo valor de **b**

A los métodos que utilizan algoritmos se les denomina **métodos algorítmicos**, en oposición a los métodos que implican algún juicio o interrupción que se denominan métodos heurísticos. Las computadoras están consideradas como maquinas

algorítmicas; en ellas, los algoritmos se expresan como programas. Estos son algoritmos codificados con un lenguaje no ambiguo, cuya sintaxis y semántica entiende el ordenador. Hay muchos lenguajes de programación de ordenadores, entre ellos C y Java.

Una vez descubierto un algoritmo para efectuar una tarea, la realización de esta ya no requiere entender los principios en que se basa dicho algoritmo, pues el proceso se reduce a seguir las instrucciones del mismo. Sin embargo, existen algunos que requieren una organización bastante compleja de los datos que van a manejar, como aquellos algoritmos para la compra, el conteo de inventarios, las ventas, las ofertas y las demás actividades de una cadena de supermercados.

Los algoritmos se pueden expresar de dos formas:

Gráfica: con símbolos, utilizando diagramas de flujo.

No gráfica: describiendo las operaciones que llevará a cabo, utilizando un pseudocódigo

Los algoritmos también pueden ser:

Deterministas: en cada paso del algoritmo se determina, de forma única, el siguiente pasó.

No deterministas: deben decidir en cada paso de la ejecución entre varias alternativas y agotarlas todas antes de encontrar la solución.

1.2 Características de los algoritmos

Las características fundamentales que debe cumplir todo algoritmo:

Finito. Él algoritmo debe acabar tras un número finito de paso. Es casi fundamental que sea un número razonable de pasos. Por ejemplo, en el caso de la lavadora son cinco pasos; en el de la receta, siete pasos.

Definido. El algoritmo debe definirse de formas precisa para cada paso, es decir si se sigue dos veces, se debe obtener el mismo resultado cada vez. Esto quiere decir que cada vez que pongamos a trabajar la lavadora o cada vez que hagamos el pay de queso, la ropa debe tener la misma limpieza y el pay el mismo sabor y consistencia.

Preciso. Todas las operaciones que el algoritmo realizará deben ser lo suficientemente claras, es decir, hay que evitar toda ambigüedad al definir cada paso, de modo que puedan ser llevadas a cabo en forma exacta y en un tiempo finito por una persona.

En la definición de un algoritmo se deben determinar tres partes:

Entrada. El algoritmo tendrá cero o más entradas, es decir, cantidades dadas antes de empezar. Estas cantidades pertenecen, además, al conjunto, especificado de objetos.

Por ejemplo, pueden ser cadenas de caracteres, enteros, naturales, fraccionarios, etc. Se trata siempre de cantidades representativas del mundo real expresadas de tal forma que sean aptas para su interpretación por la computadora.

Proceso. Descripción de las operaciones que se llevarán a cabo con las entradas como datos para obtener la salida de acuerdo con la tarea para la cual fue diseñado.

Salida. El algoritmo tiene una o más salidas, en relación con las entradas.

En el algoritmo de receta de cocina citado anteriormente se tendrá:

Entrada. Ingrediente y utensilios empleados.

Proceso. Elaboración de la receta en la cocina.

Salida. Terminación del plan, (por ejemplo, cordero).

Un cliente solicita un pedido a una fábrica. Ésta examina en su banco de datos la ficha del cliente; si el cliente es solvente, entonces la empresa acepta el pedido; en caso contrario, lo rechaza.

Al redactar el algoritmo correspondiente a este caso, se escribirían los siguientes pasos:

1. Inicio
2. Leer el pedido (entrada)
3. Examinar la ficha del cliente (proceso)
4. Si el cliente es solvente, aceptar pedido; en caso contrario, rechazar pedido (salida).

Ante la pregunta ¿cuáles son las características de un buen algoritmo?, las siguientes respuestas reflejan, en cierta medida, los factores que identifican la calidad en ellos:

1. **Corrección.** El algoritmo debe estar bien planteado, de tal modo que siempre lleve al resultado correcto.
2. **Funcionalidad.** Nunca se debe olvidar que la característica más simple e importante de un algoritmo es que funcione.
3. **Eficiencia.** La eficiencia de un algoritmo se mide por los recursos que éste consume. En particular, se habla de la memoria y el tiempo de ejecución. A pesar de que con la reducción de los costos y del hardware es posible diseñar computadoras más rápidas y con más memoria, no hay que desperdiciar estos recursos, sino tratar de desarrollar algoritmo más eficiente.
4. **Claridad.** El algoritmo debe estar bien documentado. La documentación ayuda a comprender el funcionamiento de los algoritmos. Ciertos detalles o algunas partes especiales de los mismos pueden olvidarse fácilmente o quedar oscuros si no están adecuadamente documentados.

En realidad, de acuerdo con los puntos de vista anteriores, la calidad de un algoritmo tiene muchas facetas, todas ellas importantes. **Resumiendo: lo ideal es que nuestro algoritmo resulte correcto, eficiente, claro, fiable y fácil de mantener.**

1.3 Metodología para la solución de un problema.

Definición de problema

Etimológicamente, la palabra problema deriva del griego *proballein* y significa algo lanzado hacia delante, pero en nuestro caso lo definimos como:

Un problema es un asunto o conjunto de cuestiones que se plantean para ser resueltas.

Los **problemas** son situaciones nuevas que requieren que la gente responda con comportamientos nuevos, igual que en situaciones de nuestra vida cotidiana. Su naturaleza varía con el ámbito o con el contexto donde están plantearlas: así, existen problemas matemáticos, químicos, filosóficos, etcétera.

Un problema existe cuando hay tres elementos, cada uno claramente definido:

1. Una situación inicial.
2. Una situación final u objetivo a alcanzar
3. Restricciones o pautas respecto de métodos, actividades, tipos de operaciones, etc., sobre los cuales hay acuerdos previos.

Pasos para la solución de problemas

Resolver un problema implica realizar tareas que demandan procesos de razonamiento relativamente complejos y no simplemente una actividad asociativa y rutinaria. No existe un método universal que permita resolver cualquier problema. En general, la solución de problemas es un proceso creativo en el que el conocimiento, la habilidad y la experiencia tienen un papel significativo.

La **metodología** necesaria para resolver problemas mediante programas se denomina metodología de la programación y su eje central es el **algoritmo**. Así pues, si queremos que una computadora que se denomina metodología efectúe una tarea, primero debemos crear un algoritmo para llevarla a cabo y posteriormente programarlo en la máquina para representarlo, de modo tal que puede realizar esa tarea o hasta comunicarse con otra máquina. En otras palabras, debemos transformar el algoritmo conceptual en un conjunto de instrucciones y representar estas últimas en un lenguaje de programación de alto nivel.

Fases de resolución de un problema

De manera general, las fases de resolución de un problema con computadoras que deben seguir todos los programadores son siete, cada una de las cuales se encuentra entrelazada con la siguiente de tal modo que se complementan:

1. **Definición del problema** Es el enunciado del problema, el cual debe ser claro y completo, es fundamental conocer y delimitar por completo el problema, saber qué es lo que se desea realice la computadora.
2. **Análisis del problema.** El problema se analiza teniendo presente la lista de los requisitos dados por el cliente de la empresa o por la persona que encarga el programa.

Cada programador tiene estrategias diferentes para proyectar las soluciones de los programas de aplicación. Sin embargo, normalmente debe considerar ciertas preguntas, sin importar el enfoque del desarrollo de programa que se utilice.

- **¿Han sido escritas en forma clara y completa las especificaciones del problema?** Una especificación del tipo “escribir un programa para preparar las facturas de los clientes” desde luego no es adecuada. Antes de que pueda realizarse cualquier progreso significativo, debemos tener disponible el siguiente tipo de especificaciones revisadas: escribir un programa para imprimir facturas de clientes; cada factura contendrá el nombre del cliente, su dirección, ciudad, estado, código postal, cantidad de compras de un producto y el precio unitario de éste.
- **¿Está familiarizado con un método de solución que resuelva ese problema?** el algoritmo para calcular el número de hectáreas dentro de un lote rectangular, por ejemplo, consta de los siguientes pasos: a) multiplique la longitud del lote por su anchura para obtener los metros cuadrados de la propiedad y b) divida estos metros cuadrados por 10000 (el número de metros cuadrados dentro de una hectárea) para obtener la superficie en hectáreas de la propiedad.

En algunos casos, un programador tiene conocimientos personales de un algoritmo o procedimiento que le ayudará a resolver un problema que tiene a la mano. En este caso, la solución del problema puede ser codificada en un lenguaje seleccionado; si éste no es el caso, debemos considerar la siguiente pregunta.

- **¿Puede encontrar un método de solución para este problema, preguntando a otra persona o buscando en libros y revistas?** A menudo hay soluciones totales o parciales para diferentes problemas. Después de todo, el programador a quien se le da la tarea de escribir un programa de facturación no es el primero que tiene que enfrentarse con este problema. si otros recursos pueden proporcionar un método de solución, el programa necesario puede, entonces, codificarse en un lenguaje seleccionado. si no se tiene disponible un método de solución, debemos ver la siguiente pregunta.
- **¿Cómo desarrollar un algoritmo o procedimiento que resuelva este problema?** Es una pregunta que constituye un reto para los programadores. Un enfoque útil en la etapa de análisis de la programación dentro del desarrollo de un programa es dividir un gran problema (y por lo tanto, difícil de manejar) en una serie de tareas (o subproblemas) más pequeñas y más fáciles de entender.

Por lo tanto, a estos subproblemas aislados les podemos aplicar cada una de las preguntas anteriores.

Ejemplo:

La especificación revisada del programa de facturación, vista en la pregunta número 1 puede dividirse en las siguientes tareas principales:

- a) Introducir en la computadora el nombre, la dirección, la ciudad, el estado, el código postal, la cantidad comprada y el precio unitario del producto adquirido por el cliente.
- b) Calcular la cantidad neta que adeuda el cliente.
- c) Imprimir el nombre del cliente, su dirección, la ciudad, el estado, el código postal y la cantidad neta que debe.
- d) Si debe procesarse otra factura, bifurcar el control del programa de regreso al inicio a); de otra forma, continuar con la siguiente tarea.
- e) Detener el proceso.

En muchos casos (quizá en la mayoría) será conveniente dividir algunas de las tareas identificadas inicialmente en unidades todavía más pequeñas.

Por ejemplo, la tarea b) debe ser reducida como sigue:

- b) Calcular la cantidad neta que el cliente pagará.
 - b1) Calcular la cantidad neta que debe el cliente de acuerdo con la fórmula $A = Q \times P$ (donde A es la cantidad neta, Q es la cantidad comprada y P es el precio unitario).

La etapa de análisis de la programación continúa hasta que cada pequeña tarea, o subproblemas, sea reducida al punto de que el programador confíe en tener un método de solución para ella. La codificación del programa en forma efectiva puede comenzar solamente después de que el programador tiene esa confianza. Nótese que no es necesario o que la primera metodología de solución sea la mejor posible. A menudo pueden mejorarse los programas en su periodo de funcionamiento y probablemente nunca se descubra el mejor método posible.

3. **Diseño del algoritmo.** Una vez analizado el problema, se diseña una solución que conducirá a un algoritmo que resuelva el problema. En este tercer paso nos enfrenta a la *búsqueda de representaciones operacionales*, en la cual se debe analizar actividades que se efectuarán y realizar consensos para la solución del problema.

El diseño de un algoritmo que resuelva un problema es, en general, una tarea difícil. Una forma de facilitar esta labor consiste en recurrir a técnicas conocidas de diseño de algoritmos, es decir a esquemas muy generales que pueden adaptarse a un problema particular al detallar las partes generales del esquema.

Si el hábil programador dispone de un conjunto de algoritmos de donde seleccionar el más adecuado para cada problema, su tarea se simplifica, supongamos que disponemos de una especificación precisa, completa y consiste del problema a resolver y queremos obtener un algoritmo en el que, dados unos datos de entrada válidos, se produzca cierto resultado. Si no nos importa la eficiencia del algoritmo, podríamos utilizar un algoritmo general llamado del museo británico. Se programa una computadora de manera que parta de un conjunto de axiomas matemáticos y los use para reducir aleatoriamente teoremas validos.

Al aprender los principios básicos del diseño de algoritmos podemos preguntarnos por un método aceptable. El más entendido, y quizá el mejor, es organizar el diseño sobre un esquema de algoritmo o una técnica de diseño que haya demostrado su utilidad puesto que existe problemas. Este método de trabajo es el más utilizado puesto que existe un número reducido de esquemas y técnicas de diseño. El conocimiento de estas es solo un primer paso para el diseñador, que debe completarse con otros conocimientos y, sobre todo, con la experiencia.

4. **Codificación.** La solución se escribe en un lenguaje de alto nivel y se obtiene un programa. Durante esta etapa se pone en funcionamiento el algoritmo en un código escrito en un lenguaje de programación, reflejando las ideas desarrolladas en las fases de análisis y diseño.
5. **Prueba y Depuración.** El programa se ejecuta, se comprueba rigurosamente y se eliminan todos los errores que puedan aparecer. Con la verificación y la depuración el programador busca errores, y los elimina. Se comprobaba que mientras más tiempo se gaste en la fase de análisis y de diseño, menos se gastara en la depuración del programa.
6. **Documentación.** Es la guía o comunicación escrita que sirve como ayuda para usar un programa o facilitar futuras modificaciones. A menudo un programa escrito por una persona es usado por muchas otras, por eso la documentación es muy importante; esta debe presentarse en tres formas: INTERNA, EXTERNA y AL USUARIO FINAL.
 - a) Internas. Consiste en los comentarios o mensajes que se agregan al código del programa, que explican las funciones que realizan ciertos procesos, cálculos o fórmulas para el entendimiento del mismo.
 - b) Externas. También conocida como Manual Técnico está integrada por los siguientes elementos: Descripción del problema, Nombre del Autor, Diagrama de Flujo o Pseudocódigo, lista de variables y constantes, y Codificación del Programa, esto con la finalidad de permitir su posterior adecuación a los cambios.
 - c) Manual del Usuario. Es la documentación que se le proporciona usuario final; una guía que le indica como navegar en el programa, presentando todas las pantallas y menús que se va a encontrar y una explicación de los mismos.
7. **Mantenimiento.** El programa se actualiza y modifica cada vez que sea necesario, de modo que se cumplan todas las necesidades de cambio de sus usuarios.

Es importante destacar que hasta aquí que hemos llegado a construir un algoritmo, uno todavía independiente de lenguajes de programación y, en su desarrollo, nos ha permitido focalizarnos más en el problema mismo a resolver que en las herramientas destinadas a su implementación. Eso, es un logro importante. Asimismo, la toma de decisiones para resolver un problema resulta ser una tarea difícil; sus razones o motivos pueden ser tres:

1. Exige la elección entre varias opciones o alternativas y por lo tanto, lleva consigo posibilidad de equivocarse.
2. Esta muy unida en todas las organizaciones humanas al principio de autoridad. Las decisiones que se tomen pueden y deben afectar a personas, equipos de trabajo o estructuras de organización.
3. Lleva consigo, muchas veces, consecuencia en personas y equipos de trabajo que al sentirse afectados, dan lugar a conflictos.

Métodos de solución de problemas

Algunos de los principales métodos para solucionar problemas con algoritmos son los siguientes:

Método de fuerza bruta.

Comenzamos el estudio de esquemas de solución de problemas con un método sencillo, que debe evitarse siempre que se pueda, dada su ineficacia: la fuerza bruta. En realidad no es un esquema algo rítmico sino más bien un calificativo.

Es una forma de diseñar algoritmos tomando una solución directa, poco reflexionada. En principio, esto no es malo, pero dado que apenas se está analizando el problema considerable. Es muy probable que no se hayan aprovechado propiedades deducibles y que la solución sea ineficaz.

Una solución por fuerza bruta también puede resultar adecuada como primera aproximación a la solución final, porque su desarrollo puede permitir profundizar más sobre el problema y conocer propiedades que sean utilizadas para obtener otra versión más eficiente.

Un ejemplo de este método sería tratar de abrir una caja fuerte electrónica cuya clave consta de tres dígitos, ingresando una a una todas las posibles combinaciones de tres dígitos que existen, utilizando los números del 0 al 9. Los procedimientos por fuerza bruta, dado que utilizan el método de prueba y error, son muy costosos en tiempo computacional.

Divide y vencerás.

Consiste en descomponer un problema en subproblemas, o resolver independientemente cada uno de éstos para luego combinar sus soluciones y

obtener la solución del problema original. Esta técnica (divide y vencerás) es quizá la más utilizada debido a su sencillez: si un problema es demasiado grande para resolverlo de una vez, se descompone en varias partes más fáciles de resolver. Al dividir la basura en orgánica e inorgánica dividimos lo que se tira y es más fácil la solución de este problema sanitario.

Se puede aplicar con éxito a problemas matemáticos como la multiplicación de matrices, la ordenación de vectores, la búsqueda en estructuras ordenadas, etc. En tu vida escolar lo puedes percibir cuando tienes que estudiar alguna materia; es más fácil dividir el material a estudiar en partes y a su vez resumirlas en conceptos más concretos.

Método voraz

Este método trata de producir un mejor resultado a partir del conjunto de opciones. Para ello, se va procediendo paso a paso, realizándose la mejor elección de entre las posibles. Puede emplearse en problemas de optimización, en la planificación para el orden de la ejecución de programas en una computadora, etcétera.

Si el algoritmo voraz se ha diseñado correctamente. La primera solución encontrada es óptima. Por tanto, la dificultad principal al diseñar un algoritmo voraz reside en encontrar un criterio de selección que garantice la viabilidad de la solución.

Según esta descripción, el problema parte de una función objetivo que da el valor de una solución. Obviamente, ésta es la función por optimar, así como por un conjunto de restricciones sobre el valor de los datos de entrada y sobre la solución final del problema.

A su vez, la solución consta de los siguientes requisitos:

1. Un conjunto de candidatos.
2. Una función de selección que en cada momento determine qué candidato de los no usados parece ser el mejor.
3. Una función que determine si cierto conjunto de candidatos es válido; es decir, si permite tomar alguna solución del problema.

Observe que las funciones de validez y completitud no se preocupan de lo óptimo de la solución, pero si la función de selección es la adecuada. Cada solución válida y completa óptima.

Estos algoritmos se llaman voraces porque en cada paso toman el mejor trozo de la solución: es decir, el mejor «candidato». Además, nunca cambian de opinión; una vez que un candidato es aceptado o rechazado en la solución. La decisión es definitiva.

La función objetivo no suele aparecer en el algoritmo final sino que se utiliza durante el análisis del problema y es determinante en la elección de la función de selección. De toda, formas, debe recordarse que puede haber varios criterios alternativos de selección y de elegir correcto depende que la solución calculada sea **Óptima**.

El Ejemplo del desglose en monedas ayuda a entender mejor este método. Hay que desglosar una cantidad un conjunto de monedas, tratando de cumplir alguna condición; en esto caso, utilizar el menor número de monedas. Para ello se parte de un conjunto de tipos de monedas validas (de las que se supone que hay cantidad suficiente para realizar el desglose) y de un importe. Se trata de indicar la cantidad (menor) de monedas de los Opus considerados, tales que sumados sus valores equivalgan al importe.

Para simplificar, suponemos que manejamos moneda mexicana y, en particular, podemos, utilizar solo monedas de 1, 2, 5, y 10 pesos para el desglose. Estos valores lo definen por medio de un tipo enumerado MONEDAS. Asimismo, se declaran los tipos VALORES y CANTIDADES para representar el valor asignado a cada unidad monetaria y la cantidad de cada moneda que se devolverá en el desglose.

Se supone que inicialmente se asignaron los valores a cada uno de los tipos de moneda. Los elementos de la técnica voraz están presentes en este problema de la siguiente forma:

- El conjunto de candidatos está constituido por cada una de las monedas de los diferentes tipos que se pueden usar para realizar el desglose del importe dado.
- Una solución está dada por un conjunto de monedas devuelto tras el desglose, y cuyo valor total es igual al importe a desglosar.
- La condición de factibilidad de la solución, siendo construida, establece que el desglose debe ser menor o igual que el importe a desglosar.
- La función de selección establece que hay que elegir, mientras sea posible, la moneda de mayor valor de entre los candidatas.
- La función objetivo consiste en minimizar la cantidad total de monedas utilizadas en el desglose.

Se puede comprobar que en el problema están presentes los distintos elementos del método voraz. Además, cuando un candidato (moneda) se incorpora al conjunto solución, este no será nunca excluido de él.

Programación dinámica.

La programación dinámica (o planificación dinámica) es un método de programación que también permite resolver problemas mediante una secuencia de decisiones, pero de una manera más directa que en el método voraz. En esta se necesita producir varias secuencias de decisiones. Solamente al final se sabe cuál es la mejor de todas.

La idea del método de divide y vencerás. Es llevada al extremo con la programación dinámica. Si en el proceso de resolución del problema resolvemos un subproblemas, debemos almacenar la solución en una tabla ya que puede ser necesaria para la resolución final. Con el uso de esta tabla se evitan hacer cálculos idénticos reiteradamente, mejorando así eficiencia en la obtención de la solución. También

sirve para convertir ciertas funciones recursivas en iterativas. Por ejemplo, el cálculo de los 10 primeros números primos, donde almacenamos cada primo encontrado en una tabla y dividimos cada nuevo número solo por los que hay en la tabla (y no por todos los menores que él) para saber si es primo.

No es fácil establecer una definición de la programación dinámica: una característica es que el programa, “*aprende*” dinámicamente de las decisiones que toma. Además, todo problema resuelto con esta técnica debe satisfacer el principio de *optimalidad*. Este principio establece “*una secuencia optima de decisiones que resuelve un problema y debe cumplir la propiedad de que cualquier subsecuencia de decisiones también debe ser con respecto al subproblemas que resuelva*”.

En la programación dinámica todos los subproblemas se resuelven de acuerdo con el criterio de tamaño creciente y los resultados de subproblemas más pequeños se almacenan en algún tipo de estructura de datos (normalmente tablas) para facilitar la solución de los problemas más grandes. De esta forma se reduce al número total de subsecuencia generadas. Consiguiéndose soluciones más eficientes en cuanto tiempo de ejecución.

En resumen, la aplicación del método de programación dinámica a un problema significa: *comprobar primero el principio de **optimalidad** y desarrollar después ecuaciones recurrentes.*

Algoritmos de vueltas atrás o Backtracking.

Los algoritmos de vuelta atrás no siguen reglas para la búsqueda de la solución, simplemente efectúan una búsqueda sistemática, la cual significa que hay que probar todo lo posible hasta encontrar la solución o hallar que no existe. Para conseguir este propósito se separa la búsqueda en varias búsquedas parciales o subtareas.

¿Por que se llaman algoritmos de vuelta atrás? Porque en el caso de no encontrar una solución en una solución en una subtareas se retrocede a la subtareas original y se prueba otra cosa distinta una nueva subtareas distinta a las probadas anteriormente. Puesto que a veces nos interesa conocer múltiples soluciones de un problema, este algoritmo se puede modificar fácilmente para obtener una única solución (si existe) o todas las soluciones posibles (si existe más de una) al problema dado.

Estos algoritmos se asemejan al recorrido en profundidad dentro de un árbol siendo cada subtareas un nodo del árbol. El caso es que el árbol no está definido de forma explícita (como lista o matriz de adyacente) sino de forma implícita, es decir, que se irá creando según el avance el recorrido. A menudo dicho árbol es otro árbol, o no contiene ciclos, es decir, al buscar una solución es, en general, imposible llegar a una misma solución partiendo de dos subtareas distintas a y b o de la subtareas a es imposible llegar a la subtareas b y viceversa.

En ocasiones es imposible demostrar que al hacer una poda no se está ocultando una buena solución. Sin embargo, puede ser que el problema no pida la mejor

solución, sino una que sea razonablemente buena y cuyo costo sea bastante reducido. Ésa es una buena razón para aumentar las restricciones a la hora de recorrer un nodo. Tal vez se pierda la mejor solución, pero se encontrará una aceptable en un tiempo reducido.

1.4 Diagramas

Como se ha comentado en páginas anteriores, una de las formas de representar los algoritmos es de manera gráfica a través de los diagramas de flujo, pero ¿sabes que es un diagrama?, ¿para qué sirve? y ¿cómo se elabora?

Diagrama es una representación gráfica que muestra las relaciones entre las diferentes partes de un conjunto o sistema.

