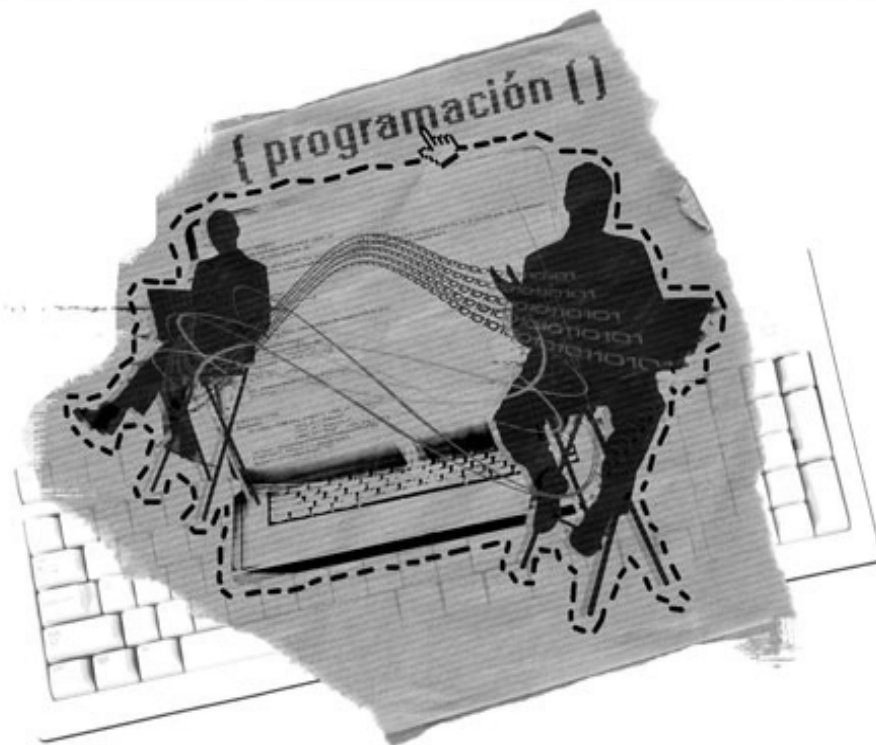


**4º AÑO
DE
BACHILLERATO**

Informática I



PRESENTACIÓN

Este libro es el resultado de la recopilación del trabajo que la Academia de Informática del Colegio Indoamericano ha venido realizando durante 25 años. Se basa en el programa de la asignatura Computación de la Escuela Nacional Preparatoria de la UNAM. y ha sido creado con la finalidad de proporcionar apoyo al estudiante y al profesor durante el curso escolar.

Esta obra cubre todos los temas y términos utilizados en dicho curso.

El libro está organizado en siete capítulos. Al final de cada uno de ellos se plantean una serie de preguntas y problemas, según corresponda el tema, para que el alumno pueda retroalimentar conceptos y terminologías.

El Capítulo 1, aborda todos los aspectos a desarrollar en la metodología del trabajo en equipo. Es el punto de partida para que el alumno se familiarice con los elementos que debe contemplar al integrarse a un equipo de trabajo así como las partes que deberá abordar en la elaboración del Proyecto Anual de Informática.

Conocer un lenguaje de programación no significa tener la capacidad de desarrollar una aplicación compleja usando dicho lenguaje. Para depurar el estilo de programación es necesario seguir una metodología, independientemente del lenguaje que utilice y de la plataforma donde lo vaya a desarrollar. El Capítulo 2 proporciona, a modo de iniciación, algunos de los pasos a seguir para aprender a programar. Estructuras secuenciales es el contenidos que se abordan en esta sección, todo esto asociado a los algoritmos y diagramas de flujo.

El crecimiento del uso de las tecnologías de Internet ha supuesto un significativo avance para los lenguajes que permitan generar páginas Web cada vez más dinámicas. Los Capítulos 3 y 4 proporcionan los fundamentos del lenguaje JavaScript para la creación de páginas Web dinámicas usando programación estructurada.

En la sección que corresponde al Capítulo 5 se describen los diferentes tipos de dispositivos mecánicos y electromecánicos de cálculo; de computadoras existentes: analógicas y digitales. Además se brindan antecedentes históricos. Primera, segunda, tercera, cuarta, y quinta generación de computadoras así como la importancia de algunos precursores como son Charles Babbage con sus dos inventos que marcaron sin duda alguna el camino del desarrollo de la computación moderna y el modelo de Von Neumann vigente en la actualidad.

El Capítulo 6, aborda los aspectos más destacados del hardware y software como los elementos más relevantes que conforman una computadora.

Las redes se hacen más y más populares. Cualquier computadora puede perfectamente trabajar como servidor de archivos o correr máscaras de IP para dar acceso a Internet a varias computadoras desde lugares diferentes al mismo tiempo. El Capítulo 7 tiene como finalidad proporcionar los conceptos básicos necesarios para entender qué es una red de computadoras. Explica el vocabulario que se requiere para comprender sobre redes.

Este libro fue escrito con mucho esfuerzo y dedicación para los alumnos del Colegio Indoamericano que cursan la materia de Informática I.

Academia de Informática.

CONTENIDO

Capítulo 1. Trabajo en Equipo

I. Introducción	5
II. Metodología para desarrollar Trabajo en Equipo	6
III. Guía para la realización de proyectos	13

Capítulo 2. Metodología de la Programación

I. Fases para solucionar un problema	19
II. Algoritmos	20
III. Estructuras Básicas y Técnicas para Representar Algoritmos.....	23
IV. Estructuras Básicas Expresadas Técnicamente.....	26
V. Identificación de Variables.....	30
VI. Técnicas para Representar Algoritmos.....	37

Capítulo 3. JavaScript

I. Páginas dinámicas	42
II. ¿Qué es JavaScript?	42
III. Código JavaScript en una página HTML	43
IV. Formas y formularios	44
V. Variables y constantes	57

Capítulo 4. Sentencias de Control

I. Operadores lógicos	69
II. Sentencia If Then	73
III. Sentencia If Then Else	77
IV. If Anidado.....	83
V. Sentencia Switch Case	86

Capítulo 5. Antecedentes de la Informática

I. Los Pioneros	93
II. Primera Generación	98
III. Segunda Generación	103
IV. Tercera Generación	104
V. Cuarta Generación	105
VI. Quinta Generación	105
VII. Cuestionario	107

Capítulo 6. Estructura Física y Lógica

I. Hardware	109
II. Software	117
III. Funciones de una computadora	118
IV. Cuestionario	119

Capítulo 7. Redes

I. ¿Qué es una red?	121
II. Tipo de Redes	122
III. Servidores de Red	124
IV. Topología de Red	125
V. Transmisión de Datos	127
VI. Protocolos de Red	129
VII. ¿Qué es Internet?	131
VIII. Sistemas Operativos de Red	134
IX. Cuestionario	135

Capítulo 1

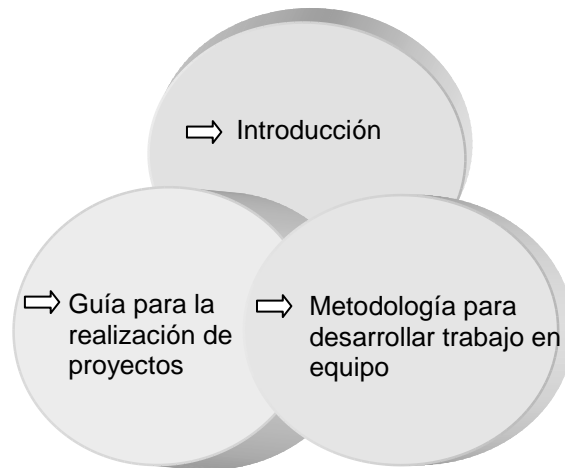
Trabajo en Equipo

Temas:

- I. Introducción.
- II. Metodología para Desarrollar Trabajo en Equipo.
- III. Guía para la Realización de Proyectos.



TRABAJO EN EQUIPO



I. INTRODUCCIÓN

Alguna vez te has preguntado ¿qué sacó al hombre de las cavernas? o ¿qué hizo posible su humanización?



¡TRABAJO EN EQUIPO!

El ser humano vive en sociedad; a través de los grupos. Hombres y mujeres se relacionan para sobrevivir, crecer y desarrollarse. Es por esta razón que los graves problemas que enfrenta la sociedad contemporánea difícilmente se solucionarán con acciones individuales aisladas; es decir, se necesita de una *comunidad de esfuerzos* para superar la difícil situación y lograr el desarrollo, necesitamos del trabajo en equipo.

Un **equipo** es un conjunto de personas interrelacionadas que se organizan para llevar a cabo una determinada tarea y que tienen un objetivo en común.

El trabajo en equipo **no** significa trabajar juntos y que uno de los integrantes decida lo que van a hacer los demás, esperando al final reunir las partes.

El trabajar en equipo es mucho más que eso, significa que todos los integrantes:

1. Aporten información y conocimientos acerca del tema.
2. Muestren sus puntos de vista u otras soluciones para resolver los problemas.
3. Escuchen a los demás de tal manera que tengan varias opciones para escoger la que mejor se adapte a sus necesidades.
4. Se ayuden entre sí, lo que reduce el tiempo empleado en cualquier tarea.
5. Se comprometan y responsabilicen con su aprendizaje y el de sus compañeros.
6. Den prioridad a los objetivos del equipo dejando aparte los objetivos individualistas y de competencia.

II. METODOLOGÍA PARA DESARROLLAR EL TRABAJO EN EQUIPO

Los pasos que se tienen que seguir para lograr el trabajo en equipo son:

1. Integración de equipos

El número ideal para conformar los equipos de trabajo es de 4 a 5 integrantes.

2. Definir funciones de los integrantes

Dependiendo del número de integrantes de cada equipo y de la complejidad del trabajo, existen diferentes papeles que deben ser desempeñados por los mismos; estos papeles se deben asignar desde el principio especificándose si las funciones van a ser rotativas o fijas, para todo el ciclo escolar.

Estas funciones sirven para que el equipo trabaje con orden y tenga mejores resultados en su proyecto.

Las funciones de los integrantes son:

Líder:

- Se encarga de coordinar las juntas de trabajo.
- Debe asegurarse que la documentación del equipo esté lista en cualquier instante.
- Programa y se ocupa de tener lo necesario para cada junta.
- Coordina y supervisa que se cumplan las funciones entre los miembros del equipo.
- Resuelve conflictos entre los miembros del equipo.
- Dirige la acción del equipo hacia la meta planteada.

Secretario:

- Redacta y resguarda la bitácora del equipo.
- Recolecta la información proporcionada por cada integrante.
- Redacta las minutas en las reuniones.
- Capta todas las aportaciones de los miembros del equipo, debido a que son espontáneas dentro de las reuniones y se dan de forma rápida y seguida.

Miembros del equipo:

- Se les designan diferentes tareas dependiendo de sus habilidades.
- Realizan solo una tarea en específico.
- Hacen investigaciones.
- Redactan textos.
- Elaboran gráficas, códigos y programas.

Nota: El líder y el secretario también son miembros del equipo, por lo tanto, también deben realizar estas funciones.

Asesor (Profesor):

- Orienta, dirige y supervisa el trabajo y desempeño del equipo.
- Motiva y apoya al desempeño colectivo.
- Proporciona asesorías por equipo.
- Proporciona asesorías individuales.
- Se encarga de la revisión de: bitácoras, cronogramas, producto final y de los conocimientos adquiridos por cada uno de los integrantes del equipo.
- Resuelve conflictos internos del equipo cuando éste no es capaz de resolverlos por si mismo.

I. Introducción

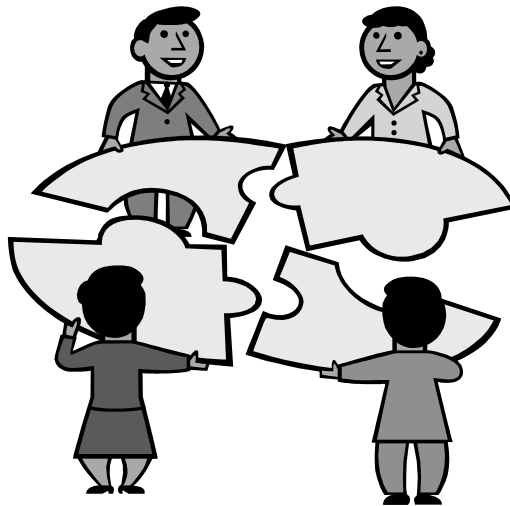
- El ser humano es un ser sociable por naturaleza
- EQUIPO es un conjunto de personas interrelacionadas que se ORGANIZAN para llevar a cabo una tarea determinada
- EQUIPO no significa dividir el trabajo y al final reunir las partes

3. Cédula de equipo

Es un instrumento de identificación en el que se encuentra la información general del equipo.

Partes de la cédula del equipo:

- Nombre del Equipo: Asignado por todos los miembros del equipo.
- Fecha de la Cédula: Fecha en la que se creó la misma.
- Miembros del Equipo: Nombres de los integrantes y la función que desempeña cada uno de ellos.
- Misión: Objetivo principal por el que se organizó el equipo.
- Antecedentes: Opiniones respecto al trabajo en equipo, de manera individual.
- Producto Final: Meta a la que se quiere llegar.
- Reglas Básicas: las establecen los integrantes para realizar el trabajo con gran calidad, además también se debe establecer si las funciones de los integrantes del equipo serán fijas o se rotarán durante el ciclo escolar.



La cédula de equipo debe ser elaborada por **todos** los integrantes. Todos deben estar de acuerdo en lo establecido.

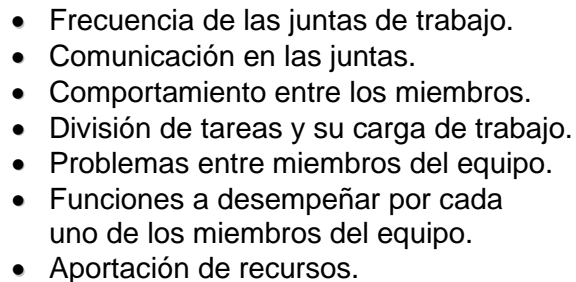
A continuación se muestra un ejemplo de cédula de equipo.

"Cédula de Equipo"	
Nombre del equipo.	<i>Informáticos.</i>
Fecha de la cédula.	<i>27 de Agosto de 2011.</i>
Miembros del equipo.	<i>Eva Maya Martínez – Secretario. Marco Antonio Cabrera Manuel – Líder. Arturo Hernández Ramírez de Arellano. Viridiana Ugalde Davalos. Tanya Macareno Prado.</i>
Misión.	<i>Formar alumnos con las capacidades y habilidades necesarias para el desarrollo de la cultura del trabajo colaborativo, aplicando los conocimientos adquiridos durante el curso en el desarrollo de un proyecto.</i>
Antecedentes.	<i>Marco: El trabajo en equipo es una de las mejores formas de trabajar. Eva: Me gusta, para conocer otras opiniones. Arturo: Es más fácil porque nos dividimos el trabajo. Tanya: No me gusta por la falta de integración. Viridiana: Me gusta porque podemos juntar varias ideas y así lograr un trabajo que nos guste a todos los miembros del equipo.</i>
Producto Final	<i>Desarrollar la metodología para la creación de un proyecto utilizando los procesos del trabajo colaborativo.</i>
Reglas básicas	<ol style="list-style-type: none"> <i>1) Planear el trabajo.</i> <i>2) Ser puntual.</i> <i>3) Respetar ideas.</i> <i>4) Ser proactivo.</i> <i>5) Aportar ideas.</i> <i>6) Respetar roles.</i> <i>7) Todos deben cumplir las reglas.</i> <i>8) Funciones del Líder.</i> <i>9) Funciones del Secretario.</i> <i>10) Funciones de los miembros.</i>

4. Reglas Básicas

Son las reglas que ponen todos los integrantes del equipo para poder trabajar con orden y disciplina durante el tiempo que trabajen juntos, además se escribirán las sanciones que se impondrán a los miembros del equipo que no cumplan con las reglas.

Se deben contemplar los siguientes puntos:

- 
- Frecuencia de las juntas de trabajo.
 - Comunicación en las juntas.
 - Comportamiento entre los miembros.
 - División de tareas y su carga de trabajo.
 - Problemas entre miembros del equipo.
 - Funciones a desempeñar por cada uno de los miembros del equipo.
 - Aportación de recursos.

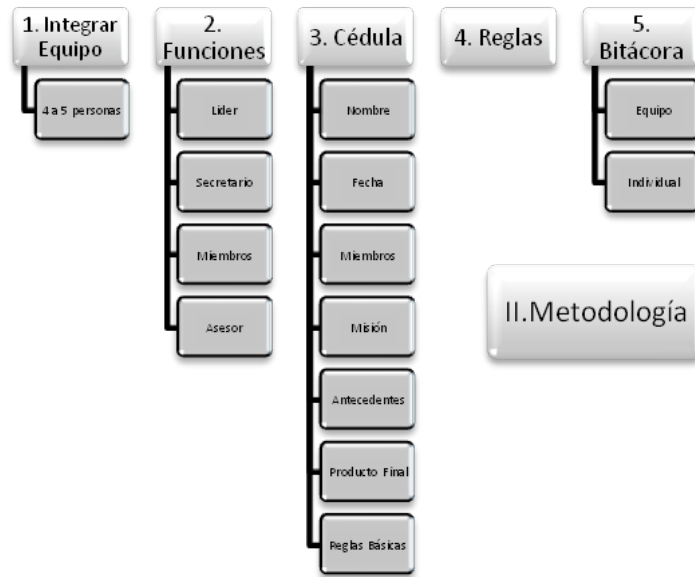
5. Cronograma de actividades

Una herramienta poderosa en la elaboración de un proyecto son los cronogramas de actividades. Te ayudarán a realizar un plan de trabajo, con un cronograma podrás determinar la duración de cada actividad a desarrollar para el proyecto, analizando, organizando y calendarizando cada actividad del proyecto.

Para realizar el cronograma lo primero que debes hacer es escribir todas las actividades relacionadas con la realización del proyecto. Posteriormente debes definir cuánto tiempo se requiere para completar cada tarea, en días, semanas o meses. También puedes establecer quién estará a cargo de cada tarea.

Actividad. Una vez que el profesor formó los equipos de trabajo, elabora en tu cuaderno de trabajo la cédula del equipo al que perteneces.

"Cédula de Equipo"	
Nombre del equipo:	
Fecha de la cédula	
Miembros del equipo	
Misión	
Antecedentes	
Producto Final	
Reglas básicas (Establece mínimo 15 reglas)	



III. GUÍA PARA LA REALIZACIÓN DEL PROYECTO

Para la realización del proyecto final se tienen que llevar a cabo los siguientes pasos:

1. Título del proyecto

Es la frase en la que se engloba el tema que el equipo desarrollará dentro del proyecto; obviamente el tema debió haber sido elegido de manera democrática, tomando en cuenta las opiniones y preferencias de todos los integrantes, y también de modo objetivo, recordando que el tema debe ser *académico*.

Si no tienen definido el tema de su proyecto, pueden pedir ayuda a su profesor para que les sugiera temas, pero la decisión final es del equipo.

2. Desglose de subtemas

En esta sección el equipo debe seleccionar los subtemas que se contemplarán en el desarrollo del proyecto, además es conveniente (para ustedes y su profesor) escribir una breve descripción (tres líneas) de lo que tratarán en cada subtema, así tendrán una idea global de los alcances del proyecto y de los aspectos que van a estudiar.

3. Acopio del material

Es la búsqueda o investigación de elementos, ideas, tesis, etc. sobre el tema que eligieron trabajar. Los principales centros de información que se tienen que consultar son:

- Biblioteca. Organización donde se clasifican publicaciones de todo tipo.
- Hemeroteca. Espacio destinado a guardar periódicos, revistas, catálogos y anuncios, se pueden encontrar publicaciones semanales, catorcenales y mensuales.
- Centros de Información Virtual. Sitios Web a los que se puede recurrir para consultar información.

Recuerden que para desarrollar un trabajo bien fundamentado, deben basarse en fuentes bibliográficas primarias y secundarias.

- Fuentes bibliográficas primarias. Son obras en las que el autor trate sobre su tema de investigación de manera directa no incidental.
- Fuentes bibliográficas secundarias. Son obras de otros autores acerca del personaje objeto de trabajo, pero estas no deben ser la mayoría; además, deben tener presente que *Internet* no siempre es una fuente confiable.

Una vez que tengan toda su información impresa, se la mostrarán a su profesor para que revise que el material que consiguieron es el adecuado para el proyecto.

4. Organización de la información

Aquí, deben separar y dividirse la información entre todos los integrantes del equipo, para que cada uno lea y analice qué es lo más importante sobre el tema o subtema.

5. Síntesis de la información recopilada

Para hacer la síntesis de su información, es conveniente que utilicen fichas de trabajo, generalmente son de cartulina, tamaño media carta y con márgenes de 2.5 cm.

Las fichas deberán contener la siguiente información:

- En cada ficha se desarrolla una idea.
- En la parte superior, se escribe el tema y subtema al que pertenece la información.
- En la parte posterior, se escriben los comentarios y al final la bibliografía.

Ejemplo de Ficha de Trabajo:

<p>TEMA 1: Trabajo en equipo. Subtema 1.1: Importancia y Trascendencia.</p> <hr style="border: 0.5px solid black;"/> <p>El trabajo en equipo permite que los alumnos:</p> <ul style="list-style-type: none"> • Logren un aprendizaje significativo. • Obtengan un aprendizaje con perdurabilidad. • Obtengan una perspectiva integral de lo aprendido. • Contextualizar su aprendizaje. • Tengan una mayor calidad en su aprendizaje. • Adquieran: conocimientos, habilidades, actitudes y valores. • Aprendan más y mejor. • Aprendan de lo social, lo afectivo y lo volitivo.

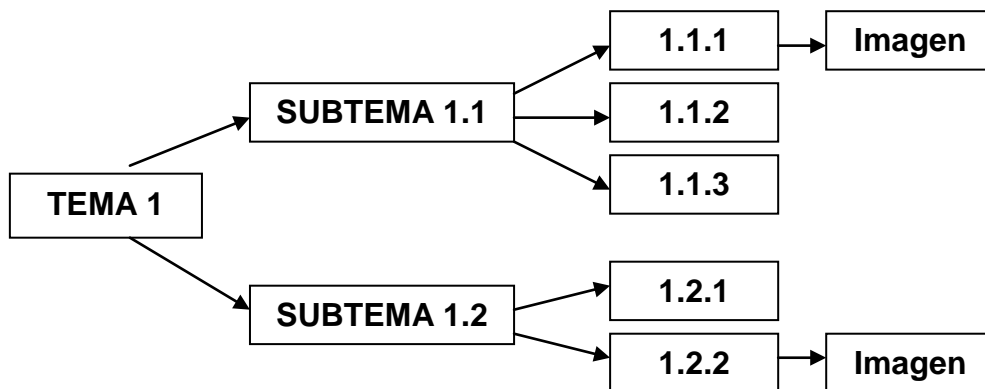
Nota: Sería una buena idea que cada ficha de trabajo representara una pantalla de la computadora, esto dará un idea clara de la extensión de su proyecto, por lo que también deben agregarse las imágenes que contendrá en trabajo.

Comentarios:

- En el trabajo en equipo todos los integrantes se involucran en el proyecto, se comprometen con su aprendizaje y el de los demás, por lo tanto, todos aprenden más y en menos tiempo.
- Al trabajar en equipo aprendes valores como el respeto, ya que tienes que respetar a los otros integrantes y sus ideas aunque no siempre estés de acuerdo con ellos.
- “volitivo” se refiere a la voluntad que los integrantes deben mostrar para aprender y ayudar a los demás.

Bibliografía:

Ferreiro,R., Calderón, M. (2000), El ABC del aprendizaje cooperativo: trabajo en equipo para enseñar y aprender. México: Trillas.



Conclusiones

Deben destacar los aspectos esenciales del trabajo; es muy importante que sea una descripción lógica del cuerpo del trabajo. Todos los integrantes deben realizar las conclusiones.

6. Introducción

En esta sección deben destacar en qué consiste el proyecto y la relevancia de éste; hay que explicar cuál es el tema del trabajo, bajo qué enfoque se va a desarrollar, y porqué es importante.

La introducción debe redactarse después de haber terminado el trabajo porque de esta manera se tiene una visión conjunta del trabajo, y se puede escribir sobre él para enunciarlo en forma sintetizada.

7. Índice

Es el contenido de su trabajo y la localización por temas de cada una de las partes.

8. Fuentes Bibliográficas

Se mencionan las obras que sirvieron para elaborar el proyecto.

Y de acuerdo al tipo de fuente consultada se debe organizar la información de dicha obra, así como sigue:

a) Bibliográficas.

» Libros.

Apellidos del Autor, Letra inicial del nombre. (Año de publicación),
Título del libro. (Volumen o número de edición).
 País: Editorial.

Ejemplo:

Ferreiro, R. (2000),
El ABC del aprendizaje cooperativo: trabajo en equipo para enseñar.
 México: Trillas.

b) Hemerográficas

» Artículos de revista.

Apellidos del Autor, Letra inicial del nombre. (Año de publicación).
 Título del artículo.
Título de la Revista, número de volumen, (Número de revista), páginas.

Ejemplo:

Jonson, C.L. (1980).
 Adolescent pregnancy: Intervention into the poverty cycle.
Adolescente, 9, (3), 391-406.

» Artículos de periodísticos.

Apellidos del Autor, Letra inicial del nombre. (Año de publicación, mes, día).
 Título del artículo.
Título del periódico, páginas.

Ejemplo:

Odrozola Urbina, A. (1994, Diciembre, 13).
 Se inicia programa de psicología clínica en el Hospital de Ensenada.
El Universal, pp.1, 7.

c) Internet

- » Sitios Web.
Nombre del sitio (día, mes y año de consulta).
Título de la página. [En red].
Especificar dirección electrónica.

Ejemplo:

Colegio Indoamericano (12 de agosto de 2007).
Colegio Indoamericano. [En red].
<http://www.indoamericano.edu>.

Ya con toda la información reunida y estructurada, se debe entregar en el siguiente orden:

- 1) Portada.
- 2) Índice.
- 3) Introducción.
- 4) Cuerpo del trabajo (temas y subtemas desarrollados y con imágenes).
- 5) Conclusiones.
- 6) Bibliografía.

9. Cronograma

Se trata de llevar una programación de actividades en las que se establecen tiempos, de acuerdo a las metas que se quieren lograr y la fechas de entrega estipuladas por el profesor.

10. Interfaz Gráfica

Se trata de la representación gráfica de cada una de las páginas a desarrollar en el proyecto.

11. Diseño sitio web del tema seleccionado

Conjunto de páginas HTML, dedicadas a un tema en particular.