Los diagramas tienen las siguientes funciones:

1. Muestran gráficamente las acciones que se realizarán de acuerdo con el programa.
2. Ayudan a pensar claramente.
3. Ayudan a detectar errores.
4. Facilitan cambios en el programa.

Existen diferentes tipos de diagramas, algunos más sencillos y otros un tanto complicados. Veamos los más representativos.

Diagramas de entrada-proceso-salida

Todos los programas tienen algún grado de entrada (desde un usuario, un dispositivo o un programa), algún grado de procesamiento de entrada y algún tipo de salida, que puede ser una pantalla, una impresora, otro programa, etcétera (Figura 1.2).

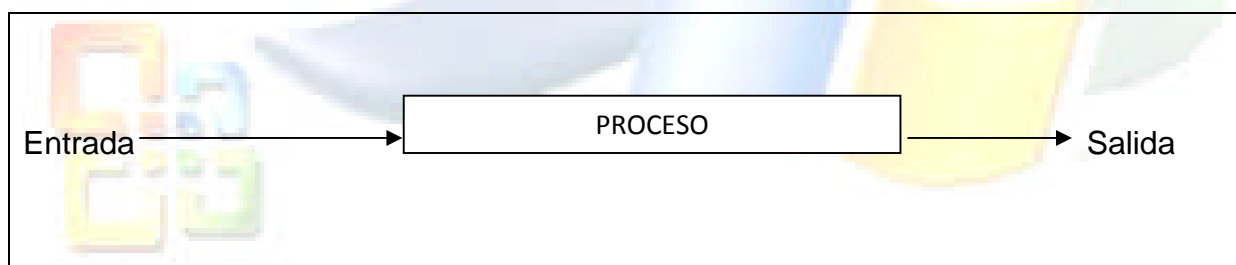


Diagrama IPO

Este diagrama debe su nombre a las siglas del inglés input-process-output (IPO); en castellano, entrada-proceso-salida (EPS). Es, por mucho, la más simple y práctica forma para planear la programación desarrollada. Su simplicidad envuelve el modelo **entrada-proceso-salida** que es bien conocido. La manera en que funciona es realmente clara, todo lo que hay que hacer es comenzar con un plan básico, así podemos desarrollar un programa, o parte de él, con alguna mutación en el proceso. A

continuación, se explica brevemente la descripción de un programa típico.

La entrada es la información que se obtiene para resolver el problema; incluye tres momentos:

1. Leer datos desde un archivo de configuración (CFC).
2. Pedirle al usuario el ingreso de algún dato (por ejemplo, su nombre).

Leer datos que ingresan desde un escáner, una cámara o cualquier otro dispositivo de entrada conectado al sistema.

El proceso involucra todo aquello que se hace para manipular o alterar los datos de entrada recibidos (por ejemplo ordenarlos, operarlos matemáticamente, etc.). Son los pasos necesarios para obtener el resultado de salida por medio de la información de entrada. Esto es, por lo general, la mayor parte del programa.

La salida es la fase inversa de la entrada; es la meta de la solución del problema. Podemos grabar la configuración actual, mostrar al usuario algún mensaje, imprimir algo, enviar los datos al disco o a la entrada de alto programa.

Los diagramas IPO son usados para dividir un sistema y difieren de atlas técnicas usadas en el análisis en que buscan dividir un sistema en subsistemas más que en funciones. El proceso inicia concibiendo al sistema como un todo.

El modelo será una simple secuencia de entrada-proceso-salida. La regla de descomposición consiste en preguntarse si el proceso "P" es un simple transformación de la entrada para obtener la salida, como a continuación se detalla:

La primera pregunta que debe hacerse es: para el conjunto de datos (entrada por teclado, archivo o base de datos), ¿"P" procesa cada entrada secuencial para producir una salida? Si la respuesta es no, como cuando solo un subconjunto de la entrada será procesada, entonces el sistema es dividido en dos subsistemas. Se presentan dos casos: cuando algún procesamiento debe ser completamente terminado antes de que el proceso principal sea llevado a cabo; por ejemplo: ordenar los datos. Y cuando la salida del primer subsistema sea directamente al segundo; por ejemplo: búsqueda de un registro.

$$E1 \begin{cases} P \text{ contrar} & \rightarrow S1 \\ P \text{ sumar} & \rightarrow S2 \\ P \text{ promedio} & \rightarrow S3 \end{cases}$$

Figura 1.3

La segunda pregunta es: ¿El formato de la salida depende sólo de los datos de entrada? Se da este caso cuando, por ejemplo, la entrada del teclado es mostrada directamente en la pantalla o cuando es capturada, pero solo una parte es desplegada. La respuesta puede ser "no", en casos como el desplegado en formato multipantalla o cuando hay múltiples salidas. En este caso, el sistema debe ser subdividido.

La tercera pregunta trata con el proceso «P» para determinar si es simple o compuesto. ¿Podrá «P» ser subdividido en dos o más procesos que individualmente procesen la entrada para generar una salida parcial? Por ejemplo: un proceso para leer un conjunto de datos y calcular el valor promedio puede ser dividido en un proceso para contar el número de datos, otro para sumarlas un tercero para calcular el promedio, como se observa en la Figura 1.3

Una vez que el problema ha sido dividido en simples subsistemas, la red de sus sistemas debe ser fraccionada en unidades de implementación. No se puede llenar un diagrama IPO al comenzar con las entradas, listar los procesos y, finalmente, poner las salidas. Generalmente, primero se encuentran las salidas o los resultados; después se determinan las entradas y, por último, se definen los pasos necesarios para convertir la información de entrada en la salida deseada.

Por ejemplo: suma de tres números.

¿Cuál es el resultado que se desea obtener?

¿Cuál es la información de entrada?

La salida o el resultado es la suma de los tres números. La entrada son los tres números. Los diagramas IPO no cuentan con una simbología especial, se representan mediante una tabla dividida en tres panes, como se muestra a continuación en la Tabla 1.1

Entrada	Proceso	Salida
Número - 1		Total
Número - 2		
Número - 3		

Tabla 1.1 Planteamiento de un diagrama IPO

¿Qué pasos son necesarios por convertir las entradas y obtener el resultado o la salida? proceso requiere:

1. Leer los Valores de los números.
2. Sumas los números guardar el resultado.

La salida es mostrar el resultado. Comúnmente, los programadores principales olvidan hacer la parte de la entrada y salida de un programa. El diagrama total se muestra a continuación en la Tabla 1.2.

Entrada	Proceso	Salida
Numero - 1	Total= número - 1 +	Total
Numero - 2	Número - 2 + numero - 3	
Numero - 3		

Tabla 1.2 Proceso IPO

El nombrado de variables es muy importante y no tiene límite. El nombre de la variable o constante debe describir los datos que se almacenan en ella. Por ejemplo: el resultado del cálculo de una nomina puede ser llamado total denomina, en lugar de resultado o total.

Puede haber pasos en un diagrama IPO antes de que se tenga todo lo que se necesita para crearlo en su totalidad.

Se hace el algoritmo al escribir la parte del proceso de un diagrama IPO en pseudocódigo. Hay que tener cuidado de no poner espacios entre palabras; es preferible usar el guion bajo para unir varias palabras. No hay que olvidar que un algoritmo comienza con un título y palabra **INICIO**, para terminar con la palabra **FIN**.

Es mejor usar la palabra Programa, seguida del nombre del mismo, para comenzar el pseudocódigo y finalizar el mismo con Fin del Programa. Por ejemplo: pseudocódigo de la suma de tres números.

Algoritmo

El nivel de un diagrama IPO mostrara el ¿Qué?, y el nivel inferior el ¿Cómo? Los detalles del ¿Cómo? se considerarán hasta que el ¿Qué? sea completamente definido.

Programa: suma de tres números

INICIO

Leer número_1, número_2, número_3

Total = número_1 + número_2 + número_3

Imprimir Total

FIN

ENTRADA	PROCESO	SALIDA
Numero _1	Total=número_1 +	
Numero _2	Número_2 + numero_3	
Numero _3		

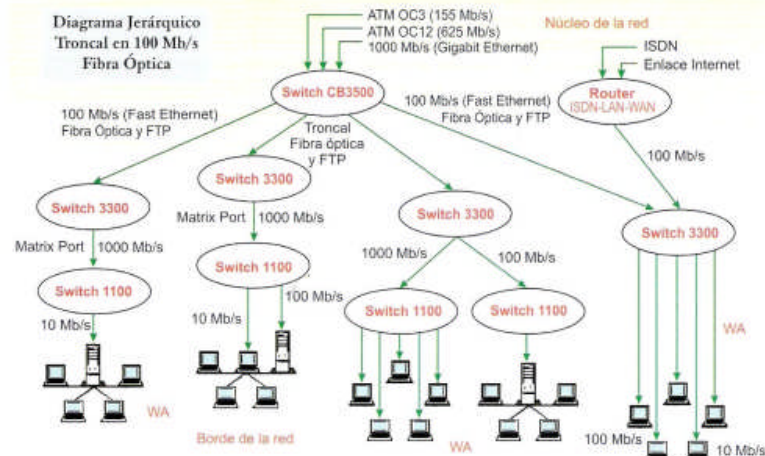


Figura 1.4 Ejemplo de un diagrama tipo HIPO.

Los diagramas generales especifican los procesos de un sistema de manera funcional; cada diagrama describe las entradas, los pasos de proceso y las salidas para la función en cuestión; un diagrama general puede indicar la localización de los diagramas de detalles subordinados necesarios. Ambos tipos tienen el mismo formato.

Diagramas de flujo

Los diagramas de flujo son representaciones gráficas de los pasos a seguir para lograr el resultado. Son una de las técnicas de representación de algoritmos más antigua y, a la vez, más utilizada, aunque su empleo ha disminuido considerablemente, sobre todo desde la aparición de lenguajes de programación estructurados. Se utilizan para representar algoritmos pequeños ya que abarcan mucho espacio y su construcción es laboriosa. Por su facilidad de lectura son usados como

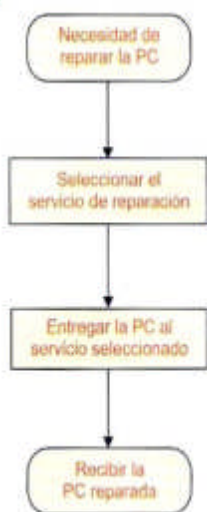


Figura 1.5 Diagrama de flujo.

introducción a los algoritmos, descripción de un lenguaje y descripción de procesos a personas ajenas a la computación.

Un diagrama de flujo debe ilustrar gráficamente los pasos procesos a seguir para alcanzar la solución de un problema. Por lo general, un diagrama de flujo está constituido de la siguiente manera: inicia, recibe datos, realiza el procesamiento, muestra resultados y finaliza.

Imagina que quieres hacer un programa en la computadora para que las amas de casa puedan preparar un pastel.



Figura 1.6 Ejemplo de un diagrama de flujo.

Lo primero que debes hacer es el diagrama de flujo, para que las instrucciones para preparar el pastel sean las correctas y estén en orden.

1.5 Simbología para elaborar diagramas de flujo

Desde que surgieron los diagramas de flujo se inventaron diversas simbologías. Los datos de los algoritmos se encierran en diferentes símbolos llamados figuras lógicas. Las principales son cinco: **Inicio, Proceso, Pregunta, Ciclo y Fin**.

Actualmente, La simbología se ha ido estandarizando y dentro de

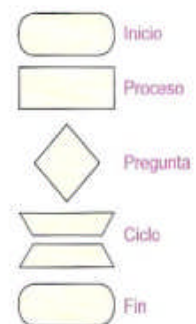


Figura 1.7 Figuras lógicas.

los símbolos más comunes podemos mencionar los siguientes:

Proceso. Se emplea un rectángulo para representar operaciones de proceso. Por supuesto, el proceso descrito es solamente un segmento pequeño de un paso de proceso mayor aplicado en el diagrama del sistema. Las instrucciones aritméticas y de movimiento de los datos generalmente se colocan en este tipo de caja. Los símbolos de entrada/salida y de Proceso tienen dos Líneas de flujo (una entrada y una salida).

Terminal. El símbolo terminal, como su nombre lo dice, representa el comienzo (Inicio) y el final (Fin) de un programa. Se usa también para señalar la interrupción de un programa cuando la información puede entrar o salir. Por ejemplo, para detectar cierto tipo de errores en los datos de entrada, el programador puede proporcionar una terminación especial en el programa que bifurque el control a un símbolo terminal etiquetado con «ALTO». Tiene una sola línea de entrada o de salida.


Decisión. El símbolo de decisión con forma de rombo, por otro lado, tiene una entrada y por lo menos dos salidas o transferencias. Este símbolo indica operaciones de lógica comparación. Las rotas de salida a menudo determinan un si o un no para alguna instrucción condicional que se escriba en forma de pregunta.


Conector. Este símbolo se representa por un círculo y se utiliza para no confundirnos con líneas adicionales de flujo y reducir la claridad del diagrama. Dos conceptos con idéntica etiqueta tienen la misma función que una línea de flujo larga. Esto es, muestra una salida a alguna otra sección del diagrama o indican una entrada proveniente de otra parte del mismo. ¿Cómo es posible determinar Si un conector se utiliza como entrada o salida? Es muy simple: si una flecha entra pero no sale de un conector, es un punto de salida, y el control del programa se transfiere hacia otro conector etiquetado idénticamente, que tenga una salida.

Proceso predeterminado. Con mucha frecuencia los programadores encuentran que ciertos tipos de operaciones de proceso se repiten en sus programas. En vez de volver a escribir un modulo cada vez que se necesite, el programador puede prepararlo una sola vez y después integrarlo dentro de otros programas, según se requiera. Las bibliotecas de estos procesos predeterminados o subrutinas, a menudo, se mantienen para reducir el tiempo y el costo de la programación. Por lo tanto, un solo símbolo de proceso predeterminado reemplaza un número de operaciones que no necesitan detallarse en ese punto en particular del diagrama. En pocas palabras, la subrutina es el módulo más utilizado que recibe entrada proveniente del control maestro del programa, realiza su tarea limitada y después regresa la salida al programa principal

Anotación. Las anotaciones se emplean para añadir comentarios, clarificando a otros símbolos del diagrama de flujo. Pueden ser dibujados a un lado de los mismos.

Otros símbolos de diagramas de flujo menos utilizados y de mayor detalle que los anteriores son:

Un  trapezoide indica que se va a ejecutar un proceso manual, en contraste con el rectángulo.

El símbolo general de entrada/salida  estándar es que se puede subdividir

en otros símbolos:

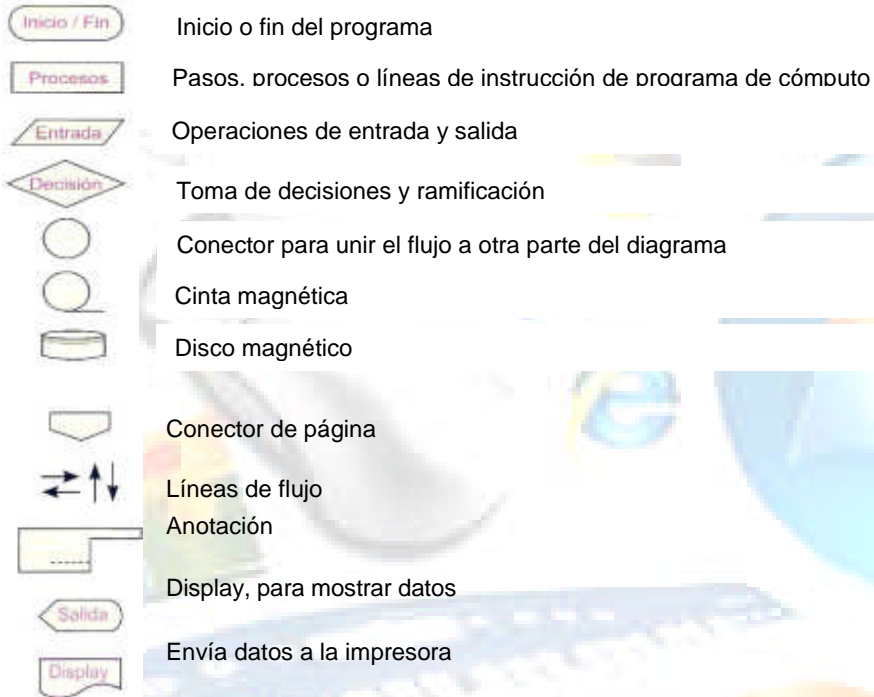


Pantalla magnética cinta



impresora magnética disco

Algunos autores señalan como símbolos estándares, que se requieren con frecuencia para diagramar programas de computadora, los siguientes:



Dentro de los símbolos fundamentales para la creación de diagramas de flujo, los símbolos gráficos son utilizados específicamente para operaciones aritméticas y relaciones condicionales. Veamos la siguiente lista:

+	Sumar	≥	Mayor o igual que
-	Menos	≤	Menor o igual que
*	Multiplicación	≠ o <>	Diferente de
/	División		Si
±	Más o menos		No
=	Equivalente a		True
>	Mayor que		False
<	Menor que		

Etapas en la construcción de un diagrama de flujo

1. Todo diagrama de flujo debe tener un Inicio y un Fin.
2. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser Rectas Verticales y Horizontales. NO pueden ser inclinadas o cruzadas.



3. Todas las líneas que indiquen la dirección del flujo deberán estar conectadas por medio de un símbolo que exprese lectura, proceso, decisión, impresión o fin.
4. La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación.
5. El diagrama de flujo debe ser construido de arriba hacia abajo y de izquierda a derecha.
6. Si el diagrama requiere más de una hoja, debemos utilizar los conectores adecuados y enumerar las páginas convenientemente.
7. No puede llegar más de una línea a un símbolo.

Ejemplo

Diseñar un algoritmo correspondiente que dando como dato la calificación de un alumno en un examen escriba "Aprobado" si su calificación es mayor que ocho y "Reprobado" en caso contrario.

Solución:

Dato: CAL, donde CAL es una variable de tipo real que expresa la calificación del alumno.

- a) Leer CAL
- b) Si $CAL > 8$

ENTONCES escribir 'probado'
SINO Escribir "Reprobado"

- c) FIN SI

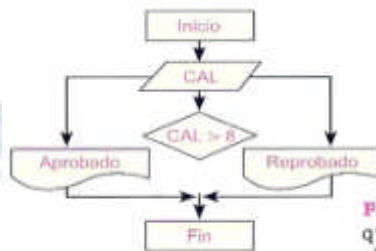


Figura 1.9 Diagrama de flujo que obtiene como resultado si un alumno está aprobado o reprobado con base a su calificación.

Diagrama de flujo que encuentra la suma de los primeros 50 números naturales.

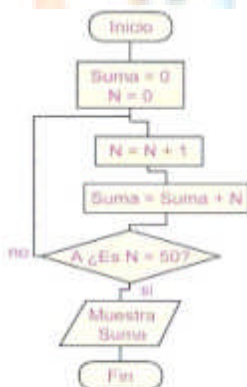


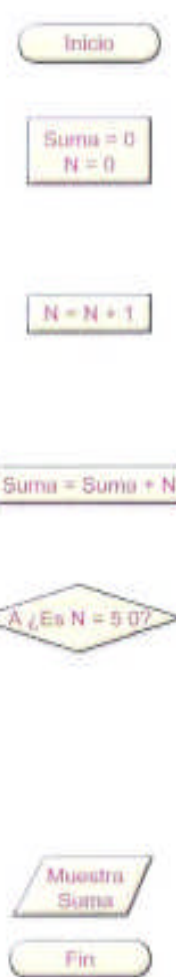
Figura 1.10 Diagrama de flujo que suma de 1 en 1 y muestra el resultado hasta que este sea 50.

Suma es la variable a la que se le va agregando el valor de cada número natural. N es el contador. Este recorrerá los números hasta llegar al 50.

El **primer bloque** indica el inicio del diagrama de flujo.

El **segundo bloque** es un símbolo de procesos. En este bloque se asume que los variables suma y N han sido declaradas previamente y las inicializa en 0 para comenzar el conteo y la suma de valores (para declararlas existe el bloque tarjetas perforadas).

El **tercer bloque** es también un símbolo de



procesos. En este paso se incrementa en 1 la variable N ($N = N + 1$), por lo que en primer fase esta N esta N valdrá 1, ya que estaba inicializada en 0.

El **cuarto bloque** es exactamente lo mismo que el anterior, pero en éste ya se le agrega el valor de N a la variable que contendrá la suma (en el primer caso contendrá 1, ya $N= 1$).

El **quinto bloque** es un símbolo que toma de decisiones y ramificación. Lo que hay dentro del bloque es una pregunta que se le hace a los valores que actualmente influyen en el proceso. ¿Es $N= 50$? Obviamente la respuesta es no, ya que N todavía es 1, por lo que el flujo de nuestro programa se dirigirá hacia la parte en donde se observa la palabra no: Tercer Bloque, éste le sumará 1 ($N =N+ 1$) y vuelve a llegar a este bloque, donde preguntará Es $N = 50$? ...¡No!, todavía es 2. De esta forma, regresa al Tercer bloque y vuelve a hacer lo mismo. Y así hasta llegar a 50, obteniendo así la suma de los primeros 50 primeros números naturales.

Por último, indicamos que el resultado será mostrado en la impresora.

Fin del programa (o diagrama).

1.6 El pseudocódigo: características

El **pseudocódigo** se puede definir como un lenguaje de especificaciones de algoritmos. Es la representación narrativa de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. Utiliza palabras que indican el proceso a realizar.

Es decir, cuando un algoritmo se representa mediante la sucesión de enunciados ordenados de manera lógica se le llama pseudocódigo. Veamos cómo se representaría del problema anterior:

El Pseudocódigo debe comenzar siempre con la palabra inicia y termina con la palabra fin.

1.7 Componentes de un pseudocódigo

Asignación de datos

Consiste en el paso de valores o resultados a una zona de la memoria, la cual será reconocida con el nombre de la variable que recibe el valor. Se puede clasificar en:

- **Simple:** consiste en pasar un valor constante a una variable ($a \leftarrow 15$).
- **Contador:** consiste en usarla como no verificador del número de veces que se realiza un proceso ($a \leftarrow a + 1$).
- **Acumulador:** consiste en usarla como sumador en un proceso ($a \leftarrow a + b$).
- **De trabajo:** donde puede recibir el resultado de una operación matemática que involucre muchas variables ($a \leftarrow c + b*2/4$).

En general. El formato a utilizar es el siguiente:

<nombre_de_variable> ← <expresión>

El símbolo ← debe leerse "asigne"

Escritura o salida de datos.

Consiste en mandar por un **dispositivo de salida** (por ejemplo: monitor o impresora) un resultado o mensaje. Esta instrucción presenta en pantalla el mensaje escrito entre comillas o el contenido de la variable y se representa como:

salida: (<lista_de_expresiones>)

Lectura o entrada de datos

La lectura o entrada de datos consiste en recibir desde un **dispositivo de entrada** (por ejemplo: el teclado) un valor o dato. Este dato va a ser almacenado en la variable que aparece a continuación de la instrucción. Esta operación se representa

Pseudocódigo

LEA <variable>

Ventajas de utilizar un pseudocódigo frente a un diagrama de flujo:

1. Ocupa menos espacio en una hoja de Excel.
2. Permite representar en forma fácil operaciones repetitivas complejas.
3. Es muy fácil pasar de pseudocódigo a un programa en algún lenguaje de programación.
4. Si se siguen las reglas se puede observar claramente los niveles que tiene cada operación.

1.8 Estructuras de control o algorítmicas

Se refiere a las estructuras con las que podemos llegar a cabo nuestros programas y algoritmos. Son un grupo de formas de trabajo que permiten. Mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas estructuras las podemos dividir en tres grupos:

Secuencial

Es una estructura con una entrada y una salida en la cual figura una serie de acciones cuya ejecución es lineal y en el orden en que

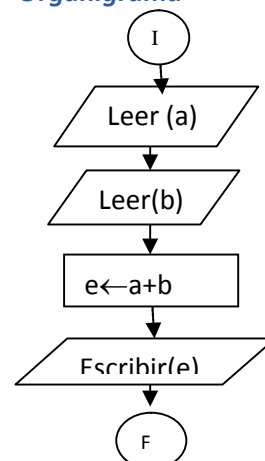
Pseudocódigo.

Inicio

Leer(a)
Leer(b)
 $e \leftarrow a+b$
escribir e

Fin

Organigrama



aparecen, a su vez, todas las acciones tienen una sola entrada y salida.

Como ya se ha comentado, el sistema para describir un algoritmo consiste en realizar una descripción paso a paso con un lenguaje propio. Recordemos que un algoritmo es un método o conjunto de reglas para solucionar un problema. En cálculos elementales estas reglas tienen las siguientes propiedades:

- Deben estar seguidas de alguna secuencia definida de pasos hasta que se obtenga un resultado coherente.
- Sólo puede ejecutarse una operación a la vez.

El flujo de control usual de un algoritmo es secuencial; consideremos el algoritmo que responde a la pregunta:

¿Qué hacer para ver la película de “La era de hielo 3”?

La respuesta es muy sencilla, y puede ser descrita en forma de algoritmo general de modo similar al siguiente:

1. Ir al cine
2. Comprar una entrada (boleto o ticket)
3. Ver la película
4. Regresar a casa

El algoritmo consta de cuatro acciones básicas, cada una de las cuales debe ser ejecutada antes de realizar la siguiente. En términos informáticos, cada acción se codificará en una o varias sentencias que ejecutan una tarea particular.

El algoritmo descrito es muy sencillo: sin embargo, como ya se ha indicado en párrafos anteriores, el algoritmo general se descompondrá en pasos más simples mediante un procedimiento denominado **refinamiento sucesivo**, ya que cada acción puede descomponerse, a su vez, en otras acciones simples. Así, por ejemplo, un primer refinamiento del algoritmo ir al cine se puede describir de la forma siguiente:

- 1 inicio
- 2 ver la cartelera de cine en el periódico
- 3 **si** no proyectan La Era de Hielo 3
entonces
 - 3.1 decidir otra actividad
 - 3.2 bifurcar al paso 7**si—no**
 - 3.3 Ir al cineFin_si
- 4 si hay cola entonces
 - 4.1. ponerse en ella
 - 4.2. **mientras** hay personas delante
Hacer
 - 4.2.1 avanzar en la cola**fin-mientras**

- fin_si
- 5. si hay localidades entonces
 - 5.1. comprar un bolero
 - 5.2. pasar a la sala
 - 5.3. localizar la(s) butaca(s)
 - 5.4. **mientras** aprovechan la película
 - hacer**
 - 5.4.1. ver la película
 - Fin_ mientras**
 - 5.5. abandonar el cine
- Sí_no**
- 5.6. refunfuñar
- fin si**
- 6. volver a casa
- 7. fin

En el algoritmo anterior existen diferentes aspectos a considerar. En primer lugar, ciertas palabras reservadas se han escrito deliberadamente en negritas (mientras, si, no, etc.). Estas palabras describen las estructuras de control fundamentales y procesos de toma de decisión en el algoritmo. Incluyen los conceptos importantes de selección (expresadas por si-entonces-si_no: **if – then – else**) y de repetición (expresadas con mientras-hacer o a veces repetir-hasta e iterar-fin_iterar, en inglés **while-do** y **repeat-until**) que se encuentran en casi todos los algoritmos, especialmente los de proceso de datos. La capacidad de decisión permite seleccionar alternativas de acciones a seguir o bien la repetición una y otra vez de operaciones básicas:

si proyectan la película seleccionada ir al cine
si_no ver la televisión, ir al futbol o leer el periódico
Mientras hay personas en la cola ir avanzando constantemente
hasta llegar a la taquilla.

Para terminar estas consideraciones iniciales sobre algoritmos, describiremos las acciones necesarias para refinar el algoritmo objeto de nuestro estudio; para ello analicemos la acción siguiente:

Localizar la(s) butaca(s).

Si los números de los asientos están impresos en el boleto, la acción compuesta se resuelve

1. Inicio/algoritmo para encontrar la butaca del espectador.
2. Caminar hasta llegar a la primera fila de butacas.
3. **Repetir**
 comparar número de filas con número impreso en boleto si no son iguales, entonces pasar a la siguiente fila **hasta_que** se localice la fila correcta.
4. **Mientras** número de butaca no coincida con número de bolero hacer avanzar a naves de la fila a la siguiente butaca
fin_mientras.
5. Sentarse en la butaca.
6. Fin

En este algoritmo la repetición se ha mostrado de dos modos, utilizando ambas notaciones, repetir... **hasta_que y mientras... fin_mientras**. Se ha considerado también, como ocurre normalmente, que los números de butaca y fila coinciden con los rotulados en el boleto.

Alternativa

Es una estructura con una sola entrada y una sola salida, en la cual se realiza una acción elegida de entre varias, según una condición, o se realiza una acción según el cumplimiento o no de una determinada condición. Ésta puede ser simple o compuesta.

Se clasifican en:

1. Estructura de dos salidas, en la que una de ellas puede ser la acción nula.
2. Estructura de tres o más salidas, también llamada múltiple.

Repetitiva

Es una estructura con una entrada y salida en la cual se repite una acción de un número determinado o indeterminado de veces, dependiendo en este caso del cumplimiento de una condición. Las estructuras repetitivas pueden ser:

1. **Estructura para (for)** Una acción se repite un número fijo de veces, representado normalmente por **N**.
2. **Estructura mientras (while)** - En esta estructura se repite una acción, mientras se cumpla la condición que controla el **bucle (ciclo)**. Su característica principal es que la condición se evalúa siempre antes de cada repetición. El número de repeticiones oscila entre cero e infinito; depende de la evaluación de la condición, cuyos argumentos en los casos de repetición, al menos una vez. Deberán modificarse dentro del bucle, pues de no ser así, el número de repeticiones será infinito y nos encontraremos en un bucle sin salida.

BLOQUE 2 HOJA DE CÁLCULO

2.1 Definición de la hoja de cálculo

Para los fines de este bloque, una hoja de cálculo es un programa que presenta una hoja electrónica cuadrículada en cuyas casillas se pueden realizar cálculos independientes de forma similar a una calculadora.



Ilustración 2.1 Vista de una calculadora y una hoja de cálculo.

¡Lo asombroso es que cada una de estas casillas funciona como una calculadora independiente!, pero más aún, la gran potencia de las hojas de cálculo viene de la capacidad de vincular todas estas calculadoras entre sí para resolver cálculos muy complejos que manualmente te llevarían mucho tiempo.

Para poder trabajar con cada una de las casillas es necesario identificarlas. Para ello utilizaremos un sistema muy similar al de las coordenadas cartesianas. Si no recuerdas qué es eso, tal vez alguna vez hayas jugado al submarino, donde para hundir el barco dices en qué columna y en qué fila pones una bomba. A continuación explicamos de qué se trata:

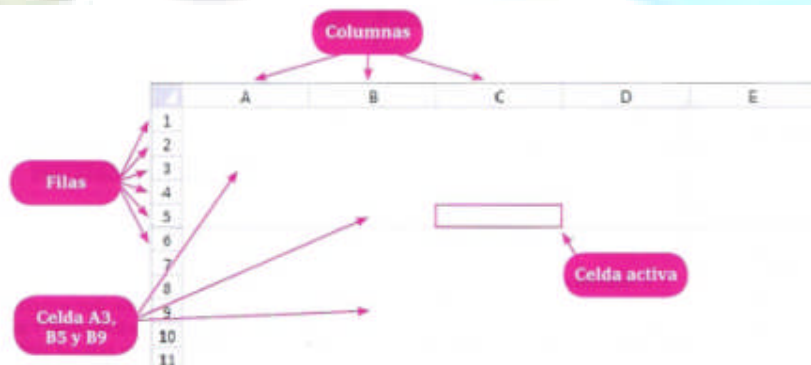


Ilustración 2.2 Columnas y filas de una hoja de cálculo.

Las columnas y las filas forman una cuadrícula con muchas casillas. Vamos a llamar celda a la casilla rectangular que se forma entre una columna y una fila. Cada celda va a tener un nombre único al que llamaré identificador, que se forma al unir la letra de la columna con el número de la fila.

Observa los siguientes ejemplos de identificadores y rangos de celdas.

	A	B	C	D	E
1	A1	B1	C1	D1	E1
2	A2	B2	C2	D2	E2
3	A3	B3	C3	D3	E3
4	A4	B4	C4	D4	E4
5	A5	B5	C5	D5	E5
6	A6	B6	C6	D6	E6
7	A7	B7	C7	D7	E7
8	A8	B8	C8	D8	E8
9	A9	B9	C9	D9	E9
10					
11					
12					

Ilustración 2.3 Identificadores y rangos de celdas en una hoja de cálculo.

Un **identificador de celda** es el nombre de la celda que se forma con la letra de la columna y la fila correspondiente. Una celda es, entonces, una casilla rectangular que se forma entre la unión de una columna y una fila: la celda se identifica con una letra (columna) y un número (fila). Un **rango de celdas** es un conjunto de celdas agrupadas para realizar algún cálculo u operación con ellas. Se representan uniendo la celda inicial del rango y la celda final y colocando dos puntos (:) entre ellas. *Ejemplo E1:E9* (rango vertical de E1 hasta E9), *A8:C8* (rango horizontal del A8 hasta C8) y *B1:C6* (rango rectangular desde B1 hasta C6).

	A	B	C	D	E
1					
2		5	3	=B2+C2	8
3					
4					
5					

Ilustración 2.4 Uso de los identificadores de celdas para hacer una suma en la celda D2.

(Ilustración 2.4).

	A	B	C	D	E	F	G	H
1								
2		18,2°C	22,9°C	28,4°C		18,2°C	23,5°C	28,4°C
3								
4	0	0,003	0,009	0,011		4,84058383	4,864536	4,87268656
5	0,5	0,003	0,012	0,011		4,84058383	4,87690629	4,87268656
6	1	0,003	0,012	0,012		4,84058383	4,87690629	4,87671028
7	1,5	0,003	0,012	0,014		4,84058383	4,87690629	4,88500908
8	2	0,007	0,014	0,019		4,85718172	4,88523967	4,90606249
9	2,5	0,007	0,014	0,026		4,85718172	4,88523967	4,93630238
10	3	0,007	0,017	0,029		4,85718172	4,89787142	4,9495476
11	3,5	0,01	0,017	0,034		4,86981347	4,89787142	4,97202046
12	4	0,01	0,02	0,036		4,86981347	4,91066477	4,98115294
13	4,5	0,01	0,023	0,042		4,86981347	4,92362392	5,00906173
14	5	0,013	0,026	0,044		4,88260682	4,93675321	5,01854047
15	5,5	0,013	0,026	0,049		4,88260682	4,93675321	5,04263603
16	6	0,013	0,029	0,052		4,88260682	4,95005717	5,05738031
17	6,5	0,016	0,029	0,057		4,89556596	4,95005717	5,08244428
18	7	0,016	0,032	0,06		4,89556596	4,96354052	5,09778985
19	7,5	0,02	0,036	0,065		4,91311027	4,97720816	5,12390099
20	8	0,02	0,037	0,068		4,91311027	4,98642482	5,13990133
21	8,5	0,02	0,037	0,073		4,91311027	4,98642482	5,16715097

Ilustración 2.5 Hoja de cálculo que utiliza identificadores de celda para construir una tabla matemática.

que usan identificadores de celdas para relacionar unas con otras. Sería muy laborioso tener que hacer todas las cuentas a mano.

Para eso sirve una hoja de cálculo: para ahorrarte trabajo al hacer cuentas y para que organices tus datos de manera más ordenada en tablas. Un poco más adelante veremos otros ejemplos con más detalle.

También puedes entrar a Excel 2003 haciendo doble clic sobre un archivo con extensión .xls (Microsoft Excel File Format); se entra de manera automática al programa cargando la hoja de trabajo respectiva.

Insertar, modificar y borrar datos

¿Para que sirven los identificadores y los rangos de celdas? Entender la respuesta a esta pregunta es el secreto de las hojas de cálculo, por lo que diremos que sirven para usar el contenido de una celda o de un rango desde otra celda.

Como puedes observar, se utilizan los identificadores de celda B2 y C2 para sumar los números 5 y 3 que están en dichas celdas.

Mediante los identificadores de celda y los rangos puede utilizar el resultado de una celda lejana y usarlo en otra celda para acumular resultados y hacer algo como en la siguiente tabla:

Tiene muchos números ¿verdad?; muchos de ellos se calculan automáticamente con formulas

Para insertar datos en una celda, primero se coloca el indicador de celda en la celda correspondiente y luego se pueden realizar los siguientes métodos

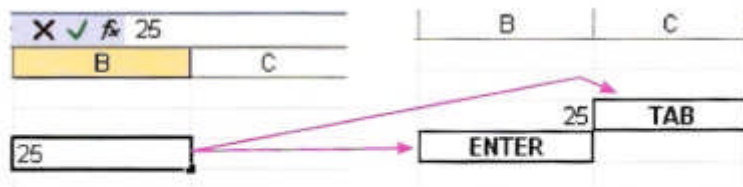


Ilustración 2.7 Diferencia entre ENTER y TAB al insertar un número.

Para insertar números. Se escribe iniciando con un número directamente, mediante el teclado numérico o mediante los números del teclado alfabético. Para terminar se presiona <ENTER>, <TAB> o se hace clic en . Observa que el

indicador de celda queda en un lugar distinto dependiendo de qué tecla presiones.

Sin embargo, si utilizas el botón , el indicador de celda se queda en el mismo lugar. Esto es bueno que lo sepas dependiendo de la dirección de captura porque convendrá utilizar una u otra tecla, si vas a capturar números en forma de lista conviene utilizar Enter.

Pero si vas a capturar en forma horizontal, te conviene utilizar **Tab** (Ilustración 2.9)



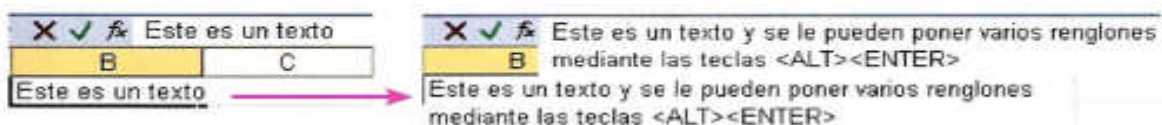
Ilustración 2.9 Captura de datos con TAB.



Ilustración 2.8 Captura de datos con ENTER.

Para insertar texto. Se escribe iniciando con una letra, un espacio en blanco o con las comillas ("). Los datos de tipo texto no sirven para hacer cálculos, sólo se usan para nombres de personas, direcciones o para encabezados de columnas. Se pueden insertar varios renglones en una celda; para ello antes de terminar se presiona <Ctrl><Enter> dentro de la celda y se escribe en no renglón distinto.

Ejemplo:



Para insertar formulas o funciones. Se inicia con el signo de igual (=) y se escribe la formula o la función. Esto lo veremos con detalle más adelante. Ejemplo = 5+6.

Para insertar fechas y horas. Para fechas se inicia con el numero de día, se separa con la diagonal (/), luego el mes (número o mes abreviado con sus primeras tres letras), se separa con la diagonal (/) y luego el año completo (2009). En algunas computadoras puede requerir iniciar con el mes.

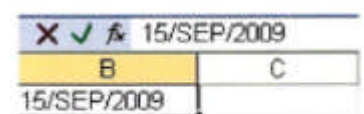


Ilustración 2.10 Insertar fecha.

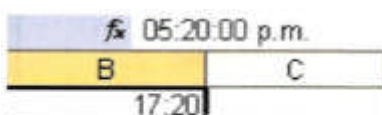





Ilustración 2.11 Insertar hora.

Para insertar horas, en el formato de 00:00, por ejemplo, las 5 de la tarde con 20 minutos se escribe: 17:20. Un truco para insertar la fecha actual es presionar **Ctrl-;** y para la hora **Ctrl-:**

Para **Entrar a modo de edición** (para corregir o cambiar los datos de una celda) se presiona la tecla F2 y para confirmar los cambios se presionan las teclas **ENTER, TAB** o se hace clic en .

Para cancelar la edición de una celda sin realizar cambios se presiona la tecla ESC o se hace clic en el botón: .



Para **borrar datos** de una celda se presiona la tecla **Supr**; esto elimina el dato y deja indicador en la misma celda. Con la tecla **Retroceso** () se elimina el dato o la formula y se entra en modo de edición en la misma celda.

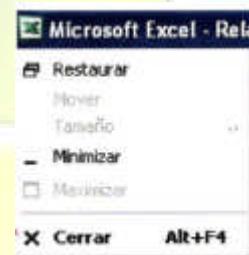
Guardar los cambios.

Si realizas algún cambio en algunas de las hojas de cálculo del *libro de trabajo* lo puedes grabar de manera rápida mediante el menú " *Archivo-Guardar* o el método rápido del teclado <CTRL> -G.

Si prefieres cambiar el nombre de la hoja de cálculo con el menú **Archivo. Guardar coma...** te permitirá hacer un duplicado de la presentación con el nuevo nombre que decidas ponerle.

El guardar los cambios efectuados en el documento es una medida preventiva que toma el programa para evitar que salgas sin grabar las últimas modificaciones realizadas a tu presentación y tienes que decidir si guardas o no. Si eliges cancelar, regresas a la presentación a seguir trabajando. Otras formas de salir de Excel son:

- Hacer clic en el menú de control  y seleccionar cerrar.
- b) Hacer clic en el botón cerrar de la ventana principal: 
- c) Mediante el método abreviado del teclado <ALT> - F4



2.2 Reconoce la aplicación de las hojas de cálculo y las ventajas de su uso.

Compara el uso de la hoja de cálculo con otros métodos de solución considerando el problema a solucionar.

Una hoja de cálculo es una herramienta para extender nuestras capacidades mentales de realizar Cálculos y es útil en la medida en que nos aporta ventajas adicionales a otros métodos alternativos dependiendo del problema a resolver. Veamos algunos otros métodos de cálculo que tenemos disponibles y que debemos valorar:

Calculo mental

El *cálculo* mental consiste en realizar cálculos matemáticos utilizando solo las capacidades intelectuales, sin ayuda de otros instrumentos manuales o electrónicos. Se ha observado que una persona con habilidades matemáticas desarrolladas tiende a pensar con claridad en otras áreas del conocimiento.

Es con mucho el mejor método que podemos utilizar para nuestros problemas aritméticos de la vida cotidiana, pues ejercita las habilidades mentales básicas de atención y concentración.

Por ejemplo, si nuestro problema es realizar sumas, un método bueno es el de sumar de derecha a izquierda mediante aproximaciones:

$$579 + 123 = 579 + 100 + 23 = 679 + 20 + 3 = 699 + 1 + 2 = 702$$

Observa cómo la suma original se fue convirtiendo en sumas más simples para obtener el resultado de manera rápida. Esto lo puedes hacer en tu mente sin mayor problema; inténtalo sin escribir.

Puede haber muchas soluciones; lo importante es que se te haga más fácil. El truco es separar alguno de los sumandos en números más fáciles de sumar como 100, 10, 5, etc. Por supuesto, hacer esta suma en Excel es muy sencillo, pero piensa que no todo el tiempo dispondrás de la computadora para hacerlo, y además considera el tiempo de encender la computadora, entrar a Excel, capturar la fórmula ya te ganaron. Si no, haz la prueba; juega una competencia con alguno de tus compañeros, tú con la mente y el con la computadora, y verás quién gana. Selecciona tus números y a sumar.

¿Será más rápido el cálculo mental que utilizar una hoja de cálculo?

Para responder a esta pregunta debemos pensar en el tipo de problema a resolver; si se trata de realizar un cálculo aritmético muy específico, es posible que sea mejor utilizar sólo tu mente, pero si se trata de llevar un registro amplio de muchos datos para guardarlos en tablas y después poder hacer operaciones con ellos, definitivamente habrá que considerar como mejor opción llevar una hoja de cálculo, pues nuestra memoria tiene dificultades para conservar grandes cantidades de datos de manera consciente, *aunque tal vez no sea imposible*.

Dos grandes calculistas mentales son Alberto Coto (www.albertocoto.com) y el cubano Yusnier Viera. Tal vez te interese participar en la Olimpiada Internacional de Cálculo Mental; puedes buscar información en la página: <http://www.eudactica.com/eudactica.html>

Cálculo manual

El cálculo manual lo realizamos con las manos o con ayuda de un lápiz y papel o algún otro método mecánico. Uno de los métodos más antiguos para cálculo manual es el ábaco. El ábaco es un instrumento poco utilizado en América y Europa, pero es de gran importancia. Durante milenios se ha utilizado en Asia y hasta nuestros días sigue siendo usado en China y Japón para cálculos cotidianos. Una persona adiestrada en el

uso del ábaco puede realizar operaciones aritméticas más rápido que otra persona que incluso haga uso de una calculadora.

¿Será más rápido, usar el ábaco que una hoja de cálculo? Depende de los cálculos a realizar. Es muy posible que para cálculos aritméticos de la vida cotidiana sea mucho más práctico para un comerciante asiático usar el ábaco que usa Excel, pues además el ábaco no usa electricidad y no requiere de una computadora. Sin embargo, para llevar el registro a largo plazo de las ventas realizadas y para hacer estadísticas y gráficos con estos datos definitivamente será mucho mejor usar una hoja de cálculo.

Cálculos automatizados

En nuestro mundo moderno no podría faltar la calculadora electrónica. Es un hecho que mediante ella se simplifican muchas operaciones y nadie niega su utilidad. Sin embargo, como ya te habrás *empezado* a percatar de las páginas anteriores, siempre existen mejores métodos, y como toda herramienta debes aprender a utilizarla cuando en verdad sea necesario.

También están las cajas registradoras, que mediante códigos de barras identifican los productos, sus precios y realizan la suma total a pagar en una tienda comercial. Por supuesto, ante aplicaciones tan específicas, la hoja de cálculo no es la solución adecuada, pero sí podría ayudar a un pequeño comerciante a llevar el registro de sus ventas diarias y ser más preciso en su control de ventas, de entradas y de salidas.

2.3 Elabora hojas de cálculo

Planea la hoja de cálculo con base en las características del problema a resolver.

Para poder planear una hoja de cálculo es necesario conocer el entorno de trabajo que nos permita comprender los elementos que nos serán de utilidad para un problema a resolver.

Describe el entorno de trabajo de la hoja de cálculo.

Empecemos por analizar los elementos que necesitamos para trabajar el programa Excel.

Barra de título

Contiene el nombre del libro de trabajo. Un libro de trabajo se compone de una o más hojas de cálculo. Al crear un nuevo libro en blanco, se crean por default tres hojas de cálculo en blanco y al libro en blanco le asigna automáticamente el nombre provisional Libro 1. Al momento de guardar el libro se le puede cambiar el nombre. A la derecha de la barra de título están los botones para minimizar, maximizar y cerrar la ventana. Al hacer clic sobre esta barra y arrastrarla se puede mover toda la ventana de la aplicación.

Barra de menús

Contiene los comandos para realizar operaciones con el libro y las hojas de cálculo, agrupados con palabras clave en *menús desplegables*. Se activan haciendo clic sobre alguno de los comandos o mediante la tecla ALT y subrayada del comando. Las opciones que aparecen en un tono más claro son opciones que no se pueden aplicar en el contexto actual. En esta barra se encuentra el comando de la Ayuda (?), que también se activa con F1. El icono X, del extremo derecho de la barra de menús, nos permite cerrar el libro de trabajo actual.

Un menú desplegable contiene tres tipos básicos de elementos:

Otro menú desplegable. Al situarse sobre este se abre un nuevo menú a su lado con más opciones para que elijamos una de ellas. Se reconocen porque tienen un triángulo a la derecha. Por ejemplo, en el menú *Formato* → *Hoja*.

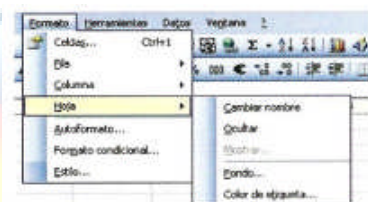


Ilustración 2.12 Ejemplo de menú de comandos con ventana.

Comandos inmediatos. Se ejecutan al hacer clic sobre ellos. Se reconocen porque a la derecha del nombre de comando, o no aparece nada o bien aparece la combinación de teclas para ejecutarlo desde el teclado. Por ejemplo, en el menú *Formato* → *Hoja* → *Ocultar*.

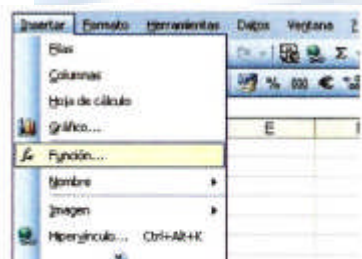


Ilustración 2.13 Ejemplo de menú de comandos con ventana.

Comandos con ventana. Al hacer clic aparece una ventana donde nos pedirá que rellenemos algunos campos y que tendrá botones para aceptar o cancelar la acción. Se distinguen porque el nombre acaba en puntos suspensivos.