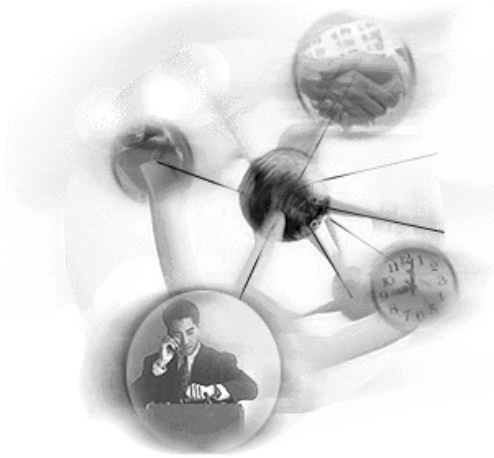


Capítulo 2

Metodología de la Programación

Temas:

- I. Fases para Solucionar un Problema.
- II. Algoritmos.
- III. Estructuras Básicas para Representar un Algoritmo.
- IV. Estructuras Básicas Expresadas Técnicamente.
- V. Identificación de Variable.
- VI. Técnicas para Representar Algoritmos.
- VII. Variables con Funciones Específicas.



METODOLOGÍA DE LA PROGRAMACIÓN

Un método se define como el camino por el que se llega a cierto resultado.

Una metodología es un procedimiento general para lograr de una manera precisa un objetivo.

La metodología de la programación se refiere a los procedimientos y técnicas que nos permitirán desarrollar programas para computadoras, entendiéndose como programa al conjunto de instrucciones explícitas para obtener un fin determinado o resolver problemas.

I. FASES PARA SOLUCIONAR UN PROBLEMA

1. Análisis y definición del problema

Es la descripción precisa del problema y delimita los alcances que tendrá la solución. Se debe responder a las preguntas:

¿Qué datos tenemos para resolver el problema?

¿Cuál es el resultado del problema?

La respuesta a la primera pregunta indicará los resultados deseados o las salidas del problema. La respuesta a la segunda indicará qué datos se proporcionan o las entradas del problema.

Ejemplo. *Calcular el área de un rectángulo con base y altura conocidas.*

2. Definición de los datos de entrada / salida

Aquí se definen los elementos que serán transformados durante el proceso de solución (entradas), así como el resultado final que se desea (salidas). Es de gran importancia, porque si no se define con exactitud qué se desea como producto final, pueden obtenerse resultados erróneos.

Ejemplo:

Entradas: *los valores de altura y base del rectángulo.*

Salida: *valor del área del rectángulo.*

3. Proceso de solución

En esta fase se definen los pasos para encontrar la solución al problema planteado y se realiza la descripción detallada de estos. Deben ser muy precisos. Esta fase se definirá como algoritmo.

Ejemplo:

1. *Inicio.*
2. *Leer altura rectángulo.*
3. *Leer base rectángulo.*
4. *Área es igual a multiplicar la base por la altura.*
5. *Imprimir Área.*
6. *Fin.*

4. Comprobación

Aquí se realizan las pruebas del o de los procesos y se verifica la efectividad de los pasos.

Ejemplo. *Tome los valores de altura = 1, base = 1 y debe obtener área = 1.*

II. ALGORITMOS

Comúnmente, las personas efectuamos una serie de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema. Por ejemplo, al tomar el desayuno, al transportarse de la casa a la escuela, al bañarse, entre otros. Al realizar esto, estamos utilizando un algoritmo.

Formalmente se define un algoritmo como:



El conjunto ordenado de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema.

Características de un algoritmo:

- Preciso. Indicar el orden en el que se llevará a cabo cada uno de los pasos.
- Definido. Al seguir el algoritmo varias veces se debe llegar al mismo resultado.
- Finito. Siempre debe terminar.

Por otro lado, existen problemas que necesitan de un análisis profundo y de un pensamiento flexible y estructurado para encontrar su solución. Indudablemente surgen preguntas como:

- ¿Puede resolverse el problema?
- ¿Puede analizarse el problema?

Las respuestas a estas interrogantes son difíciles de contestar. No existen reglas específicas que nos permitan resolver problemas. Sin embargo, podemos desarrollar un conjunto de técnicas y herramientas metodológicas que nos permitan ser flexibles

y estructurados al razonar en la solución de un problema. Esto nos conduce finalmente a la construcción de algoritmos eficientes.

Uno de los objetivos de este curso es aprender a resolver problemas mediante una computadora. Una persona que programa computadoras resuelve problemas de una manera más rigurosa y sistemática.

A lo largo de esta sección nos referiremos a la metodología necesaria para resolver problemas mediante programas. El eje central de esta metodología es el concepto de algoritmo. Los algoritmos son independientes del lenguaje de programación en que se expresan, así como de la computadora que los ejecuta.

El diseño de la mayoría de los algoritmos requiere creatividad y conocimientos. En esencia, la solución de un problema puede expresarse mediante un algoritmo.

1. Conceptos fundamentales

Para la construcción de algoritmos es necesario analizar los siguientes conceptos:

- Tipos de datos.

Los tipos de datos se subdividen en los siguientes:

a. Datos numéricos.

Dentro de los tipos de datos numéricos se encuentran los *enteros* y los *reales*. Siguen las reglas de las Matemáticas, es decir, tienen magnitud y signo (+ ó -).

Los *enteros* no tienen parte decimal, por ejemplo:

128 1528 -714

Los *reales* tienen una parte decimal, por ejemplo:

7.5 128.05 -3.1

b. Datos alfanuméricos.

Son datos cuyos contenidos son solo texto (sin significado numérico). Pueden ser letras (a, b, c, ..., z), dígitos (0, ..., 9) o símbolos especiales (#, \$, %, etc.). Generalmente se escriben encerrados entre comillas. Con este tipo de datos no pueden realizarse operaciones aritméticas.

c. Datos lógicos.

También se les conoce como *booleanos* y solo pueden tomar los valores verdadero (true) o falso (false).

- **Identificador.**

Un identificador se forma de acuerdo a las siguientes reglas:

- a. El primer caracter debe ser una letra.
- b. Los demás caracteres pueden ser letras o dígitos.

- **Constante.**

Es una cantidad que no cambia su valor durante el proceso de solución. Su nombre debe seguir las mismas reglas que un identificador. Su tipo de dato puede ser numérico, alfanumérico o lógico.

- **Variables.**

Son cantidades que pueden modificar o no su valor durante un proceso. Para nombrar las variables se utilizan identificadores y al igual que las constantes tienen un tipo de dato.

Ejemplo 1. Realizar una llamada telefónica a una amiga.

Solución:

- 1. Análisis y definición del problema.**

Llamar por teléfono a una amiga.



- 2. Definición de los datos de entrada / salida.**

Entrada: Número telefónico de amiga, teléfono.

Salida: Llamada.

- 3. Proceso de solución (Algoritmo).**

1. Inicio.
2. Leer número telefónico.
3. Tomar el auricular.
4. Marcar número telefónico.
5. Esperar tono.
6. Esperar respuesta, pedir comunicación con mi amiga.
7. Establecer comunicación con amiga.
8. Fin.

- 4. Comprobación.**

Haber realizado la conversación vía telefónica con nuestra amiga.

III. ESTRUCTURAS BÁSICAS Y TÉCNICAS PARA REPRESENTAR ALGORITMOS

1. El Concepto de Estructura

Una estructura se define como un esquema que nos permite representar de manera simplificada alguna idea que bajo condiciones normales es constante. Esto significa que de alguna manera el pensamiento del ser humano, en lo que se refiere a los algoritmos, está enmarcado en algún tipo de *Estructuras* que no solo le permiten tener un medio más simplificado y a la mano para expresar las ideas sino que además permite “restringir” un poco el horizonte de la *Lógica Algorítmica*.

Es pertinente, pues, hacer un breve paréntesis para explicar el porqué es importante “restringir” un poco el horizonte de la *Lógica Algorítmica*.

Comencemos con dos breves preguntas:

- a. ¿Con cuántos algoritmos las señoras de la casa pueden preparar arroz guisado?
- b. ¿Cuántas personas vemos que lleven puesta una camisa y un pantalón exactamente igual al tuyo?

La primera pregunta la resolvemos preguntando a tres o cuatro señoras acerca de su forma de preparar arroz guisado. Ten la seguridad de que todas van a tener una manera diferente (o sea un algoritmo diferente) para prepararlos, pero lo que va a ser coincidente en todas, es lograr el mismo objetivo: dejar listo el arroz guisado. Ello nos demuestra, en primera instancia, que cada persona concibe algorítmicamente el mismo proceso de manera diferente pero que pueden llegar al mismo objetivo sin importar el camino que se haya escogido para lograrlo.

La segunda pregunta es más reveladora ¿Sabes porqué ninguna o casi ninguna persona lleva puesta una camisa y un pantalón exactamente igual al tuyo? Pues sencillamente porque todas las personas están cumpliendo, en condiciones normales, con el objetivo de estar vestidos más no exactamente de la misma forma.

Esa variabilidad en cuanto a la concepción de un determinado algoritmo es lo que llevó a pensar en que la parte técnica también podría llegar a ser igualmente variable. ¿Qué pasa cuando una persona concibe un algoritmo computacional en condiciones lógicas que prácticamente solo ella la entiende? Si esta persona se tiene que retirar de la empresa, la empresa se verá en un gran problema.

Los diferentes puntos de vista en cuanto a la concepción del mundo es lo que le ha permitido a éste avanzar, de ahí se han podido crear tecnologías, modas, teorías y muchos avances del mundo moderno pero a nivel técnico si resulta ser muy importante que la lógica para desarrollar un algoritmo computacional sea tan clara y

tan “estándar” de tal forma que un programa desarrollado por una persona sea fácilmente entendible por cualquier otra, dado que haciendo uso de la lógica propia de cada individuo podemos llegar a encontrarnos con programas tan confusos que solo llegarían a ser entendibles por su creador.

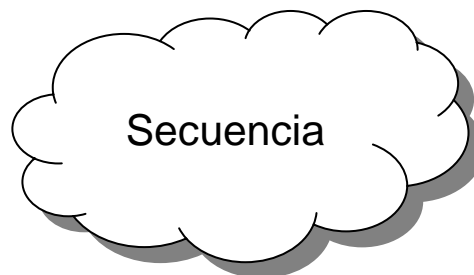
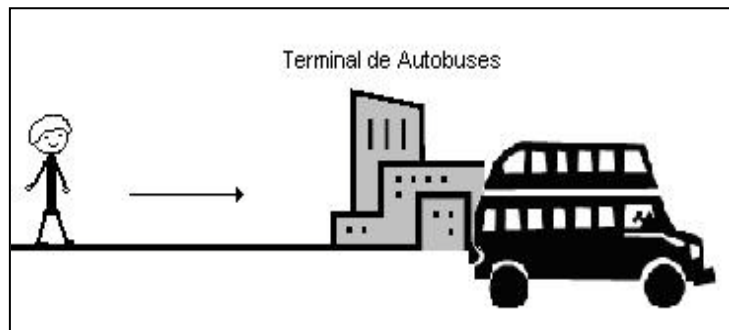
Esta es la razón fundamental por la cual se busca “uniformar” la lógica para desarrollar algoritmos computacionales y poder llegar a estructuras básicas sobre las cuales se pueda decir que está fundamentada dicha Lógica.

2. Consideraciones Algorítmicas Sobre el Pensamiento Humano

Después de analizar desde muchos ángulos el pensamiento humano y teniendo en cuenta los conceptos de algoritmo informal y algoritmo computacional se llegó a la conclusión de que dicho pensamiento se mueve entre tres estructuras básicas.

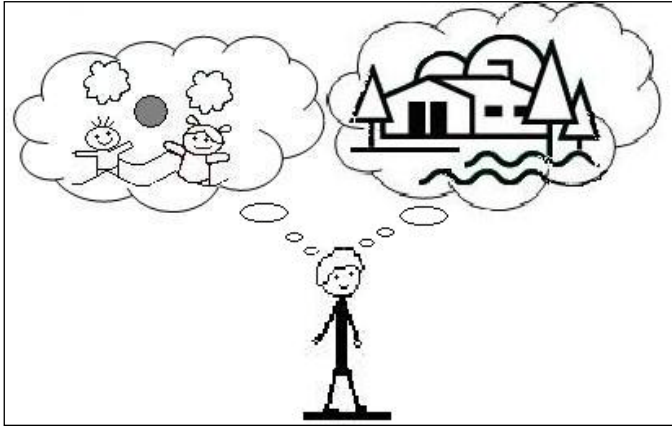
Cuando estás planeando ir este fin de semana a pasear con tus amigos. Lo que en tu mente se va dibujando poco a poco es una secuencia de acciones a realizar que te permitan pasar un fin de semana agradable. Cuando tienes que pensar que debes ir

a la Terminal de autobuses a tomar el transporte lo que vas organizando en tu mente es una secuencia de acciones que permitan acercarte a dicha Terminal, verificar la línea del autobús correcto y tomarlo para irte. Pues bien esa es la primera estructura sobre la cual se mueve el pensamiento humano y es la estructura de:



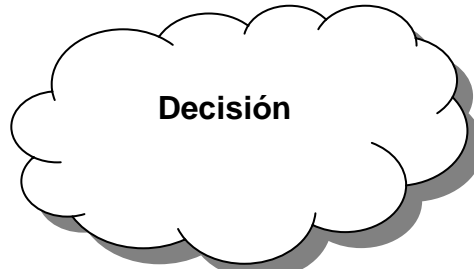
Permanentemente estamos inmersos en esta estructura y generalmente primero se planea cada secuencia de acciones (consciente o inconscientemente) antes de ejecutarlas. Cada una de las cosas que hacemos diariamente no son más que secuencias de acciones que hemos planeado para poder cumplir con nuestros

objetivos en la sociedad. Esta estructura se caracteriza porque una acción se ejecuta detrás de otra.



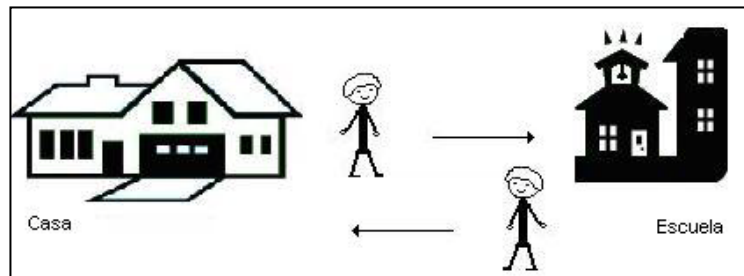
Estás planeando el fin de semana pero no sabes si pasar el domingo en el balneario que queda a una hora de la ciudad en donde vives o aprovechar e ir hasta la finca de aquel tío que hace mucho tiempo no visitas y que queda también a una hora. En cada alternativa encuentras ventajas y desventajas, visitar al tío es bueno porque hace mucho tiempo no lo haces y te gustaría ver a tu familia, pero en la

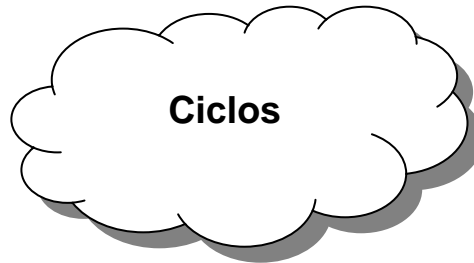
finca del tío no tiene piscina y el balneario sí. Esta estructura es la segunda sobre la cual se basa el pensamiento (o razonamiento) humano. Estructura de:



Gracias a la cual podemos escoger lo que según nosotros sea la mejor alternativa. La decisión se da siempre que tengas que escoger de entre, por lo menos, dos caminos lógicos.

Acostumbras ver tu programa favorito todos los días a las 9:30 de la noche, acostumbras ir a la escuela a la misma hora y esperas el autobús en el mismo paradero, acostumbras saludar de la misma forma a tu mejor amiga y acostumbras dormir en el mismo lado de la cama y en la misma posición. Tú sencillamente vives practicando la tercera estructura:





Es la estructura que nos permite repetir una o varias acciones una cantidad definida de veces. Todos los días almuerzas en tu casa según lo cual estarás en el ciclo de ir a almorzar a tu casa siempre. En pleno almuerzo el hecho de que lleves muchas veces la cuchara del plato a tu boca representa que estarás haciendo lo mismo mientras en el plato exista todavía algo más para comer. Puedes notar que permanentemente (e infortunadamente) estás también realizando tareas cíclicas. Cuando nos aferramos mucho a estos ciclos de vida es cuando la vida se nos vuelve tremendamente monótona. Por ahora lo que importa es que tengas claro que todo lo que haces, sin importar que sea, cualquier acción o conjunto de acciones que hagas siempre estarán enmarcadas en alguna de estas tres estructuras:

1. **Secuencias de acciones.**
2. **Decisión de acción.**
3. **Ciclos de acciones.**

Tomar una decisión depende de una determinada condición y repetir un conjunto de acciones depende también de que se cumpla o se deje de cumplir igualmente una condición.

IV. ESTRUCTURAS BÁSICAS EXPRESADAS TÉCNICAMENTE

Precisamente y con el fin de facilitar estándares técnicos que permitan describir las ideas lógicas de una manera uniforme se han desarrollado esquemas que te permitirán escribir las estructuras mencionadas anteriormente.

1. Las Secuencias de Órdenes

Para escribir una secuencia de órdenes o acciones todo lo que tenemos que hacer es colocar una nueva orden o una nueva acción después de la última que hayamos colocado. De esta manera se entiende la secuencialidad y la ordinalidad en la ejecución de esas acciones.

Ejemplos Secuencias de Órdenes.

Ejemplo1: Vamos a desarrollar un algoritmo que nos permita asomarnos a la ventana, pero asumiremos que la ventana donde queremos asomarnos ya está abierta y que no estamos muy distantes de la ventana.

Solución:

Algoritmo para asomarnos a la ventana.

Inicio.

- Ubicar la ventana por la que deseamos asomarnos.
- Levantarnos del lugar en donde estemos sentados.
- Avanzar hacia la ventana.
- Llegar hasta tener la ventana muy próxima.
- Asomarnos por la ventana.

Fin.

En el ejemplo puedes ver que cada acción está antes de una y después de otra. También puedes notar que para que este algoritmo nos permita asomarnos a la ventana todo lo que se hace es realizar cada acción en el orden en que están planteadas y realizando una a la vez. Eso permitirá lograr el objetivo propuesto.

Ejemplo 2: Si queremos realizar el algoritmo para colocarnos una camisa (asumimos que la camisa está en nuestro ropero doblada y abrochada) entonces:

Solución:

Algoritmo para colocarnos una camisa.

Inicio.

- Dirigirnos a nuestro ropero.
- Abrir el ropero.
- Tomar una camisa.
- Desabrocharla.
- Abrir la camisa.
- Meter un brazo por una de sus mangas.
- Meter el otro brazo por la otra de sus mangas.
- Ajustar la camisa al tronco.
- Abotonarla (botón a botón).

Fin.

Al igual que en el ejemplo anterior todo lo que tenemos que hacer es ejecutar cada acción en el orden indicado y hacerlo paso a paso, entonces podremos lograr el objetivo.

Podemos notar que para utilizar la estructura de secuencia (que a veces parece ser tan obvia) todo lo que tenemos que hacer es ir colocando una acción tras otra y, por supuesto, ser muy racional en el orden de dichas acciones porque de seguro, hasta el momento, hemos podido notar que en cuestión de algoritmos “El orden de los factores sí altera el resultado”.

2. Las Decisiones

Siempre que tengamos que utilizar la estructura de decisiones vamos a depender de una **condición**. La condición es la que nos permite poder decidir cuál es el camino lógico correcto a tomar.

Ejemplo Decisiones.

Ejemplo 1: Desarrollemos el mismo algoritmo de asomarnos a la ventana pero esta vez le vamos a colocar las condiciones: estamos sentados y se encuentra abierta la ventana. Para ello vamos a incorporar unas líneas de decisión que permitan tener un algoritmo más genérico y lograr mejor el objetivo, entonces:

Solución:

Algoritmo para asomarnos a la ventana.

Inicio.

Ubicar la ventana por la que nos queremos asomar.

Si estamos sentados.

Levantarnos del lugar en donde estemos sentados.

Orientarnos hacia la ventana.

Sino.

Orientarnos hacia la ventana.

Avanzar hacia la ventana.

Llegar hasta tener la ventana lo más cerca posible.

Si está cerrada.

Abrirla.

Asomarnos por la ventana.

Fin.

Observa, el algoritmo ha cambiado, ahora tiene unas condiciones que le permiten ser una secuencia de acciones más racional. En estas condiciones el algoritmo se convierte en algo depurado y, por tanto más próximo a la realidad.

Algunos puntos importantes en este algoritmo son:

- a. Las palabras **Si** que aparecen son exclusivamente condicionales y no afirmativas como pudiera pensarse en algunos casos.

- b.** Después de cada **Si** (condicional) va una condición que es la que permite que se haga una selección de un proceso u otro. La condición regula las actividades o acciones que se realizarán posteriores a él.

En la decisión:

Si estamos sentados.

Levantarnos del lugar en donde estemos sentados.

Orientarnos hacia la ventana.

Sino.

Orientarnos hacia la ventana.

Estar sentados es la condición de la cual depende si realizáremos dos acciones: Levantarnos del lugar en donde estemos sentados y orientarnos hacia la ventana, o si sólo haremos la acción orientarnos hacia la ventana.

- c.** Podrás notar que una decisión completa involucra una pregunta que evalúa una condición.

Un conjunto de acciones a realizar en caso de que la condición sea verdadera.

Un conjunto de acciones a realizar en caso de que la condición sea falsa. Esta última parte es razón de ser de la existencia de la acción **Sino**.

- d.** No siempre que exista un condicional **Si** debe existir un **Sino** asociado a él. Tal es el caso de la Decisión:

Si esta cerrada.

Abrirla.

Sin embargo siempre que exista un **Sino** es porque está asociado a un **Si** condicional.

Si la ventana está abierta no hay que hacer más que asomarse por ella pero si está cerrada debemos primero abrirla para poder asomarnos por ella.

Ejemplo 2: Retomando el segundo ejemplo, y sabiendo que contamos con una estructura para mejorar los algoritmos, podremos adecuarlo de manera que el algoritmo para colocarnos una camisa quede de la siguiente forma:

Solución:

Algoritmo para colocarnos una camisa.

Inicio.

Dirigirnos a nuestro ropero.

Si esta cerrado.

Abrirlo.

Tomar una camisa.

Si está abrochada.

Desabrocharla.

Abrir la camisa.

Si está doblada.

Desdoblarla.

Meter un brazo por una de sus mangas.

Meter el otro brazo por la otra de sus mangas.

Ajustar la camisa al tronco.

Si es una camisa de botones.

Abotonarla (botón a botón).

Ajustarla al cuerpo.

Sino.

Ajustarla de manera que quede bien puesta.

Fin.

En este ejemplo podemos notar una utilización alta de condicionales **Si** que no tienen necesidad de tener un **Sino** por las razones lógicas del mismo algoritmo. Es obvio que podremos tener muchas “objeciones” a este algoritmo porque en algún sentido alguno o algunos de los pasos aquí escritos no coinciden con nuestra lógica, pero tengamos en cuenta que todos los algoritmos planteados aquí son solo una idea de quien lo escribe y si se logran los mismos objetivos, aunque el algoritmo sea diferente al que nosotros estemos pensando estará completamente correcto.

Con seguridad existen muchas diferencias de concepción sobre todo en cuanto a este tipo de algoritmos informales pero lo importante es que nos vayamos acostumbrando a una **filosofía propia de los algoritmos** para expresar cualquier idea.

V. IDENTIFICACIÓN DE VARIABLES

Una gran cantidad de problemas involucran el uso de una o más variables, para la solución de este tipo de problemas es fundamental la localización de dichas variables, esto implica que la interpretación del problema la hagamos con cuidado para que sea clara y podamos identificar las variables involucradas y, de esta manera podamos usar adecuadamente los datos y las relaciones existentes entre ellos para encontrar satisfactoriamente la solución buscada.

Ejemplo: Luis pesa menos que Antonio, pero más que Pablo. Pablo pesa menos que Luis, pero más que Esteban. ¿Quién pesa más y quién le sigue en esta variable? ¹

Varios de los problemas se pueden resolver usando algoritmos semejantes, los cuales difieren en las variables involucradas, en el tipo de representación que se utilizará (gráfica, algebraica, etc.) o en los procesos involucrados. A continuación se presenta la propuesta de un algoritmo general que de solución a este tipo de problemas:

Algoritmo general para resolver problemas con variables.²

Inicio.

- Leer todo el problema.
- Identificar las variables.
- Identificar lo que se pide en el problema.
- Decidir el tipo de representación o los procesos que se requieren utilizar.
- Leer el problema, parte por parte, y representar en el diagrama los datos que se dan en cada parte o aplicar los procesos seleccionados.
- Observar el diagrama o los resultados obtenidos en los procesos y formular la respuesta del problema.

Fin.

¹ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 165.

² DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas.

Aplicaremos este algoritmo para encontrar la solución al ejemplo:

Algoritmo General para Resolver Problemas con Variables.	Algoritmo General Aplicado al Ejemplo.
Inicio.	Inicio.
Leer todo el problema.	Luis pesa menos que Antonio, pero más que Pablo. Pablo pesa menos que Luis, pero más que Esteban. ¿Quién pesa más y quién le sigue en esta variable?
Identificar las variables.	Variables: Peso de Luis = p_L Peso de Antonio = p_A Peso de Pablo = p_P Peso de Esteban = p_E
Identificar lo que se pide en el problema.	Indicar que persona es la que pesa más y quién lo sigue en peso.
Tipo de representación.	Signo de representación matemática menor que (<).
Leer el problema parte por parte y representarlo.	Luis pesa menos que Antonio: $p_L < p_A$ pero más que Pablo. Pablo pesa menos que Luis: $p_P < p_L$ pero más que Esteban: $p_E < p_P$
Observar el diagrama o los resultados obtenidos.	$p_E < p_P < p_L < p_A$
Formular la respuesta del problema.	La persona de mayor peso es Antonio, le sigue Luis, luego Pablo y finalmente Esteban.
Fin.	Fin.

Ejemplos Algoritmo General para Resolver Problemas con Variables

Ejemplo 1: El Distrito Federal y Nueva York tienen mayor grado de contaminación que Monterrey. Nueva York tiene más contaminación que Tokio, pero menos que el Distrito Federal y Monterrey tiene menos contaminación que Tokio. ¿Cuál es la ciudad más contaminada y cuál es la menos contaminada?¹

Solución	Aplicando el Algoritmo
Inicio.	Inicio.
Leer todo el problema.	Leer el problema.
Identificar las variables.	Variables: Contaminación del Distrito Federal = c_{DF} . Contaminación de Nueva York = c_{NY} . Contaminación de Monterrey = c_M . Contaminación de Tokio = c_T .
Identificar lo que se pide en el problema.	Indicar cuál es la ciudad más contaminada y cuál la menos contaminada.
Tipo de representación.	Signo de representación matemática mayor que ($>$).
Leer el problema parte por parte y representarlo.	El Distrito Federal y Nueva York tienen mayor grado de contaminación que Monterrey. $c_{DF} > c_M$ y $c_{NY} > c_M$ Nueva York tiene más contaminación que Tokio, pero menos que el Distrito Federal. $c_{DF} > c_{NY} > c_T$ Monterrey tiene menos contaminación que Tokio. $c_T > c_M$
Observar el diagrama o los resultados obtenidos.	$c_{DF} > c_{NY} > c_T > c_M$
Formular la respuesta del problema.	La ciudad más contaminada es: el Distrito Federal. La ciudad con menor contaminación es: Monterrey.
Fin.	Fin.