Ilustración 2.14 Ejemplo de insertar función.

Por ejemplo, en el menú **Insertar** → **Función**.

El comando con ventana también puede llevar asociada también puede llevar asociada una combinación de teclas como los comandos inmediatos, como en el caso de **Insertar** → **Hipervínculo...** (Ctrl + Alt + K).

Barra de herramientas estándar

Contiene los comandos más frecuentes representados en forma de iconos para manejo de los libros de trabajo y las hojas de cálculo, como por ejemplo Nuevo Documento en blanco, Abrir, Guardar, Imprimir, etc. Se activan de manera directa al hacer Clic sobre alguno de ellos. Adicionalmente, cuentan con los botones AutoSuma, Orden Ascendente y Orden Descendente, respectivamente.



Barra de herramientas de formato

Contiene los comandos más frecuentes para dar formato a los datos dentro del libro

de trabajo y las hojas de cálculo, el relleno de las celdas, el número de decimales y formatos numéricos. También cuentan con un comando para combinar y centrar utilizado con frecuencia.

Barra de fórmulas



Se utiliza para editar y observar la fórmula que da origen al resultado que se observa en las celdas. También sirve para que mediante el botón fx se inserte una función para realizar cálculos de funciones matemáticas predefinidas. Cuando se entra en modo de edición (al escribir una fórmula) se activan dos botones adicionales:

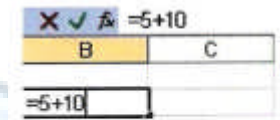


Ilustración 2.15 Ejemplo de edición de fórmulas.

- ✓ Este botón cancela y borra la fórmula que se está capturando.
- ✗ Este botón se utiliza para confirmar la fórmula y que se calcule el resultado. Es equivalente a presionar ENTER.

Cuadro de nombres

Se encuentra a la izquierda de la Barra de fórmulas y tiene dos funciones, dependiendo de la operación que se esté realizando.

Cuando se selecciona una celda o un rango, presenta el identificador de la celda activa o el nombre de un rango de celdas. Si el rango no tiene nombre, se puede nombrar escribiendo en este cuadro alguno que servirá para identificar el rango completo.



Cuando se escribe una fórmula, presenta una lista desplegable de las funciones de uso más común para poder seleccionarla de manera rápida.



Etiquetas de hojas

Se encuentran ubicadas en la parte inferior de la hoja de cálculo y se utilizan para el desplazamiento a través de las hojas. Al hacer clic con el botón derecho del mouse sobre estas etiquetas es posible cambiar el nombre de las hojas, eliminarlas o agregar nuevas hojas al libro de trabajo. De manera predeterminada se crean tres etiquetas de hoja al abrir un nuevo libro de trabajo, aunque siempre se pueden agregar más. Cuando el número de etiquetas es demasiado grande para el espacio disponible se activan los botones de desplazamiento para navegar entre las hojas, al inicio una hoja a la izquierda o una hoja a la derecha.

Barra de herramientas de estado



Contiene información de referencia para el usuario. Mediante ella Excel notifica de algunas acciones realizadas. Por ejemplo, al momento de activar la tecla Bloq Mayús, Bloq. Num, o bien al momento de copiar información, se podrá apreciar el mensaje «Seleccione el destino y presione ENTRAR o elija PEGAR», entre otros mensajes que dependen de las acciones que se estén ejecutando. Contiene también un área de autocalcular en la que aparecen resultados preliminares de un rango de celdas seleccionadas.



Ilustración 3.17 Panel de tareas para nuevo libro.

El Panel de Tareas permite seleccionar un conjunto de opciones para las tareas más comunes de Excel y las clasifica en secciones. Esta selección se realiza mediante el botón de selección ▼ de la parte superior derecha del panel.

Por ejemplo si seleccionamos **Nuevo libro** las opciones disponibles se presentan en el panel de tareas.

Si al seleccionar una opción en el panel de tareas observas en la parte inferior la barra ▼ quiere decir que hay más opciones que se pueden ver si colocas al apuntador sobre esta barra.

Emplea formulas y funciones.

La verdadera potencia en una hoja de cálculo se descubre cuando comprendemos y usamos las fórmulas y las funciones. Iniciemos primero por comprender qué es una fórmula y algunas aplicaciones básicas.

Podríamos definir una **fórmula** como una expresión aritmética que representa una serie de operaciones a realizar en cierto orden.

Formulas	Resultados	Descripción
= 5+10	15	Fórmula simple que expresa la suma de dos valores. Sólo se realiza una operación que es la suma.
= 2 +5 * 8	42	Fórmula simple que expresa una suma y una multiplicación. En Excel la multiplicación tiene más importancia que la suma y se realiza primero, aunque aparezca primero la suma de izquierda a derecha.
= (2 + 5) * 8	56	La misma fórmula anterior, pero al tener paréntesis se da más importancia a la suma. Primero se suman el 2 y el 5 y luego se multiplica por el ocho.
= 35-2*30/5+1	24	Fórmula simple que expresa una suma, una multiplicación y una división. La multiplicación y la división son más importantes que la suma y la resta. ¿Y cuál se hace primero de la multiplicación y la división? De estas dos, se realiza primero la operación que aparece primero de izquierda a derecha. Primero se multiplica 2 x 30, luego se divide el resultado entre 6 y luego a 35 se le resta ese resultado.

Operadores, operandos y precedencia

Como se puede observar en los resultados del cuadro anterior, el orden en que se realizan las operaciones es muy importante.

Orden	Operador	Descripción del funcionamiento
1	()	a) Se realiza primero la operación que aparece entre paréntesis. Si hay varios paréntesis, primero el más interno.
2	Función	b) Luego se calcula la función Si existe.
3	^	c) Luego se realiza la exponenciación
4	*, /	d) Luego se realizan la multiplicación y la división.
5	+, -	e) Luego se realiza la suma y la resta.
6		Operaciones de igual precedencia se realiza de izquierda a derecha y no afecta el orden el resultado.

Tabla 2.2 Reglas de precedencia

NOTA: Un operador es un signo o función que realiza una operación específica; los operandos son los valores sobre los cuales actúan los operadores, y la precedencia, el orden en que deben realizarse.

Aplicación de referencias relativas y absolutas

Las fórmulas usan el identificador *de celda* para hacer uso de los contenidos de una celda. A estos identificadores se les llama *referencias de celda*. Existen dos tipos de referencias de celda: las *referencias relativas* y las *referencias absolutas*. Veamos sus diferencias; observa la siguiente imagen:

a) Referencias relativas. Son identificadores de celda que cambian al copiar una fórmula de una celda a otra. En el siguiente ejemplo, al copiar la fórmula de A1 a B1, observa cómo las referencias A2/A3 se convierten de manera automática en **B2/B3**. Los identificadores que hemos utilizado en nuestros ejemplos, todos, han sido relativos.

Ilustración 2.18 Las referencias relativas cambian al copiar una fórmula.

b) Referencias absolutas. Son identificadores de celda que *no cambian* al copiar una fórmula de una celda a otra. Se identifican porque se les agrega un signo de pesos (\$) a las referencias para que permanezcan fijas. En el siguiente ejemplo, al copiar la fórmula de A1 a B1, observa cómo las referencias **\$A\$2/\$A\$3** permanecen sin cambio al copiar la fórmula en B1.

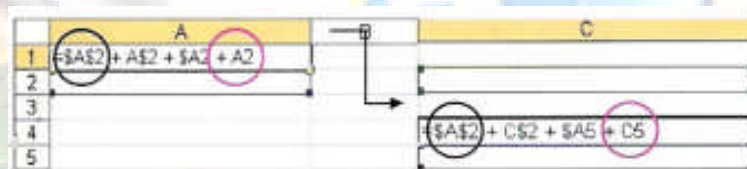
Ilustración 2.19 Las referencias absolutas no cambian al copiar una fórmula.

Puede haber referencias absolutas *por fila* cuando únicamente permanece fija la fila pero la *columna* cambia. Por ejemplo la referencia (A\$1) puede copiarse a otra celda y sólo cambiará la columna, pero la fila (1) permanecerá sin cambio. También hay referencias absolutas por columna (\$A1); en este caso, al copiar la fórmula a cualquier otra celda, la fila (1) cambiará, pero la columna (\$A) permanecerá sin cambio. Este tipo de referencias se denominan **Referencias mixtas**. Observa la siguiente tabla con ejemplos de las distintas referencias.

Ejemplo	Tipo de Referencia
\$A\$2	Referencia absoluta a la celda A2.
A\$2	Referencia relativa en la columna A y absoluta en la fila 2. (Referencia mixta)
\$A2	Referencia absoluta en columna y relativa en fila. (Referencia mixta)
A2	Referencia relativa

Tabla 2.3 Tipos de referencias

Ahora observa lo que ocurre con las distintas referencias, al copiar la fórmula del A1 a C4:



Referencia original	Al copiar la fórmula a C4 se convierte en:	¿Por qué?
\$A\$2	\$A\$2	Se mantiene sin cambio porque es una referencia absoluta.
A\$2	C\$2	Cambia la A por C (referencia relativa en columna) y se mantiene sin cambios el \$2 (referencia absoluta en fila), porque es una referencia mixta.
\$A2	\$A5	Cambia el 2 por 5 (referencia relativa en fila) y se mantiene la columna \$A (referencia absoluta en columna) porque se trata de una referencia mixta.
A2	C5	Cambia completamente la referencia, porque es una referencia relativa.

Tabla 2.4 Aplicación de referencias.

	A	B	C	D	E
1				% IVA	0.15
2					
3	Descripción	Cantidad	Costo	IVA	Importe
4	Computadora HP Sempron	3	4500	=C4*\$E\$1	=C4+D4
5	Disco Duro 320 Gb	2	960	=C5*\$E\$1	=C5+D5
6	Impresora HP DeskJet D1460	1	1500	=C6*\$E\$1	=C6+D6
7					
8	TOTAL:			=C4+C5+C6	=C8*\$E\$1
9					=C8+D8
10					

Ilustración 2.20 Cálculo de una factura sencilla.

Representar fórmulas matemáticas. Veamos un ejemplo más complejo; observa cómo se escribe en Excel la siguiente fórmula de una función cuadrática de la forma:

$$f(x) = ax^2 + bx + c$$

a) La constante **a** está colocada en la celda **A4**, pero para evitar que esta

referencia cambie al copiar la fórmula, se le ha colocado como *referencia absoluta* mediante los símbolos de (**\$**). (**\$A\$4**).

- b) La X^2 se ha representado como **A8^2**. Observa que los valores de «x» se toman del rango A8:A12.
- c) La constante **b** está colocada en la celda **B4**: para evitar que esta *referencia* cambie al copiar la fórmula, se le ha colocado como *referencia absoluta* mediante los símbolos de (**\$**). (**\$B4\$**).
- d) El primer valor de la **X** se toma de A8.
- e) La constante **c** está colocada en la celda **C4**; para evitar que esta referencia cambie al copiar la fórmula, se le ha colocado como *referencia absoluta* mediante los símbolos de (**\$**). (**\$C4\$**).

IMPORTANTE. No te confundas; la fórmula en B6 sólo es para representar visualmente la función cuadrática pero no hace nada. La fórmula que verdaderamente funciona está en B8 y de ahí se copió al rango B9:B12. ¿Te das cuenta de eso?

	A	B	C	D	E
1	Incremento en X				
2	1.00				
3	a	b	c		
4	4	4	-5		
5					
6	x	f(x)=a*X^2+b*X+c			
7					
8	-25.00	2395			
9	-24.00	2203			
10	-23.00	2019			
11	-22.00	1843			
12	-21.00	1675			

Ilustración 2.2: Aplicación de fórmulas.

Representación de cálculos en Excel.

Área de un triángulo	Aceleración			
<p>Área triángulo = $\frac{b \cdot h}{2}$</p>	$a = \frac{v - v_0}{t - t_0}$	t=0 t= 1s t= 2s t= 3s t= 4s	v= 0 v= 9.81 m/s v= 19.6 m/s v= 29.4 m/s v= 39.2 m/s	x= 0 x= 4.91 m x= 19.6 m x= 44.1 m x= 78.5 m

Figura 2.1: Ejemplos de cálculo de área y aceleración.

NOTA: La efectividad de las fórmulas es mayor cuando se usa en combinación con otra herramienta fundamental de la hoja de cálculo, que son las funciones.

Función

Una *función* es una instrucción predefinida que calcula de manera automática y directa un resultado, lo que facilita las operaciones más complejas. Las funciones ya están predefinidas en Excel, a diferencia de las fórmulas que son expresiones aritméticas definidas por usuario.



- Siempre se empieza con el signo=
- Luego sigue el nombre de la función
- Y entre paréntesis los datos sobre los que va a trabajar la función, que puede ser una lista de números, un rango o incluso una función. A estos datos se les llama Argumentos. Puede haber funciones que no lleven argumentos, en cuyo caso se pone los

Función	Descripción
a) =MAX(5,3,2,9)	Máximo, da como resultado 9, el número más grande.
b) =MIN(5,3,2,9)	Mínimo, da como resultado 2, el número más chico.
c) =SUMA(30,4,5,7,8)	Da como resultado 54, que es la suma de los valores.
d) =PROMEDIO(10,9,8,10,8)	Da como resultado 9, el promedio de la lista de valores.
e) =RAÍZ(49)	Da como resultado 7, que es la raíz cuadrada de 49.
f) =PRODUCTO(B5:B10)	Multiplica los datos del rango B5:B10.
g) =HOY()	Inserta la fecha del día actual. No tiene argumentos.
h) =CUENTA(C5:C10)	Cuenta los elementos dentro de un rango.

Tabla 2.5 Diferentes tipos de funciones.

paréntesis vacíos.

Las funciones se clasifican en diferentes tipos según su aplicación (Tabla 2.6).

Matemáticas	Estadísticas	Financieras	Búsqueda	Fecha y Hora
ABS(valor)	CONTAR	AMORTIZLIN	ÁREAS	SEGUNDO
ALEATORIO()	CONTARA	DURACIÓN	BUSCAR	MINUTO
COCIENTE(valor,divisor)	CRECIMIENTO	EFECTO	BUSCARH	MES
CONTAR(rango)	CUARTIL	MONEDA.DEC	BUSCARV	HOY
CONTAR.SI(condición, rango)	CURTOSIS	PAGO	COINCIDIR	HORA
ENTERO(valor)	DESVEST	PAGOINT	COLUMNA	FECHA/NÚMERO
EXP()	DESVIA2	PAGOPRIN	COLUMNAS	FECHA/MES
LN(valor)	MEDIANA	PRECIO	DESREF	FECHA
SUMA(rango)	MODA	RENDTO	DIRECCIÓN	DÍA/SEM
POTENCIA(valor,exponente)	PROMEDIO	TASA	ELEGIR	DÍAS/360
PRODUCTO(rango)	TENDENCIA	TIRM	FILA	DÍA
RAÍZ()	VAR	VNA	FILAS	AÑO
RESIDUO(valor,divisor)				

Tabla 2.6 Lista detallada de las principales funciones de Excel.

Bases de datos	Información	Texto y Datos	Lógicas	Ingeniería
BDCONTAR	CELDA	ASC	FALSO	BIN.A.DEC
BDESVEST	ES.IMP	CARÁCTER	NO	BIN.A.HEX
BDEXTRAER	ES.PAR	CÓDIGO	O	BIN.A.OCT
BDMAX	ESBLANCO	CONCATENAR	¡	COMPLEJO
BDMIN	ESERR	DECIMALES	VERDADERO	CONVERTIR
BDFPRODUCTO	ESERROR	DERECHA	Y	DELTA
BDCONTARA	ESLÓGICO	DESPEJAR		HEX.A.BIN
BDESVESTP	ESNOD	ENCONTRAR		HEX.A.DEC
BDSUMA	ESNOTEXTO	ESPACIOS		HEX.A.OCT
BDVAR	ESREF	EXTRAE		OCT.A.BIN
BDVARP	ESEXTO	HALLAR		OCT.A.DEC
BDFPROMEDIO	INFO	IGUAL		OCT.A.HEX

Tabla 2.6 Lista detallada de las principales funciones de Excel.



Ilustración 2.23 Explicación de la función S.I.

En la ayuda de encontrar los cada una de encontrar un detalladas que no dudes en usar la Ayuda.

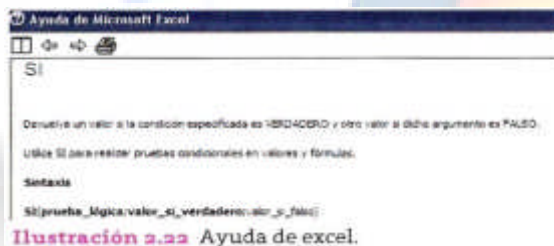


Ilustración 2.22 Ayuda de excel.

	Gastos reales	Gastos previstos
1	1900	900
2	300	900
3	500	925
4		
	Fórmula	Descripción (Resultado)
	=SI(A2>B2;"Presupuesto excedido";"Aceptar"); Comprueba si la primera fila sobrepasa el presupuesto (Presupuesto excedido)	
	=SI(A3>B3;"Presupuesto excedido";"Aceptar"); Comprueba si la segunda fila sobrepasa el presupuesto (Aceptar)	

Ilustración 2.24 Ejemplo de la función S.I.

Excel puedes detalles de cómo usar ellas pues es difícil libro que tenga todas y cada una; así

Y la podemos probar de la siguiente manera; sólo observa que en lugar de “;” debe usar “,”

SUMA	A	B	C	D	E
	Gastos Reales	Gastos Previstos	Dictamen		
	1500	900	=SI(A2>B2;"EXCEDIDO";"ACEPTADO")		EXCEDIDO
	500	900	=SI(A3>B3;"EXCEDIDO";"ACEPTADO")		ACEPTADO
	700	950	=SI(A4>B4;"EXCEDIDO";"ACEPTADO")		ACEPTADO

Formato a la hoja de cálculo

Las opciones de formato de la hoja de cálculo se utilizan para mejorar la representación de los datos desde una celda hasta una tabla.

Observa la siguiente tabla de datos sin formato:

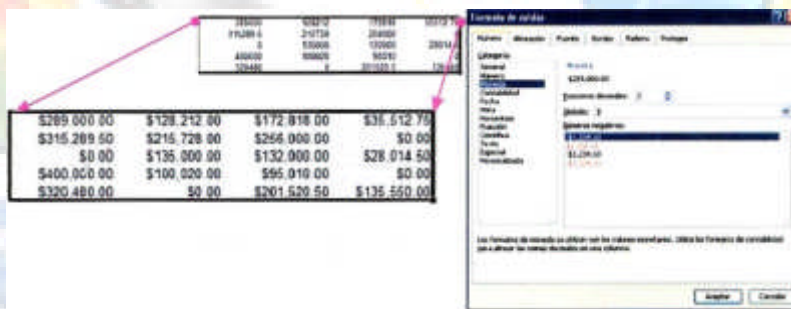
	A	B	C	D	E	F
1						
2	VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL	
3						
4	JOSE JIMENEZ	289000	128212	172818	35512.75	
5	CARMEN MEZURA	315289.5	215728	256000	0	
6	SANTIAGO PEREZ	0	135000	132000	28014.5	
7	LUISA LANE	400000	100020	95010	0	
8	CARLOS MARTINEZ	320480	0	201520.5	135550	
9						
10	TOTAL PRODUCTO	1324769.5	578960	657348.5	199077.25	
11						

Y la misma tabla pero con formato:

	A	B	C	D	E	F
1						
2	VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL	
3						
4	JOSE JIMENEZ	\$ 289,000.00	\$ 128,212.00	\$ 172,818.00	\$ 35,512.75	
5	CARMEN MEZURA	\$ 315,289.50	\$ 215,728.00	\$ 256,000.00	\$ -	
6	SANTIAGO PEREZ	\$ -	\$ 135,000.00	\$ 132,000.00	\$ 28,014.50	
7	LUISA LANE	\$ 400,000.00	\$ 100,000.00	\$ 95,010.00	\$ -	
8	CARLOS MARTINEZ	\$ 320,480.00	\$ -	\$ 201,520.50	\$ 135,550.00	
9						
10	TOTAL PRODUCTO	\$1,324,769.50	\$578,960.00	\$857,348.50	\$199,077.25	
11						

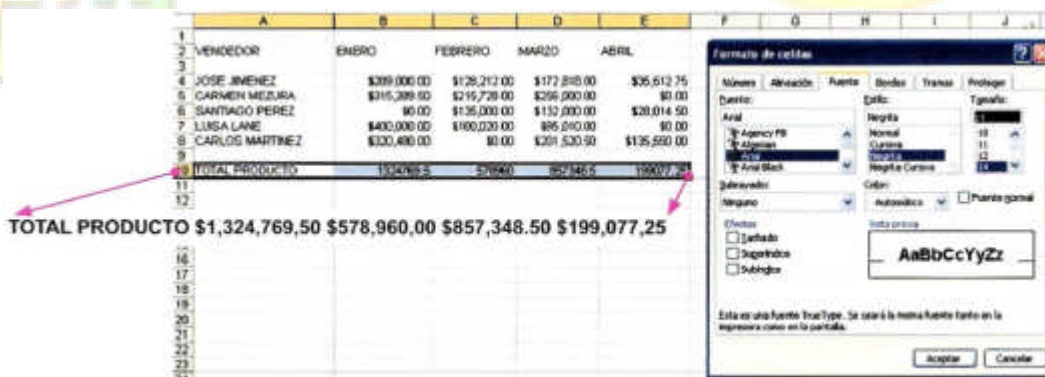
Como puedes notar, los cambios son significativos. Veamos cómo hacerlo. Primero cambiemos el formato de los números para separar los miles y poner el signo de pesos:

1.1 **Formato de celda.** Para activar este menú se utiliza la opción **formato** → **celda** a través del comando de teclado Ctrl + 1. Mediante este cuadro de diálogo podemos determinar distintas maneras de cómo se verán los datos. En la



pestaña número se establece la forma como se verán los números, fechas o datos de textos en una celda, en este caso seleccionaremos la categoría moneda observa.

Bien, esto mejoró un poco. Ahora cambiemos el tamaño de los totales a 14 puntos, lo ponemos en negrita y formato moneda.

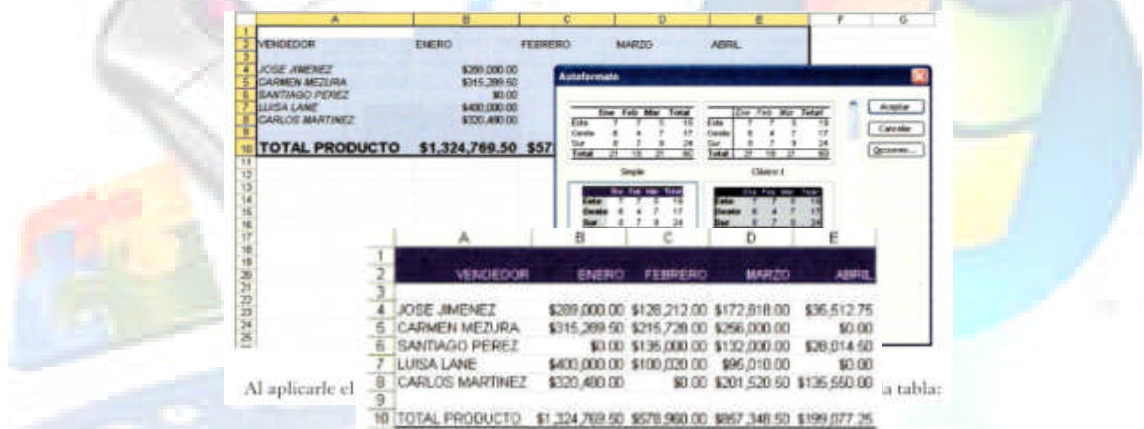


Con este mismo procedimiento, se aplica cursiva a los nombres de los vendedores para obtener el siguiente resultado:

	A	B	C	D	E
1					
2	VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL
3					
4	JOSE JIMENEZ	\$289,000.00	\$128,212.00	\$172,818.00	\$35,512.75
5	CARMEN MEZURA	\$315,289.50	\$215,728.00	\$256,000.00	\$0.00
6	SANTIAGO PEREZ	\$0.00	\$135,000.00	\$132,000.00	\$28,014.50
7	LUISA LANE	\$400,000.00	\$100,020.00	\$95,010.00	\$0.00
8	CARLOS MARTINEZ	\$320,480.00	\$0.00	\$201,520.50	\$135,550.00
9					
10	TOTAL PRODUCTO	\$1,324,769.50	\$578,960.00	\$857,348.50	\$199,077.25

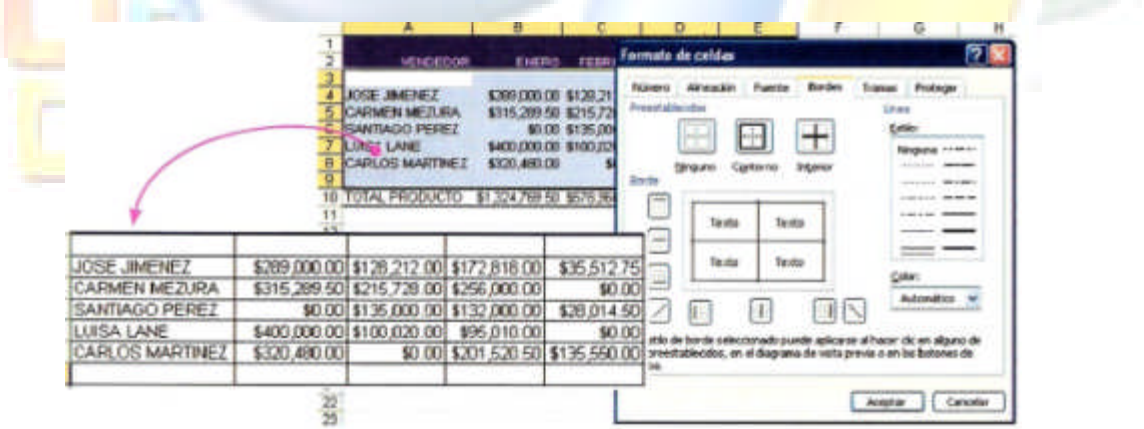
La mayor parte de los formatos se establecen de este modo, seleccionando un rango de celdas y aplicándoles un formato determinado. Una manera de simplificar este proceso es la opción autoformato.

1.2 **Autoformato.** Con esta opción, que se activa en el menú **Formato** → **Autoformato..**, podemos mejorar aún más la apariencia de la tabla con estilos predefinidos para aplicar sobre una tabla de datos. Observa lo siguiente:



El autoformato de tablas las opciones de los tipos de letra y apariencia de una tabla, pero no los formatos numéricos. Observa que los tipos de letra negrita y, a no aparecen en este formato. Sin embargo, es posible volver a colocarlos después de aplicar el *Autoformato*. Como una recomendación, es mejor aplicar primero el *Autoformato* y después los detalles de negritas y cursivas para mejorar la apariencia.

1.3 **Bordes de Celdas.** Ahora apliquemos bordes a la hoja. Volvamos al menú **Formato- Celda**. En la pestaña **bordes** se definen las líneas que se dibujan alrededor de las celdas para formar cuadrículas. Observa lo siguiente:



Al aplicar bordes de contorno se establece un marco alrededor de todo el rango y al especificar borde interior se cuadrícula, todas las celdas del interior del rango. Observa lo

que sucede después de aplicar este formato.

Alineación. Define la orientación del texto, horizontal, vertical o inclinada. Esta opción resulta muy útil cuando queremos poner en poco espacio las descripciones o títulos. Dentro de esta opción las más. Utilizadas son:

- Ajustar texto. Permite que un texto muy largo para una celda se ajuste al ancho de la celda, haciendo más ancha la fila.
- Reducir hasta ajustar. Reduce hasta ajustar. Reduce el tamaño de la fuente para que el texto quepa en una celda. La desventaja es que lo puede volver demasiado pequeño e ilegible.
- Combina celdas. al seleccionar dos o más celdas las integra para formar una más grande.



Ilustración 2.25 Formato de celdas.

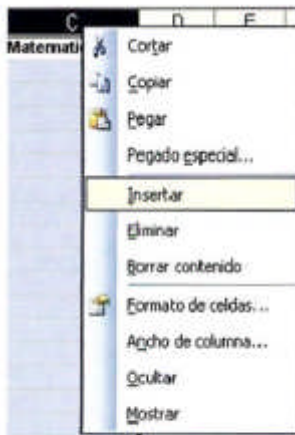


Ilustración 2.26 Menú insertar.

Tramas. Define rellenos o colores en el interior de las celdas. Esta opción resulta útil para resaltar los datos o títulos.

Proteger. En esta opción se puede bloquear o proteger una celda. Se verá con más detalle en el apartado de protección de hojas de cálculo.

1.4 Insertar celdas, filas y columnas. Insertar una fila o una columna puede hacerse con el menú Insertar y seleccionando el elemento a insertar (celda, fila o columna) o con el menú contextual que se activa con el botón derecho del mouse.

1.5 Eliminar celdas, filas y columnas. Se puede eliminar con el menú Edición – Eliminar o con el menú contextual y seleccionando la opción eliminar.

1.6 Cambiar el ancho de una fila o columna. Se puede utilizar el menú Formato-Columna-Alto, o seleccionando el elemento deseado y con el menú contextual.

1.7 Ocultar una fila o columna. Se selecciona la fila o columna haciendo clic sobre la letra o número de la fila o columna a ocultar y luego se utiliza el menú formato-fila-Ocultar o con el número contextual.

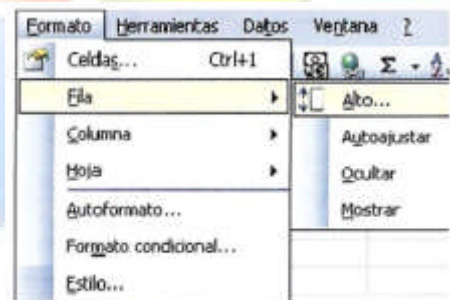


Ilustración 2.27 Formato de filas.

Orden los datos de una hoja de cálculo

Para ordenar los datos se utiliza el menú Datos-Ordenar: mediante esta opción se pueden organizar por orden alfabético, por totales o algún otro criterio.

1	A	B	C
2	VENDEDOR	ENERO	FEBRERO
4	JOSE JIMENEZ	\$289,000.00	\$128,212.00
5	CARMEN MEZURA	\$315,289.50	\$215,728.00
6	SANTIAGO PEREZ	\$0.00	\$135,000.00
7	LUISA LANE	\$400,000.00	\$100,020.00
8	CARLOS MARTINEZ	\$320,480.00	\$0.00
10	TOTAL PRODUCTO	\$1,324,769.50	\$578,960.00

Ordenar

Ordenar por: Columna A

Luego por: Ascendente

Luego por: Ascendente

El rango de datos tiene fila de encabezamiento: Sí No

Opciones... Aceptar Cancelar

SANTIAGO PEREZ	\$0.00	\$135,000.00	\$132,000.00	\$28,014.50
LUISA LANE	\$400,000.00	\$100,020.00	\$95,010.00	\$0.00
JOSE JIMENEZ	\$289,000.00	\$128,212.00	\$172,818.00	\$35,512.75
CARMEN MEZURA	\$315,289.50	\$215,728.00	\$256,000.00	\$0.00
CARLOS MARTINEZ	\$320,480.00	\$0.00	\$201,520.50	\$135,550.00

En este ejemplo se utilizó la opción eje *ordenar* de manera descendente, lo que propicia que se ordene de la Z.A. es decir, en orden alfabético inverso. Si ahora se decide hacerlo por orden *ascendente*, entonces se ordena de la A.Z, es decir, en orden alfabético normal. Observa q tic para ordenar es necesario primero *seleccionar* un rango de celdas con los (latos y después ordenarlos. En orden alfabético, utilizando e1 botón de la *Barra de herramientas estándar*, queda de la siguiente manera:

1	A	B	C	D	E
2	VENDEDOR	ENERO	FEBRERO	MARZO	ABRIL
4	CARLOS MARTINEZ	\$320,480.00	\$0.00	\$201,520.50	\$135,550.00
5	CARMEN MEZURA	\$315,289.50	\$215,728.00	\$256,000.00	\$0.00
6	JOSE JIMENEZ	\$289,000.00	\$128,212.00	\$172,818.00	\$35,512.75
7	LUISA LANE	\$400,000.00	\$100,020.00	\$95,010.00	\$0.00
8	SANTIAGO PEREZ	\$0.00	\$135,000.00	\$132,000.00	\$28,014.50
10	TOTAL PRODUCTO	\$1,324,769.50	\$578,960.00	\$867,348.50	\$199,077.25

En ocasiones se requiere ordenar por más de una columna un ejemplo es cuando queremos organizar una lista de nombres ordenados por apellido paterno y apellido materno, a modo de tener tina lista en orden alfabético. Veamos el siguiente ejemplo: se selecciona el rango de datos y después se entra a Datos Ordenar.

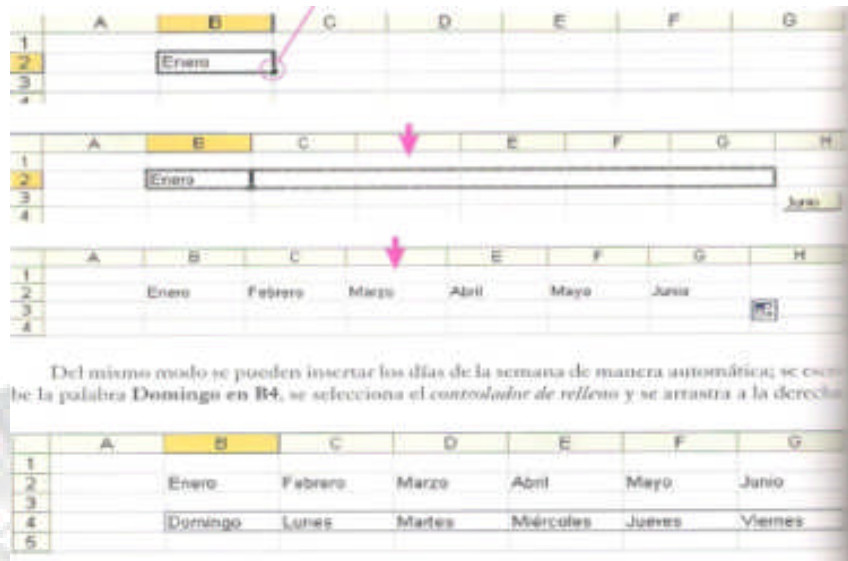
Apellido_Paterno	Apellido_Materno	Nombre
Alaud	Bianchi	Angela Isabel
Contreras	Rivera	Claudia
Cruz	Agarín	Zuleica Ceia
Cruz	Martínez	Rosario Magali
Díaz	Arroyo	Carlos Manuel
Martínez	Ceballos	Angélica Sarahi
Martínez	Loyo	Miriam Guadalupe
Palmeros	Acuña	Javier de Jesús
Palmeros	Martínez	Mariana
Robles	Cruz	Claudia
Rodríguez	Freyre	Angélica Sarahi
Rodríguez	Guerrero	Victoria
Sáez	Cristina	San Luis
Zepeda	Rodríguez	Rosario Magali
		Zuleica Ceia
		Carlos Manuel
		Alejandra Denisse

Genera series

Una herramienta para simplificar la elaboración de una hoja de cálculo es la realización de serie de manera automática, mediante lo cual es posible hacer los siguientes ejemplos.

Serie de meses o días. Se escribe el nombre del primer mes (enero) se selecciona el pequeño cuadro negro (controlador de relleno) en el borde inferior derecho de la celda

y se arrastra para generar la serie:



Series numéricas

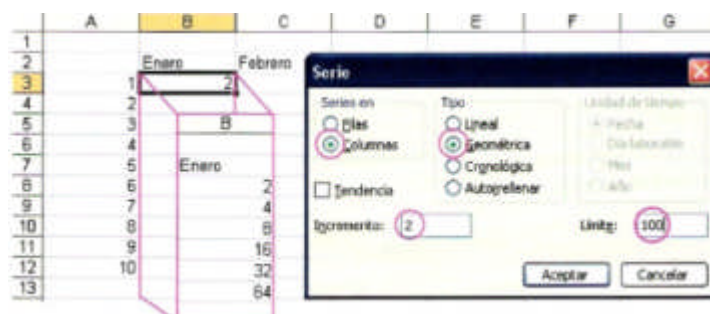
En este cuadro de diálogo se determina si la serie va en la fila (horizontal) o en la columna (vertical), se determina el tipo de la serie: *lineal* (se suma el incremento), *geométrica* (se multiplica el incremento) y el *límite superior* (valor máximo de la serie).

El valor máximo ayuda a de Excel sepa hasta dónde escribir los datos de la serie sin pasarse. Si no se pone valor límite, no se hace nada.

Observa lo que sucede al realizar una serie en columna, que parte de 1(B3=1), con incremento de uno en uno (1), de tipo lineal (se va sumando el incremento) y con límite superior 10.



Si ahora hacemos una serie geométrica, que tenga con número inicial 2; incremento 2 y límite 100:



Otra manera de realizar series de manera rápida sin recurrir al menú es escribiendo los dos primeros valores de la serie; se seleccionan estos valores y se arrastra del cuadro inferior izquierdo para formar la serie:

	A	B	C	D	E	F	G	H
1								
2		Enero	Febrero					
3								Julio
4								

Esta estrategia también funciona para números.

	A	B	C	D	E	F	G	H
1								
2		1	2					
3								
4								2

Si por ejemplo en lugar de escribir 1,2..., se escribe 2,4..., se obtendrá una serie con incrementos de dos en dos. Esta estrategia nos permite series geométricas.

2.5 Elabora y edita gráficos con las características solicitadas

Ventajas del empleo de gráficos.

Un gráfico es una representación visual de un conjunto de un número. Esta representación visual nos permite comprender con más claridad un fenómeno, natural, social o económico. Algunas de las ventajas de usar los gráficos es la facilidad de comparar resultados; otra ventaja es la capacidad de observar tendencias y adicionalmente, la Facilidad de representar funciones matemáticas.

Sin embargo, los gráficos también pueden representar modelos matemáticos complejos, como pueden ser representaciones moleculares y químicas, representación del ADN, modelos de la creación del universo del clima, los gráficos. Cuáles podemos observar visualmente mediante gráficos para facilitar su comprensión. Una de las ramas más fascinantes de la Informática son los gráficos por computadora, que requieren profundos estudios en matemáticas y programación de computadoras, y, la base de los juegos de video, que no son otra cosa que modelos matemáticos del movimiento, uso de coordenadas y geometría combinadas con la programación de computadoras.

Protege la información de la hoja de cálculo con contraseña

Excel proporciona dos opciones principales de seguridad y protección para controlar quien puede modifica los datos o formulas de una hoja cálculo.

- Una protección *global* para todo el Ebro mediante una contraseña, lo cual permite que solo los usuarios autorizados puedan consultar o modificar los datos. Las contraseñas en Excel puede tener hasta 255 caracteres.

- Una *protección de chaos* para proteger elementos específicos de la hoja de cálculo o del libro. Esta protección de datos específicos puede ser con contraseña o sin ella.

Protección Global con contraseña



Ilustración 2.41 Protección de contraseña.

Para tener Una *protección global* de todo el libro se entra al menú Herramientas → Opciones y se selecciona la ficha Seguridad.

En esta ficha se establecen contraseñas de apertura (para poder abrir el archivo) y contraseña de escritura (para poder modificar su contenido). Las contraseñas son la primera Línea de defensa para proteger tus documentos contra cambios deseados. Con las contraseñas fuertes se dificulta que otras personas obtengan que otras personas obtengan. Una contraseña fuerte es aquella que combina tres tipos de caracteres: letras mayúsculas y minúsculas, números y símbolos. Por ejemplo: **JM7qe^c/cp9uk** es una contraseña fuerte mientras que jose20 es una contraseña débil. Las contraseñas fuertes son más difíciles de romper, mientras que las contraseñas débiles son relativamente fáciles de adivinar.

Algunas recomendaciones para que construyas tus propias contraseñas seguras o fuertes son:

1. Que tengan, como mínimo, siete caracteres de longitud.
2. Que estén formadas tanto con letras mayúsculas como minúsculas, números, y al menos un símbolo especial como #, \$, %, & colocado entre el segundo y sexto carácter de la contraseña.
3. Utiliza una secuencia al azar de caracteres. Ejemplo rkth74%4, mientras menos sentirlo tenga, más difícil será de adivinar. (Una vez que la hayas formado, inventa tu propia regla para recordarla).
4. No uses caracteres repetidos.
5. No uses secuencias, fáciles de escribir como 4, 5, 6, 7..., cdef o asdf
6. Procura no cambiar letras por números o símbolos. por ejemplo, @ en lugar de "a" o = en lugar de "o", pues esto facilita adivinar la contraseña.
7. No uses partes del nombre de usuario que tienes en tu cuenta de correo o para acceso de alguna cuenta de Internet.

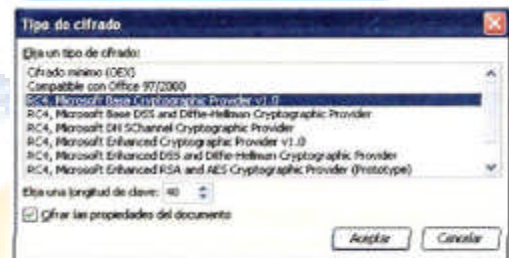


Ilustración 2.42 Tipos de cifrado.

En la opción Avanzada..., se puede seleccionar el tipo de cifrado para aumentar la protección la protección de los datos: el cifrado hace más difícil adivinar la contraseña.

Protección específica de datos

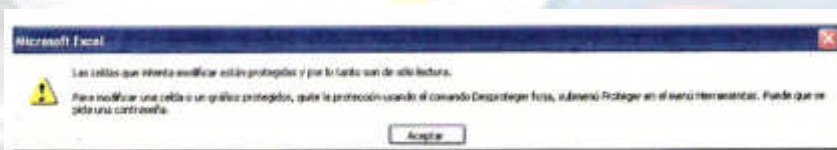
Para evitar que los usuarios realicen cambios por descuido u omisión en elementos específicos del libro o la hoja de cálculo, Se protege (o bloquea) de manera específica ciertas áreas, celdas o rangos. También puede especificar una contraseña para permitir que ciertos usuarios tengan permisos para modificar estas áreas.

El primer paso para proteger una hoja de cálculo es bloquear todas las celdas. Para esto, se utiliza el menú Herramientas- Proteger,

Al seleccionar la opción Proteger toda la hoja se activa el cuadro de dialogo; en él se especifica que se va a proteger la hoja y los permisos de los usuarios:

En este mismo cuadro de dialogo se puede establecer una contraseña para desproteger la hoja o rango específico de la hoja que un usuario podrá modificar, siempre y cuando conozca la contraseña.

Una vez protegida toda la hoja, si intentamos modificar el contenido de alguna celda, aparecerá el siguiente mensaje.



Para permitir que algunos usuarios modifiquen rangos específicos de la hoja de cálculo es necesario especificar los rangos y una contraseña para el rango. Para esto entramos a la opción Herramientas-Proteger-Permitir que los usuarios modifiquen



Ilustración 2.45 Modificación de rangos.

rangos..., se hace clic en Nuevo y se selecciona el rango específico la contraseña para modificarlo.

Una vez realizada esta acción, cuando un usuario quiere modificar el contenido de estas celdas con permiso de modificación.



Ilustración 2.46 Desbloqueo de rangos.

Al escribir la contraseña se obtiene el permiso de efectuar cambios a esta celda específica.

Para quitar la protección a la hoja de cálculo se utiliza el menú **Herramientas** → **Proteger** → **Desproteger la hoja...** y aparece el cuadro de diálogo, donde se escribe la contraseña para quitar la protección.

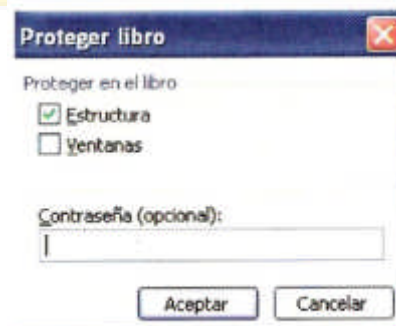


Ilustración 2.47 Protección de libros.

Además de proteger una hoja, también podemos proteger un libro completo contra modificaciones; se hace mediante el menú **Herramientas** **Proteger** **Proteger Libro...**, y especificar que elementos del libro se van a proteger y su contraseña. Al proteger la estructura de un libro se

desactivan los menús que permiten borrar hojas completas del libro de trabajo.



Ilustración 2.48 Protección compartida de libro.

La opción **Herramientas** → **Proteger** → **Proteger y compartir libro**. Permite prestar el libro a otros usuarios y llevar un control (le lo que cada usuario le modifica al regresar el archivo modificado. También puede establecerse una contraseña para permitir los cambios de los usuarios.

Como puedes ver tienes varias opciones para proteger tus datos. Es tu responsabilidad hacerlo.

BLOQUE 3 BASE DE DATOS

3.1 Definición de base de datos.

Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la gestión de los datos. Hace tiempo se utilizaban los ficheros secuenciales para almacenar datos. Estas nuevas formas de organización de datos permitían acceder la información, pero solo de forma secuencial, y, por ende, bastante tardía la consulta de los mismos. Se mejora un poco con los ficheros indexados, con los que el acceso ya podía ser aleatorio; así, el sistema de ficheros era el sistema más común de almacenamiento de datos.

Pero a medida que el volumen de los datos y la complejidad de los programas era cada vez mayor, surge lo que conocemos como los sistemas de gestión de bases de datos. Con esto, el propósito general de un sistema de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada en conjunto de datos que posteriormente se convertirán en información importante para la toma de decisiones.

Una base de datos (cuya abreviatura es *BD*) es una entidad en la cual se pueden almacenar datos de manera estructurada con la menor redundancia posible. Diferentes programas y usuarios deben poder utilizar estos datos. Por in tanto, el concepto de base de datos generalmente esta. Relacionado con el de red, a la que se debe poder compartir ésta información. De allí el término base. «Sistema de información» es el termino general empleado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

3.2 Características de la base de datos.

Funciones y ventajas de los sistemas de gestión de base de datos

Funciones

8. Debe proporcionar a los usuarios la capacidad de almacenar datos, acceder a ellos y actualizarlos.
9. Proporcionar un catalogo en el que se almacena la descripción de los datos y que sea accesible para los usuarios. A este catálogo también se le llama *diccionario de datos*.
10. Asegurar la integridad y validez de los datos.
11. Asegurar y mantener los datos originales.
12. Proporcionar un mecanismo que asegure la actualización de los datos cuando varios usuarios la están actualizando constantemente.
13. Debe contar con un sistema de seguridad en el acceso de la información solo a personal autorizado a través de estrategias (autenticación, contraseñas, etc.) que se implementan en el sistema.

Ventajas

14. Menor espacio de almacenamiento.
15. Mayor seguridad en los datos.
16. Datos más documentados.
17. Acceder a los datos más eficientes.
18. integridad de los datos
19. Independencia de los datos de los programas y procesos.
20. Menor redundancia
21. Acceso simultaneo a los datos.

Como podemos darnos cuenta. Un «sistema de gestión de bases de datos» es sumamente útil para poder manipular datos de una forma clara a sencilla, pero sobre todo de una manera ordenada. Para esto contamos con software que nos brinda herramientas para poder llevar a cabo todo el proceso referente a las bases de datos.

Se cuenta con varios programas para crear y *manipular* bases de datos. Entre ellos podemos encontrar a uno de los más populares tanto a nivel empresarial como doméstico, como Microsoft Access 2003, Otros programas que son Sistema de Gestión de Bases de datos son: Personal Oracle 7, Microsoft SQL Server, Oracle7, Workgroups Server.

3.3 Elementos básicos de Access

Abrir y Cerrar.

Son dos las formas básicas de iniciar Access 2003.

22. Desde el botón inicio colocado en la parte inferior izquierda de la pantalla. Al colocar el cursor y hacer clic sobre el botón Inicio se despliega un menú; al colocar el cursor sobre Programas, aparece otra lista con los programas que hay instalados en tu ordenador. Busca Microsoft Office y luego Microsoft Access; haz clic sobre él y se iniciará el programa.
23. Acceso directo, desde el icono Access 2003 del escritorio dando doble clic sobre

éste.

Para cerrar o salir del programa de Access 2003 tenemos las siguientes operaciones:

24. Hacer clic en el botón cerrar.
25. Pulsar las teclas de atajo ALT + F4
26. Ir al menú Archivo y seleccionar la opción salir.

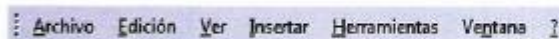
Pantalla inicial

Una vez iniciada el programa de Access se mostrará una pantalla inicial como la de la ilustración 3.2, en la cual se identificarán los componentes que la conforman.