¹ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 169.

Ejemplo 2: Se desea encontrar la longitud y el área de un círculo de radio 5.

Solución	Aplicando el Algoritmo
Inicio.	Inicio.
Leer todo el problema.	Se desea encontrar la longitud y el área de un círculo de radio 5.
Identificar las variables.	Variables: L = longitud A = área R = radio $Pi \approx 3.1416$
Identificar lo que se pide en el problema.	Calcular la longitud y el área del círculo.
Tipo de representación.	Matemática: $A = (Pi)(R^2)$ $L = 2 (Pi) (R)$
Leer el problema parte por parte y representarlo.	$A = (3.1416) (5^2)$ $L = 2 (3.1416) (5)$
Observar el diagrama o los resultados obtenidos.	$A = (3.1416) (5^2)$ $= 78.54$ $L = 2 (3.1416) (5)$ $= 31.416$
Formular la respuesta del problema.	El valor del área es 78.54 u. La longitud encontrada es 31.416 u.
Fin.	Fin.

Ejemplo 3: Las personas que asistieron a una reunión se estrecharon la mano. Uno de ellos advirtió que los apretones de manos fueron 66. ¿Cuántas personas concurrieron a la reunión?²

Solución	Aplicando el Algoritmo
Inicio.	Inicio.
Leer todo el problema.	Lectura del problema.
Identificar las variables.	Variables: X , para representar a las personas que asistieron a la fiesta.
Identificar lo que se pide en el problema.	Cuántas personas asistieron a la reunión.
Tipo de representación.	Algebraica.
Leer el problema parte por parte y representarlo.	X ; personas asistieron a la fiesta. (X-1) ; cada persona dio la mano a X – 1 personas. El apretón entre dos personas debe ser considerado como uno.
Observar el diagrama o los resultados obtenidos.	$\frac{X(X-1)}{2} = 66$
Formular la respuesta del problema.	$X^2 - X - 132 = 0$ Resolviendo: $X_1 = 12$ y $X_2 = -11$ La respuesta X_2 carece de sentido, por lo que la solución es: A la fiesta asistieron 12 personas.
Fin.	Fin.

² ALGEBRA RECREATIVA Y. PERELMAN EDICIONES QUINTO SOL.

Ejercicios Algoritmo General para Resolver Problemas con Variables.

Instrucciones: Resuelve en tu cuaderno de trabajo los siguientes problemas, usa el algoritmo general para resolver problemas con variable.

1. José es más rápido que Tomás. Pedro es más rápido que Samuel, pero a diferencia de José más lento que Tomás. Por otra parte, se sabe que José es más lento que Miguel y Samuel más rápido que Jacobo. ¿Quién es el más rápido?³
2. Rubén obtuvo un promedio de calificaciones inferior a Marcelo. Isidro, por el contrario, obtuvo un promedio más alto que Iván. Marcelo, a pesar de haber sacado buenas calificaciones en la escuela, obtuvo un promedio inferior al de Iván. ¿Quién obtuvo el mayor promedio y quién le sigue?⁴
3. Gloria es mayor que Josefina, pero no tanto como Carmen, quien a su vez es mayor que Pamela y que Natalia. Natalia, por otra parte, es más joven que Pamela, pero mayor que Josefina y Gloria. ¿Quién es la más Joven?⁵
4. El Sol es la fuente calorífica del sistema planetario solar. El calor que reciben los planetas está en relación directa con su distancia al Sol. De los siguientes planetas: Neptuno, Plutón, Urano y Saturno, ¿Cuál es el menos caliente y cuál el menos frío? Si Neptuno es más frío que Saturno. Éste a su vez es más caliente que Urano. Plutón es el planeta más frío y saturno es mucho más caliente que Neptuno. Urano es más frío que Neptuno.⁶
5. Regocijándose los monos divididos en dos bandos: su octava parte al cuadrado en el bosque se solaza. Con alegres gritos 12 astronando el campo están. ¿Sabes cuántos monos hay en la manada, en total?⁷
6. Un caballo y un mulo caminaban juntos llevando sobre sus lomos pesados sacos. Lamentábase el jamelgo de su enojosa carga, a lo que el mulo le dijo: ¿De qué te quejas? Si yo te tomara un saco, mi carga sería el doble que la tuya. En cambio, si te doy un saco, tu carga se igualará a la mía. ¿Decidme, doctos matemáticos, cuántos sacos llevaba el caballo, y cuántos el mulo?⁸

³ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 173.

⁴ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 176.

⁵ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 174.

⁶ DESARROLLO DE HABILIDADES DEL PENSAMIENTO. Pensamiento Verbal y Solución de Problemas. Margarita A. de Sánchez. Editorial Trillas. Pág. 176.

⁷ ALGEBRA RECREATIVA Y. PERELMAN EDICIONES QUINTO SOL. PAG. 171.

⁸ ALGEBRA RECREATIVA Y. PERELMAN EDICIONES QUINTO SOL. PAG. 47.

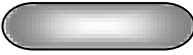
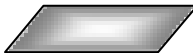
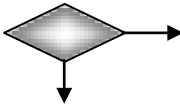

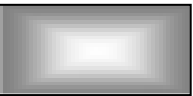
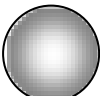
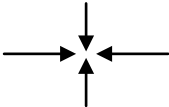

VI. TÉCNICAS PARA REPRESENTAR ALGORITMOS

1. Diagramas de Flujo

Un diagrama de flujo es la representación gráfica de un algoritmo. Muestra gráficamente los pasos a seguir para alcanzar la solución de un problema. Esto significa que la solución de un problema la podemos expresar por medio de símbolos gráficos.

Simbología:

La organización ANSI (*American National Standard Institute*) y la ISO (*International Standardization Organization*) han estandarizado los símbolos utilizados en los diagramas de flujo. A continuación se muestran estos elementos:

Símbolo	Descripción
	Principio/Fin. Representa el <i>inicio</i> y el <i>final</i> de un programa o algoritmo.
	Entrada de Datos. Símbolo utilizado para introducir los datos de entrada. Expresa <i>lectura</i> .
	Toma de Decisión. Representa operaciones de comparación entre datos en las que se requiere determinar qué opción se debe seguir.
	Impresión. Se utiliza para representar la salida de datos en un programa. Expresa <i>escritura</i> .
	Proceso. Indica las instrucciones para la manipulación de datos. Expresa <i>proceso</i> . En su interior se expresan asignaciones, operaciones aritméticas, cambios de valor entre variables, etc.
	Conector. Enlaza dos partes del diagrama de flujo que se encuentran en la misma página. Expresa conexión dentro de una misma página.
	Indicador de dirección o Línea de Flujo. Indica el sentido de ejecución de las instrucciones. Expresa la dirección del flujo del diagrama.
	Proceso repetitivo o iterativo. Indica la repetición de un proceso en un intervalo determinado.

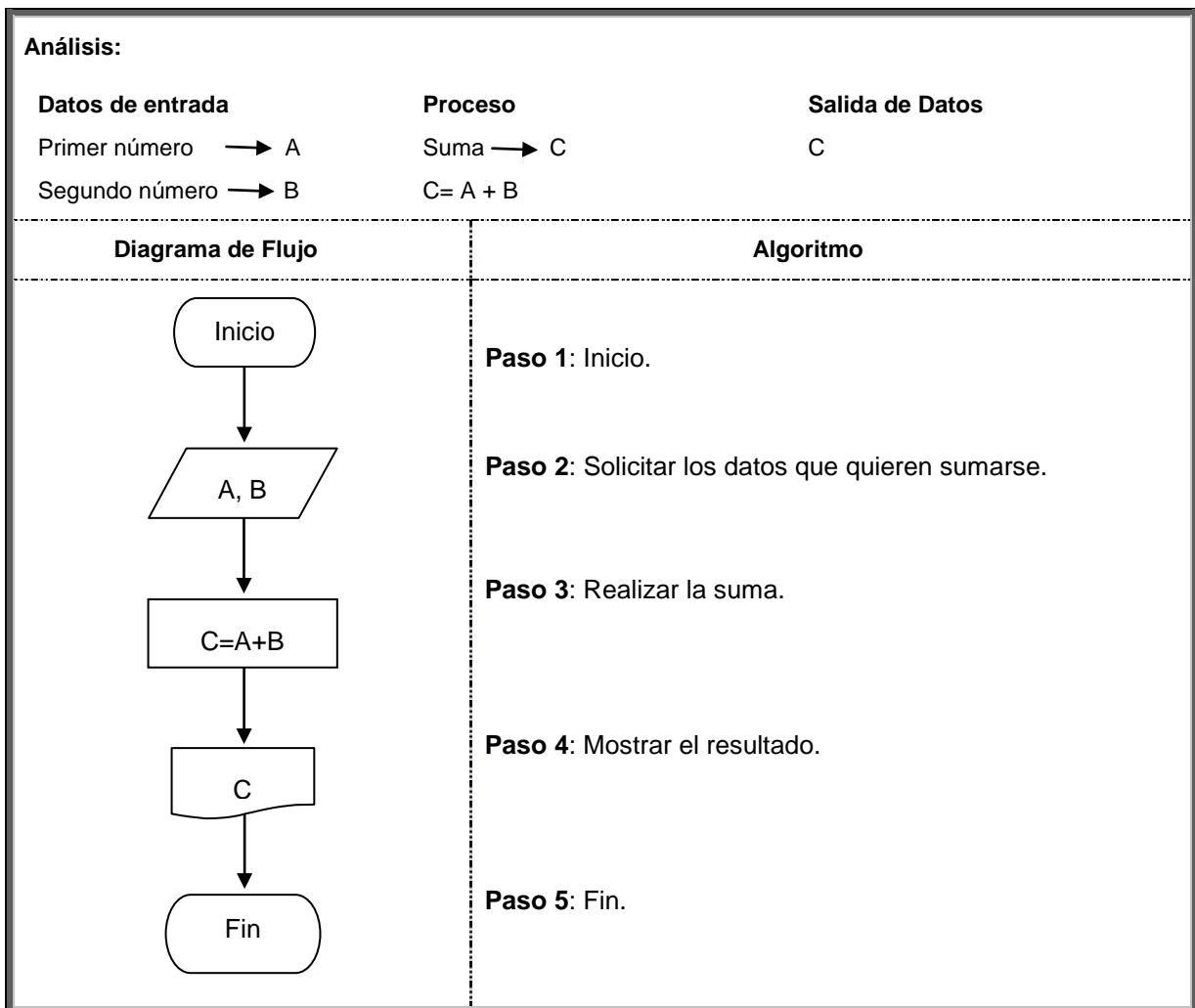
Reglas para construir un diagrama de flujo

- a. Los diagramas de flujo deben escribirse de arriba hacia abajo.
- b. Los símbolos se unen con líneas.
- c. Se usan flechas para indicar el flujo del programa.
- d. Sólo los símbolos de decisión pueden y deben tener más de una línea de flujo de salida.
- e. Todos los símbolos pueden tener más de una línea de entrada.

A continuación mostraremos el uso de estas técnicas usadas en la solución de problemas.

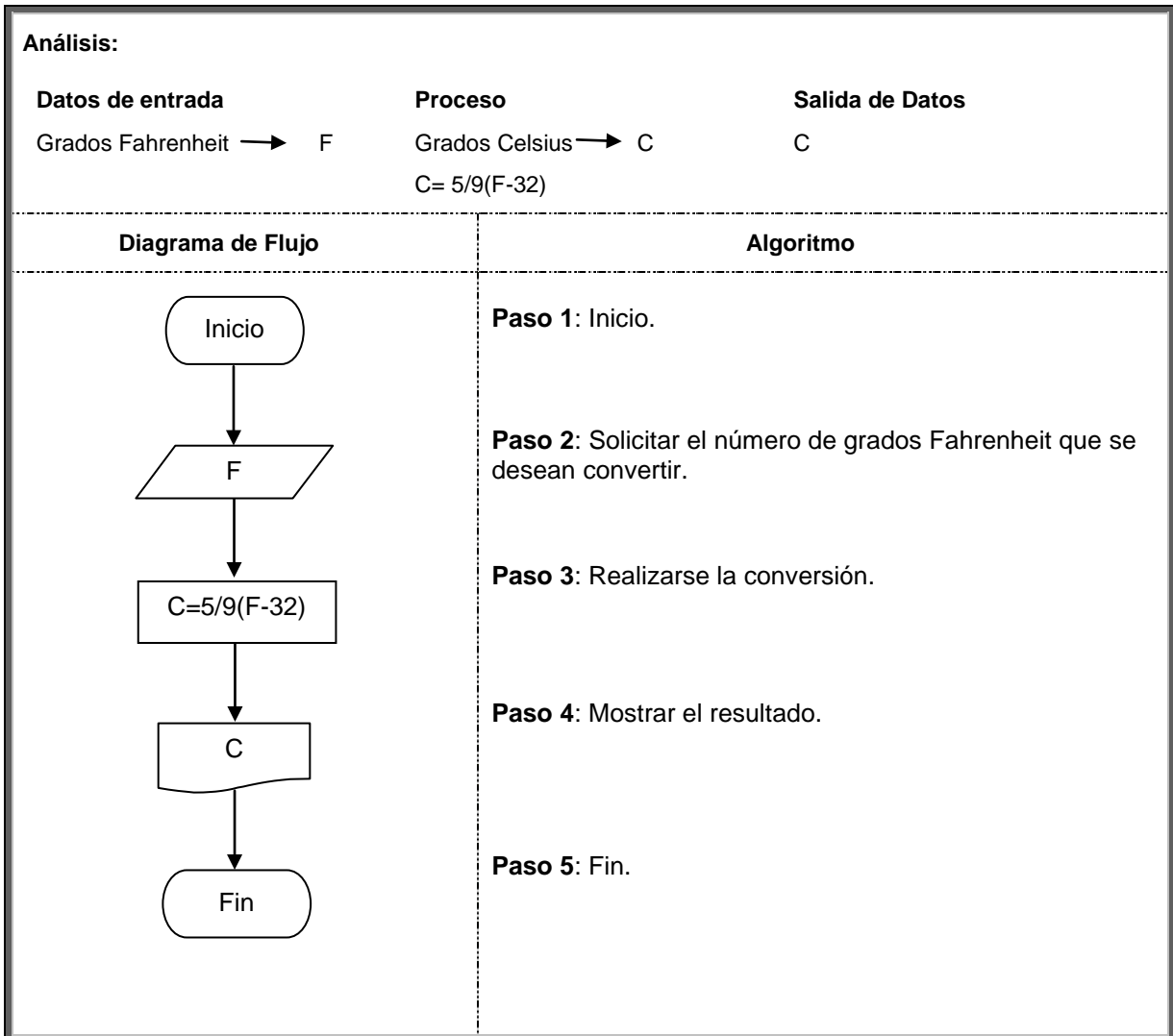
Ejemplos estructura secuencial:

Ejemplo 1: Realizaremos el diagrama de flujo para el proceso de suma de dos números:



Como podemos observar en el ejercicio anterior cada uno de los pasos correspondientes al algoritmo tiene su equivalente en el diagrama de flujo. Y en ambos se pueden identificar los siguientes elementos: Entrada, Proceso y Salida.

Ejemplo 2: Realizaremos el diagrama de flujo para calcular la conversión de grados Fahrenheit a Celsius.



Ejercicios estructura secuencial:

Instrucciones: En los siguientes ejercicios realiza el análisis del problema, el diagrama de flujo y el algoritmo en la siguiente tabla:

1. Se tienen tres números enteros, calcula la multiplicación, la suma y el promedio de los mismos.
2. Calcular perímetro de un romboide dada su Base (b) y Altura (a).
3. Dada las horas trabajadas de una persona y su tarifa de pago. Calcular su salario.
4. Escribe un programa que lea una cantidad depositada en un banco y que calcule la cantidad final después de aplicarle un 3 % de interés.
5. Un maestro desea saber qué porcentaje de hombres y que porcentaje de mujeres hay en un grupo de estudiantes.
6. Dado el costo de un artículo y el porcentaje de ganancia que se quiere obtener calcular el costo al que se debe vender dicho artículo.
7. Calcular el área de un cuadrado.
8. Calcular el valor de la variable y para la ecuación $y=3x^2$
9. Calcular el costo total de la compra de 1110 bates de béisbol cuyo costo es de \$10 cada uno.
10. Dos pueblos distan X Km, están unidos por una carretera recta. Un ciclista viaja de un pueblo al otro con una velocidad constante de 10 m/s. Calcula el tiempo que emplea el ciclista para trasladarse de un pueblo al otro.

Capítulo 3

JavaScript

Temas:

- I. Páginas Dinámicas.
- II. ¿Qué es JavaScript?
- III. Código JavaScript en una Página HTML.
- IV. Formas y Formularios.
- V. Variables y Constantes.



I. PÁGINAS DINÁMICAS

El lenguaje HTML visto en los capítulos anteriores es suficiente para visualizar documentos, imágenes, sonidos y otros elementos multimedia, pero el resultado es siempre una página estática.

Entonces, ¿qué se puede hacer para construir una página dinámica?, entendiendo por página dinámica una página que actualiza su contenido mientras se visualiza.

Aunque HTML ha sido ampliado para soportar contenidos dinámicos, inicialmente no fue pensado para esto. En cualquier caso hay lenguajes como JavaScript que permite incorporar instrucciones para tomar el control sobre los contenidos de una página WEB.

Lo que hay que tomar en cuenta:

- El conjunto de instrucciones no puede ser interpretado por todos los navegadores que existen, el único navegador que interpreta las instrucciones es el Explorer de Microsoft, a partir de la versión 3.0.
- Para evitar que el explorador interprete las líneas entre las etiquetas anteriores, como instrucciones HTML, es necesario definir las entre las siguientes etiquetas:
<!-- Código en JavaScript -->
- En este lenguaje no existen separadores de instrucciones, por lo que cada una de éstas deberá ir en una línea aparte (Excepto en aquellas que se compongan de más de una línea, ej. FOR-NEXT, IF-THEN-ELSE, etc.).
- El carácter “_” sirve para delimitar el tamaño de una línea de tal forma que lo que se escriba en la siguiente forme parte de la anterior, básicamente con el fin de hacer más clara la programación del código.
- Para manipular una página dinámica es necesario crear interfaces. Las interfaces son pantallas gráficas donde el usuario va a tener contacto con el programa, es decir, va a poder ingresar datos o seguir las instrucciones que el programa solicite. A través de ésta, se podrán realizar las aplicaciones, por ejemplo: manejo de bases de datos, operaciones matemáticas y simular problemas a través de la computadora así como darles solución.

II. ¿QUÉ ES JAVASCRIPT?

JavaScript es un lenguaje de programación basado en EMACScript. Siendo tan popular en la web, está presente en todos los exploradores modernos, por lo cual en cualquier equipo con un sistema operativo que cuente con explorador puedes desarrollar y probar código con JavaScript. Es un lenguaje que cumple lo siguiente:

- a. Se ejecuta como apoyo al lenguaje HTML.
- b. No necesita compilación.

- c. Únicamente se ejecuta dentro de un programa mayor, en este caso el navegador.

III. CÓDIGO JAVASCRIPT EN UNA PÁGINA HTML

Para incluir instrucciones JavaScript en una página HTML hay que utilizar la etiqueta `<script>` con el atributo LANGUAGE.

Por ejemplo:

```
<script language="JavaScript">
    <!--
        Conjunto de instrucciones en JavaScript
    -->
</script>
```

Observa que la propiedad **language** de la etiqueta `<script>` indica que el código que hay hasta `</script>` es código JavaScript.

Para empezar a trabajar con código JavaScript en una página HTML es necesario conocer las tres secciones que contiene el documento HTML:

1. `<head></head>`
2. `<body></body>`
3. `<form></form>`

Las dos primeras ya las conoces; la tercera permite introducir formularios. Cuando se carga una página, lo primero en ejecutarse es la sección que se encuentra entre las etiquetas `<head>` `</head>`; por lo tanto, este es el lugar idóneo para colocar código JavaScript que es accesible desde otras partes de la página. El código JavaScript que se incluya en la sección `<head>` tiene ámbito global, es decir, el código es reconocido en todo el documento; en cambio, el que se incluya en la sección `<body>` y `<form>` tiene ámbito local, solo será reconocido dentro de esta sección.

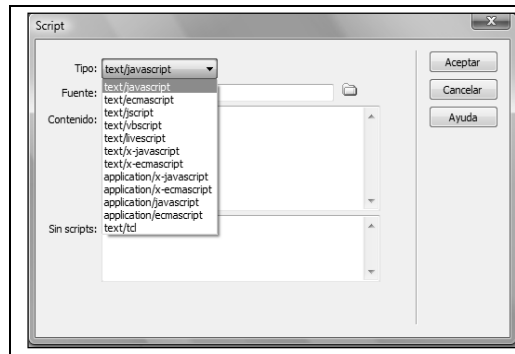
Puedes utilizar bloques `<script>` en cualquier lugar dentro de una página HTML. Puedes situarlos en las secciones `<body>` y `<head>`. Sin embargo, seguramente desearás situar todo el código de ejecución de secuencias de propósito general en la sección `<head>` para mantener todo el programa junto.

Al mantener tú programa en la sección `<head>` asegurarás que sea leído y decodificado todo el código antes de que se realice la llamada desde la sección `<body>`.

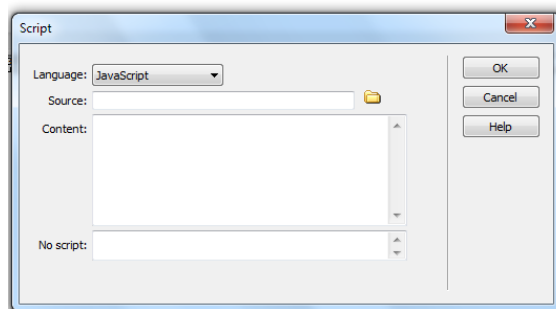
En Dreamweaver, incluyes el código JavaScript a partir de la etiqueta HTML o Común (según sea la versión):



En el icono , el cual permite insertar el script, conforme a lo siguiente:



En la opción “Tipo” seleccionas JavaScript y en “Contenido”, se escribe el código correspondiente.



IV. FORMAS Y FORMULARIOS

Los formularios se ven en páginas Web para completar una encuesta, enviar un mensaje, capturar información, mostrar datos, seleccionar información clasificada, ejecutar acciones, entre otras.

Los formularios son los elementos en HTML que permiten la interactividad con el usuario al estar destinados a la recolección de datos para posteriormente enviarlos a un servidor.

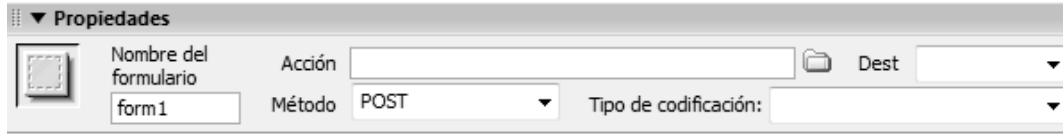
Un formulario se compone por una serie de campos que el usuario puede llenar desde una página Web. Todos los campos que componen al formulario se encierran entre las etiquetas `<form></form>`.

Propiedades del formulario: Etiqueta `<form></form>`

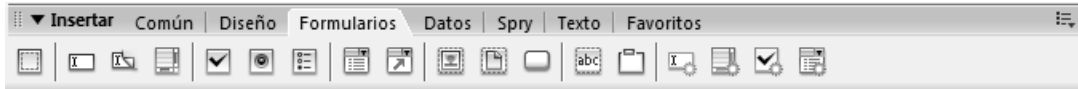
Propiedad	Valor	Función
name	text	Indica el nombre del formulario y sirve para poder hacer referencia a él desde el código JavaScript.


En Dreamweaver, trabajar con formularios se convierte en sencillos pasos:

- a. En el menú Insertar → Formulario → Formulario
- b. En la ventana de propiedades se le asigna nombre al formulario



O desde la pestaña de Formularios:



Y el icono , para insertar el formulario.

1. Etiqueta input

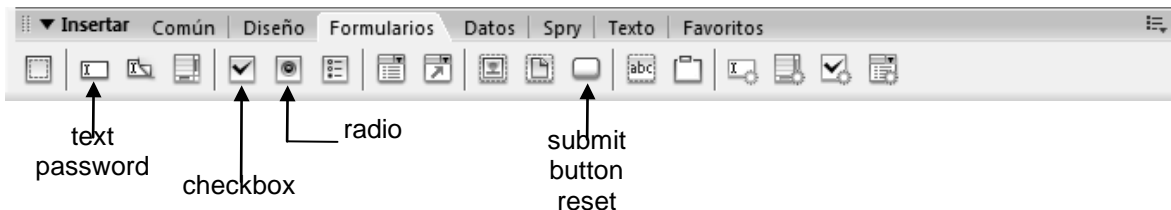
Varios de los campos de un formulario (campos de texto, botones, selección, opción) se crean con la etiqueta <input>.

Propiedades de los campos de un formulario: Etiqueta <input ...>

Propiedad	Valor	Función	Elemento muestra
type	text	Crea un campo de texto donde el usuario puede escribir desde un número hasta una línea de texto.	<input type="text"/>
type	password	Campo de texto donde el usuario puede escribir desde un número hasta una línea de texto, sustituyéndolo por “*”.	<input type="password"/>
type	checkbox	Es una caja de selección en la que el usuario, al activarla, escribe una marca.	<input type="checkbox"/>
type	radio	Es un círculo pequeño que puede ser activado, y no pueden activarse al mismo tiempo varios de ellos.	<input type="radio"/>
type	submit	Es un botón que cuando el usuario le da clic provoca el envío de los datos del formulario al servidor	<input type="submit" value="Enviar consulta"/>
type	button	Es un botón cuya finalidad es la de provocar una acción al darle clic y que debe ir acompañado por el control del evento “OnClick”	<input type="button"/>
type	reset	Botón que borra todos los datos que el usuario haya incluido en el formulario, devolviendo todos los elementos al estado por defecto.	<input type="reset" value="Restablecer"/>


Cada uno de estos elementos del formulario tiene características propias, las cuales se pueden asignar con diferentes propiedades:

Para insertar cada uno de los elementos del formulario, lo puedes hacer desde la barra de Formularios:

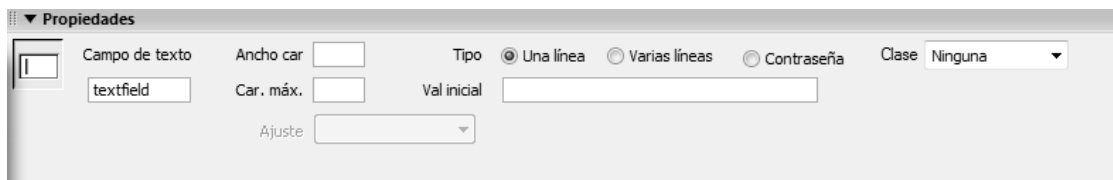


Propiedades campo de texto: Etiqueta <input type="text">

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre del campo de texto
size	número	Determina la longitud del campo de texto en la pantalla.
maxlength	número	Determina el máximo número de caracteres que se pueden teclear.
value	alfanumérico	Asigna un contenido predeterminado al campo.