A continuación se encuentran los siguientes elementos que la integran las barras:

Barra de menús

Contiene las operaciones de Access agrupadas en menús desplegables. Cada opción tiene una letra subrayada; esto nos indica que se puede acceder directamente a tal opción pulsando simultáneamente la tecla ALT y la letra subrayada del menú.



Los menús desplegables contienen tres tipos básicos de elementos para llevar a cabo otras opciones:

Comandos inmediatos. Se ejecutan de forma inmediata al hacer clic sobre ellos. Se identifican porque a la derecha del nombre del comando o no aparece nada o bien aparece la combinación de teclas para ejecutarlo desde el teclado. Por ejemplo, en el menú (?) ayuda, F1 para entrar en la ayuda de Microsoft Access.

Otro menú desplegable (▾) Al situarse sobre éste, se abre un nuevo menú a su lado con más opciones. Se reconocen porque tienen un triángulo a la derecha. Por ejemplo, en el menú (?) bases de datos.

Cuadro de dialogo Al hacer clic en la opción aparece un cuadro de dialogo que nos pedirá mas información y que tendrá botones para aceptar o cancelar in acción. Se distinguen porque el nombre acaba con puntos suspensivos. Por ejemplo, en el menú (?) Detector y reparar...

Barra de herramientas

Contienen iconos para ejecutar de forma inmediata algunas de las operaciones más usuales, como guardar, abrir, imprimir, etcétera.



También existen otras barras como la de Panel de Tareas, que se encuentra en la parte derecha de la pantalla.

Ahora conoceremos algunos conceptos necesarios para la creación de una base de datos en Access 2003, conocimientos indispensables para la consecución de la exploración del programa de Access.

Las bases de datos de Access 2003 tienen la extensión .MDB para que el ordenador las reconozca como tal. Casi todos los sistemas de gestión de base de datos modernos almacenan y tratan la información utilizando el modelo de gestión de bases de datos relacional. En un sistema de base de datos relacional los datos se organizan en Tablas.

3.4 Elementos de la tabla de datos

Tablas

Es una colección de datos sobre un tema específico, como productos o proveedores. Al usar una tabla independiente para cada tema los datos se almacenan solo una vez. Esto tiene como resultado una base de datos más eficaz y menos errores de entrada de datos.

Las tablas organizan datos en columnas (denominadas campos) y filas (denominadas registros).

Cada campo de una tabla contactos contiene el mismo tipo de información para cada Contacto, por ejemplo su nombre. Cada uno de los registros de esa tabla contiene toda la información acerca de un Contacto por ejemplo el id del Contacto, Nombre del Contacto, Apellidos, Dirección, etc.

IdContactos	Nombre	Apellidos	Dirección
1	Resalia	Pérez Muñoz	Norte 10 No. 5
2	José Alberto	López Cruz	Sur 14 No. 145
3	Pedro	Terres Montalvo	Oriente 18 No. 207

Ejemplo de tabla.

3.5 Diseño de base de datos

Durante el proceso de diseño de base de datos se deben tomar en cuenta algunos principios como evitar la información duplicada, ya que los datos redundantes ocupan mayor espacio y aumenta la probabilidad de que se produzcan errores e incoherencias; otro principio importante es que la información sea correcta y completa, ya que si la base de datos contiene información errónea, por ende, los informes mostrarán información incorrecta, y por lo tanto las decisiones que se toman a partir de estos informes estarán mal fundamentadas.

Un buen diseño de base de datos es, por tanto, aquel que:

- Determina la finalidad de la base de datos.
- Divide la información en tablas basadas (-ti temas para reducir los datos redundantes.
- Proporciona a Access la información necesaria para reunir la información de las tablas cuando así se precise.
- Define relaciones entre las tablas.
- Especifica claves principales.
- Ayuda a garantizar la exactitud e integridad de la información.
- satisface las necesidades de procesamiento de los datos y de generación de informes.

Al momento de crear una tabla de claros con su respectivo diseño, es necesario implementar las relaciones entre las diferentes tablas que conforman a la base de datos. Dichas relaciones resultan importantes para un buen manejo de los datos.

Después de esto, procedernos a realizar un esquema conceptual, que básica ti tente es detallar de forma ordenada la descripción de los datos, su tipo, relaciones con otros datos y restricciones. Esto será la base para nuestro modelo racional más adelante.

Modelo relacional

El modelo relacional es un modelo lógico de demás que utiliza registros, lo que quiere decir que la estructura de nuestra base de datos utiliza este modelo estará definida por registros de formato fijo, donde cada registro se define por medio de un número también fijo de campos o atribuidos y cada campo tiene un tamaño estático. En otras palabras, el modulo relacional es una colección de tablas con las cuales representamos demás y relaciones entre ellos.

Los componentes del modelo relacional:

Entidades. Es el componente básico y se refiere a cualquier objeto distinguible del cual podemos almacenar cierta información: puede ser una persona, lugar, cosa, evento, concepto, etc. También puede ser algo físico como un libro, carro, casa, empleado, o algo abstracto como un trabajo o un curso universitario. Algo que siempre debemos tener presente es que cada entidad es única y debe contener cierta información descriptiva.

Atributos. Después de identificar una entidad, necesitamos describirla en términos reales a través de un atributo. Un atributo es una propiedad o una característica que define una entidad. Ejemplo: asumamos que identificamos una entidad a la que llamaremos vehículo, entonces algunos de sus atributos será: marca, modelo y color. Pues son características que podrían diferenciar un vehículo de otro.

Relaciones. Después de identificar entidades y sus respectivos atributos, necesitamos determinar cómo se relacionan entre ellas. Una relación es cualquier asociación, vínculo o conexión entre las entidades de interés en un sistema. Podemos pensar en una relación como un verbo que representa la acción que una entidad ejerce sobre otra. Ejemplo: en una biblioteca, un estudiante (entidad) pide prestado (relación) un libro (entidad).

Debemos saber que cada relación debe tener un nombre, un grado y una cardinalidad. El grado se refiere al número de entidades asociadas a la relación, y las más comunes son las relaciones binarias (asocian dos entidades), pues las relaciones ternarias o de mayor grado se descomponen en dos o más relaciones binarias.

La cardinalidad de una relación determina cuántas instancias de una entidad son asociadas con cuántas instancias de otra entidad en una relación.

Existen tres tipos de cardinalidad:

De 1 a 1 (1:1) Uno a uno. A cada ocurrencia de una entidad le corresponde entre máximo una ocurrencia de la otra entidad relacionada. Ejemplo: la entidad «hombre» y la entidad «mujer» se unen por la relación «matrimonio» -

De 1 a N (1: N) Uno a varios. A cada ocurrencia de una entidad le pueden corresponder varias de la entidad relacionada. Ejemplo: la entidad «empresa», la entidad «trabajador», y entre ellos la relación «trabajar-en».

De N a N (N: N) Varios a varios. Cada ocurrencia de una entidad puede contener varias de la otra entidad relacionada y viceversa. Ejemplo: la entidad, empleados>> puede ser asignada a varios proyectos al mismo tiempo; y cada proyecto tiene menos de dos empleados asignados a él.

Estos tipos de relaciones, con la ayuda de estructura, nos permiten planear y organizar las informaciones, para realizar consultas rápidas y eficientes de nuestra base de datos. Access 2003 es un sistema de administración de bases de datos relacionales. En una base de datos relacional la información se divide en tablas distintas en función del tema.

3.6 Tipos de consulta

Una consulta puede ayudarte a recuperar los datos que cumplen las condiciones de búsqueda que se especifique. También se puede utilizar una consulta para modificar datos en una tabla, es decir, actualizar o eliminar algún registro.

Existen varios tipos de consultas:

Consultas de selección. Son las consultas que extraen o muestran datos. Nos muestran los datos de acuerdo con el criterio que se especifique. Si el usuario prefiere puede hacer modificaciones o no. Este tipo de consulta genera una tabla lógica y se le llama así porque no se encuentra almacenada en el disco duro, sino en la memoria del ordenador, y cada vez que se abre se recalcula.

Consultas de acción. Son consultas que realizan cambios a los registros. Existen varios tipos de consultas de acción, de eliminación, de actualización, de datos anexados y de creación de tablas.

3.7 La finalidad de la base de datos

Como toda tarea informática siempre es necesario determinar su objetivo para ser creada. En cuanto a las bases de datos surgen las interrogantes para que se va a crear, quién la va a utilizar y como piensan. Lo anterior con la finalidad de obtener una base de datos que nos permita ordenar, consultar y modificar nuestros datos de una forma sencilla para el usuario y así poder tomar decisiones de una manera acertada para beneficio de las organizaciones.

3.8 Aplica las bases de datos para manejar y administrar información.

Cuando se decide crear una base de datos, éstos se almacenan en tablas que usual formadas por filas y columnas: las filas son los registros, y las columnas, los campos, pero basadas en temas o entidades. Por ejemplo, podemos crear una tabla llamada contactos para tener los datos almacenados, como pueden ser los nombres, direcciones, números de teléfono, etc. Siempre debes empezar a diseñar una base de datos creando primero tus tablas.

Creación de un base de datos en Access

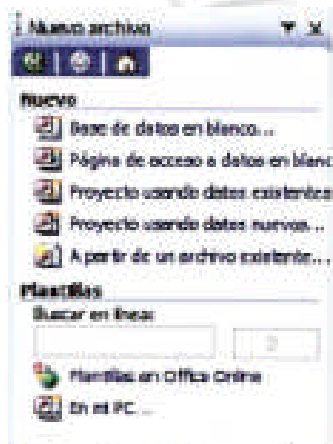


Ilustración 3.6 Crear una base de datos desde el panel de tareas.

Al iniciar el programa de Access 2003 se presentara la siguiente ventana, hacer clic en la opción Base de datos en blanco del Panel de Tareas, que se encuentra en la parte derecha de la pantalla inicial (Ilustración 3.6).

Si la pantalla interior no se encuentra visible podemos utilizar el menú archivo y dar clic en la opción Nuevo. O bien, hacer clic en el ícono de Nuevo que se encuentra en la Barra de herramientas.

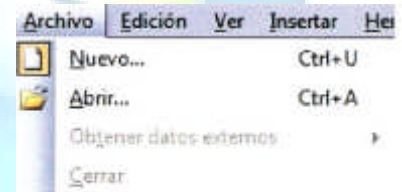


Ilustración 3.7 Crear una base de datos desde Menú archivo.

Inmediatamente aparecerá la siguiente ventana o cuadro de diálogo, en la cual colocaremos el nombre de la base de datos que se va a crear. Recordemos utilizar un nombre relacionado con el contenido de la misma para su fácil recuperación.

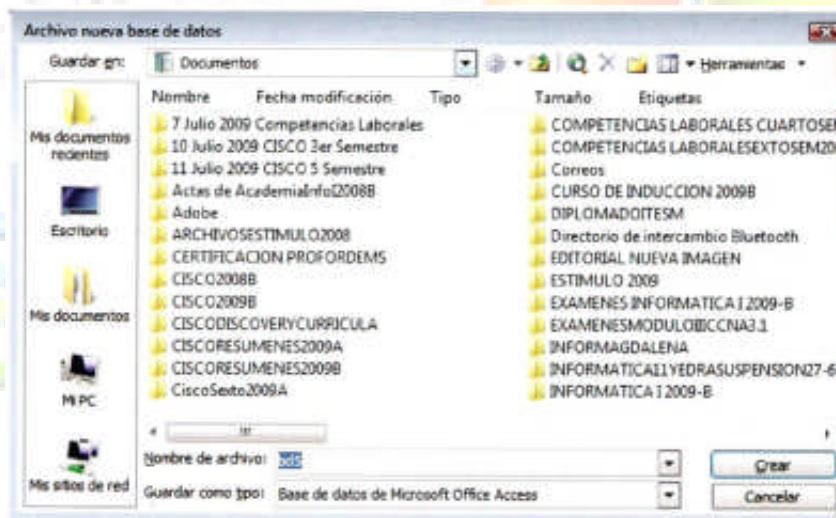


Ilustración 3.8 Cuadro de diálogo.

recuperación.

En el cuadro Guardar en.. se da un doble clic sobre la carpeta a seleccionar, en la cual se va guardar la base de datos. En lo que corresponde al cuadro Nombre de archivo., se escribe el nombre de la base de datos y posteriormente se hace clic en el botón de Crear. Inmediatamente aparecerá la ventana de Base de Datos (ilustración 3.9) y se creará la nueva base de datos, a la cual Access asignará la extensión .mdb.

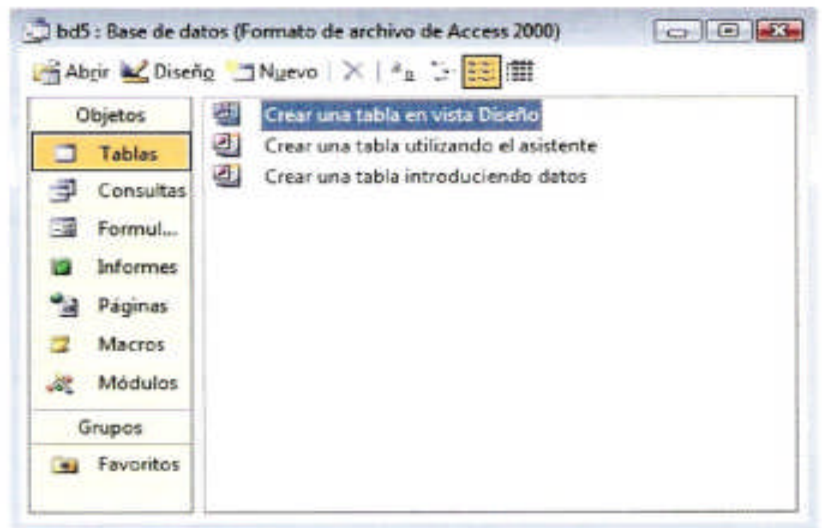


Ilustración 3.9 Ventana de base de datos.

En la misma (parte izquierda) se encuentran los tipos de objetos que se pueden crear (tablas, consultas, formularios, informes, etc.), y en la parte de la derecha, dependiendo del tipo de objeto que hayamos seleccionado, Access mostrará los objetos que se hayan creado de acuerdo con el tipo: también nos permitirá crear nuevas formas.

En este caso el tipo que seleccionamos será Tablas, que es el principal elemento a partir del cuál se crean todas las demás. Cuando ya existan tablas creadas, estas aparecerán en la parte derecha de la semana debajo de las opciones. . . Crear.

Abrir y cerrar una base de datos.

Veremos tres formas de abrir la Base de datos.

27. Ir al menú archivo y dar clic en la opción Abrir (ilustración 3.10).

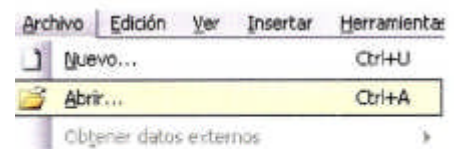



Ilustración 3.10 Abrir una base de datos desde el Menú archivo.

28. Hacer clic en el ícono de  de la Barra de herramientas.

29. O bien, desde el Panel de Tareas.

En este caso veremos dos formas de cerrar la base de datos.

30. Ir al menú Archivo y dar clic en la opción Cerrar

31. También haciendo clic en el ícono de  de la ventana de base de datos.

Un aspecto importante que se debe tomar en cuenta para continuar con la creación de nuestra base de datos es la *búsqueda, selección y organización de la información de la base de datos; es decir;* para poder dividir la información en tablas es necesario con anterioridad planear y organizar nuestra información, de tal forma que se cumpla con la finalidad de la creación de la base de datos y que por consecuencia sea útil y cumpla con los requerimientos específicos establecidos.

Creación de tablas

Vista hoja de datos. En esta opción se pueden introducir directamente los datos en la Labia, de acuerdo con el valor que se introduzca en in columna será el tipo de datos de la misma.

Vista diseño. Permite crear los campus manualmente y configurar el diseño de la tabla.

Asistente para tablas. Un asistente nos guiará paso a paso en la creación de la tabla, utilizando tablas ya predefinidas

Importar tabla. Nos permite crear una tabla a partir de otra ya existente en otra base de datos.

Vincular tabla. Consiste en crear una y vinculación entre las tablas originales y las importadas. Las modificaciones que se realicen en los datos se transmiten a aquellas.

La opción **Vista diseño** define la estructura de la tabla; es decir, el usuario puede definir las distintas columnas (campos) que ésta tendrá y otras consideraciones como claves, etc. Después, utilizando la Vista hoja de datos se puede capturar los datos. Pero



Ilustración 3.12 Vista de una hoja de datos.

también se puede empezar con la Vista hoja de datos introduciendo la información; Access creará la estructura automáticamente.

Vista hoja de datos.

Es esta vista se diseñan los campos pero no se pueden introducir datos. El menú de la Vista hoja de datos contiene lo relacionado con registro, ya que en esta vista se pueden introducir datos. En la barra de herramientas se añaden elementos de orden de los datos (ilustración 3.13).



Ilustración 3.13 Barra de herramientas.

Vista diseño

Desde esta lista no se pueden modificar los tipos de datos que contienen los campos o su descripción. Otra forma que existe de llegar a la vista diseño es desde la ventana de base de datos seleccionando el objeto tablas, pulsando doble clic en la opción crear una tabla en vista diseño, y aparecerá la ventana vista diseño.

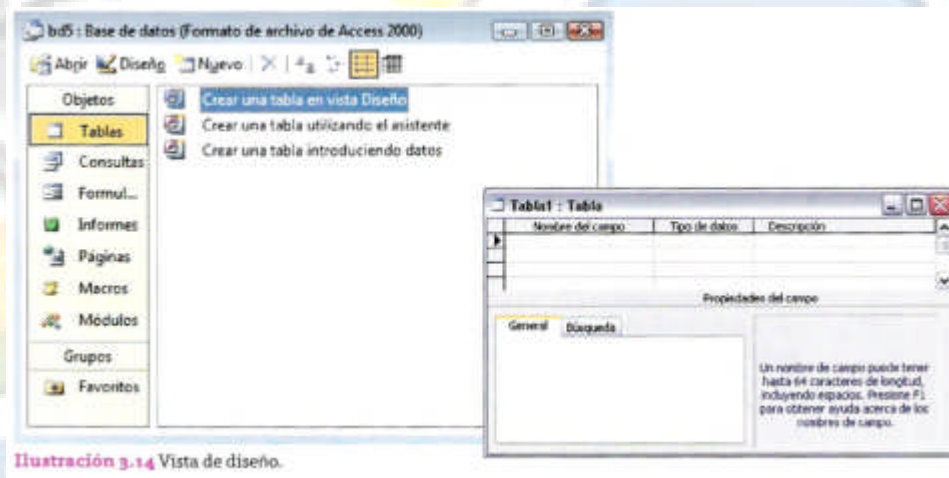



Ilustración 3.14 Vista de diseño.

Es posible definir un campo utilizando el ícono de Generador de Campos  que se encuentra en la Barra de herramientas, el cual permite definir campos a partir de unas tablas ya establecidas, o bien, el usuario puede definir directamente los campos.

Insertar el nombre del campo. Primero se escribe el nombre del campo en la primera fila, se da clic sobre la celda correspondiente y se introduce el nombre.

Elegir el tipo de datos. Si en la Vista hoja de datos no se introducen datos, Access asigna automáticamente el tipo de datos Texto. Si se hubiera introducido un dato numérico entonces el tipo de datos sería numérico. Ahora bien, si se desea cambiar el tipo de datos se hace clic en la flecha que se encuentra en la parte derecha de la celda; al pulsar sobre ella se despliega un menú en el que se puede seleccionar el tipo de datos.

3.9 Tipos de datos en Access 2003.

En Access debes especificar el tipo de dato que vas a utilizar y ser asignado a un campo. Existen varios tipos de datos, los cuales veremos a continuación:

Texto. Permite datos como caracteres, dígitos Ni caracteres especiales. Tiene una longitud máxima de 255 caracteres.

Memo. Tipo de dato que se emplea para almacenar texto de más de 255 caracteres, como comentarios o explicaciones. Puede almacenar hasta 65,535 caracteres.

Numérico. Este tipo de dato utiliza únicamente números que se usan en cálculos matemáticos

Fecha/Hora. Tipo de dato para la introducción de datos tipo fecha y horas, desde el año 100 al 9999.

Moneda. Para valores de formato monetario y datos numéricos.

Autonumérico. Número secuencial; incremento uno a uno, único, número aleatorio que Microsoft Access asigna cada vez que se agrega un nuevo registro a una tabla.

Sí/No. Tipo de dato lógico, que contenga uno de dos valores (Sí/No, Falso/Verdadero).

Objeto OLE. Objeto que puede ser un Gráfico, una Hoja de cálculo, de Microsoft Excel, imágenes, un documento de Microsoft Word, sonidos, etcétera.

Hipervínculo. Texto o combinación de texto y números que es utilizado como dirección de liga o enlace.

Los tipos de datos son de suma importancia al momento de ser empleados para ser asignados al campo ya que dependiendo del campo será asignado el tipo de dato. Access valida de alguna manera el contenido del campo para llevar un mejor orden en nuestra base de datos.

Claves principales

Surge otro concepto Importantísimo es la estructura de las tablas, que son las claves principales. Una clave principal es un valor único que nos permite identificar de manera particular un registro, de tal manera que con esta clave podamos saber si ninguna confusión el registro al que identifica, lo que también permite no tener registros duplicados o nulos.

No se puede definir más de una clave principal, pero sí puede estar compuesta por más de un campo. Los pasos para poder asignar una **clave principal** son los siguientes:


1. Hacer clic sobre el nombre del campo que va a ser la clave principal.
2. Hacer clic sobre el icono Clave Principal que se encuentra la Barra de herramientas.
3. De manera inmediata aparecerá una llave al lado izquierdo del campo que elegimos como la Clave Principal.
4. Si se quiere definir una clave compuesta; es decir, formada por varios campos, se seleccionan los campos pulsando simultáneamente la tecla CTRL, y el campo a seleccionar, así, una vez seleccionados todos los campos, se hace clic en el icono.

Es necesario recordar que el campo o combinación de campos que forman la clave principal de una tabla no puede contener valores nulos y no puede haber dos filas con el mismo valor en la clave principal. Para ello, cuando incurramos en alguna de estas fallas, Access nos mandará un mensaje.

Guardar una tabla

Hay que efectuar los siguientes pasos:

Ir al menú Archivo y dar clic en la opción Guardar.

Otra forma es hacer clic sobre el icono de Guardar  que se encuentra en la Barra de herramientas.

Si la tabla aun no tiene nombre, aparecerá el siguiente cuadro de dialogo:

*Escribir el nombre de la Tabla

*Pulsar un botón Aceptar.

Cerrar una tabla

Si quieres cerrar una tabla, realiza los siguientes pasos:

Ir al menú Archivo y dar clic en la opción Cerrar.

También puedes realizar esta acción dando clic sobre el icono de Cerrar de la ventana de tabla de datos.

Relaciones entre tablas

Una vez creadas las diferentes tablas, de acuerdo con nuestro diseño de base de datos, a veces necesitamos combinar la información contenida en las diferentes tablas para realizar informes, formularios, etc., los cuales se derivan de ellas. Para lograr este proceso, el primer paso es definir relaciones en dos tablas (recordemos los tipos de relaciones de uno a uno, de uno a varios y de varios a varios).

Crear relaciones entre tablas.

Para crear relaciones en Access 2003 se tiene que acceder a la opción de Relaciones, a la cual podemos llegar con los siguientes pasos:

1. En la ventana Base de datos ir al menú Herramientas y seleccionar la opción Relaciones.
2. También se puede realizar haciendo clic sobre el botón de la Barra de herramientas.
3. Derivado de lo anterior aparecerá el cuadro de dialogo **Mostrar tabla**, en el cual se muestran las tablas que formaran parte de la relación a establecerse.
4. Se procede a seleccionar las tablas dando clic sobre ellas y pulsar el botón de **Agregar**.
5. Se repiten los pasos anteriores hasta seleccionar todas las tablas que intervendrán en la relación a crear.

6. Hacer clic en el botón Cerrar.

Después aparecerá la ventana de **Relaciones** con las tablas agregadas en los pasos anteriores.

Para crear la relación se realiza el siguiente paso:

Seleccionar el campo de relación de la tabla principal (**Alumnos**). En este ejemplo es Asignatura, y se arrastra el campo hasta el campo Nombre del profesor de la tabla secundaria (Docentes).

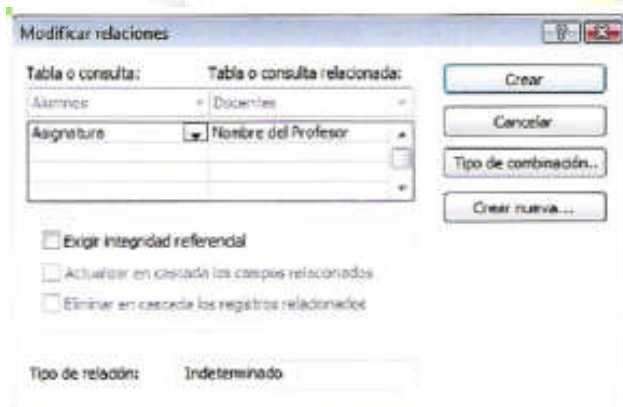


Ilustración 3.20 Cuadro de diálogo Modificar relaciones.

De igual manera aparecerá el cuadro de diálogo **Modificar relaciones**, en cuya parte superior se encuentran los nombres de las tablas relacionadas, y debajo de éstas los nombres de los campos que seleccionamos para la relación (Asignatura y Nombre del profesor).

En la Ilustración 3.20 observemos los botones del cuadro de diálogo; se encuentra uno que dice Tipo de relación, en el cual podemos definir el tipo de relación (1:1, 1:N, N:N).

De la siguiente manera se activa la opción **Exigir integridad referencial**. Con esto le decimos a Access que no permita introducir datos en la tabla secundaria si antes no se ha introducido el registro de la tabla principal.

Las otras dos opciones (Actualizar en cascada los campos relacionados y Eliminar en cascada los registros relacionados) si así lo quieres puedes activarlas.

Una vez finalizado haz clic en el botón de **Crear**. Se creará la relación y aparecerá la **Ventana de relaciones** (Ilustración 3.21). Del Menú Relaciones puedes ver todas las relaciones o las relaciones directas. El signo 1 significa **Tabla Principal**, y el signo infinito significa **Tabla Secundaria**.

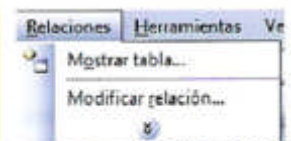


Ilustración 3.22 Menú relaciones.

Para agregar una relación entre dos tablas:

1. Ir al menú Relaciones y abrirlo.
2. Seleccionar la opción Mostrar tabla...
3. Escoger la tabla deseada y dar clic en Agregar. La tabla se mostrará junto con las otras ventanas.

También se puede **Modificar las relaciones o Eliminarlas** dando clic derecho sobre la línea de la Relación.



Ilustración 3.23 Cuadro de diálogo Modificar relaciones.

3.10 Creación base de datos en relación con los requerimientos solicitados

Hace el momento hemos visto las características generales de una base de datos, y los elementos que nos proporciona MS Access como un motor de base de datos.

Ejemplo: el profesor de Informática que desde hace mucho tiempo registra sus calificaciones en sus listas de papel, al crecer el número de alumnos a los que imparte clases decide automatizar este proceso. Para ello solicita a un grupo de alumnos avanzados en el manejo de base de datos que diseñen la correspondiente a utiliza. Juan (el líder del proyecto) pregunta al profesor qué desea registrar; él le contesta que iniciaran capturando los datos de los grupos que lleva, las materias que maneja y los alumnos que reciben tales materias. De esta información se muestra el siguiente extracto:

Grupos	
Clave	Nombre
101	101
201	201
301	301
...	...

Materias		
Clave	Nombre	Créditos
I-I	Informática I	8
I-II	Informática II	8
I-III	Informática III	8
...

Alumnos				
Clave	Folio	Nombre	Parcial	Calificación
1	00001	Esteban Pérez	1	10
2	00002	Claudia Ramirez	1	10
3	00003	Juan Garcia	1	9
...

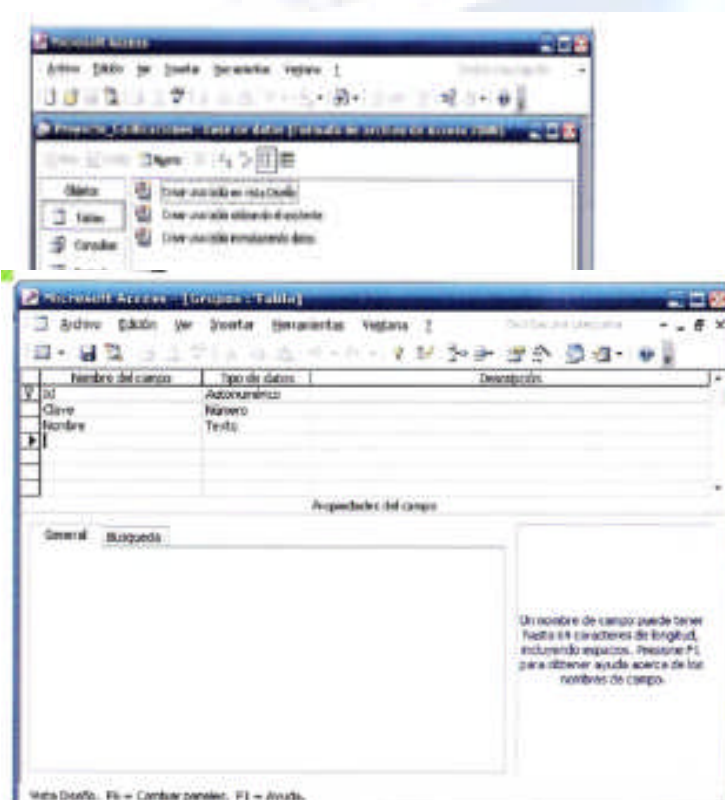


Ilustración 3.15 Creación del nombre del campo y tipo de datos en Vista diseño.

El equipo de estudiantes, con base en la información proporcionada, elabora la siguiente base de datos (Proyecto Calificaciones), selecciona los objetos Tablas y crea las tablas correspondientes.

Iniciando por la tabla que almacenará la información de los grupos, da un clic en Crear una tabla en Vista diseño.

Se dan de alta dos campos, uno denominado Clave de tipo de dato

Número, que almacenará las claves de los grupos, y un segundo denominado Nombre, de tipo de dato Texto, que almacenara los nombres de los grupos.

El tercer campo denominado Id fue creado automáticamente por la base de datos para servir como clave principal al conjunto de datos de los grupos.

La tabla de Alumnos se creó con base en la información proporcionada, como se muestra en la Ilustración 3.26.

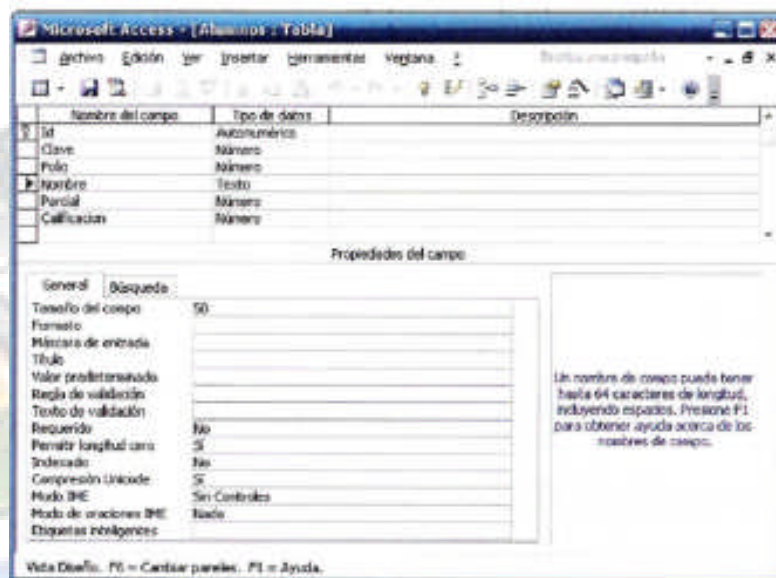


Ilustración 3.26 Creación de la Tabla alumnos.

En la Tabla de Materias (Ilustración 3.27), notemos que los alumnos «el campo Clave» es de tipo de dato Texto. Esto es porque el valor que se almacenara ahí no es un número (I-II). Los créditos en cambio sí son de tipo Número.

Es importante mencionar que las tablas creadas plasman la información solicitada; sin embargo, es importante señalar que las tablas mostradas aun deben ser analizadas para poder cumplir plenamente con los requerimientos solicitados.

El equipo de diseño analiza las tablas y se da cuenta de que debe contar con campos adicionales que permitan relacionarlas. Pedro indica que con la tabla de Alumnos, tal como está, no es posible saber a que Grupo pertenece cada alumno, por lo que propone crear un campo que almacena la clave del grupo del Alumno. Ana dice que está muy bien, y que lo mismo puede hacer para saber de qué materia es la Calificación. (Ilustración 3.28).

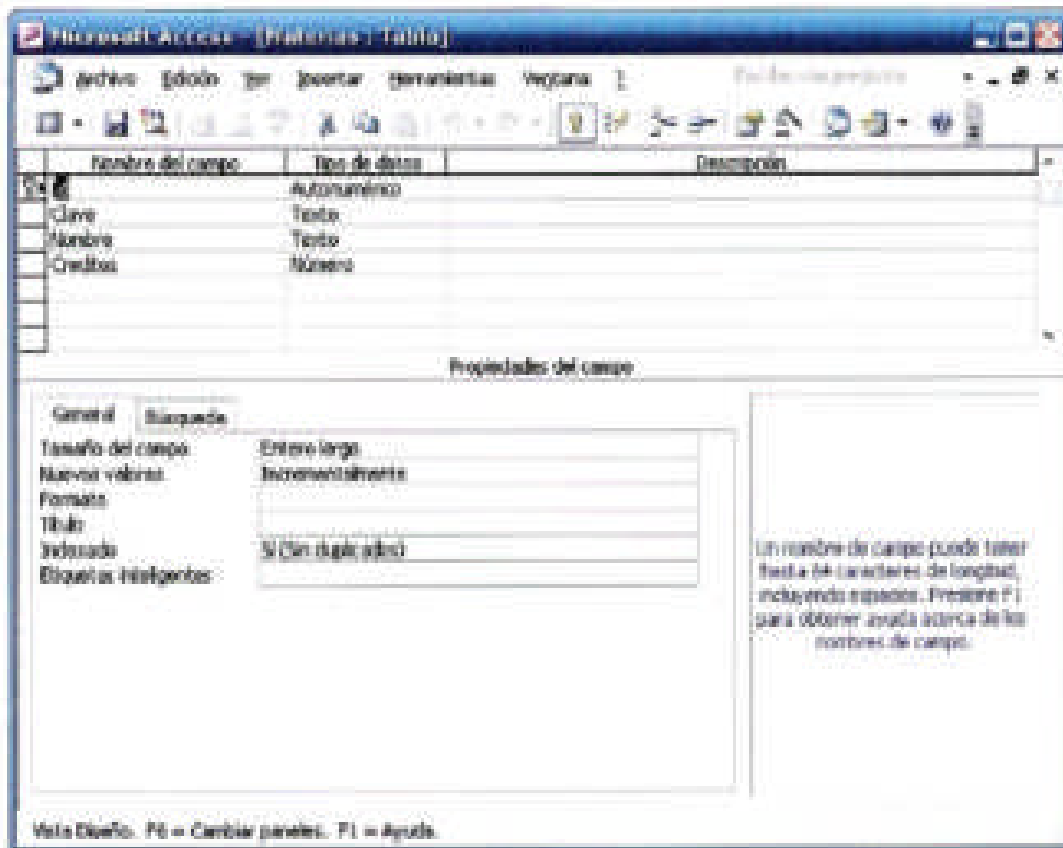


Ilustración 9.27 Creación de la tabla Materias.

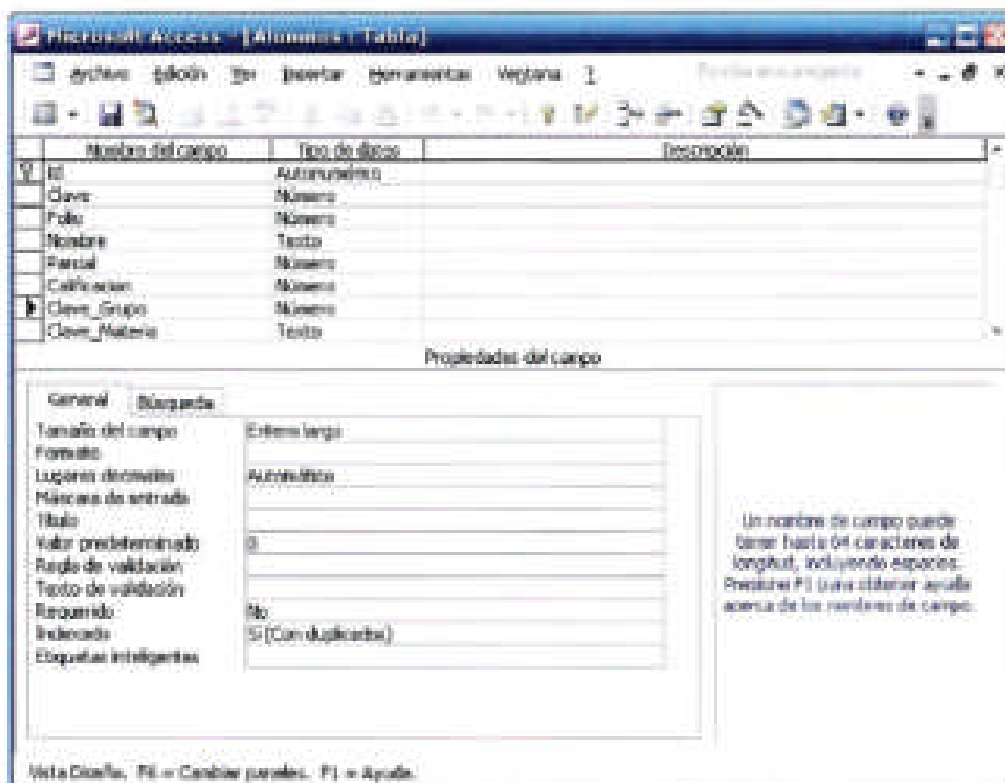


Ilustración 9.28 Creación del campo Clave_grupo.

Al

ver la

tabla, Juan expresa que ahora sí pueden establecer las relaciones entre las tablas y les muestra los pasos para hacerlo.

Primero va al menú principal y selecciona el submenú Relaciones.



Ilustración 3.29 Submenú Relaciones.

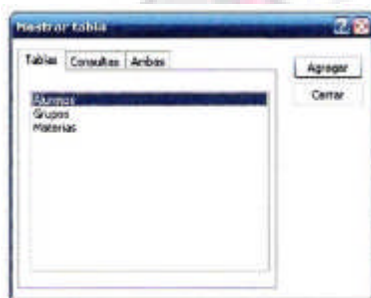


Ilustración 3.30 Pantalla mostrar tabla.

Utilizando la pantalla de mostrar tabla, con el botón agregar muestra las tres tablas de nuestro proyecto.

Para hacerlo selecciona la tabla y da clic en Agregar. Al final selecciona el botón Cerrar de la ventana Mostrar tabla.

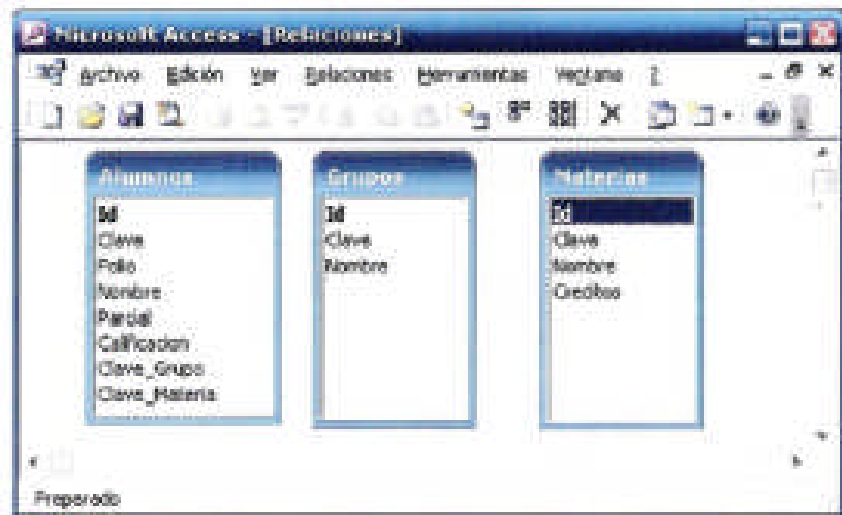


Ilustración 3.31 Ventana de relaciones (Alumnos_Grupos_Materias).

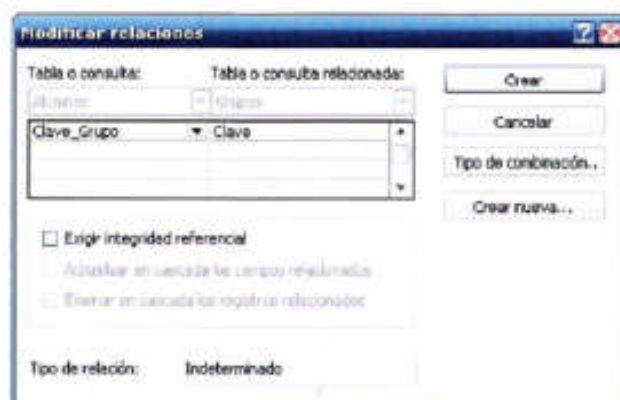


Ilustración 3.32 Ventana modificar relaciones.

Pero aún hace falta establecer las relaciones. Juan indica que el campo Clave_Grupo de la tabla Alumnos debe ser un dato que se encuentre en el campo Clave de la tabla Grupos, o de otra manera se generara una inconsistencia de información, y que lo mismo aplica para el campo Clave_Materia con el campo Clave de la tabla Materias. Eso son las relaciones, indica.

Para establecer una relación se selecciona el campo de la tabla, y manteniendo el botón derecho del mouse presionado se arrastra sobre el campo de la tabla con el que se quiere relacionar, en este caso de Clave_Grupo en Alumnos al campo Clave en Grupos (Ilustración 3.32).

Repetiendo el proceso para el campo Clave_Materia los alumnos consiguen establecer relaciones con ambas tablas.

Con esto termina la primera parte del proyecto. Juan indica que la siguiente actividad es capturar la información a la base de datos. Entusiasmado Ana, a quien le encanta el diseño por computadora, señala que llegó su turno de controlar la computadora, Será ella quien diseñe los formularios.

Primero selecciona Formularios en la Barra de objetos.

Después crea un formulario con el asistente (Ilustración 3.35). Les comenta a sus compañeros que para iniciar lo mejor es utilizar el asistente; cuando tengan mayor experiencia podrán modificar el formulario desde el modo diseño.

Para empezar escoge la tabla de Grupos y selecciona los campos de la tabla; posteriormente el diseño en columnas y un diseño estándar, logrando un excelente resultado (Ilustración 3.36).

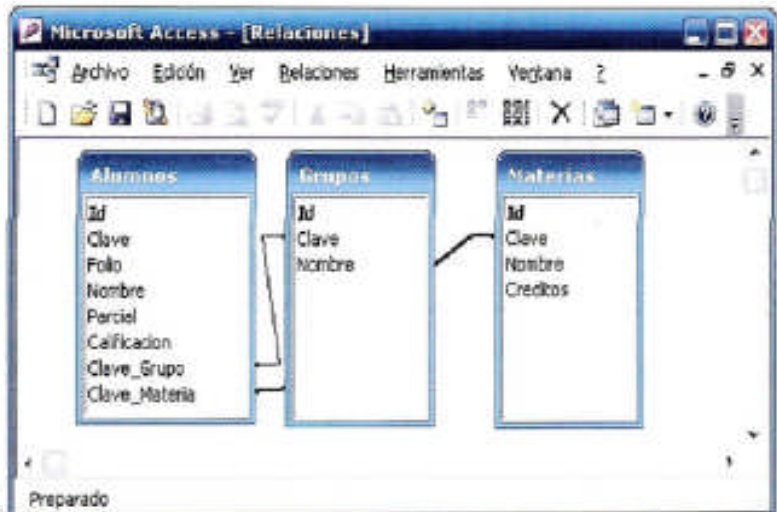


Ilustración 3.33 Relaciones creadas (Alumnos_Grupos_Materias).

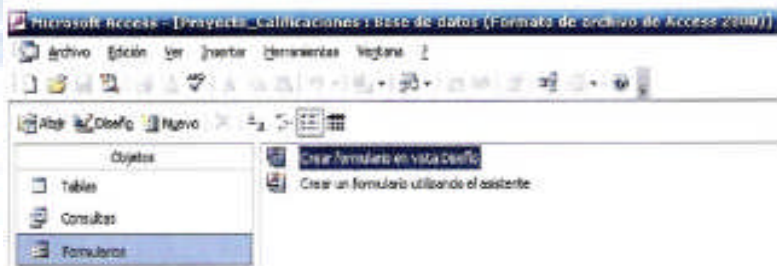


Ilustración 3.34 Formularios de la barra de objetos.

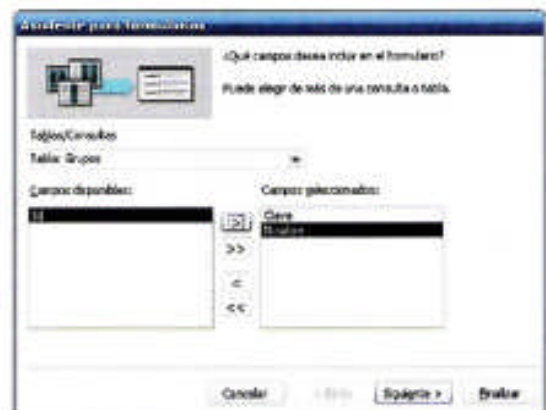


Ilustración 3.35 Creación de formularios desde el asistente.

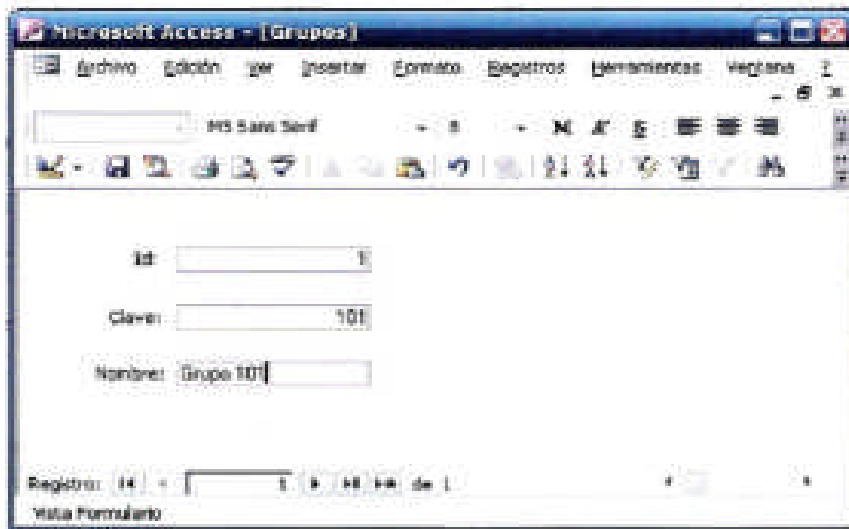


Ilustración 3.36 Ventana vista formularios.

Antes de realizar los formularios adicionales, Pedro comenta qué pasaría si la base de datos cayera en manos de alumnos que pudieran modificar sus calificaciones. Para prevenir eso Juan decide establecer una contraseña de acceso. Para hacerlo ingresa al siguiente menú.

Al llevarlo a cabo nota que primero debe abrir la base de datos en modo Exclusivo, como se muestra en la Ilustración 3.38. Al ejecutarlo, pudo establecer la contraseña de la base de datos.

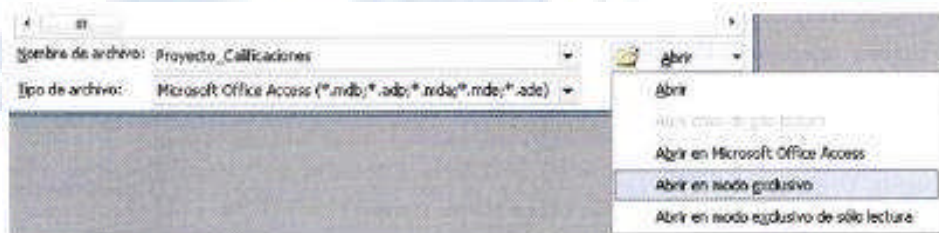


Ilustración 3.38 Abrir un modo exclusivo.