En Dreamweaver insertas el campo de texto con el icono  o desde el menú Insertar → Formulario → Campo de texto

Aparecen sus propiedades en la ventana:



Propiedades campo de texto password: Etiqueta <input type="password">

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre del campo de texto password.
size	número	Determina la longitud del campo de texto en la pantalla.
maxlength	número	Determina el máximo número de caracteres que se pueden teclear.

En Dreamweaver insertas el campo de texto con el icono  o desde el menú Insertar → Formulario → Campo de texto

Aparecen sus propiedades en la ventana:



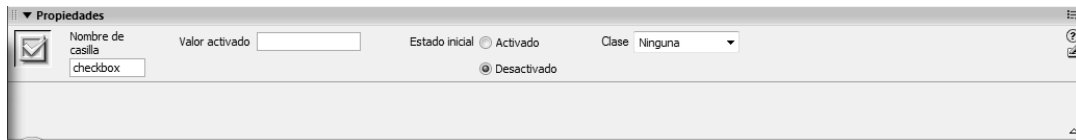
En la opción “Tipo” seleccionas “Contraseña”, para que el elemento sea de tipo password.

Propiedades de la casilla de verificación: Etiqueta <input type=“checkbox”>

Propiedad	Valor	Función
Name	alfanumérico	Determina el nombre de la casilla de verificación.
Value	Alfanumérico	Asigna un contenido predeterminado a la casilla.
Checked		En caso de que se requiera que la casilla aparezca activada al momento de abrir la página.

En Dreamweaver insertas la casilla de verificación con el icono o desde el menú Insertar → Formulario → Casilla de verificación.

Aparecen sus propiedades en la ventana:

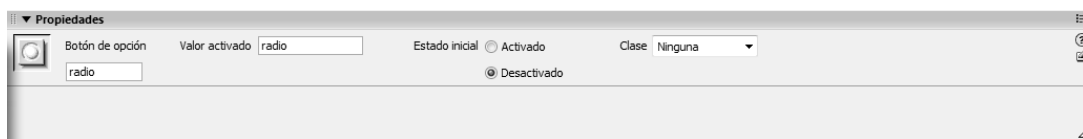


Propiedades de casilla de opción: Etiqueta <input type=“radio”>

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre de la opción.
value	Alfanumérico	Asigna un contenido predeterminado a la opción.
checked		En caso de que se requiera que la opción aparezca activada al momento de abrir la página.

En Dreamweaver insertas la casilla de opción con el icono o desde el menú Insertar → Formulario → Botón de opción.

Aparecen sus propiedades en la ventana:

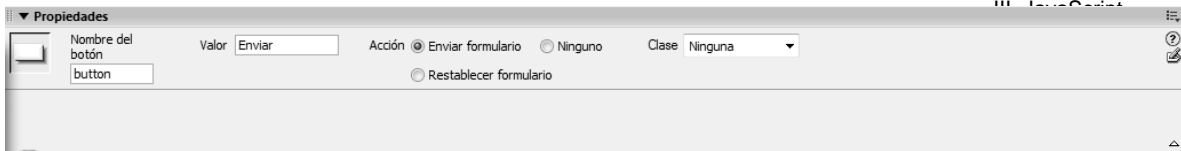


Propiedades del botón enviar consulta: Etiqueta <input type=“submit”>

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre del botón.
value	alfanumérico	Leyenda que se requiere aparezca sobre el botón.

En Dreamweaver insertas la casilla de opción con el icono o desde el menú Insertar → Formulario → Botón.


Aparecen sus propiedades en la ventana:



En la opción “Acción” se marca “Enviar formulario”

Propiedades del botón: Etiqueta <input type=“button”>

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre del botón.
value	alfanumérico	Leyenda que se requiere aparezca sobre el botón.

En Dreamweaver insertas la casilla de opción con el icono  o desde el menú Insertar → Formulario → Botón.


Aparecen sus propiedades en la ventana:

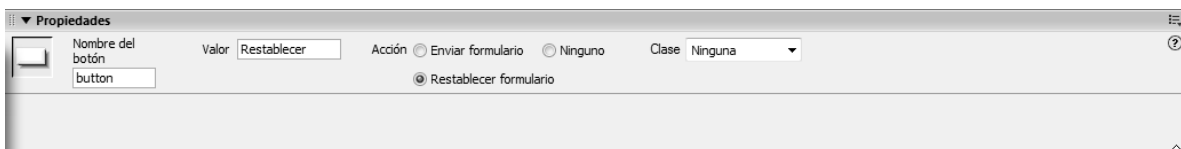


En la opción “Acción” se marca “Ninguno”

Propiedades del botón borrar: Etiqueta <input type=“reset”>

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre del botón.
value	alfanumérico	Leyenda que se requiere aparezca sobre el botón.

En Dreamweaver insertas la casilla de opción con el icono  o desde el menú Insertar → Formulario → Botón.



Aparecen sus propiedades en la ventana:

En la opción “Acción” se marca “Restablecer formulario”

Ejemplo. Desarrolla el código HTML para generar una página que contenga como título “Ficha de registro” y dentro del documento el encabezado de tamaño 2 “REGISTRO”; genera cada uno de los campos necesarios dentro del formulario de nombre “miformulario” para mostrar en la página una ficha de registro.

Solución:

Diseño Dreamweaver:



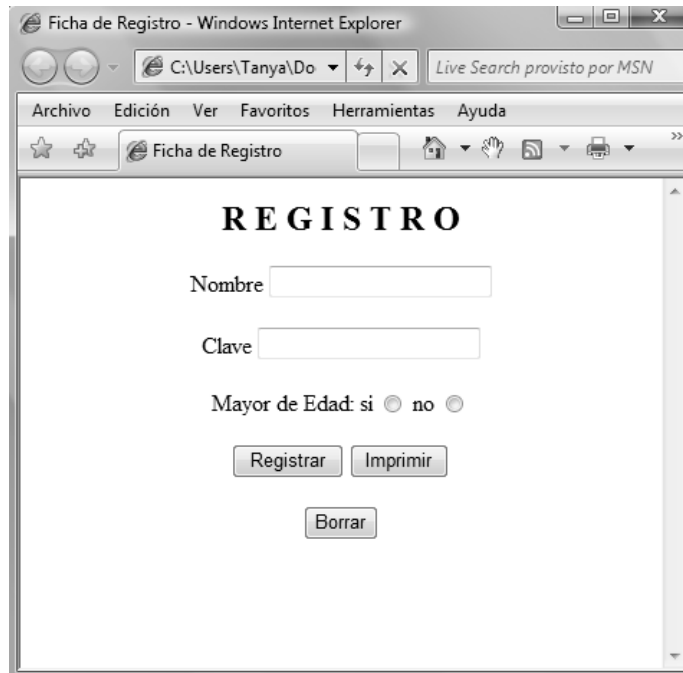
Código Dreamweaver:

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>Ficha de Registro</title>
7 </head>
8 <body>
9 <form id="miformulario" name="miformulario" method="post" action="">
10 <h2 align="center">R E G I S T R O</h2>
11 <p align="center">Nombre
12 <input type="text" name="Nombre" id="Nombre" />
13 </p>
14 <p align="center">Clave
15 <input type="password" name="clave" id="clave" />
16 </p>
17 <p align="center">Mayor de Edad:
18 <input type="radio" name="edad" id="radio" value="radio" />
19 <input type="radio" name="edad" id="radio2" value="radio2" />
20 </p>
21 <p align="center">
22 <input type="submit" name="button" id="button" value="Registrar" />
23 <input type="button" name="button2" id="button2" value="Imprimir" />
24 </p>
25 <p align="center">
26 <input type="reset" name="button3" id="button3" value="Borrar" />
27 </p>
28 </form>
29 </body>
30 </html>

```

La página en el explorador:




Otros elementos que componen un formulario son:

- **Área de texto: <textarea></textarea>**

Este tipo de campo es utilizado para dar la oportunidad al usuario de escribir un mensaje largo. Es una caja de texto con barras de desplazamiento. Para generar un campo de este tipo se utiliza la etiqueta <textarea></textarea>.

Propiedades del área de texto: Etiqueta <textarea></textarea>

Propiedad	Valor	Función
name	alfanumérico	Determina el nombre de la caja de texto.
cols	Número	Expresa el ancho, en caracteres, de la caja de texto.
rows	Número	Expresa el número de filas que se muestran en pantalla.
Wrap	Virtual/physical	Ajusta el texto que se escriba en la caja de texto.

En Dreamweaver insertas la casilla de opción con el icono  o desde el menú Insertar → Formulario → Área de texto.

Aparecen sus propiedades en la ventana:




• **Listas desplegables: <select></select>**

Este tipo de campo se emplea para elegir una o más opciones de una lista. Una lista desplegable se define entre las etiquetas <select></select>. Dentro se define cada una de las opciones necesarias entre las etiquetas <option></option>

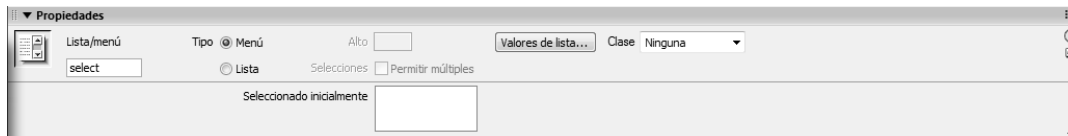
```
<select>
  <option> Texto </option>
  ...
</select>
```

Propiedades de la lista desplegable: Etiqueta <select></select>

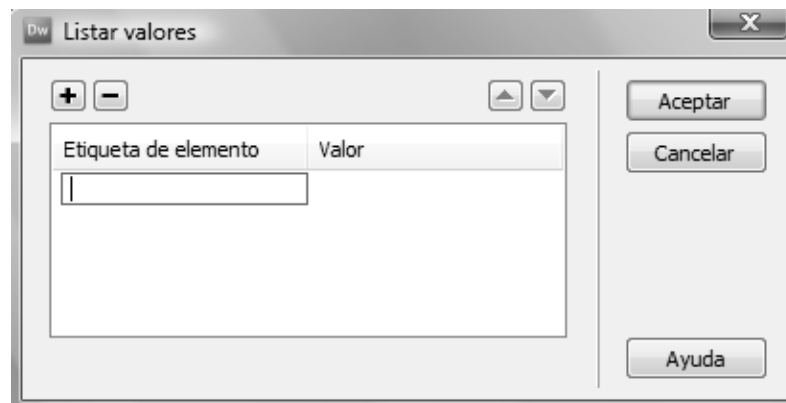
Propiedad	Valor	Función
name	alfanumérico	Determina el nombre de la lista.
size	Número	Indica cuántas opciones se ven simultáneamente en la pantalla.

En Dreamweaver insertas la casilla de opción con el icono  o desde el menú Insertar → Formulario → Lista/menú

Aparecen sus propiedades en la ventana:



Y en el botón “Valores de lista...”, escribes las opciones de la lista desplegable:



Ejemplo. Desarrolla el código HTML para generar una página que contenga como título “Ficha de registro” y dentro del documento el encabezado de tamaño 2 “REGISTRO”; genera cada uno de los campos necesarios dentro del formulario de nombre “miformulario” para mostrar en la página una ficha de registro, incluyendo los últimos 2 elementos mencionados (guárdalo como “formulario2.html”)

Solución:

Diseño Dreamweaver:



Código Dreamweaver:


```

formulario2.html
Código Dividir Diseño Título: Ficha de Registro Comprobar página
"http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8" /><title>Ficha de Registro</title></head><body>
2 <form id="miformulario" name="miformulario" method="post" action="">
3   <h2 align="center">R E G I S T R O</h2>
4   <table width="439" border="2" align="center" cellpadding="5" cellspacing="5">
5     <tr><td width="106">Nombre:</td>
6       <td colspan="2"><input type="text" name="Nombre" id="Nombre" /></td>
7     </tr><tr><td>Clave:</td>
8       <td colspan="2"><input type="password" name="clave" id="clave" /></td>
9     </tr><tr>
10      <td>Mayor de Edad:</td><td width="133">si
11      <input type="radio" name="edad" id="radio" value="radio" /></td>
12      <td width="140">no
13      <input type="radio" name="edad" id="radio2" value="radio2" /></td>
14    </tr><tr><td>Estado:</td>
15      <td colspan="2">
16        <select name="select" id="select">
17          <option value="0" selected="selected">Selecciona un Estado</option>
18          <option value="1">Aguascalientes</option>
19          <option value="2">Baja California</option>
20          <option value="3">Durango</option>
21        </select>
22    </td></tr><tr><td valign="top">Comentarios: </td>
23      <td colspan="2"><textarea name="comentarios" cols="45" rows="5" wrap="physical" id="comentarios"></textarea></td>
24    </tr>
25  </table><div align="center">
26    <input type="submit" name="button" id="button" value="Registrar" />
27  </div><div align="center">
28    <input type="button" name="button2" id="button2" value="Imprimir" />
29  </div><div align="center">
30    <input type="reset" name="button3" id="button3" value="Borrar" />
31  </div></div></tr></table>
32 </form>
33 </body>

```

El generar el formulario dentro de una tabla ayuda a organizar y dar un mejor aspecto a la página; otro recurso que se puede utilizar para mejorar dicho aspecto es organizar nuestro formulario en conjunto de campos con la etiqueta **<fieldset></fieldset>**

Se puede dividir el formulario en tantos fieldsets como se quiera, pero para que la separación sea visible, se puede rotular cada conjunto de campos con la etiqueta **<legend></legend>**. Cada fieldset lleva un rótulo que hace que el formulario sea más agradable a la vista y cómodo de comprender al usuario.

En Dreamweaver insertas la casilla de opción con el icono  o desde el menú Insertar → Formulario → Juego de campos.

Ejemplo. Para el ejemplo anterior organiza el formulario utilizando tablas y la etiqueta fieldset.

Solución:

Diseño Dreamweaver:



Código Dreamweaver:

```

<form id="miformulario" name="miformulario" method="post" action="">
  <h2 align="center">R E G I S T R O</h2>
  <fieldset><legend>Datos Personales</legend>
  <table width="439" border="2" align="center" cellpadding="5" cellspacing="5">
    <tr><td width="106">Nombre:</td>
      <td colspan="2"><input type="text" name="Nombre" id="Nombre" /></td>
    </tr><tr><td>Clave:</td>
      <td colspan="2"><input type="password" name="clave" id="clave" /></td>
    </tr>
    <tr><td>Mayor de Edad:</td><td width="133">si
      <input type="radio" name="edad" id="radio" value="radio" /></td>
      <td width="140">no
      <input type="radio" name="edad" id="radio2" value="radio2" /></td>
    </tr>
  </table>
</fieldset>
<fieldset><legend>Residencia</legend>
<table width="439" border="2" align="center" cellpadding="5" cellspacing="5">
  <tr><td width="106">Estado:</td>
    <td colspan="2"><select name="select" id="select">
      <option value="0" selected="selected">Selecciona un Estado</option>
      <option value="1">Aguascalientes</option>
      <option value="2">Baja California</option>
      <option value="3">Durango</option>
    </select></td></tr></table>
</fieldset>
<fieldset><legend>Comentarios</legend>
<table width="439" border="2" align="center" cellpadding="5" cellspacing="5">
  <tr><td width="106" valign="top">Comentarios: </td>
    <td colspan="2"><textarea name="comentarios" cols="45" rows="5"
      id="comentarios"></textarea></td>
  </tr><tr><td><div align="center">
    <input type="submit" name="button" id="button" value="Registrar" />
  </div></td><td width="133"><div align="center">
    <input type="button" name="button2" id="button2" value="Imprimir" />
  </div></td><td width="140"><div align="center">
    <input type="reset" name="button3" id="button3" value="Borrar" />
  </div></td></tr></table>
</fieldset>
</form>

```


Página de resultado en el explorador:

The image shows a screenshot of a web browser window titled "Ficha de Registro - Windows Internet Explorer". The address bar shows the path "C:\Users\Tanya\Documer" and the search engine "Live Search provisto por MSN". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar contains various icons for navigation and actions. The main content area displays a registration form with the following sections:

- REGISTRO**: The main title of the form.
- Datos Personales**: A section containing three input fields:
 - Nombre: [Text input field]
 - Clave: [Text input field]
 - Mayor de Edad: [Radio button labeled "si"] [Radio button labeled "no"]
- Residencia**: A section containing one input field:
 - Estado: [Dropdown menu labeled "Selecciona un Estado"]
- Comentarios**: A section containing one input field:
 - Comentarios: [Text area]

At the bottom of the form, there are three buttons: "Registrar", "Imprimir", and "Borrar".

V. VARIABLES Y CONSTANTES

El uso de variables y constantes en el desarrollo de un programa es muy útil ya que por medio de éstas podemos realizar operaciones matemáticas: sumas, restas, multiplicaciones, entre otras. También podemos realizar operaciones lógicas, como comparaciones.

En el entorno de programación una variable es el nombre que se le asigna a un espacio de memoria que almacenará un valor que puede o no cambiar en el transcurso de un programa.

Las variables tienen tres características:

1. Nombre

Es el identificador con el que se podrá hacer referencia al espacio de memoria asignado. Tiene las siguientes reglas:

- El primer carácter debe ser una letra o los símbolos especiales "\$" y "_", pero no un número.
- Puede estar formada por letras, números y los caracteres especiales "\$" y "_".
- El número máximo de caracteres que lo forman es de 255.
- No debe declararse más de una vez la misma variable en una misma secuencia de comandos.

2. Tipo de dato

Se refiere a la clase de información que puede almacenar el espacio de memoria asignado: *Alfanumérico* (Texto o cadena de caracteres) o *Numérico* (Entero o con punto decimal)

3. Ámbito

Se refiere al área donde es reconocida la variable: en todo el documento (*Global*) o dentro de un procedimiento o sección específica (*Local*)

1. Declaración de variables

Todas las variables que vayamos a utilizar en el transcurso de un programa deben ser declaradas (Declarar una variable significa indicarle a la computadora el *nombre*, *tipo de dato* y *ámbito* de ésta), esto es con el fin de que el programa asigne un espacio de memoria donde pueda almacenar los valores dados.

Sintaxis para declarar una variable:

Var Nombre_Variable

Cabe mencionar que al trabajar con código JavaScript todas las variables al ser declaradas son del tipo **Variant** y es el uso y aplicación de estas por medio de funciones de conversión las que definirán otro tipo.

Ejemplos de nombres de variables:

A	Es una variable. Cumple todas las reglas.
H_2O	Es una variable. Cumple con todas las reglas.
\$Nombre:	Es una variable. Cumple con todas las reglas.
@marco99	No es una variable. No cumple la regla i.
20Veces	No es una variable. No cumple la regla i.

2. Tipos de Datos

Los tipos de datos permiten clasificar la información que ingresa a las variables, es decir, es la clase de valor que almacenará una variable.

Existe un solo tipo de dato: **Variant**, este tipo de dato a su vez, se convierte en un subtipo de dato de acuerdo a la información que ingresa a la memoria.

3. Implementación de variables

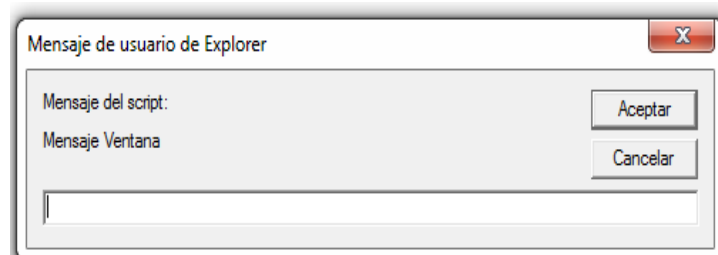
Después de conocer la forma de declarar una variable y qué tipo de datos se pueden manejar en JavaScript, es importante conocer las instrucciones necesarias para su implementación (asignación, procesamiento y escritura de datos).

- **Asignación de datos**

Consiste en pedir desde el teclado el valor dado a la variable o bien asignarlo directamente en el código.

Las instrucciones son:

Variable=**Prompt** ("Mensaje Ventana", " ")



O la instrucción:

Variable =document.**Formulario.Elemento.Value**

Donde:

document. → Indica que se trata de un documento que contiene un formulario (nunca cambia).

Formulario → Es el nombre dado al formulario diseñado.

Elemento → Es el nombre dado al elemento del formulario de donde se va a tomar el valor-

Value → Indica que se trata del valor que se va a asignar a la variable (nunca cambia).

Nota: Cada una de las palabras se separa por un punto.

- **Escritura de datos**

Consiste en imprimir en pantalla el valor dado a la variable que mostrará la salida de la aplicación una vez procesada la información.

Las instrucciones son:

alert(variable)

O la instrucción:



document.**Formulario.Elemento.Value=variable**

Donde:

document. → Indica que se trata de un documento que contiene un formulario (nunca cambia).

Formulario → Es el nombre dado al formulario diseñado.

Elemento → Es el nombre dado al elemento del formulario a donde se va a imprimir la variable.

Value → Indica que se trata de tomar el valor de la variable y mandarlo al elemento del formulario correspondiente (nunca cambia).

Nota: Cada una de las palabras se separa por un punto

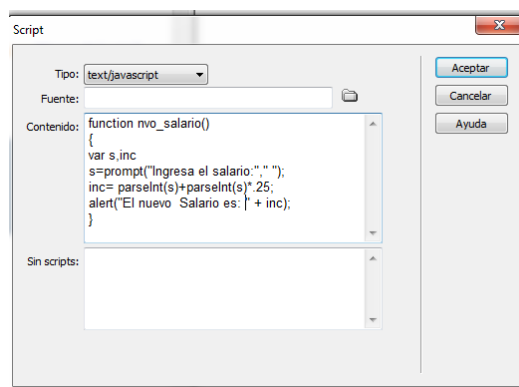
Ejemplo. Escribiremos una secuencia de código para que calcule el nuevo salario de un empleado que obtuvo un incremento del 25%. Utiliza el proceso de solución visto en el capítulo de metodología de la programación.

Solución:

1. **Análisis y definición del problema:**
Realizar el incremento del 25%
2. **Definición de los datos de entrada / salida:**
Entrada: Salario (s)

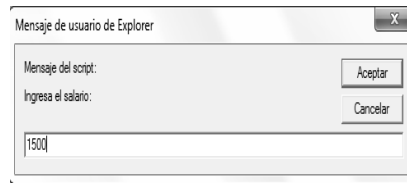
Algoritmo	Diagrama de Flujo	Interfaz Gráfica y Código
1. Inicio. 2. Leer salario (s). 3. Incremento (inc) = $s+s*.25$ 4. Imprimir inc. 5. Fin.	<pre> graph TD INICIO[INICIO] --> s[/s/] s --> inc_calc[inc=s+s*.25] inc_calc --> inc[/inc/] inc --> FIN[FIN] </pre>	<pre> <body> <form id="form1" name="form1" method="post" action=""> <div align="center"> <input type="button" name="salario" id="salario" value="Nuevo Salario" onclick="nvo_salario()" /> </div> </form> </body> </pre>

3. Código JavaScript: Código Dreamweaver Script (Forma 1)

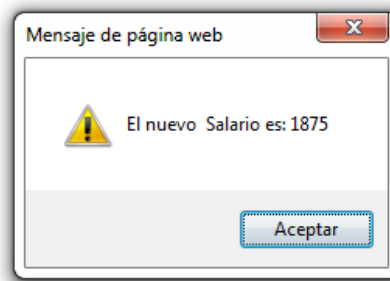


4. Comprobación:

Página HTML como se ve en el explorador después de dar clic en el botón Nuevo Salario:



La página que muestra el resultado del nuevo salario:



NOTA: La instrucción **ParseInt** permite convertir el valor de la variable a un tipo numérico entero (Numérico sin punto decimal).

Aunque se ha definido una variable para contener un tipo de datos determinado, JavaScript nos permite cambiar sobre la marcha el tipo asociado a la variable, y pone a nuestra disposición las siguientes funciones:

Función	Acción
ParseInt	Convierte un dato a tipo Integer
parseFloat	Convierte un dato a tipo Float

Es más, JavaScript permite cambiar el subtipo de dato asociado a una variable con tan sólo asignarle directamente un nuevo valor de otro tipo, por ejemplo:

```
miVariable = 35
...sentencias de código...
miVariable = "Hola Amigos"
```

y con esto hemos cambiado directamente el tipo de dato que contenía la variable. Es ésta una fuerte ventaja de JavaScript frente a otros lenguajes de alto nivel.

4. Constantes

Las constantes son variables que nunca cambian de valor, por lo que una vez definidas no es posible modificarlas directamente. Pueden contener datos de cualquiera de los tipos vistos antes. Para declararlas se usa la sentencia **Const**:

```
Const Pi = 3.1416
Const Saludo = " Bienvenidos "
```

5. Operadores Aritméticos

Un operador aritmético es un símbolo especial que permite, en JavaScript, ejecutar una operación aritmética: Suma, resta, división, multiplicación, entre otras.

En el lenguaje JavaScript se tienen los siguientes:

Símbolo	Descripción
+	Permite realizar la suma entre dos datos numéricos.
-	Permite realizar la resta entre dos datos numéricos.
*	Permite ejecutar la multiplicación entre dos datos numéricos.
/	Permite ejecutar la división entre dos valores numéricos.
%	Devuelve el residuo de una división.
+	Permite concatenar o unir caracteres.

Ejemplos:

Instrucción	Comentarios
10 + 5	Calcula la suma de 10 con 5, es decir, devuelve el valor 15.
10 - 5	Calcula la resta de 10 con 5, es decir, devuelve el valor 5.
13 / 2	Calcula la división de 13 con 2, es decir, devuelve el valor 6.5.
13 * 2	Calcula la multiplicación de 13 con 2, es decir, devuelve el valor 26.
17 % 5	Calcula el residuo de dividir 17 y 5, es decir, devuelve el valor 2.
"Hola" + "Buen día"	Une los textos <i>Hola</i> y <i>Buen día</i> para formar uno solo: <i>Hola Buen día</i> .

NOTA: Para calcular raíces recuerda que $\sqrt[x]{y}$ también puede escribirse como $x^{1/y}$

Ejemplo. Desarrolla el código para simular una calculadora de operaciones básicas entre dos números enteros. Recuerda utilizar la metodología para resolverlo.

Solución:

1. Análisis y definición del problema:

Realizar la suma, resta, división y multiplicación de dos números dados.

2. Definición de los datos de entrada / salida:

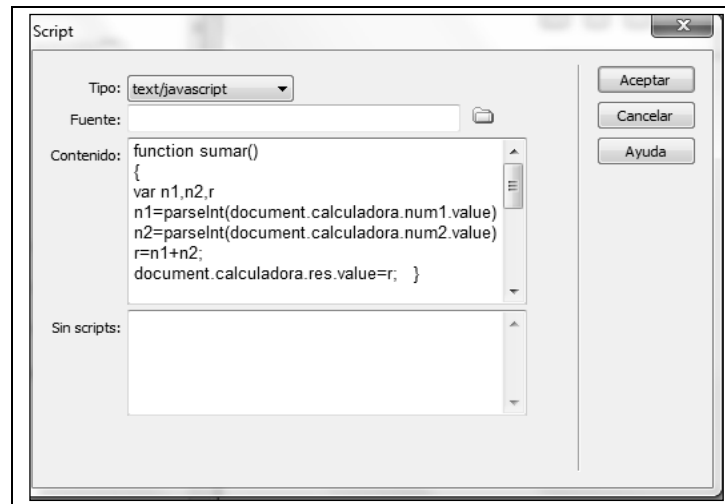
Entrada: Número 1 (n1), Número 2 (n2).

Salida: Resultado operación correspondiente (r).

3. Proceso de solución:

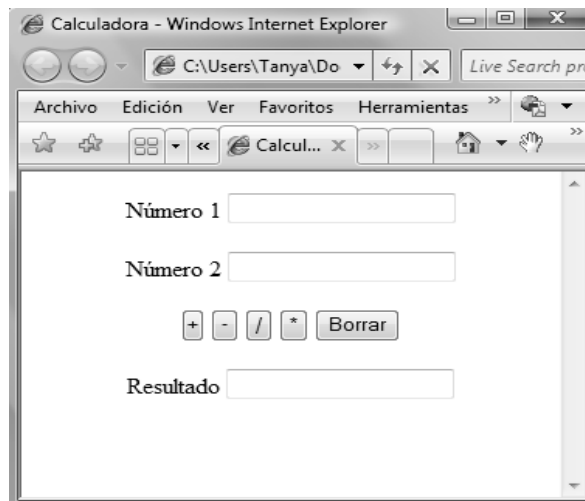
Algoritmo	Diagrama de Flujo	Interfaz Gráfica y Código
<ol style="list-style-type: none"> 1. Inicio. 2. Leer n1. 3. Leer n2. 4. $r = n1 + n2$. 5. $r = n1 - n2$. 6. $r = n1 * n2$. 7. $r = n1 / n2$. 8. Imprimir r. 9. Fin. 	<pre> graph TD INICIO[INICIO] --> Input[/n1, n2/] Input --> Suma[r = n1 + n2] Input --> Resta[r = n1 - n2] Input --> Multiplicacion[r = n1 * n2] Input --> Division[r = n1 / n2] Suma --> Output[/r/] Resta --> Output Multiplicacion --> Output Division --> Output Output --> FIN[FIN] </pre>	<pre> <body> <form id="calculadora" name="calculadora" method="post" action=""> <p align="center">Número 1: <input type="text" name="num1" id="num1" /> </p> <p align="center">Número 2: <input type="text" name="num2" id="num2" /> </p> <p align="center"> <input type="button" name="suma" id="suma" value="+" onclick="sumar()" /> <input type="button" name="resta" id="resta" value="-" onclick="restar()" /> <input type="button" name="multi" id="multi" value="*" onclick="multiplicar()" /> <input type="button" name="dividicion" id="dividicion" value="/" onclick="dividir()" /> </p> <p align="center">Resultado: <input type="text" name="res" id="res" /> </p> </form> </body> </html> </pre>

5. Código JavaScript: Código Dreamweaver Script (Forma 2)



6. Comprobación:

Página HTML como se ve en el explorador:



Las instrucciones,

```
n1=document.calculadora.num1.value;
n2=document.calculadora.num2.value;
```

permiten tomar el contenido de la cajas *num1* y *num2* con la propiedad *value* y pasarlo a las variables *n1* y *n2* respectivamente. Observa que “*calculadora*” es el nombre de nuestro formulario. La notación *document.* es una instrucción que hace referencia a todo lo que se encuentra dentro del formulario y este a su vez, dentro del documento web.

De la misma manera la instrucción,

```
document.calculadora.res.value=r;
```

permite que el contenido de la variable *r* se envíe a la caja *res* utilizando la propiedad *Value*.

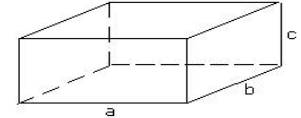
Por último, recuerda que la función **ParseInt** es una función que permite convertir un valor alfanumérico (texto) a otro de tipo numérico, en este caso, tipo Integer.

EJERCICIOS

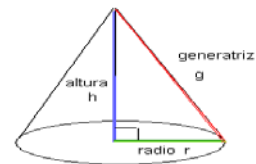
Para cada uno de los siguientes ejercicios realiza el análisis de la solución, algoritmo, diagrama de flujo, la interfaz gráfica del usuario y el código para crear la página dinámica.

1. Elabora una página que permita leer una cantidad en metros. Posteriormente que muestre su equivalente en milímetros, centímetros, kilómetros y pulgadas.

2. Calcular el área y el volumen de un paralelepípedo recto.

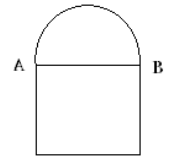


3. Calcular el volumen de un cono.

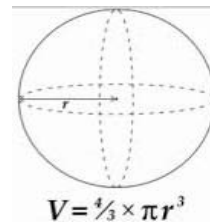


4. Elabora una página que permita el ingreso de los valores de la masa (en Kg) y la aceleración (en m/s^2) y devuelva como resultados el valor de la fuerza (en N) de una partícula.
5. Se conoce el tiempo (en segundos) en que una partícula cae libremente. Elabora una página que permita conocer la altura de la que cae y la velocidad con que choca en el suelo.

6. Si se sabe el valor del segmento AB, calcular el área de la siguiente figura:



7. Calcular el volumen de una esfera.



8. Calcular el número de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, si la fórmula es: $\text{num.pulsaciones} = (220 - \text{edad}) / 10$.
9. Elabora un documento que permita calcular el valor de la variable Y para cualquier valor de X si $Y = 5X^2 - 3X + 6$.

10. Tres personas deciden invertir su dinero para fundar una empresa. Cada una de ellas invierte una cantidad. Obtener el porcentaje que cada quien invierte con respecto a la cantidad invertida.
11. Se tiene la ecuación $Y=2X-6$. Realizar un programa que calcule el valor de Y para 5 valores de X y los muestre en una tabla, se desea que el usuario proporcione los valores de X durante la ejecución del programa.
12. Un vendedor compra manzanas a un determinado precio. Obtener el precio en que debe vender cada una para obtener una ganancia del 27 %.
13. Una persona tiene a monedas de 20 pesos, b monedas de 10 pesos, c monedas de 5 pesos, d monedas de 2 pesos, e monedas de 1 peso y f monedas de 50 centavos. Diseña un programa que le permita a esta persona conocer la cantidad exacta en pesos que tiene.
14. Realizar un programa en el cada una determinada cantidad de días diga cuantos años, meses y días existen. Considere que son 365 días para un año y 30 días para un mes. Ejemplo: 420= 1 año, 1 mes, 25 días.

Capítulo 4

Sentencias de Control

Temas:

- I. Operadores Lógicos.
- II. Sentencia If - Then.
- III. Sentencia If -Then - Else.
- IV. If Anidado.
- V. Sentencia Switch Case.
- VI. Ejercicios.



La programación exige en muchas ocasiones la repetición de acciones sucesivas o la elección de una determinada secuencia y no de otra, dependiendo de las condiciones específicas de la ejecución del programa.

Las sentencias de control se usan para controlar el flujo de ejecución del programa, mediante la ejecución de distintas alternativas en función de una condición o más condiciones.

Definiremos una **condición** como la comparación entre dos datos. Solo puede tomar uno de dos valores **Verdadero** (equivalente al valor **1**) o **Falso** (equivalente al valor **0**). Esto quiere decir que una condición siempre es una cantidad del tipo Boolean. La condición es una expresión que se evalúa con operadores de comparación y operadores lógicos.

Los operadores de comparación son símbolos que permiten comparar dos cantidades y son:

Operador	Nombre	Ejemplo	Resultado verdadero cuando
==	Igual	a == b	a es igual b
!=	Distinto	a != b	a es distinto b
<	Menor que	a < b	a es menor que b
>	Mayor que	a > b	a es mayor que b
<=	Menor o igual	a <= b	a es menor o igual que b
>=	Mayor o igual	a >= b	a es mayor o igual que b

Ejemplos.

Condición	Valor
5 < 10	Trae
-5 > 10	False
3 <= 3	Trae
3 >= 4	False
1 != 1	False
1 == 3	False
A == 4	Depende del valor de la variable A

I. OPERADORES LÓGICOS

Son operadores que permiten realizar algunas operaciones lógicas con las condiciones como son la Negación, Conjunción y Disyunción, entre otras. Los operadores lógicos son **AND**, **OR** y **NOT**:

1. Operador AND (&&)

La expresión que se evalúa es verdadera, siempre y cuando las dos condiciones evaluadas sean verdaderas; de lo contrario es falso.

Por ejemplo:

Condición 1	Condición 2	RESULTADO
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Falso
Falso	Verdadero	Falso
Falso	Falso	Falso

Ejemplo.

(5 < 10) && (10 < 17)

Devuelve el valor True porque (5 < 10) tiene el valor True y (10 < 17) tiene el valor True.

2. Operador OR (||)

La expresión que se evalúa es verdadera cuando al menos una de las condiciones que se evalúa es verdadera.

Por ejemplo:

Condición 1	Condición 2	RESULTADO
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Verdadero
Falso	Verdadero	Verdadero
Falso	Falso	Falso

Ejemplo.

(5 < -10) || (10 < 17)

Devuelve el valor True porque (5 < -10) tiene el valor False y (10 < 17) tiene el valor True.

3. Operador NOT (!)

El resultado de la expresión se niega. Por ejemplo:

Condición	RESULTADO
Verdadero	Falso
Falso	Verdadero

Ejemplo.

Not (5 > 10)

Devuelve el valor True.

ACTIVIDAD:

Si tenemos lo siguiente, el resultado es verdadero cuando:

Operador	Nombre	Ejemplo	Resultado verdadero cuando:
And	Y	(7>2) && (2<4)	_____
Or	O	(7>2) (2<4)	_____
Not	No	!(7>2)	_____

4. Prioridad de los operadores (Jerarquía de operaciones)

Cuando aparecen varias operaciones en una expresión, cada parte se evalúa y se resuelve en un orden determinado llamado "prioridad de los operadores". Puedes utilizar paréntesis para alterar el orden de prioridad y obligar a evaluar algunas partes de una expresión antes que otras. Las operaciones entre paréntesis se realizan siempre de dentro hacia fuera. Sin embargo, entre los paréntesis también se mantiene la prioridad estándar de los operadores.

Cuando las expresiones contienen operadores de más de una categoría, los operadores aritméticos se evalúan primero, los de comparación después y los lógicos al final. Todos los operadores de comparación tienen la misma prioridad, es decir, se evalúan en el orden de izquierda a derecha en el que aparecen. Los operadores aritméticos y lógicos se evalúan en el siguiente orden de prioridad.

Jerarquía de operaciones					
Aritmético		Relacional		Lógico	
Descripción	Símbolo	Descripción	Símbolo	Descripción	Símbolo
Negación unaria	-	Igualdad	==	Negación lógica	!
Multiplicación	*	Desigualdad	!=	Conjunción lógica	&&
División	/	Menor que	<	Disyunción lógica	
Módulo aritmético	%	Mayor que	>		
Suma	+	Menor o igual que	<=		
Resta	-	Mayor o igual que	>=		
Concatenación de cadenas	+				

Así, cuando aparecen la multiplicación y la división juntas en una expresión, se evalúa cada operación según aparece de izquierda a derecha. De la misma forma, cuando aparecen la suma y la resta juntas en una expresión, cada operación se evalúa en orden de aparición de izquierda a derecha.

El operador de concatenación de caracteres (+) no es un operador aritmético, pero por omisión JavaScript lo coloca después de todos los operadores aritméticos y antes que todos los operadores de comparación.

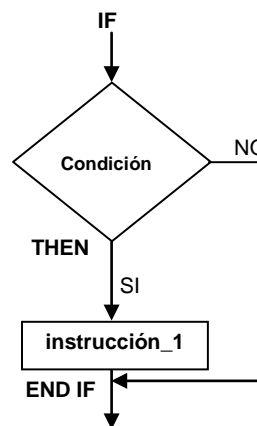
II. SENTENCIA IF THEN

Hasta este momento hemos visto como las instrucciones de un programa se ejecutan en forma continua (una seguida de otra) y de arriba hacia abajo. Algunas veces es necesario alterar esta secuencia normal de ejecución. Las sentencias de control permiten realizar tal actividad.

La sentencia **IF** te permite realizar alternativas dentro del código en función de una condición y de acuerdo al resultado de ésta, ejecuta la instrucción.

La sintaxis más simple de **IF** es:

```
IF (condición_1)
{
    instrucción_1
}
```



La condición se evalúa; si es verdadera, la instrucción_1 se ejecuta.

Ejemplo. Realiza un programa que al solicitar tu calificación te diga si exentas la materia de informática sólo si la calificación es mayor que 8.5. Recuerda utilizar la metodología para encontrar la solución.

Solución:

1. **Análisis y definición del problema:**
Al solicitar una calificación dada, decir si se exenta la materia de informática.
2. **Definición de los datos de entrada / salida:**
Entrada: calificación (cal)
Salida: Mensaje "Felicidades"

3. Algoritmo	4. Diagrama de Flujo	5. Interfaz Gráfica y su Código
<ol style="list-style-type: none"> 1. Inicio. 2. Pedir calificación del alumno. 3. ¿Calificación mayor a 8.5? 4. Si, exentas. 5. Imprime mensaje "Felicidades". 6. Fin. 	<pre> graph TD INICIO([INICIO]) --> C[/C/] C --> IF{IF} IF --> DEC{C > 8.5} DEC -- Si --> THEN[THEN] THEN --> EX[Exentas] EX --> ENDIF[END IF] ENDIF --> FEL[Felicidades] DEC -- No --> FIN([FIN]) </pre>	<pre> <body> <form id="formulario" name="formulario" method="post" action=""> <div align="center"> <p>Calificación</p> <input type="text" name="cal" id="cal"> </p> <p> <input type="button" name="exentar" id="exentar" value="¿Exentar?"> <input type="reset" name="borrar" id="borrar" value="Borrar"> </p> <p>Mensaje</p> <input type="text" name="men" id="men"> </p> </div> </form> </body> </pre>

6. Código JavaScript:

```

function exentar()
{
    var c, m
    c=ParseInt(document.formulario.calificacion.value)
    if (c>8.5 )
    {
        m="Felicidades exentaste";
    }
    document.formulario.men.value=m
}

```

7. Comprobación:

Página HTML como se ve en el explorador y funcionando con datos dados por el usuario. Escribimos un valor en el cuadro de calificación y damos clic en el botón; el resultado obtenido es el siguiente:

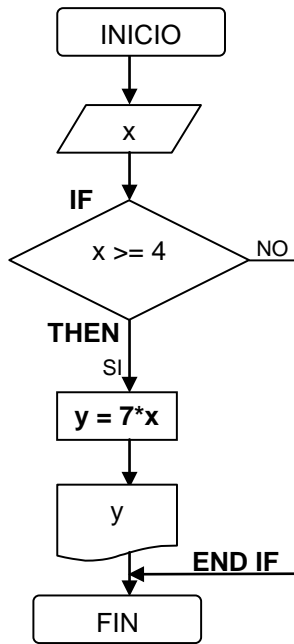


¿Cómo trabaja la Sentencia **IF**?:

- a. Evalúa el valor de la calificación ($\text{calificación} > 8.5$)
- b. Si la calificación es mayor que 8.5; entonces muestra en la segunda caja de texto el mensaje "FELICIDADES".

ACTIVIDAD:

- Desarrolla el código del ejercicio anterior utiliza las instrucciones prompt y alert para solicitar la calificación e imprimir el mensaje respectivamente.
- ¿Cuál es el valor de y? Explica el diagrama de flujo y realiza el código correspondiente para la sentencia IF.



Analiza las líneas de código y explica cuál es el resultado, realiza el diagrama de flujo correspondiente a dicha sentencia.

```

IF (i>7 )
{
    document.writeln("<font size=" + i + ">Hola</font>")
}
    
```

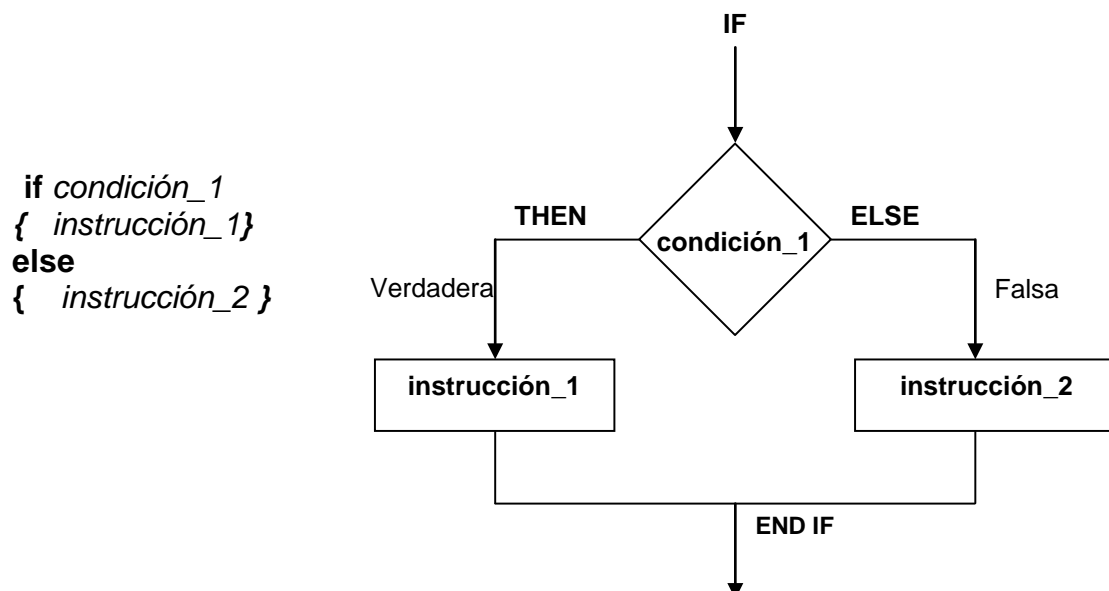
EJERCICIOS PROPUESTOS CON IF THEN

1. Calcular el valor de $c=a/b$ únicamente cuando b sea diferente de cero.
2. Desarrolla un programa que Calcule el valor de Y para toda X mayor a 1.
3. Escribe un programa que solicite un año e indique si es bisiesto. Un año bisiesto es múltiplo de 4 o de 400 pero no de 100.
4. Elabora un programa que permita saber si una persona puede inscribirse a la Universidad, la edad mínima para ingresar a la Universidad es 17 años. El dato que se le pide es el año en el que nació.
5. Realiza un programa que calcule el valor de $C = a/b$ únicamente cuando b sea diferente de cero.

III. SENTENCIA IF THEN ELSE

Si quieres que el programa, llegado a un cierto punto, tome un camino determinado en determinados casos y otro diferente si las condiciones de ejecución difieren, nos servimos del conjunto de instrucciones **IF**, **THEN** y **ELSE**.

La estructura base de este tipo de instrucciones es la siguiente:



La condición se evalúa: si es verdadera, la Instrucción_1 se ejecuta; de lo contrario, si la condición es falsa la instrucción que se ejecuta es la instrucción_2.

Ejemplo. Actualiza el ejemplo anterior utilizando la sentencia IF – ELSE. Ahora, cuando tu calificación sea menor de 8.5 debe aparecer un nuevo mensaje indicando que no puedes exentar la materia.

Solución:

1. **Análisis y definición del problema:**

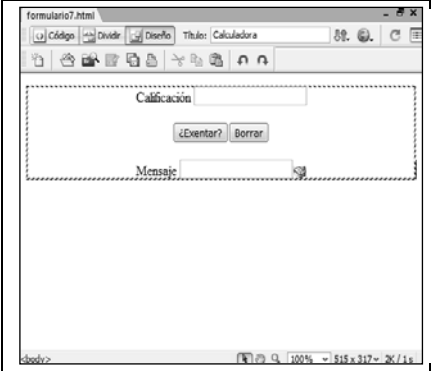
Al solicitar una calificación dada decir si se exenta o no la materia de informática.

2. **Definición de los datos de entrada/salida:**

Entrada: calificación (cal)

Salida: Mensaje

3. Algoritmo	4. Diagrama de Flujo
<p>1. Inicio. 2. Pedir calificación del alumno. 3. ¿Calificación mayor a 8.5? Si, "Felicidades Exentas". No, "Presentas Examen Final". 4. Imprime Mensaje. Fin.</p>	<pre> graph TD INICIO([INICIO]) --> C[/C/] C --> IF{IF} IF --> D{c > 8.5} D -- THEN --> T["Felicidades Exentas"] D -- ELSE --> E["Presentas Examen Final"] T --> ENDIF{END IF} E --> ENDIF ENDIF --> IM[Imprimir Mensaje] IM --> FIN([FIN]) </pre>

5. Interfaz Gráfica y su Código	6. Código en JavaScript
	<pre>function exentar() { var c, m c=parseInt(document.formulario.cal.val ue) if (c>8.5) { m="Felicidades exentaste"; } else { m="Presentas examen"; } document.formulario.men.value=m }</pre>

7. **Comprobación:**

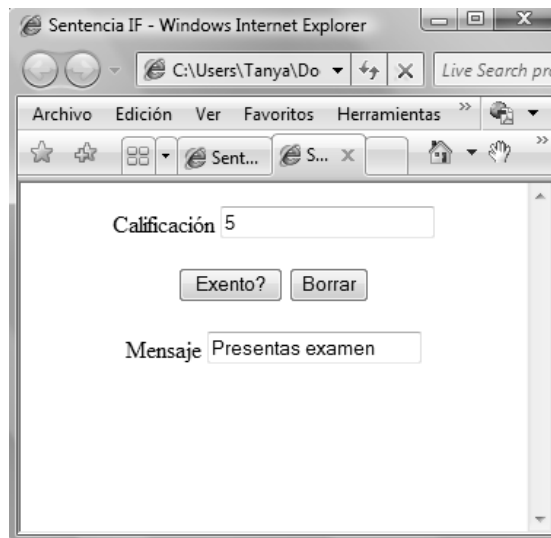
Página HTML como se ve en el explorador y funcionando con datos dados por el usuario.

Después de completar el código en el bloc de notas, probamos nuestro programa. Escribimos un valor en el cuadro de calificación y damos clic en el botón; el resultado obtenido es el siguiente:



- a. Evalúa el valor dado a la calificación (calificación>8.5).
- b. **Si** la calificación es mayor que 8.5; entonces muestra en la segunda caja el mensaje correspondiente, porque se cumple la condición.

- c. **Otro** la calificación es menor que 8.5; en la segunda caja se imprime el mensaje correspondiente, cuando no se cumple la condición.



ACTIVIDAD:

- a. Desarrolla el código del ejercicio anterior pero ahora utiliza las instrucciones `prompt` y `alert` para solicitar la calificación e imprimir el mensaje respectivamente.
- b. Analiza las líneas de código, y realiza el diagrama de flujo correspondiente.

```
a = 8
b = 3
IF (a < b)
{ alert ("a es menor que b"); }
ELSE
{ alert ("a no es menor que b"); }
```

EJERCICIOS PROPUESTOS CON IF THEN ELSE

1. Se conoce los valores de los catetos de un rectángulo, realiza un programa que permita calcular el área o el perímetro según lo solicite el usuario.
2. A un trabajador le pagan por las horas trabajadas y la tarifa tiene un valor por hora. Si la cantidad de horas trabajadas es mayor a 40 horas, la tarifa por hora se incrementa en un 30 %. Calcular el salario del trabajador dadas las horas trabajadas y la tarifa por hora.
2. Solicitar tres números e indicar si el tercero es igual o no a la suma de los dos primeros.
3. Una empresa de bienes raíces ofrece casas de interés social, bajo las siguientes condiciones. Si los ingresos del comprador son de 8000 o menos el enganche será del 15% del costo de la casa y el resto se distribuirá en pagos mensuales, a pagar en diez años. Si los ingresos del comprador son mayores de 8000 el enganche será del 30% del costo de la casa y el resto se distribuirá en pagos mensuales a pagar en 7 años. La empresa quiere obtener cuanto debe pagar un comprador por concepto de enganche y cuanto por cada pago parcial.
4. Una empresa quiere hacer una compra, dependiendo del monto total de la compra, decidirá qué hacer para pagar al fabricante.
 - a. Si el monto total de la compra excede de \$50000 la empresa tendrá la capacidad de invertir de su propio dinero un 55% del monto de la compra, pedir prestado al banco un 30% y el resto lo pagará solicitando un crédito al fabricante.
 - b. Si el monto total de la compra no excede de \$ 50000 la empresa tendrá la capacidad de invertir de su propio dinero un 70% y el restante 30% lo pagara solicitando un crédito al fabricante.

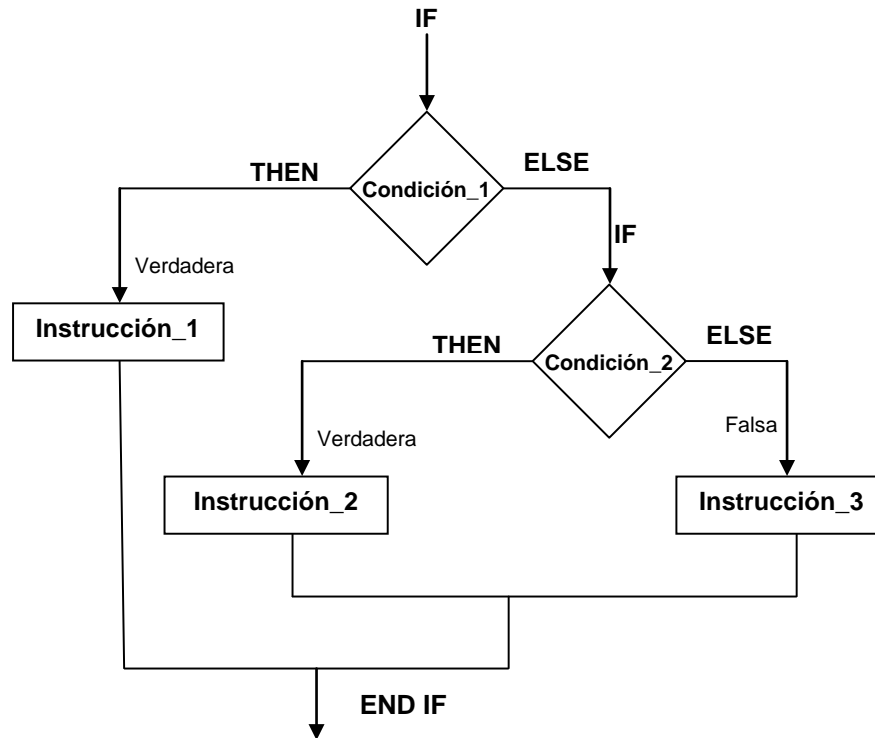
El fabricante cobra por concepto de interés un 20% sobre la cantidad que se le pague a crédito. Realizar un programa que muestre la cantidad invertida, el préstamo solicitado al banco y el interés que pagara por el crédito solicitado a la fábrica.
5. Una agencia automotriz está ofreciendo promociones en la compra de autos, la cual consiste en lo siguiente: si el auto vale 150,000 o menos se pide que se dé un enganche del 12% y el resto a pagar en mensualidades a 3 años; si el auto vale más de 150,000 se deberá dar un enganche del 22% y el resto a pagar en mensualidades a 4 años. Elaborar un programa que muestre cuanto se debe dar de enganche y cuanto mensualmente.

IV. IF ANIDADO

La sentencia **IF**, puede complicarse un poco más si tomas en cuenta que no necesariamente todo es blanco o negro y que muchas posibilidades pueden darse. Es por ello que puedes plantear otras condiciones dentro de la condición principal. Esto es, condiciones anidadas que tienen una estructura del siguiente tipo:

```

IF (condición_1)
    { Instrucción_1; }
ELSE
    IF (condición_2)
        { Instrucción_2; }
    ELSE
        { Instrucción_3; }
    
```



De este modo puedes introducir tantas condiciones como quieras dentro de una condición principal. En este tipo de estructuras es importante cerrar correctamente cada uno de los **IF** con sus { } (llaves) correspondientes.

ACTIVIDAD:

1. Analiza las líneas de código, ¿Cuál es el valor de la variable tasa si estado es igual a "Ingresos"? Explica la secuencia de los datos y realiza el diagrama correspondiente.