En esos momentos acudió el profesor, interesado en el avance del proyecto, e hizo la observación al equipo de que aun no había alguna forma de poder imprimir la información contenida, y les mostro como crear un informe. Hizo clic en Informes en la lista de objetos, seleccionando el asistente para la creación de informes.



Ilustración 3.39 Creación de contraseñas.

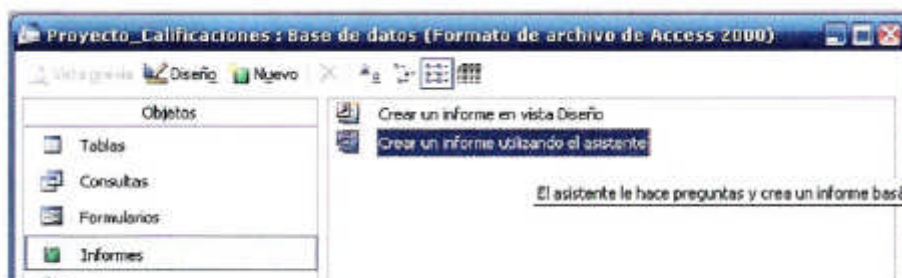
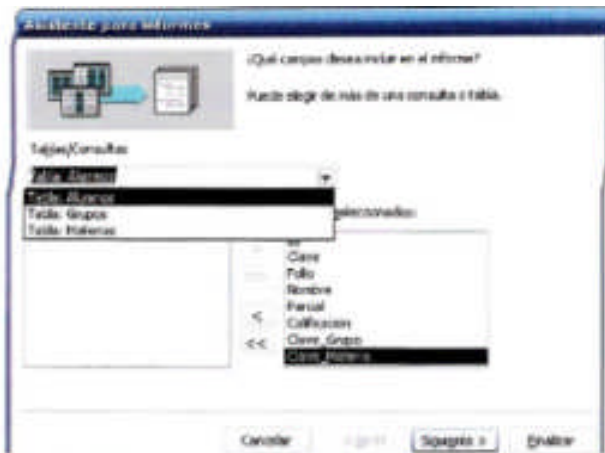


Ilustración 3.40 Informes de la barra de objetos.

El asistente muestra las tablas de la base de datos. Ahora bien seleccionarse aquellas que queremos incluir en nuestro informe; al hacerlo se muestran sus campos, y podemos elegir de entre ellos la información a mostrar.



El profesor eligió un nivel de agrupamiento, ordenó los datos por clave y dio una distribución tabular. Al final obtuvo el siguiente informe.

Ilustración 3-41 Creación de informes desde el asistente.

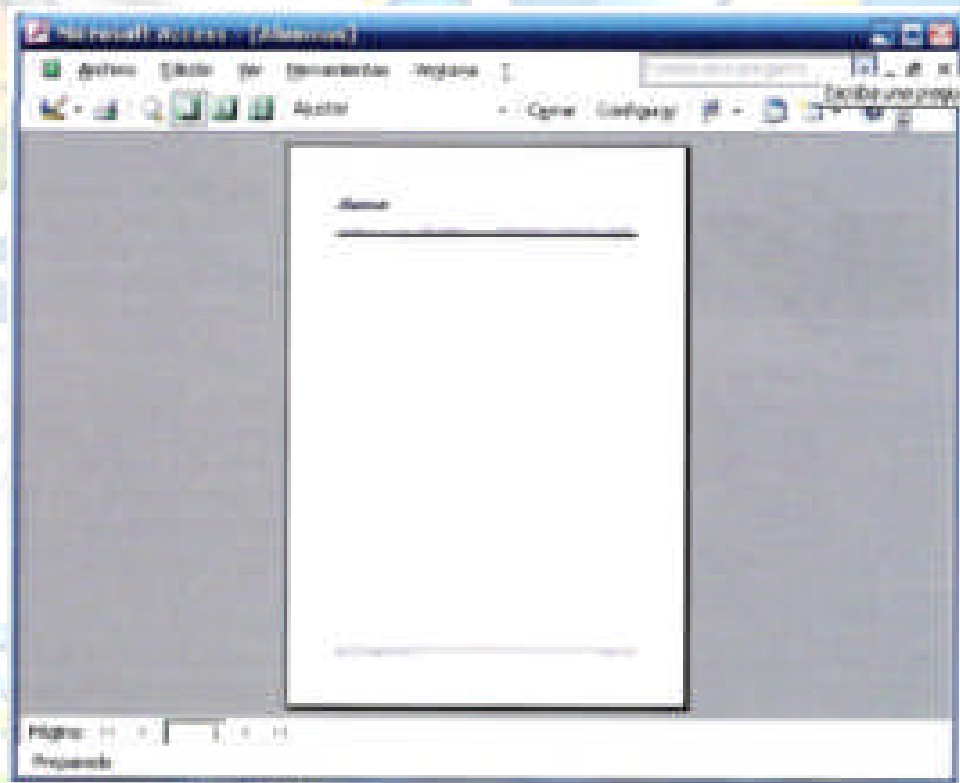


Ilustración 3-42 Previsualización del informe.

BLOQUE 4 SOFTWARE EDUCATIVO.

4.1 Definición de software educativo.

El software educativo es un conjunto de programas que tienen como propósito favorecer o facilitar el aprendizaje de una determinada asignatura, conocimiento o el desarrollo de una habilidad o competencia en los estudiantes.

Veamos otra definición:

«Se denomina **software educativo** al destinado a la enseñanza y el autoaprendizaje, y además permite el desarrollo de ciertas habilidades cognitivas».

El **software educativo** está concebido con la intención de facilitar la realización de actividades educativas que elaboradas manualmente podría consumir un tiempo excesivo. También para hacer más atractivas estas actividades y presentar de manera visual ciertos contenidos que de otro modo resultarían más difíciles de comprender.

De tal modo que en la actualidad existe software educativo prácticamente para cualquier asignatura y para temas de difícil aprendizaje. A diferencia de los bloques anteriores en que nos concentramos en aprender un programa en particular, en este bloque no es posible centrar nuestro estudio en un software educativo en particular. Tú tienes que aprender a buscar de acuerdo con tus necesidades un programa que te ayude a mejorar tu nivel de aprendizaje en aquellos temas que te son particularmente difíciles. Para apoyarte en tu búsqueda, estudiaremos las categorías de software educativo que te faciliten este proceso y revisaremos páginas de Internet con contenidos de software educativo.

4.2 Tipos de software educativo

a) Por su funcionalidad

Una manera práctica de clasificar el software actual desde el punto de vista de cómo trabaja el software y como accesa el usuario es por su tipo de *funcionalidad*: *software en línea (aplicaciones web)* y *software de escritorio*, es decir, hablamos de los programas que funcionan en línea (on-line) y los programas que funcionan *independientemente* en computadoras individuales.

Software en línea (aplicaciones web)

Los programas *on-line* son aquellos que funcionan sobre Internet en una plataforma Web y que, por lo tanto, requieren que tengas este servicio. Estos programas no se encuentran físicamente instalados en la computadora del usuario; se cargan y en algunos casos se ejecutan desde otra máquina. Depende para su ejecución que estén encendidos otros equipos, que exista conexión de Internet y que el ancho de banda de comunicación sea suficiente para que el rendimiento sea aceptable. Una tecnología que se utiliza para hacer aplicaciones educativas pequeñas son los *applets* de Java, que permiten hacer un software educativo ligero que se carga desde Internet y se utiliza en la máquina del usuario sin instalarse; por ejemplo, en la página <http://www.walter-fendt.de/ph11s/index.html>

Puedes encontrar diversos programas con ejercicios de física simples que te facilitaran el estudio de esta materia. Observa las ilustraciones 4.3 a y b.

Pero no son los únicos programas. Y como las páginas de Internet cambian constantemente, mejor aprenderemos a buscar nuestras propias paginas.

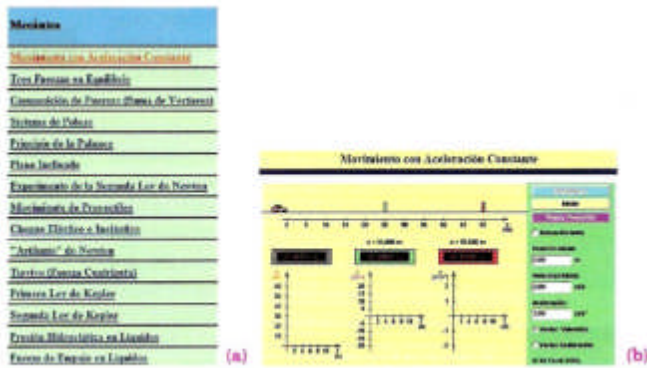


Ilustración 4.3 a y b. Ejemplos de temas de estudio de física.

De modo similar puedes buscar otras asignaturas, de física, matemáticas o biología. Prueba a hacer tus propias búsquedas y encuentra un tema que te interese reforzar.



Ilustración 4.5 Ejemplo software educativo.

La gran ventaja de los applets que son programas ligeros, fáciles de usar, en su mayoría gratuitos y se encuentran sin mucha dificultad. Se requiere tener instalado un complemento de JAVA para que se puedan ver en el navegador.

Otra página interesante para buscar software educativo en Línea es www.todoeducativo.com

Cada día crece más el número de páginas web que funcionan en línea, incluso procesadores de texto, hojas de cliente y otros tipos de software que ya funcionan en Internet, de tal modo que no necesitamos tenerlo instalado en la computadora. Dado que con la misma velocidad con que se crea páginas nuevas otras desaparecen, es importante que tú aprendas a buscar lo que necesitas y hacer tus propios descubrimientos.

Otro ejemplo de software en línea que aunque no es propiamente educativo es importante que conozcas es www.google.com/docs en el puedes hacer tus trabajos de procesador de palabras, presentaciones electrónicas y hoja de cálculos que se vieron en bloques anteriores sin tener instalado Word, Excel y PowerPoint. Es una alternativa interesante que vale la pena empezar a explorar; además puedes grabar en tu cuenta de Internet tus archivos y puedes tener acceso a ellos sin traer una memoria USB; todo será desde Internet.



Ilustración 4.6 Aplicaciones en línea.

Software de escritorio (stand-alone).

Hasta hace relativamente poco tiempo, unos 10 años, la mayor parte de las aplicaciones de una computadora eran de escritorio stand-alone. La gran mayoría de software debía primero instalarse en el disco duro y configurarse antes de poder ser utilizado. Hoy las cosas están cambiando lentamente, pero de manera constante hacia aplicaciones web. Sin embargo, las aplicaciones de escritorio todavía están muy difundidas y son necesarias para lugares donde no se cuenta con el servicio de Internet.

Los programas de escritorio, también conocidos como stand-alone (independientes), son aplicaciones que se ejecutan y cargan de manera independiente en la máquina en la que trabaja el usuario. Requieren de haber sido instaladas previamente o de ser aplicaciones *portables*. La ventaja de este tipo de aplicaciones es que se cargan y se ejecutan de manera rápida, pues su acceso es *local* y no dependen de que exista servicio de Internet o de una conexión de banda ancha para su adecuado funcionamiento.

Dentro de esta categoría se encuentran diferentes tipos de software educativo como las Enciclopedias de consulta en CD-ROM. Este tipo de aplicaciones tienen requisitos especiales de instalación para que puedan funcionar adecuadamente.

b) Por el tipo de licenciamiento

Se le llama *software propietario* al software por el cual se requiere pagar una licencia para poder utilizarlo por un tiempo determinado. Este tipo de software tiene un periodo de vencimiento y requiere que el usuario renueve su licencia para seguir teniendo derecho a utilizarlo. Por ejemplo, el proyecto Galileo en México: <http://www.galileo2.com.mx> tiene una amplia variedad de programas educativos por los cuales se requiere pagar una licencia, además de examinar las necesidades y valorar que en ciertas ocasiones si vale la pena pagar el costo un programa, pues se adquiere la certeza de tener un software seguro, libre de virus y por el cual se puede exigir un servicio adecuado.

El *software libre* es el tipo de software para el cual no se requiere necesariamente el pago de una licencia para su utilización; sin embargo, se aceptan contribuciones para mantener este tipo de proyectos. Un catálogo rápido de software libre de diferente tipo para Windows es <http://www.cdlibre.org/consultar/catalogo/index.html> (Ilustración 4.9 a y b). Otro catálogo importante de software educativo libre es el proyecto KDE de educación <http://edu.kde.org/> en el que puedes encontrar una buena cantidad de software educativo para física, química, matemáticas, astronomía y otras asignaturas, con la salvedad que estos programas están en inglés.



Ilustración 4.9 a y b Ejemplos de catálogo de software libre.

Como puedes observar en la ilustración, las categorías de software libre que puedes encontrar en esta página son *lenguajes*, *matemáticas*, *misceláneos (diversos)*, *ciencia* y *herramientas para la enseñanza*. Aunque las versiones originales son para ejecutarse en el sistema operativo LINUX, con el ambiente de ventanas conocido como KDE es posible descargar un programa especial que permite probar estos programas en Windows; este programa se llama el **KDE-Installer** y se puede descargar en la página <http://windows.kde.org/download.php>

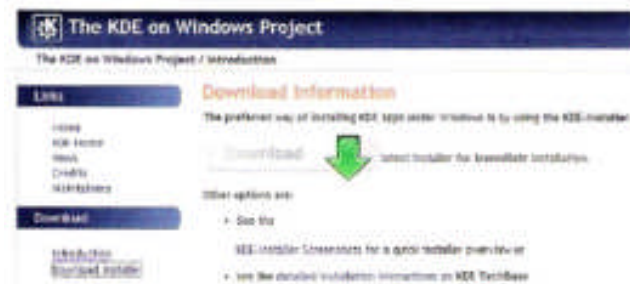


Ilustración 4.10 Descarga de Geogebra

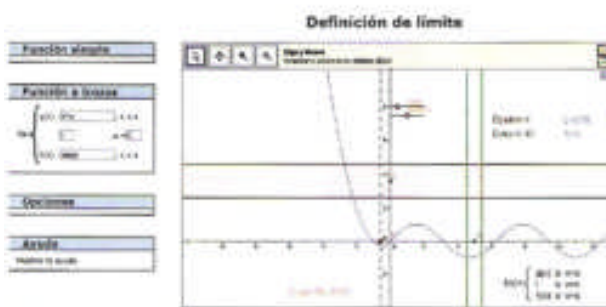


Ilustración 4.12 Ejemplo de Geogebra.

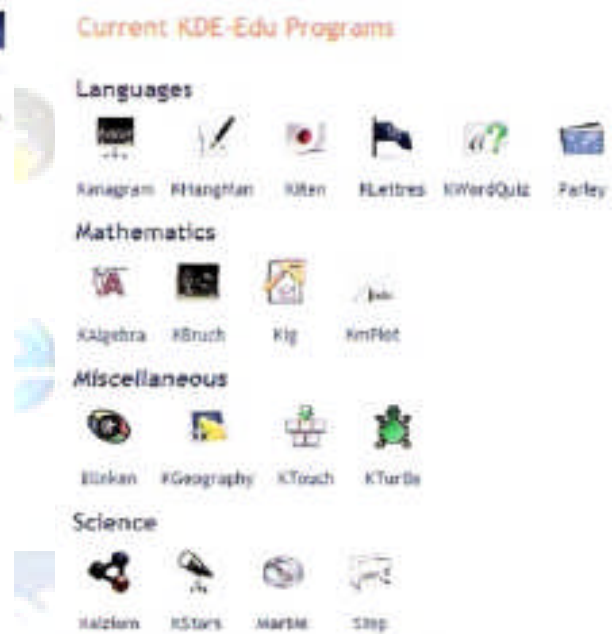


Ilustración 4.11 Programas educativos.

Geogebra es un software matemático interactivo libre para la educación en preparatorias y Universidades que ofrece la posibilidad de trabajar tanto en línea como poder descargarlo; está escrito en lenguaje java. Puedes encontrar recursos de cómo usarlo en la página <http://www.geogebra.org/en/wiki/index.php/Spanish>.

4.3 Universidad virtual

Definición de universidad virtual

La universidad virtual es una institución educativa que a diferencia de las universidades tradicionales no tiene una presencia física, es decir, no cuenta con un edificio, salones, sillas. existe como una creación de software a través de Internet y los estudiantes acuden a sus estudios desde cualquier computadora que tenga conexión al mismo. En México, una de las universidades virtuales más reconocidas es la del Instituto Tecnológico de Monterrey, campus virtual: <http://www.tecvirtual.itesm.mx/>

Es natural preguntarse si el aprendizaje en los programas a distancia que ofrecen las universidades virtuales tiene la misma efectividad que en los programas escolarizados. Estamos tan acostumbrados a la presencia física y social que da un salón de clases, la seguridad de tener unos compañeros con quienes trabajar, del profesor que nos guía y

un espacio donde estar, que todos estos factores que dan seguridad se diluyen en una universidad virtual. Debido a la distancia física que existe, la relación interpersonal entre el maestro y el alumno y entre sus compañeros es totalmente virtual; no se conocen en persona, sino a través de Internet. Esto puede ser una ventaja para cierto tipo de estudiantes.

Por otra parte, en la universidad virtual el principal responsable de su aprendizaje es el propio estudiante. Se requiere tener cualidades de autoestudio y capacidad de proponerse metas y alcanzarlas. Es todo un reto estudiar en una universidad virtual, pero puede ser muy gratificante.

En universidades tradicionales el maestro es el encargado de transmitir los contenidos, y el alumno sigue sus orientaciones, toma notas y consulta sus dudas. Esto mismo debe hacerse en los programas a distancia, pero sin el apoyo directo del maestro, el cual pasa a ser un tutor que guía al alumno en el proceso.

Consulta de universidades virtuales

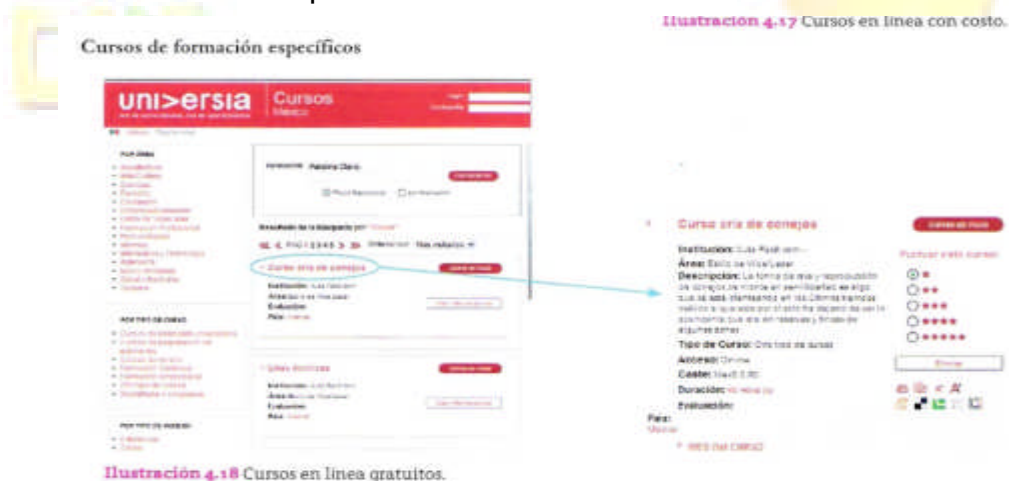
Aunque todavía estás en segundos semestre y aún falta tiempo para que tomes una decisión, es importante empieces a familiarizarte con las distintas opciones que pueden ser determinantes en tu futuro, así que realizaremos unas visitas virtuales a las universidades que sean de tu interés y comparemos las carreras que ofrecen, sus servicios y sus costos.

Alternativas de formación virtual

La formación virtual es una alternativa real para hacer realidad la competencia de «Aprender a lo largo de la vida». Nos ofrece la posibilidad de podernos organizar de acuerdo con nuestros tiempos y continuar en este proceso continuo de crecimiento personal, no solo para estudiar una carrera profesional, sino también cursos sobre temas específicos, tutoriales, simuladores, laboratorios virtuales o para aprender una lengua extranjera. Todo depende de lo que tú quieras.

Veamos algunos ejemplos:

Cursos de formación específicos



Este tipo de cursos especializados son de corta duración, de entre 40 a 80 horas, y tratan sobre un tema en particular. Por ejemplo, en la página <http://cursos.universia.net/> puedes encontrar cursos en línea, desde cómo poner uñas acrílicas hasta la crianza de conejos. La mayor parte de estos cursos son de paga, pero algunos no tienen costo.

Por supuesto, no es la única página que contiene cursos especializados. Existen muchas más; basta que entres a Google y escribas cursos en línea para que te aparezcan páginas de cursos diversos.

Tutoriales

Otra alternativa de formación que podemos encontrar en Internet son los programas tutoriales. Un programa tutorial es aquel cuya finalidad es guiar al estudiante durante la participación en un curso. Existen diferentes tipos de tutoriales. A continuación se presenta una clasificación muy completa de los tutoriales del autor Peré Marqués, de la Universidad de Barcelona, en su artículo Software Educativo:

Son programas que en mayor o menor medida dirigen, tutorizan el trabajo de los alumnos. Pretenden que a partir de unas informaciones y mediante la realización de ciertas actividades previstas de antemano los estudiantes pongan en juego determinadas capacidades y aprendan o refuercen conocimientos y habilidades. Cuando se limitan a proponer ejercicios de refuerzo sin proporcionar, explicaciones conceptuales previas se denominan programas tutoriales de ejercitación, como es el caso de los programas de preguntas (drill&practice, test) y de los programas de adiestramiento psicomotor, que desarrollan la coordinación neuromotriz en actividades relacionadas con el dibujo, la escritura y otras habilidades psicomotrices.

En cualquier caso, son programas basados en los **planteamientos conductistas** de la enseñanza, que comparan las respuestas de los alumnos con los patrones que tienen como correctos, guían los aprendizajes de los estudiantes y facilitan la realización de prácticas más o menos rutinarias y su evaluación; en algunos casos una evaluación negativa genera una nueva serie de ejercicios de repaso. A partir de la estructura de su algoritmo, se distinguen cuatro categorías:

Programas lineales. Presentan al alumno una secuencia de información y ejercicios (siempre la misma o determinada aleatoriamente) con independencia de la corrección o incorrección de sus respuestas. Herederos de la enseñanza programada, transforman el ordenador en una máquina de enseñar transmisora de conocimientos y adiestradora de habilidades. No obstante, su interactividad resulta pobre y el programa se hace tedioso.

Programas ramificados. Basados inicialmente también en modelos conductistas, siguen recorridos pedagógicos diferentes según el juicio que hace el ordenador sobre la corrección de las respuestas de los alumnos o según su decisión de profundizar más en ciertos temas. Ofrecen mayor interacción, más opciones, pero la organización de la materia suele estar menos compartimentada que en los programas lineales y exigen un esfuerzo más grande del alumno. Pertenecen a este grupo los programas que estructuran los contenidos en niveles de dificultad y previenen diversos caminos, y los programas ramificados con dientes de sierra, que establecen una diferenciación entre los conceptos y las preguntas de profundización, que son opcionales.

Entornos tutoriales. En general estén inspirados en **modelos pedagógicos cognitivistas** y proporcionan a los alumnos una serie de herramientas de búsqueda y de proceso de la información que pueden utilizar libremente para construir las respuestas a las preguntas del programa. Este es el caso de los **entornos de resolución de problemas**, “problems solving”, en los que los estudiantes conocen parcialmente las informaciones necesarias para su resolución y buscan la información que falta; aplican reglas, leyes y operaciones para encontrar la solución. En algunos casos, el programa no solo comprueba la corrección del resultado, sino que también tiene en cuenta la idoneidad del camino que se ha seguido en la resolución. Sin llegar a estos niveles de análisis de las respuestas, podemos citar como ejemplo de entorno de resolución de problemas el programa MICROLAB DE ELECTRONICA.

Sistemas tutoriales expertos. Como los Sistemas Tutores Inteligentes (Intelligent Tutoring Systems), que, elaborados con las técnicas de la Inteligencia Artificial y teniendo en cuenta las teorías cognitivas sobre el aprendizaje, tienden a reproducir un dialogo autentico entre el programa y el estudiante, y pretenden comportarse como lo haría un tutor humano: guían a los alumnos paso a paso en su proceso de aprendizaje, analizan su estilo de aprender y sus errores y proporcionan en cada caso la explicación o ejercicio más conveniente.