```

IF (estado = "Balance")
    tasa = 0.20 * pago;
ELSE
    IF (estado = "Cuenta")
        tasa = 0.14 * pago;
    ELSE
        tasa=1000;
    
```

ACTIVIDAD:

Para cada uno de los siguientes ejercicios realiza el análisis de la solución y el código para crear la página dinámica.

1. Desarrollo un programa que acepte tres números por teclado, los ordene de menor a mayor, y los muestre en una tabla
2. Desarrolla un programa que solicite un número y como resultado regrese como resultado un mensaje que indique si el número es múltiplo de 5 o no.
3. Desarrolla un programa que de una serie de 10 números enteros positivos, separe en pares e impares y posteriormente imprima el resultado en una tabla.
4. Escriba un programa que pida el número de mes (del 1 al 12) e imprima el número de días que tiene el mes, donde:

El mes 2 tiene 28 días
 Los meses 4, 6, 9 y 11 tienen 30 días
 Los meses 1, 3, 5, 7, 8,10 y 12 tienen 31 días

Si da un mes diferente a los anteriores deberá imprimir el mensaje "MES ERRONEO".

EJERCICIOS PROPUESTOS CON IF ANIDADADO

1. Realizar un programa que lea dos números; si son iguales que los multiplique, si el primero es mayor que el segundo que los reste y si no que los sume.

2. Considera el siguiente juego, entre los jugadores A y B, done el jugador A es el adivino y el jugador B es el pensador. B piensa un número comprendido entre 1 y 10, y A trata de adivinarlo usando la técnica de tanteos sucesivos. Por cada tanteo de A, B da una respuesta orientando al jugador A, por ejemplo:
 - FALLASTE. El número pensado es mayor que el tuyo.
 - FALLASTE. El número pensado es menor que el tuyo.
 - ¡¡¡ ACERTASTE !!!

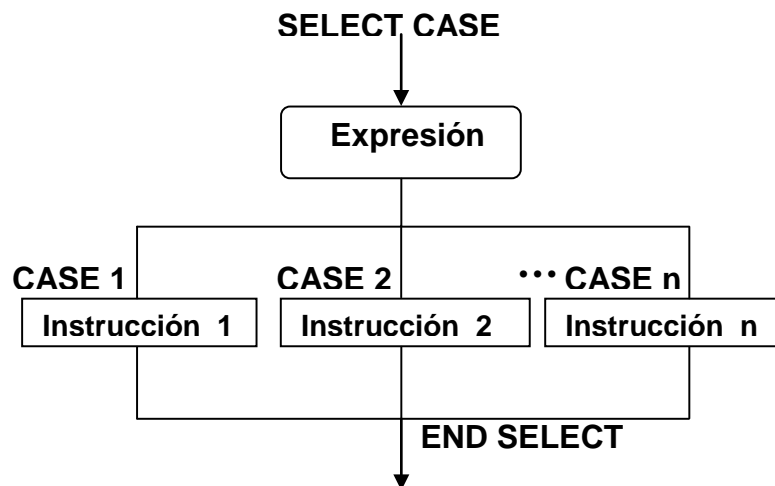
“A” solamente tiene tres oportunidades para acertar.
3. Construye un programa que calcule el índice de masa corporal de una persona (IMC), se sabe que $IMC = \text{peso} / \text{altura}^2$; peso en Kg. y altura en m. Se desea que el programa indique el estado en el que se encuentra la persona en función del valor obtenido de IMC, de acuerdo a las siguientes opciones:
 - a. Si IMC es menor a 20, “La persona tiene un peso deficiente”.
 - b. Si IMC se encuentra en un rango de 20 a 25, “La persona está en peso ideal”.
 - c. Si IMC de la persona es mayor a 25 pero menor a 30, “La persona tiene sobre peso”.
 - d. Si el IMC de la persona es mayor a 30, “La persona es obesa”.
4. Elaborar un programa que calcule el sueldo que le corresponde al trabajador de una empresa que cobra 8000 pesos mensuales, el programa debe realizar los cálculos en función de los siguientes criterios:
 - a. Si lleva más de 10 años en la empresa se le aplica un aumento del 10%.
 - b. Si lleva menos de 10 años pero más que 5 se le aplica un aumento del 7%.
 - c. Si lleva menos de 5 años pero más que 3 se le aplica un aumento del 5%.
 - d. Si lleva menos de 3 años se le aplica un aumento del 3%.
5. Elaborar un programa que calcule el gasto de agua en una vivienda dado los metros cúbicos gastados, siendo el sistema de cobro:
 - De 1 a 50 mc \$ 7.9
 - De 51 a 200 mc \$ 8.5
 - Más de 200 mc \$ 11.3

V. SENTENCIA SWITCH CASE

La Sentencia **SWITCH CASE** es una decisión múltiple. Esta estructura funciona con una expresión de prueba simple que se evalúa una vez en la parte superior de la estructura. El resultado de la expresión se compara entonces con los valores para cada instrucción **CASE** en la estructura. Si existe una coincidencia, se ejecuta el bloque de instrucciones asociadas con dicho **CASE**.

```

SWITCH (expresión)
{
  CASE 1:
    Instrucción_1 o Bloque_de_Instrucciones_1
  BREAK;
  CASE 2:
    Instrucción_2 o Bloque_de_Instrucciones_2
  BREAK;
  ...
  CASE n:
    Instrucción_n o Bloque_de_Instrucciones_n
  DEFAULT:
    Instrucción que no aplica a ninguna de las anteriores
  BREAK;
}
    
```



Ejemplo. Realiza un programa que al indicarte el número del día de la semana te regrese el nombre del día de la semana seleccionado.

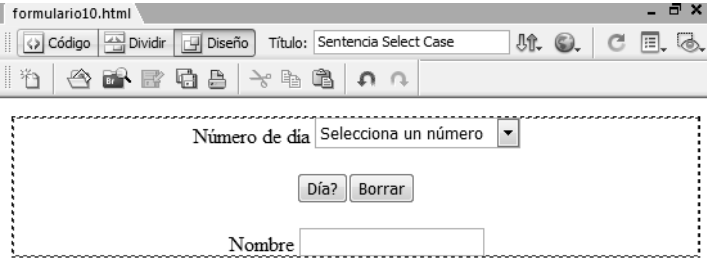
Solución:

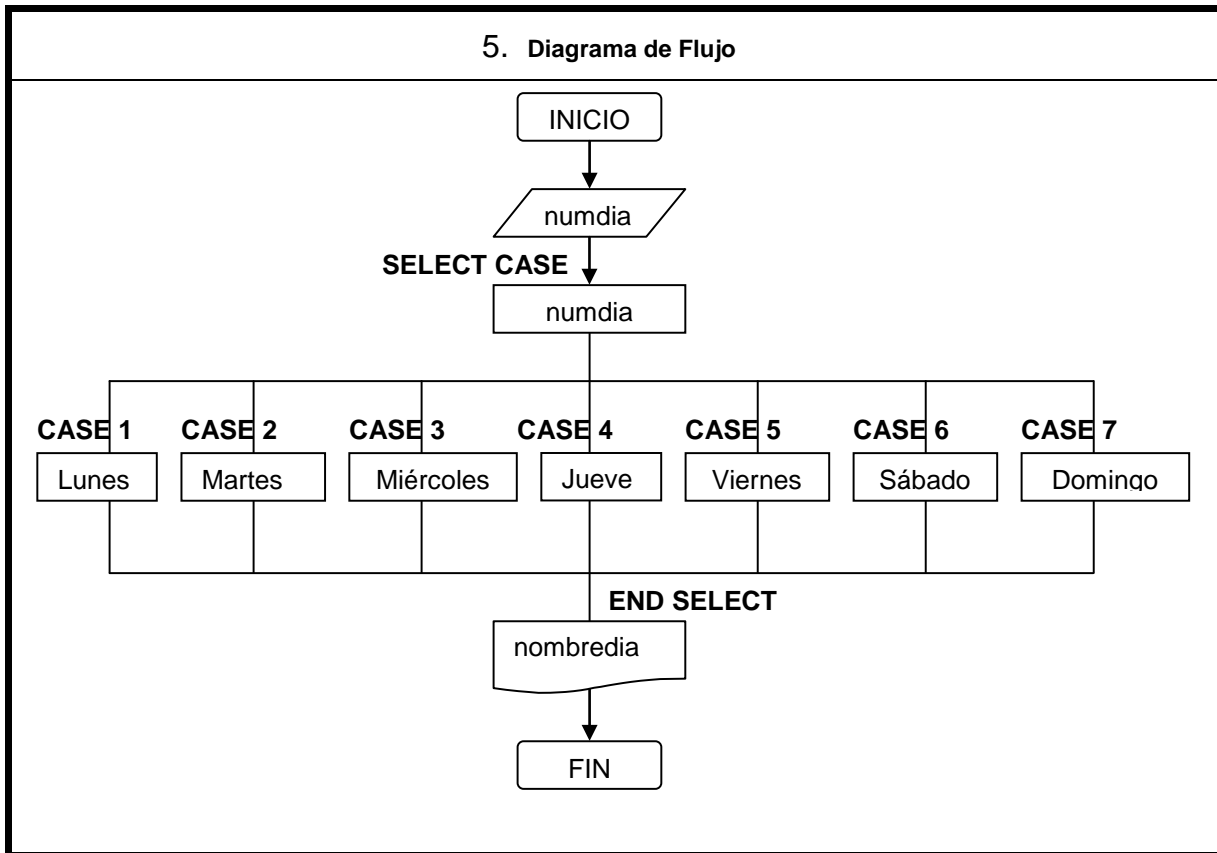
1. **Análisis y definición del problema:**
Al solicitar un número regresa de qué día se trata.

2. **Definición de los datos de entrada / salida:**

Entrada: número (numdia)

Salida: nombre día (nombredia)

3. Algoritmo	4. Interfaz Gráfica y su Código
<ol style="list-style-type: none"> 1. Inicio. 2. Pedir número de día de la semana. 3. Comparar el número de día con cada caso hasta coincidir. 4. Imprimir nombre del día. 5. Fin. 	 <pre data-bbox="597 1136 1339 1675"> <body> <form id="semana" name="semana" method="post" action=""> <p>&nbsp;&nbsp;&nbsp;</p> <p align="center">numero de semana <label for="numdia3"></label> <label for="numdia4"></label> <select name="numdia" id="numdia4"> <option>1</option> <option>2</option> <option>3</option> <option>4</option> <option>5</option> <option>6</option> <option>7</option> </select> </p> <p align="center"> <input type="button" name="mostrar" id="mostrar" value="Día?" onclick="dia()" /> <input type="reset" name="borrar" id="borrar" value="Borrar" /> </p> <p align="center"> <label for="nomdia"></label> el dia de la semana es: <input type="text" name="nomdia" id="nomdia" /> </p> <p>&nbsp;&nbsp;&nbsp;</p> </form> </pre>



6. Código JavaScript:

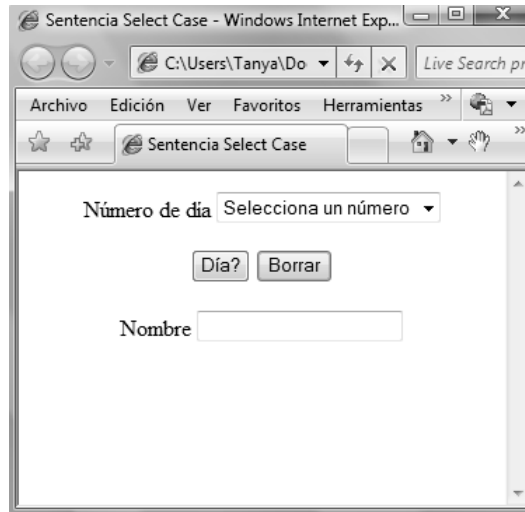
```

<script type="text/javascript">
function dia()
{
var nd;
nd=parseInt(document.semana.numdia.value);

switch (nd)
{
case 1: document.semana.nomdia.value="Lunes";
break;
case 2: document.semana.nomdia.value="Martes";
break;
case 3: document.semana.nomdia.value="Miercoles";
break;
case 4: document.semana.nomdia.value="Jueves";
break;
case 5: document.semana.nomdia.value="Viernes";
break;
case 6: document.semana.nomdia.value="Sábado";
break;
case 7: document.semana.nomdia.value="Domingo";
break;
}
}
</script>
    
```

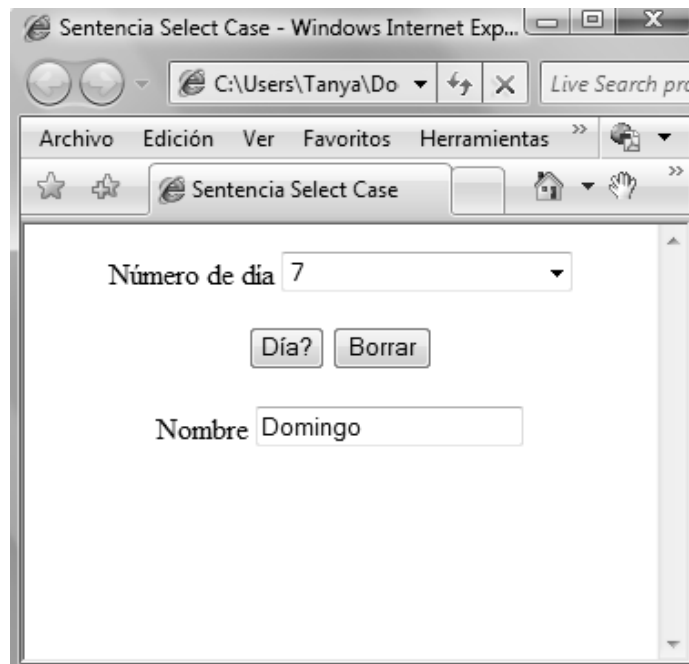
7. Comprobación

La página en el explorador es:



¿Cómo trabaja la Sentencia **SWITCH**?

- Después de elegir un número de la lista desplegable y al dar clic en el botón “¿Qué día es?”
-
- La sentencia hace la comparación con cada uno de los casos desplega en la caja de texto el nombre del día correspondiente al número seleccionado.



ACTIVIDAD:

- a. Tienes el siguiente código que utiliza la sentencia **IF**, ahora cambia las instrucciones y utiliza la sentencia **SWITCH**.

```
IF (a == 0 )  
    alert("a es igual a 0")
```

ELSE

```
IF (a == 1 )  
    alert("a es igual a 1")
```

ELSE

```
IF (a == 2)  
    alert("a es igual a 2")
```

- b. De la actividad anterior realiza el diagrama de flujo para la sentencia **SELECT CASE**.

EJERCICIOS PROPUESTOS CON SWITCH CASE

1. Ingresar 2 números y luego escoger la operación que se quiere hacer con ellos y reportar el resultado. 1) Suma, 2) Resta, 3) Multiplicación 4) División.
2. Ingresar una letra entre "A" y "E" y reportar el mensaje de acuerdo a:

A) excelente.	D) malo
B) bueno.	E) pésimo
C) regular.	
3. Realizar un programa en el que dados como datos el modelo del vehículo y su precio, determine el valor final que debe pagar un comprador. El concesionario está manejando descuentos según el modelo: Charger 8%, Corvett 5%, Camaro 6%, Caliber 9%
4. En un spa se realiza un análisis de clientes registrados en los últimos 5 años con el objeto de conocer los gastos de cada uno de ellos. Crear un programa que calcule el costo dependiendo el número de días que recibió dicho tratamiento; cada tratamiento tiene el siguiente costo: 1) \$2800, 2) \$1950, 3) \$2500, 4) \$1150
5. El costo de llamadas internacionales, depende de la zona geográfica en que se encuentre el país destino y el número de minutos hablados. Realice un programa que calcule el costo total de una llamada, considerando que los costos de acuerdo a la zona son: América \$2.20, Europa: \$3.50, Asia \$6, África \$6, Oceanía \$5.

Capítulo 5

Antecedentes de la Informática

Temas:

- I. Los Pioneros.
- II. Primera Generación.
- III. Segunda Generación.
- IV. Tercera Generación.
- V. Cuarta Generación.
- VI. Quinta Generación.
- VII. Cuestionario.



ANTECEDENTES DE LA INFORMÁTICA

Desde la aparición del hombre en la tierra ha existido la necesidad de contabilizar y efectuar cálculos; contaban sus pertenencias utilizando los medios que tenían a su alcance, como los dedos de sus manos, de los pies, piedritas o palos. La mano podría ser el primer medio utilizado para contar.

Posteriormente y con la necesidad de contabilizar y procesar mayores cantidades de objetos el hombre empezó a crear diferentes herramientas de cálculo.

I. LOS PIONEROS (Dispositivos mecánicos)

El dispositivo de cálculo más antiguo que se conoce es el *ábaco*, su creación fue realizada hace más de 5000 años. Su nombre viene del griego *abax* que significa superficie plana.

Con el nombre de *ábaco* se han reconocido tres instrumentos de cálculo:

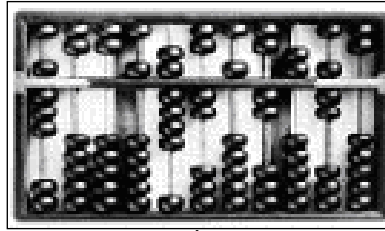
- 1) El más antiguo, era un tablero espolvoreado con una capa de arena oscura, donde se podían trazar cifras y figuras geométricas.
- 2) Un segundo tipo de *ábaco*, conocido ya desde el siglo IV A. C., era el tablero de recuento. Se trataba de un auténtico utensilio de cálculo, una computadora digital tan genuina como la regla de cálculo lo es en lo analógico. El tablero estaba grabado con líneas paralelas que representaban los lugares de valor relativo de un sistema de numeración, comúnmente, de base diez. Estas líneas podían estar trazadas sobre pergamino, esculpidas en mármol, vaciadas en madera e incluso bordadas en paño. Desplazando cuentas sueltas sobre las líneas, hacia adelante y atrás, podían ejecutarse cálculos sencillos.

Los griegos llamaban *abakion* a este tipo de instrumento, y los romanos, *abacus*. Las cuentas utilizadas eran “piedritas” redondeadas que se iban moviendo por los surcos, de ahí la palabra latina *calculus*, que significa: “piedrita”. Es por ello madre de nuestro «cálculo» y «calcular».



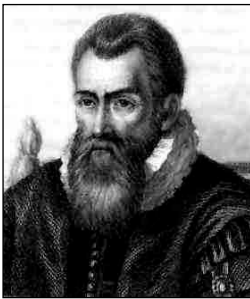
Al lado derecho de la imagen se muestra un *avacus*

- 3) El *ábaco* tal como lo conocemos actualmente está constituido por una serie de hilos con cuentas ensartadas en ellos; la efectividad de este dispositivo ha soportado la prueba del tiempo.



Ábaco

Hoy en día, el *ábaco* se emplea como método de enseñanza en las escuelas de los países orientales, aunque es usado con regularidad en muchos lugares del mundo, particularmente en los pequeños negocios de los barrios chinos (Chinatowns), en los Estados Unidos de América, en Canadá y en los países cosmopolitas.



John Napier (1550 -1617)

Durante el siglo XVI John Napier, matemático escocés, se basó en su teoría de que todos los números se podían expresar exponencialmente y en 1614 dio a conocer su invención de los logaritmos, funciones matemáticas que permiten convertir las multiplicaciones en sumas y las divisiones en restas.

Napier publicó una obra titulada: "RABDOLOGIAE", pequeño tratado sobre la forma de ejecutar multiplicaciones. En su apéndice explicaba un método para multiplicar y dividir usando varillas y placas metálicas, que puesto en la práctica se convirtió en el precursor de las modernas calculadoras de bolsillo. Este invento es conocido como los "*huesos de Napier*".

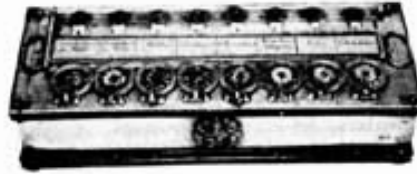
Posteriormente Blaise Pascal, a la edad de 19 años, motivado por simplificar el trabajo de su padre (funcionario de las finanzas nacionales), empezó a construir una máquina para sumar y restar; esta complicada máquina fue concluida 3 años más tarde.



Blaise Pascal (1623 – 1662)

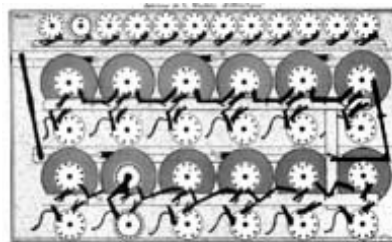
Así, la "Pascalina", fue la primera calculadora mecánica, tenía el tamaño de un cartón de tabaco y su mecanismo consistía en una serie de discos asociados a ruedas, tales que cada una de las ruedas hacía avanzar un paso a la siguiente al completar una vuelta. Las ruedas estaban marcadas con números del 0 al 9 y había dos para

los decimales y 6 para los enteros, con lo que podía manejar números de hasta 8 dígitos entre 000.000 01 y 999.999 99.



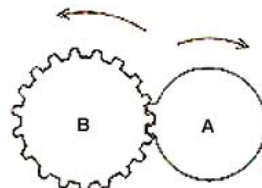
Pascalina

Las ruedas giraban mediante una manivela, por lo tanto, para sumar o restar, había que girar la manivela correspondiente, en un sentido o en otro, el número de pasos adecuado.



Estructura interna de la Pascalina

Había unos indicadores sobre los discos, que eran los encargados de dar la respuesta buscada.



Mecanismo utilizado en la Pascalina

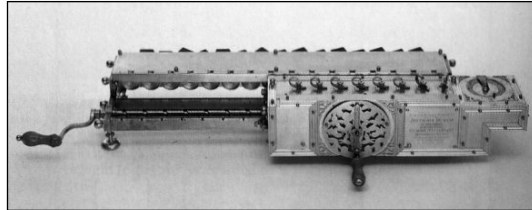
El principio básico de esta máquina calculadora se usa todavía en algunos pluviómetros (medidores de precipitación líquida) y cuentakilómetros (mide los kilómetros totales y parciales).



Gottfried Wilhelm von Leibniz
(1646 -1716)

Gottfried Wilhelm Von Leibniz, genio de su época, fue un verdadero precursor de la lógica matemática. Leibnitz fue el primero que propuso el uso de un sistema binario para realizar los cálculos.

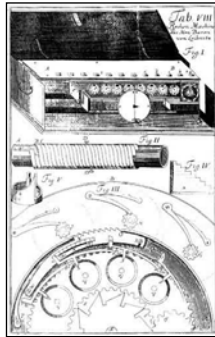
En 1671 desarrolló una máquina multiplicadora, conocida como la *calculadora universal*, mejorando la de Blaise Pascal, ya que realizaba las operaciones de suma, resta, multiplicación, división y extracción de raíces cuadradas, caracterizándose por hacer la multiplicación de forma directa.



Calculadora universal

El mecanismo de esta máquina usaba una serie de cilindros con dientes graduados.

Mecanismo de la calculadora universal



Leibnitz propuso la idea de una máquina de cálculo en sistema binario, esto es, la base de numeración empleada por las computadoras actuales.

Posterior a los trabajos desarrollados por Pascal y Leibnitz se crearon una multitud de máquinas mecánicas de cálculo, entre las más destacadas se encuentra la de Jacob Leupold (1727) quien propuso algunas mejoras sobre el mecanismo de Leibniz. En 1777, Charles Mahon (1753-1816), construyó una máquina aritmética y otra lógica, esta última llamada *demonstrador* de Stanhope. En 1825, el francés Charles Xavier Thomas de Colmar diseña una máquina calculadora que después consigue comercializar con éxito.

Entre los trabajos más destacados se encuentra el de Joseph Jacquard, que en 1801 dio un aporte fundamental al proceso de las máquinas programables al utilizar un mecanismo de tarjetas perforadas para controlar el dibujo formado por los hilos de las telas confeccionadas por una máquina de tejer.



Joseph Jacquard (1752 - 1834)

Jacquard fue el primero en emplear tarjetas perforadas para almacenar la información sobre el dibujo del tejido y además controlar la máquina. La máquina de tejer de Jacquard es considerada la antesala de la informática.



Maquina de tejer Jacquard

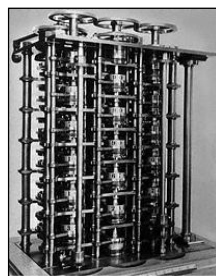
A partir del invento de Jacquard empezaron a proliferar las máquinas y equipos programados por sistemas perforados, tales como los pianos mecánicos, conocidos como pianolas, muñecos y otros novedosos juguetes mecánicos.



Charles Babbage (1791 - 1871)

La historia de las computadoras modernas empieza con Charles Babbage, considerado *El Padre de la Computación Moderna*.

Babbage, matemático e inventor inglés, predijo muchas de las teorías en que se basan las actuales computadoras. En 1822 diseñó la máquina diferencial para el cálculo de polinomios. Esta máquina se utilizó con éxito para el cálculo de tablas de navegación y artillería lo que permitió a Babbage conseguir una subvención del gobierno para el desarrollo de una segunda y mejor versión de la máquina.



Máquina diferencial

Durante 10 años Babbage trabajó en una segunda máquina sin completarla.

En 1833, mientras que la máquina diferencial era un aparato de proceso único, Babbage decidió construir una máquina de propósito general que pudiese resolver casi cualquier problema matemático. Todas estas máquinas eran por supuesto mecánicas, movidas por vapor. De todas formas, la velocidad de cálculo de las

máquinas no era tal como para cambiar la naturaleza del cálculo, además la ingeniería en ese tiempo no estaba lo suficientemente desarrollada como para permitir la fabricación de los delicados y complejos mecanismos requeridos por el ingenio de Babbage. La sofisticada organización de esta segunda máquina diferencial es lo que hace a Babbage el padre de la informática actual.

Como las modernas computadoras, la máquina de Babbage tenía un mecanismo de entrada y salida por tarjetas perforadas, una memoria, una unidad de control y una unidad aritmético-lógica. Preveía tarjetas separadas para programas y datos. Una de sus características más importantes, era que la máquina podía alterar su secuencia de operaciones basándose en el resultado de cálculos anteriores, algo fundamental en las computadoras modernas. Sin embargo, la máquina nunca llegó a construirse. Babbage no pudo conseguir un contrato de investigación y pasó el resto de su vida inventando piezas y diseñando esquemas para conseguir los fondos y construirla, por desgracia murió sin conseguirlo.

Aunque otros pocos hombres trataron de construir autómatas o calculadoras siguiendo los esquemas de Babbage su trabajo quedó olvidado hasta que inventores modernos que desarrollaban sus propios proyectos de computadoras se encontraron de pronto con tan extraordinario precedente.



II. Primera Generación (Electromecánicos y electrónicos de tubos de vacío)

Herman Hollerith fue el iniciador del procesamiento automatizado de información.

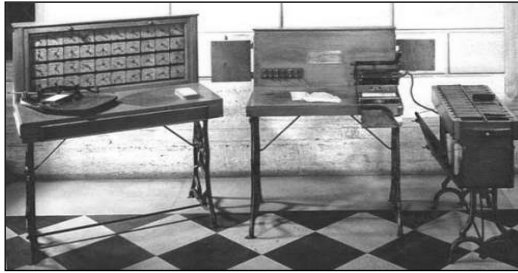
En 1879, Hollerith fue contratado como asistente en las oficinas del censo norteamericano para realizar el recuento de la población para el año de 1880. Éste tardó 7 años y medio en completarse manualmente. Hollerith desarrolló un sistema de cómputo automático para futuras tareas.



Herman Hollerith (1860 - 1929)

El sistema contenía una prensa manual que detectaba los orificios en las tarjetas perforadas, los orificios representaban el sexo, la edad, la raza, etc. Tenía un alambre que pasaba a través de los huecos dentro de una copa de mercurio debajo de la tarjeta, cerrando de este modo el circuito eléctrico. Este proceso disparaba unos contadores mecánicos y ordenaba los recipientes de las tarjetas, tabulando así

en forma apropiada la información. La máquina de Hollerith era eléctrica y procesaba los hoyos en las tarjetas.



Máquina de Hollerith

El censo de 1890 se realizó en dos años y medio gracias a la máquina de Hollerith.

En 1896 fundó su propia compañía: *Tabulating Machine Company*. En 1900 había desarrollado una máquina que podía clasificar 300 tarjetas por minuto, una perforadora de tarjetas y una máquina de cómputo semiautomática. En 1924, Hollerith fusionó su compañía con otras dos para formar la *Internacional Business Machines* hoy mundialmente conocida como IBM.

Hacia 1910 (año de censo), ante la necesidad de agilizar el proceso de datos de las oficinas del censo, el Ingeniero Ruso James Powers desarrolló el sistema de tarjetas perforadas. En 1911 constituyó la *Power's Tabulating Machine Company*, convirtiéndose en el principal competidor de Hollerith.

A inicios del siglo XX se produjeron una serie de desarrollos tecnológicos, trayendo consigo la construcción de nuevas máquinas y la generación de programas, propiciando una rápida evolución en las computadoras. Se construyó la primera computadora analógica mecánica, capaz de resolver ecuaciones diferenciales, este *Analizador Diferencial* fue construido por Vannevar Bush en 1930, usaba engranajes diferenciales que rodaban por motores eléctricos.

En 1933, el primer programa mecánico fue diseñado por Wallace J. Eckert; Alan Turing, en 1936, construyó un modelo formal de computadora, la *máquina de Turing* es el primer modelo teórico de lo que luego sería una computadora programable; en 1939 se desarrolló la primera computadora electrónica digital el *Atanasoff Berry Computer (ABC)* por el Dr. John V. Atanasoff y Clifford Baya, fue la primera máquina en hacer uso de tubos al vacío como los circuitos de la lógica; en 1943 se desarrolló "*El coloso*" la primera calculadora inglesa electrónica para criptoanálisis y muchos otros avances más .

La histórica *Mark I* de la Universidad de Harvard, se empezó a construir en 1939 y se terminó en 1944, se basó conceptualmente en Babbage. Oficialmente, se le dio el nombre de *Automatic Sequence Controlled Calculator (ASCC)*.

La *Mark I* medía aprox. 15.5 metros de largo, 2.40 metros de alto, 60 centímetros de ancho y pesaba 5 toneladas. Su funcionamiento era electromecánico y su interior estaba compuesto por unas 750,000 piezas diferentes, entre relevadores, interruptores binarios, ruedas rotatorias para los registros, interruptores de diez posiciones (para los dígitos), etc.



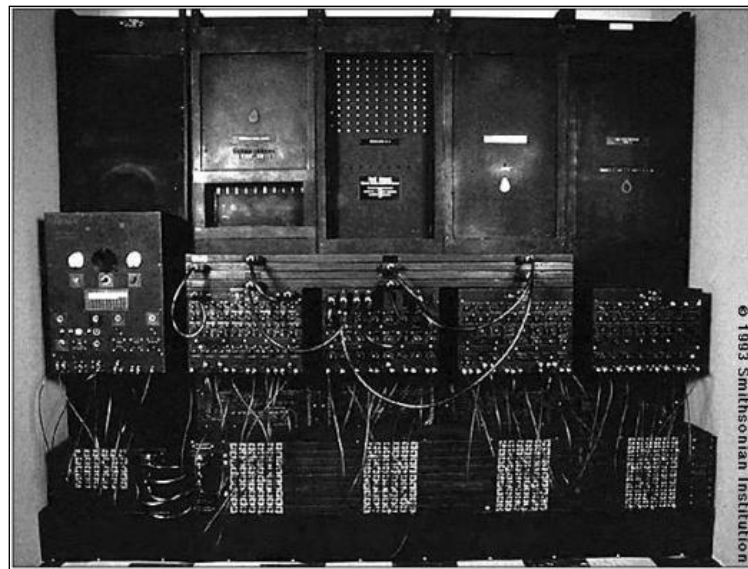
Mark I

La *Mark I* contenía 72 registros mecánicos, cada uno de los cuales podía almacenar 23 dígitos decimales más un dígito para el signo (cero para el más y nueve para el menos). La posición del punto decimal estaba fija durante la solución de un problema, pero podía ajustarse previamente de manera que estuviera entre dos dígitos cualesquiera. La *Mark I* recibía sus secuencias de instrucciones (programas) y sus datos a través de lectoras de cinta de papel perforada y los números se transferían de un registro a otro por medio de señales eléctricas.

La *Mark I* inició su operación en abril de 1944, fue usada para resolver problemas de balística y diseño naval durante el final de la Segunda Guerra Mundial. Al terminar la guerra, la *Mark I* fue utilizada principalmente para calcular tablas de las funciones de Bessel (usadas para resolver cierto tipo de ecuación diferencial).

Simultáneamente a inicios de la Segunda Guerra Mundial, la armada de los Estados Unidos de América, bajo la necesidad de calcular la mesa de tiro para la artillería se unió con la Universidad de Pensylvania para construir un analizador de cálculo diferencial digital. La persona encargada de realizar este proyecto fue John W. Mauchly quien comenzó a analizar la probabilidad de construir una computadora electrónica y para ello se reunió con el Profesor J. V. Atanasoff. Después de varias presentaciones del proyecto y la incorporación de otros asesores profesionales, en 1945 la computadora *ENIAC* (Electronic Numerical Integrator and Computer) se terminó de construir; al año siguiente, fue mostrada oficialmente.

La *ENIAC* estaba compuesta de 17,468 válvulas o tubos de vidrio al vacío (más resistencias, condensadores, etc.) con 32 toneladas de peso, 2,40 m de ancho y 30 metros de largo.

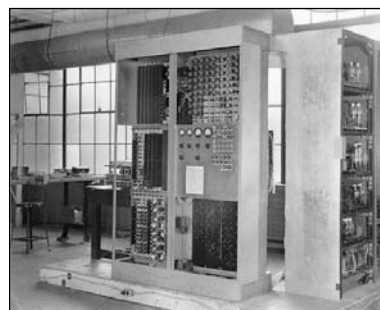


ENIAC

Para desarrollar diferentes operaciones, debían cambiarse las conexiones (como en las viejas centrales telefónicas), el proceso podía tomar varios días. Era capaz de calcular con gran velocidad las trayectorias de proyectiles, principal objetivo inicial de su construcción. La *ENIAC* podía resolver 5,000 sumas y 360 multiplicaciones por segundo, pero su programación era terriblemente tediosa.

La *ENIAC* con sus circuitos electrónicos ensombreció el desarrollo de las máquinas electromecánicas.

Antes de tener concluida la *ENIAC* John Mauchly y Prespert Eckert, empezaron a trabajar en una máquina que pudiera tener el programa almacenado en ella misma. Esto fue posible y en 1949 surgió la *EDVAC* (Electronic Discrete Variable Automatic Computer) con una mayor capacidad de almacenamiento de memoria. La memoria consistía en líneas de mercurio dentro de un tubo de vidrio al vacío, de tal modo que un impulso electrónico podía ir y venir en 2 posiciones, para almacenar los ceros (0) y unos (1). Esto era indispensable ya que en lugar de usar decimales la *EDVAC* empleaba números binarios. La *EDVAC* fue el primer equipo con capacidad de almacenamiento de memoria e hizo desear a los otros equipos que tenían que ser intercambiados o reconfigurados cada vez que se usaban.



EDVAC

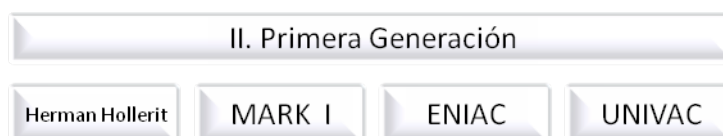
Aprovechando la idea de almacenamiento de programas en la memoria de la computadora surge el desarrollo de la *EDESAC* (Electronic Delay Storage Automatic Calculator). Desarrollada por Maurice Wilkes y sus colegas de Cambridge (Inglaterra), fue completamente operativa por primera vez en Junio de 1949. Considerada la primera computadora electrónica, por su tecnología de tubos de vacío y por su estructura funcional de programación almacenada en memoria, a esto se le conoce como arquitectura Von Neuman, definiendo así, lo que hoy consideramos una computadora.

En 1946, en colaboración con Arthur W. Burks y Herman H. Goldstine, John Louis Von Neuman escribió uno de los artículos de mayor influencia en la historia moderna de las computadoras: "*Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*". Las ideas que contiene este documento, que de forma conjunta se conocen con el nombre de Máquina de Von Neumann o Arquitectura Von Neumann, han proporcionado los fundamentos para la construcción y el desarrollo de todas las computadoras hasta el momento.

El concepto central en la Arquitectura Von Neumann es el de una programación almacenada, según la cual, las instrucciones y los datos tenían que almacenarse juntos en un medio común y uniforme, en vez de separados, como hasta entonces se hacía. De esta forma, no sólo se podían procesar cálculos, sino también las instrucciones y los datos podían leerse y escribirse bajo el control del programa. A partir de esta idea básica, se sigue que un elemento en la memoria tiene una calidad ambigua con respecto a su interpretación, esta ambigüedad se resuelve, sólo temporalmente, cuando se requiere ese elemento y se ejecuta como una instrucción o se opera como un dato. Un beneficio de esta ambigüedad es el hecho de que un dato, obtenido como resultado de algunas operaciones en la unidad aritmético-lógica de la computadora, podía colocarse en la memoria como si fuera cualquier otro dato, para entonces usarlo y ejecutarlo como si fuera una instrucción.

Además la Máquina de Von Neumann presentaba como característica importante un pequeño número de registros para mantener la instrucción del programa en curso y el registro de datos que se estaban procesando. La máquina operaba en un ciclo repetitivo de pasos para localizar y ejecutar en secuencia las instrucciones del programa. Resulta evidente que esta breve descripción puede aplicarse a casi todas las computadoras que desde 1946 se han construido, por lo que la aportación de Von Neumann a las Ciencias de la Computación es más que notable.

Con este concepto en mente se construyeron *EDVAC*, *EDSAC* y *UNIVAC*. Estas máquinas se programaban directamente en lenguaje máquina, pero a partir de mediados de los 50, se produjo un gran avance en la programación avanzada.



III. SEGUNDA GENERACIÓN (los transistores y los avances en programación)

En 1945 la máxima limitación de las computadoras era la lenta velocidad de procesamiento.

En 1947, John Bardeen, Walter Brattain y William Shockley inventan el transistor, se usó como un interruptor electrónico. Las computadoras eran más rápidas, más pequeñas y producían menos calor. Dándose la oportunidad de que se desarrollaran los microprocesadores. Algunas de las máquinas que se construyeron en esta época fueron la *TRADIC*, de los Laboratorios Bell (donde se inventó el transistor), en 1954, la *TX-0* del laboratorio LINCOLN del MIT y las *IBM 704*, *709* y *7094*. También aparece en esta generación el concepto de supercomputadora, específicamente diseñada para el cálculo en aplicaciones científicas y mucho más potentes que los de su misma generación, como el *Livermore Atomic Research Computer* (LARC) y la *IBM 7030*.

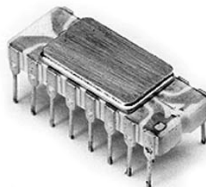
Pero esta generación se distingue también por los avances teóricos. Así, en 1950, Alan Turing publica el artículo *Computing Machinery and Intelligence* en la revista *Mind*. Este artículo estimuló a los pensadores sobre la filosofía e investigación en el campo de la Inteligencia Artificial.

En 1951, Grace Murria Hooper, da la primera noción de compilador y más tarde desarrolló el COBOL. John Backus, en 1957, desarrolló el primer compilador para FORTRAN. En 1958, John MacCarthy propone el LISP, un lenguaje orientado a la realización de aplicaciones en el ámbito de la Inteligencia Artificial. Casi de forma paralela, Alan Perlis, John Backus y Peter Naur desarrollan el lenguaje *ALGOL*.

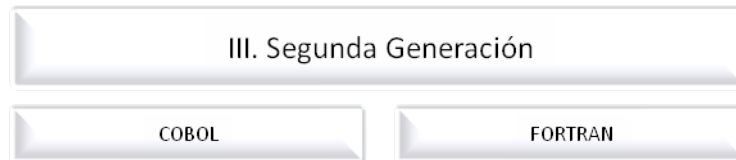
Pero el personaje más importante en el avance del campo de los algoritmos y su análisis, es Edsger Dijkstra, que en 1956, propuso su conocido algoritmo para la determinación de los caminos mínimos en un grafo, y más adelante, el algoritmo del árbol generador *minimal*. Más tarde, en 1961, N. Brujin introduce la *notación O*, que sería sistematizada y generalizada por D. Knuth. En 1957, aparece la Programación Dinámica de la mano de R. Bellman. En 1960, S. Golomb y L. Baumet presentan las Técnicas Backtracking para la exploración de grafos. Se publican en 1962 los primeros algoritmos del tipo Divide y Vencerás: el *QuickSort* de Charles Hoare y el de la multiplicación de grandes enteros de A. Karatsuba e Y. Ofman.

En 1959, Jack Kilby presenta el primer circuito integrado, un conjunto de transistores interconectados con resistencias, en una pequeña pastilla de silicio y metal, llamada chip.

Chip



Fue a partir de este hecho que las computadoras empezaron a fabricarse de menor tamaño, más veloces y a menor costo, debido a que la cantidad de transistores colocados en un solo chip fue aumentando en forma exponencial.



IV. TERCERA GENERACIÓN (El uso de los circuitos integrados)

A partir del circuito integrado, se producen nuevas máquinas, mucho más pequeñas y rápidas que las anteriores, así aparecen las *IBM 360/91*, *IBM 195*, *SOLOMON* (desarrollada por la Westinghouse Corporation) y *la ILLIAC IV*, producida por Burroughs, el Ministerio de Defensa de los EE.UU y la Universidad de Illinois.

Seymour Cray (1925 - 1996) revoluciona el campo de la supercomputación con sus diseños: en 1964, el *CDC 6600*, que era capaz de realizar un millón de operaciones en coma flotante por segundo; en 1969, el *CDC 7600*, el primer procesador vectorial, diez veces más rápido que su predecesor.

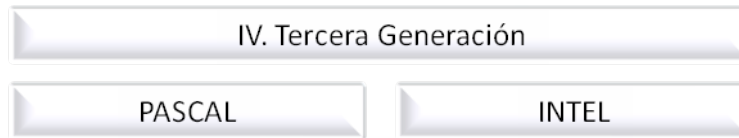
En cuanto a los avances teóricos, a mediados de los 60, un profesor de Ciencias de la Computación, Niklaus Wirth, desarrolla el lenguaje *PASCAL*, y en Berkeley, el profesor Lotfi A. Zadeh, publica su artículo *Fuzzy Sets*, que revoluciona campos como la Inteligencia Artificial, la Teoría de Control o la Arquitectura de Computadoras.

En 1971, Intel introduce el primer microprocesador. El potentísimo 4004 procesaba 4 bits de datos a la vez, tenía su propia unidad lógico-aritmética, su propia unidad de control y 2 chips de memoria. Este conjunto de 2.300 transistores que ejecutaba 60.000 operaciones por segundo se puso a la venta por 200 dólares. Muy pronto Intel comercializó el 8008, capaz de procesar el doble de datos que su antecesor y que inundó los aparatos de aeropuertos, restaurantes, salones recreativos, hospitales, gasolineras, etc.

A partir de aquí nacieron las tecnologías de integración a gran escala (LSI) y de integración a muy gran escala (VLSI), con las que procesadores muy complejos podían colocarse en un pequeño chip.

Sin embargo, hasta este momento, por motivos económicos, complejidad de uso y dificultad de mantenimiento, las computadoras habían sido patrimonio de universidades, organismos militares, gubernamentales y grandes empresas.

En 1975, Popular Electronics dedicó su portada a la primera microcomputadora del mundo capaz de rivalizar con los modelos comerciales, el *Altair 8800*.



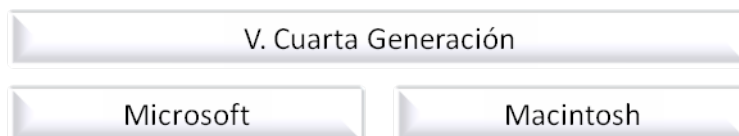
V. CUARTA GENERACIÓN (computadoras personales de uso doméstico)

El *Altair 8800*, producido por una compañía llamada Micro Instrumentation and Telemetry Systems (MITS), se vendía a 397 dólares, lo que indudablemente contribuyó a su popularización. No obstante, el *Altair* requería elevados conocimientos de programación, tenía 256 bytes de memoria y empleaba lenguaje máquina. Dos jóvenes, William Gates y Paul Allen, ofrecieron al dueño de MITS, un software en *BASIC* que podía correr en el *Altair*. El software fue un éxito y posteriormente Allen y Gates crearon Microsoft.

Paralelamente, Steven Wozniak y Steven Jobs, también a raíz de ver el *Altair 8800* en la portada de Popular Electronics, construyen en 1976, la *Apple I*. El 1 de Abril de 1976 nació Apple Computer. En 1977, con el lanzamiento de la *Apple II*, la primera computadora con gráficos a color y carcasa de plástico, la compañía empezó a imponerse en el mercado.

En 1981, IBM estrena una nueva máquina, la *IBM Personal Computer*, protagonista absoluta de una nueva estrategia: entrar en los hogares. El corazón de esta pequeña computadora, con 16 Kb de memoria (ampliable a 256), era un procesador Intel, y su sistema operativo procedía de una empresa recién nacida llamada Microsoft.

En 1984, Apple lanza la *Macintosh*, que disponía de interfaz gráfico para el usuario y un ratón, que se hizo muy popular por su facilidad de uso.



VI. QUINTA GENERACIÓN

Comprende de 1981 - 200?. En 1981, los principales países productores de nuevas tecnologías (fundamentalmente Estados Unidos y Japón) anunciaron una nueva generación.

La quinta generación de computadoras tienen las siguientes características estructurales:

- Estarán hechas con microcircuitos de muy alta integración, que funcionarán con un alto grado de paralelismo y emulando algunas características de las redes neuronales con las que funciona el cerebro humano.
- Computadoras con Inteligencia Artificial.
- Interconexión entre todo tipo de computadoras, dispositivos y redes (redes integradas).
- Integración de datos, imágenes y voz (entorno multimedia).
- Utilización del lenguaje natural (lenguaje de quinta generación).



VII. CUESTIONARIO

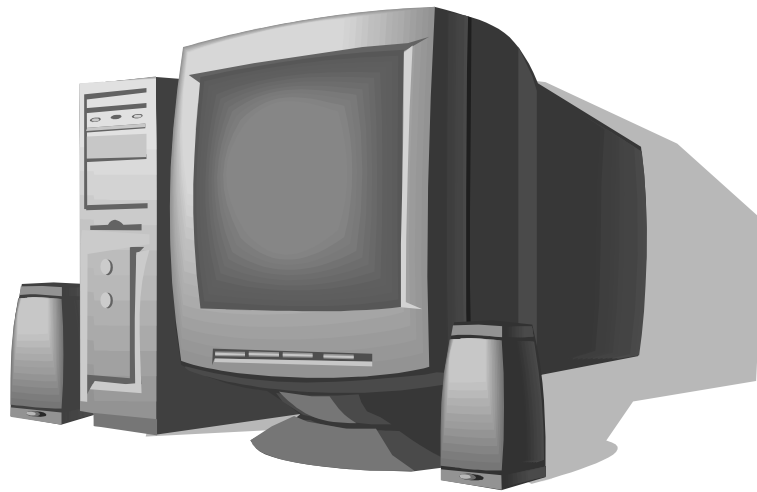
1. Menciona 3 tipos de ábacos.
2. ¿Cuáles fueron los primeros instrumentos que utilizó el hombre para contar?
3. ¿Cuáles fueron las aportaciones de John Napier?
4. ¿Cómo se llamó la primera calculadora mecánica y qué operaciones realizaba?
5. Realiza un dibujo de la primera calculadora mecánica.
6. ¿Cómo mejoró Leibnitz la máquina de Pascal?
7. ¿Cuál fue la aportación de Joseph Jacquard?
8. ¿Por qué se le denomina a Charles Babbage el padre de la informática?
9. En tu cuaderno de informática realiza un cuadro comparativo en donde se muestre la siguiente información: Pionero, invención, teorías y año de invención.
10. Menciona las principales características de las computadoras de la Primera Generación.
11. Menciona 4 máquinas que pertenezcan a la Primera Generación:
12. Describe el principio de Von Neumann.
13. ¿Cuáles son las características de las máquinas de la Segunda Generación?
14. ¿Qué otros sucesos importantes acontecieron durante el desarrollo de las máquinas de la Segunda Generación?
15. ¿Qué características tienen las máquinas de la Tercera Generación?
16. ¿Cómo se llamó el primer microprocesador?
17. ¿Cuáles son las computadoras de la Cuarta Generación?
18. Menciona las características de la primera computadora de la Cuarta Generación
19. ¿Cuáles son las computadoras de la Quinta Generación?
20. Menciona las características de una computadora de la Quinta Generación.
21. En tu cuaderno de informática realiza un cuadro comparativo en donde se muestre la siguiente información: Generación, máquinas y teorías desarrolladas en cada una de ellas.

Capítulo 6

Estructura Física y Lógica

Temas:

- I. Hardware.
- II. Software.
- III. Funciones de una Computadora.
- IV. Cuestionario.



ESTRUCTURA FÍSICA Y LÓGICA

Una computadora es una máquina capaz de efectuar una secuencia de operaciones en un tiempo corto, la secuencia de realización está asignada mediante un programa, de tal manera que se transformen datos de entrada en información (procesada) de salida. A este complejo procedimiento de transformación se le llama **procesamiento**.

Para desarrollar el procesamiento de información, una computadora está integrada básicamente por los siguientes elementos: *Hardware* y *Software*.



I. HARDWARE

Se le denomina *hardware* a todos los componentes eléctricos, electrónicos y físicos que forman una computadora. Se dice que todo lo que se puede ver y tocar de una computadora es *hardware*.

Así, a la estructura física de una computadora se le denomina *hardware*, en forma general, según las actividades que realizan sus distintas partes se clasifican en:

1. Unidad de entrada.
 2. Unidad de salida.
 3. Unidad de almacenamiento.
 4. Unidad central de procesamiento.
1. Unidad de entrada






Comprende todos los dispositivos por medio de los cuales el usuario puede tener una comunicación con la computadora. Los dispositivos de entrada aceptan datos e instrucciones del usuario, es la forma de introducir información a una computadora.

Existen muchos dispositivos de entrada pero el más común es el teclado, acepta letras, números y comandos del usuario; muy similar al teclado de una máquina de escribir.



El teclado

Algunos de los dispositivos de entrada de uso más frecuente se muestran a continuación:

Unidades del Entrada	Función
	<p>Ratón (mouse) permite mover un indicador (llamado cursor) sobre la pantalla y ubicarlo sobre determinados objetos que producen una acción al oprimir sus botones.</p>
	<p>Palancas de juegos Permite una acción similar a la del ratón, pero moviendo solo una palanca en lugar del dispositivo completo sobre el escritorio.</p>
	<p>Lápiz óptico Es un lector cuya forma es similar a la de un lápiz que, al desplazarlo sobre un código de barras, produce una señal digital que la computadora interpreta.</p>
	<p>Scanner Digitalizador de imágenes, convierte información gráfica (Fotografías, dibujos, gráficos, estadísticas, etc.) a un formato digital que puede ser leído por la computadora.</p>
	<p>Tabla digital Dispositivo que permite hacer trazos sobre una superficie sensible y ver el resultado (dibujo) en la pantalla de la computadora.</p>

Otros dispositivos de entrada son los CD-ROM, micrófonos y las cámaras digitales.

2. Unidad de salida

Todos aquellos dispositivos que permiten que la computadora transfiera datos del interior al exterior de la misma, se considera unidad de salida. El dispositivo de salida más común es la pantalla o monitor.



Monitor

Unidades de salida de uso frecuente:

Unidades de Salida	Función
	<p>Impresora, permite obtener la información en forma impresa.</p>
	<p>Bocinas, permiten la salida de sonido.</p>
	<p>Plotter, impresión de grandes dimensiones. Generalmente se utiliza para trazar planos y gráficas.</p>
	<p>Cañón, Permite proyectar sobre una pantalla las imágenes del monitor. Se le usa al exponer frente a un auditorio numeroso.</p>

3. Periféricos Duales

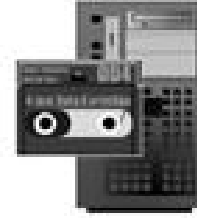
Son aquellos dispositivos que permiten que la computadora reciba y/o entregue datos, puede leerse y escribirse en ellos.



Floppy



Zip drive



Tape Backups



CD - Writer

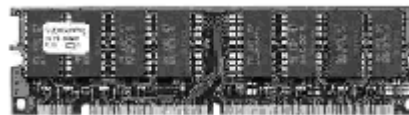


USB

4. Unidad de Almacenamiento

Es la unidad que permite a la computadora guardar información como números, palabras, párrafos, instrucciones o programas en formato digital. Existen cuatro tipos de dispositivos de almacenamiento que se usan con mayor frecuencia en una computadora personal. Los dos primeros se encuentran dentro de la computadora: son las memorias *RAM* (Random Access Memory) o memoria de acceso aleatorio y la memoria *ROM* (Read Only Memory) o memoria de sólo lectura. Los dos últimos son dispositivos de almacenamiento externo y son los discos flexibles y los discos duros.

- a. La memoria *RAM* la forman uno o varios chips en los que se guardan de forma temporal datos.



Memoria RAM

La memoria *RAM* mantiene los datos mientras la computadora está encendida; una vez apagada, se pierde toda la información contenida en esta memoria; es una memoria volátil. En la memoria *RAM* se pueden leer y escribir datos, además almacena los datos con los que el microprocesador va a trabajar. De este modo el microprocesador va tomando los datos de la *RAM*, los procesa, y escribe el resultado también en la *RAM*.

La memoria *RAM* se mide generalmente, en Megabytes y cuanto mayor sea esta cifra, significará que puedes disponer de más sitio para programas y datos sin necesidad de ir constantemente a leerlos al disco flexible o duro.

La memoria *RAM* suele mejorar el desempeño global de una computadora de forma dramática. Es el lugar donde la computadora almacena la información mientras la procesa. Si no tiene suficiente *RAM* la computadora se hará más lenta, o se colapsará cuando uses varios programas al mismo tiempo. Hay quienes opinan que "nunca se tiene suficiente *RAM*"

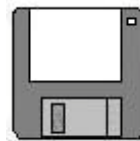
- b.** La memoria *ROM* está constituida por chips que contienen datos que no pueden ser modificados; sólo permite leer datos contenidos en ella, y en ningún caso permite que se escriba. La ventaja que tiene es que al apagar la computadora los datos contenidos en ella no se pierden. La información que contiene ha de ser escrita en el momento de su fabricación, y consiste en programas o datos de configuración que son necesarios para la puesta en marcha de la computadora.



Memoria ROM

Interviene de forma casi exclusiva al encender el equipo para ejecutar automáticamente las operaciones necesarias de arranque, como cargar el sistema operativo.

- c.** Los discos flexibles son un medio de almacenamiento removible. La información almacenada en ellos es fácilmente recuperable. Son de gran importancia para la instalación de software nuevo y para hacer copias de seguridad.



Disco flexible 3 ½

- d.** Discos duros. Es un medio de almacenamiento magnético que está compuesto por un conjunto de platos circulares que giran dentro de una estructura sellada. Almacenan más información y el acceso a la información también es más rápido.

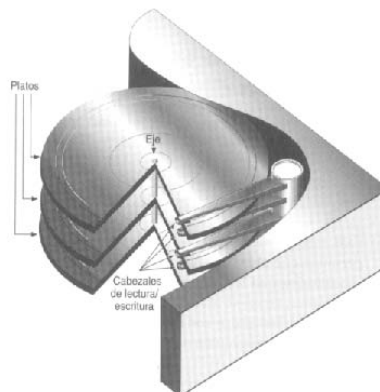


Disco duro

No importa cuál sea el medio de almacenamiento, es de gran importancia saber cómo cuantificar su capacidad. Un dígito binario (bit), puede contener un 0 ó un 1, (encendido, apagado, si o no...). Para representar números y datos más complejos es necesario utilizar una serie de bits. Una medida común en la computación es el byte (u octeto), que es un conjunto de 8 bits. En un byte es posible representar 256 combinaciones diferentes con lo que puede representarse un carácter del mismo alfabeto, o un número de 0 a 255 (sin tomar en cuenta el signo).

Unidad	Cantidad de Información	Símbolo
Bit	1 o 0	Bit
Byte	8 bits	Byte
Kilobyte	1024 bytes	Kb
Megabyte	1024 kilobytes	Mb
Gigabyte	1024 megabytes	Gb
Terabyte	1024 gigabytes	Tb

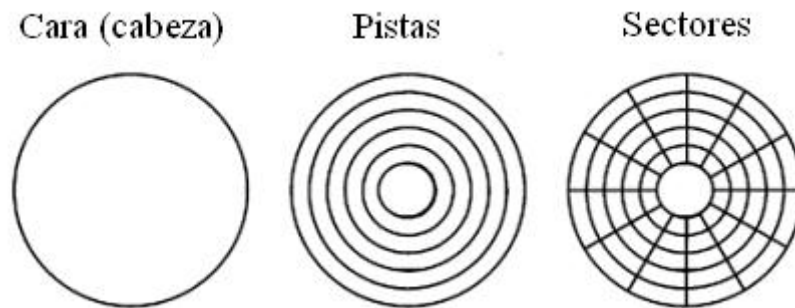
Funcionamiento de una unidad de disco duro. A continuación verás qué mecanismos permiten a la unidad acceder a la totalidad de los datos almacenados en los platos. Cada superficie magnética tiene asignada una de las cabezas de lectura / escritura de la unidad. El conjunto de cabezas se puede desplazar linealmente desde el exterior hasta el interior de la pila de platos mediante un brazo mecánico que los transporta. Por último, para que las cabezas tengan acceso a la totalidad de los datos, es necesario que la pila de discos gire. Este giro se realiza a velocidad constante y no cesa mientras esté encendida la computadora. En cambio, en los discos flexibles sólo se produce el giro mientras se está efectuando alguna operación de lectura o escritura. El resto del tiempo, la unidad de disco permanece en reposo. Con las unidades de CD-ROM ocurre algo similar, sin embargo en este caso la velocidad de giro no es constante y depende de la distancia al centro del dato que se esté leyendo.



Estructura interna de un disco duro

Cada vez que se realiza una operación de lectura en el disco duro, éste tiene que realizar las siguientes tareas: desplazar las cabezas de lectura / escritura hasta el lugar donde empiezan los datos; esperar a que el primer dato, que gira con los platos, llegue al lugar donde están las cabezas y finalmente, leer el dato con la cabeza correspondiente. La operación de escritura es similar a la anterior.

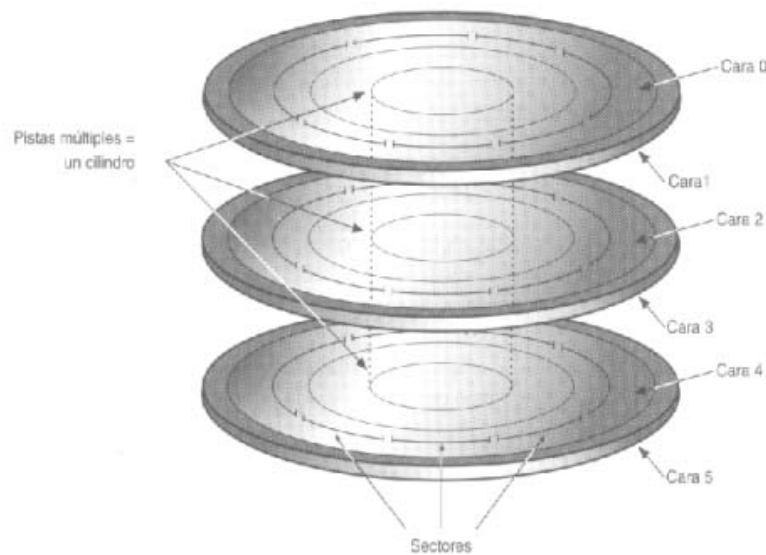
Estructura física: *cabezas*, *cilindros* y *sectores*. Cada una de las dos superficies magnéticas de cada plato se denomina *cara*. El número total de *caras* de un disco duro coincide con su número de *cabezas*. Cada una de estas *caras* se divide en anillos concéntricos llamados *pistas*. Finalmente, cada *pista* se divide en sectores.



Estructura física de un disco duro

Los *sectores* son las unidades mínimas de información que puede leerse o escribirse en un disco duro.

Las *cabezas* y *cilindros* comienzan a numerarse desde el cero y los *sectores* desde el uno. En consecuencia, el primer *sector* de un disco duro será el correspondiente a la cabeza 0, cilindro 0 y sector 1.



Estructura física de un disco duro

5. Unidad Central de Procesamiento (Central Processor Unit)

La Unidad central de procesamiento (*CPU*) es la parte que contiene la inteligencia de la máquina; en ella se realizan cálculos y las decisiones requeridas por la computadora. Es el cerebro de la computadora, la parte que interpreta y ejecuta las instrucciones.

La *CPU* completa, está contenida en un chip muy pequeño llamado microprocesador. El procesador casi siempre se compone de varios circuitos integrados o chips, estos están insertados en tarjetas de circuitos, módulos rígidos rectangulares con circuitos que los unen a otros chips y a otras tarjetas de circuitos.

Un microprocesador tiene por lo menos dos partes básicas:

a. *Unidad de control.*

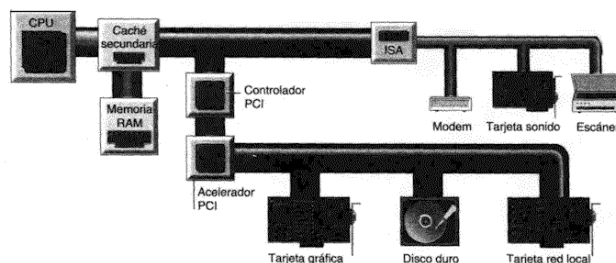
Se encarga de administrar todos los recursos de la computadora, coordina todas las actividades de la computadora, contiene las instrucciones de la *CPU* para llevar a cabo comandos. El conjunto de instrucciones, que están incluidas en los circuitos de la unidad de control, es una lista de todas las operaciones que realiza la *CPU*. Cuando la computadora corre un programa, busca los comandos del programa dentro del conjunto de instrucciones de la *CPU* y las ejecuta en orden.

b. *Unidad aritmético - lógica.*

Esta unidad realiza cálculos (suma, resta, multiplicación y división) y operaciones lógicas (comparaciones). Transfiere los datos entre las posiciones de almacenamiento.

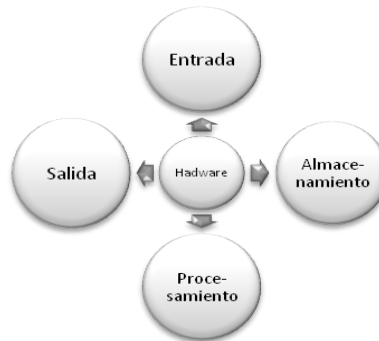
6. EI BUS

La información codificada viaja a través de la computadora por un bus, que es un conjunto de cables que sirven de "autopista" y que alcanza los componentes de la computadora entre sí.



Cada componente se comunica con el procesador (*CPU*) a través del BUS.

El BUS soporta tres tipos principales de información: un grupo de cables transporta datos, tales como la letra A codificada; otro grupo lleva la dirección del componente al que van dirigidos los datos. Cada componente acepta sólo la información que va dirigida a él; por ejemplo, la información enviada a la impresora no será aceptada de forma inadvertida por la unidad de disco. La tercera clase de información son señales de tiempo, que sincronizan todo lo que hay conectado al bus para enviar y recibir mensajes en el instante correcto.



II. SOFTWARE

Es la parte lógica de una computadora, es el conjunto de todas aquellas instrucciones que hacen que la computadora funcione, son intangibles y normalmente las encontramos almacenadas en discos, por ejemplo los programas de aplicación, lenguajes de programación, sistemas operativos, etc.

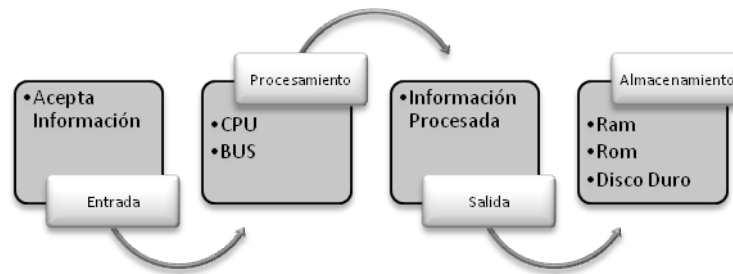
Aplicación. Una aplicación o un paquete computacional, es un programa que efectúa una tarea definida, los principales tipos son:

- Procesadores de texto o palabras.
- Gráficos.
- Hojas de Cálculo.
- Bases de Datos.
- Utilerías.
- Lenguajes de programación (lenguaje ensamblador).
- Sistema Operativo.

El sistema operativo es un software que controla todas las funciones de la computadora, está compuesto por archivos y comandos (internos y externos para el caso de MS-DOS, en la PC). Este sistema operativo actúa como traductor entre el usuario y la computadora. Los componentes que controla son: Teclado, Ratón (mouse), CPU, Monitor, Driver, Disco Duro, etc.

III. FUNCIONES DE UNA COMPUTADORA

- | | | |
|------------------------|------------------------|-------------------|
| a. Acepta información | Entrada= | Input |
| b. Procesa datos | Procesamiento= | Processing |
| c. Produce una salida | Salida= | Output |
| d. Almacena resultados | Almacenamiento= | Storage |



IV. CUESTIONARIO

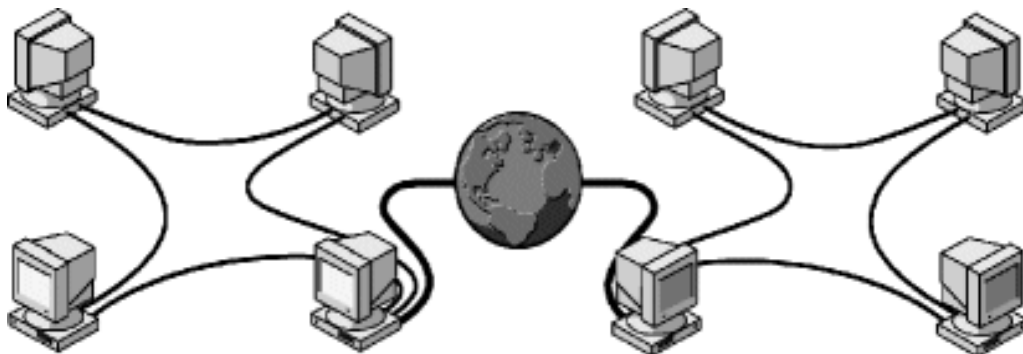
1. Define qué es Hardware.
2. ¿Cuántas partes componen el hardware de una computadora?
3. Menciona las partes que forman una computadora:
4. ¿Cuál es la utilidad de las unidades de entrada?
5. Menciona algunas unidades de entrada:
6. ¿Cuál es la utilidad de las unidades de salida?
7. Menciona algunas de ellas:
8. ¿Cuántos tipos de memoria hay?
9. Explica qué es la memoria ROM y para qué se usa.
10. Explica qué es memoria RAM y para qué se usa.
11. Describe cómo está constituido internamente el disco duro.
12. ¿Cómo se transmite la información por la computadora?
13. ¿Cuáles son los componentes principales del microprocesador?
14. Define qué es software.

Capítulo 7

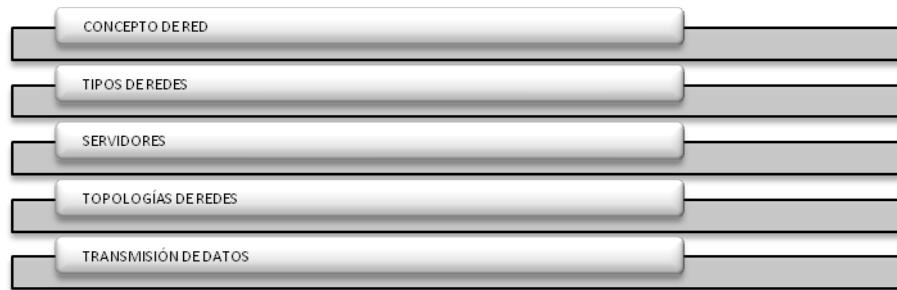
Redes

Temas:

- I. ¿Qué es una Red?
- II. Tipo de Redes.
- III. Servidores de Red.
- IV. Topología de Red.
- V. Transmisión de Datos.
- VI. Protocolos de Red.
- VII. ¿Qué es Internet?
- VIII. Sistemas Operativos de Red.
- IX. Cuestionario.



REDES



I. ¿QUÉ ES UNA RED?

Los sistemas de comunicación en red están constituidos por un conjunto de computadoras conectadas entre sí, de tal forma que sea posible compartir recursos e información. Una red de computadoras ofrece muchas ventajas, como acceso simultáneo a programas y datos, utilización de dispositivos periféricos ubicados en otros equipos, transmisión ágil de datos, respaldos de información más fáciles, comunicación entre personas a través de correo electrónico, información centralizada en el servidor y mucho más.

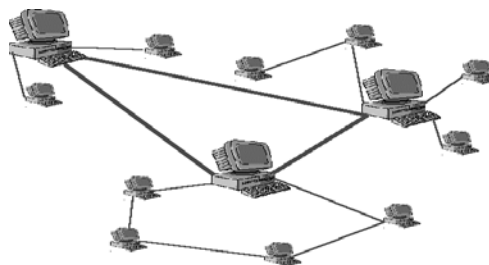
En la actualidad, la comunicación de datos por medio de redes de computadoras, es una de las áreas de la informática que ha tomado más fuerza por la gran cantidad de beneficios que ofrece a las compañías que utilizan sistemas multiusuario.

- Conjunto de computadoras conectadas entre sí, de tal forma que sea posible compartir recursos e información.

CONCEPTO



Para conectar varias computadoras deben seguirse algunos patrones y reglas mediante las que se define el *tipo*, la *topología* y los *protocolos* de comunicación de red que utilizarán.



Red de computadoras.

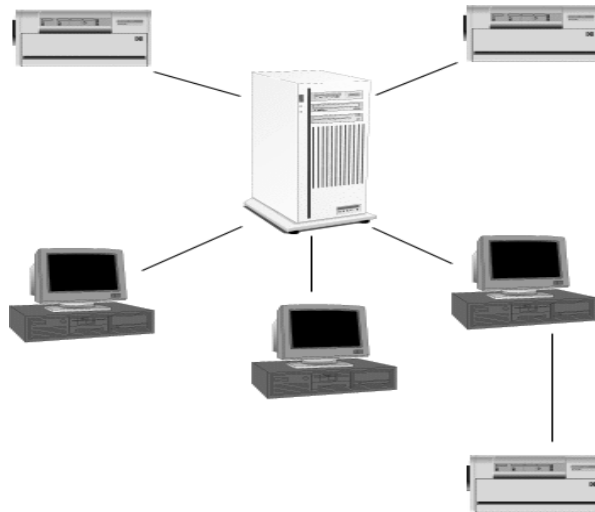
II. TIPOS DE REDES

Existen muchos tipos y tamaños de redes, desde unas pocas computadoras conectadas en un área hasta miles de ellas distribuidas alrededor del mundo, como las que poseen las grandes corporaciones multinacionales. De acuerdo con el área geográfica en la que se conecten los periféricos, existen dos grandes tipos de redes: las **LAN** (Local Area Network, red de área local) y las **WAN** (Wide, Area Network, red de área amplia).

1. Redes de área local (LAN)

Una red de área local es un grupo de computadoras y periféricos conectados por un cable o por un sistema inalámbrico, generalmente en un área geográfica muy cercana.

Una Lan está compuesta por dos o más computadoras conectadas entre sí para compartir recursos y datos.



Red de Área Local.

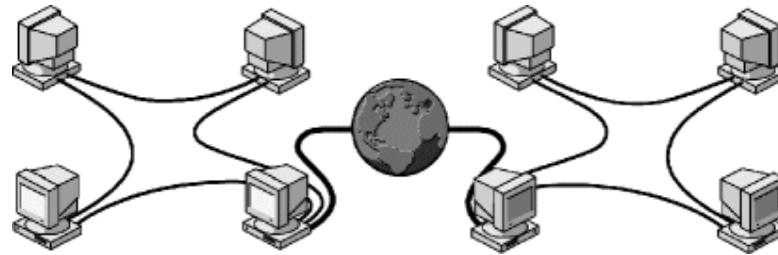
2. Redes de área metropolitana (MAN)

Generalmente llamadas MAN, más extensa que la LAN y su tecnología es similar. La cobertura de las redes MAN comprende desde unos kilómetros a varios cientos de kilómetros, como un grupo de oficinas cercanas dentro de la misma ciudad. Estas redes pueden llegar a usar hasta dos cables como medio de difusión de datos, con lo que se simplifica el diseño.

3. Redes de área amplia (WAN)

Una WAN está constituida por dos o más LAN interconectadas, que cubren una zona geográfica muy amplia. Este tipo de red es común en las empresas que tienen sucursales en varias ciudades o países. Cada sucursal tiene un sistema de red local a través del cual comparten recursos e información, pero también

necesitan datos de otras sucursales, por esta razón la comunicación es evidentemente necesaria por medio de redes de área amplia.



Red de Área Amplia.

4. Módem

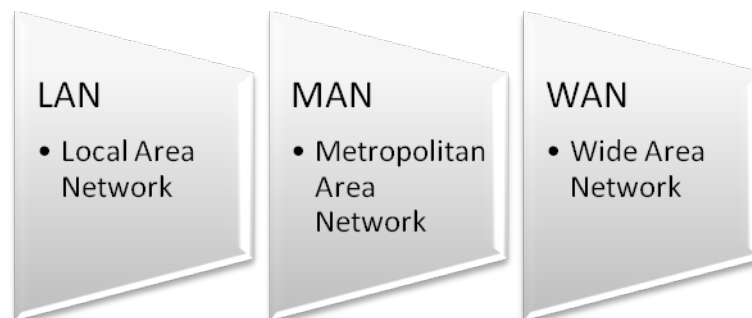
Otra manera de conectar computadoras es a través de la línea telefónica. Esto es posible utilizando un dispositivo que se conecta a la computadora y luego a la línea telefónica; este dispositivo se denomina *módem* (**Modulador/Demodulador**)

Existe una gran variedad de módems pero todos funcionan de la misma manera. La *modulación* es el proceso por el que los datos binarios son convertidos en señales electromagnéticas que la línea telefónica puede transportar.



Módems comerciales.

Por otra parte la *demodulación* es el proceso inverso, es decir, la conversión de señales electromagnéticas en señales digitales.



III. SERVIDORES DE RED

Un *servidor* es un programa que presta servicio a un solicitante, generalmente denominado *cliente*; este servicio puede ser: imprimir, administrar periféricos o respaldar información. Una misma red puede incluir varios servidores, cada uno con tareas específicas.

Existe una gran variedad de servidores, entre los cuales se destacan los siguientes: Servidor de archivos, Servidor de programas, Servidor de red, Servidor de impresión, Servidor de copias de seguridad, Servidor de correo, entre otros.

1. Servidor de archivos

Esta computadora almacena todos los archivos que se crean o se modifican en computadoras conectadas a la red, su función es guardar toda la información y mantenerla disponible para los usuarios.

2. Servidor de programas

Este se encarga de almacenar todos los programas (*software*) instalados en la red. Su labor es permitir a los usuarios que utilicen las aplicaciones para realizar los trabajos diarios.

3. Servidor de red

Este equipo es el administrador de la red, en él se almacena toda la información referente a la administración de ésta, como tiempo de conexión, tareas realizadas, recursos utilizados, etc.

4. Servidores de impresión

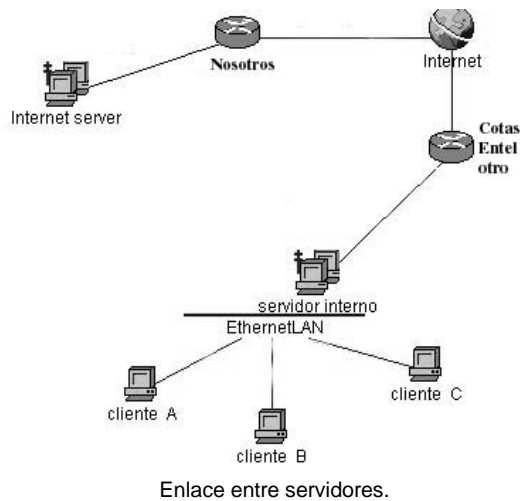
Sirve para recibir los trabajos de impresión de todos los usuarios de la red, así como para administrar las prioridades entre una o más impresoras.

5. Servidores de copias de seguridad

Se encarga de realizar los respaldos de información comúnmente llamados *copias de seguridad*. Se caracteriza porque está programado para ejecutar dicha operación de manera automática y, por lo general, en horas nocturnas.

6. Servidor de correo

Administra el correo entrante y saliente de todos los usuarios de la red, es decir, direcciona el correo de un equipo a otro.



IV. TOPOLOGÍA DE RED

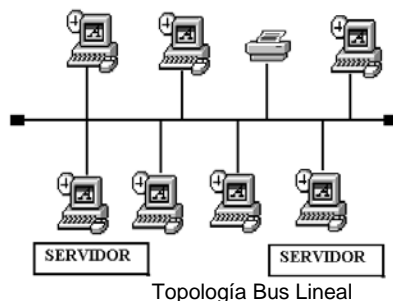
Se entiende por *topología* a la forma en que se distribuyen las líneas (cables) de una red. Las tres principales topologías son: *Bus Lineal*, *Estrella* y *Anillo*.

¿Cuál es la mejor? Realmente no es posible decirlo, debido a que cada topología se adapta según los equipos que se conectan y el cableado que se utiliza.

En ocasiones, las empresas comunicadas en red no solo utilizan una topología sino varias de ellas, por ejemplo, en una empresa que funcione en un edificio se utiliza una red en bus para conectar los pisos y a su vez cada piso utiliza una topología diferente como anillo o estrella.

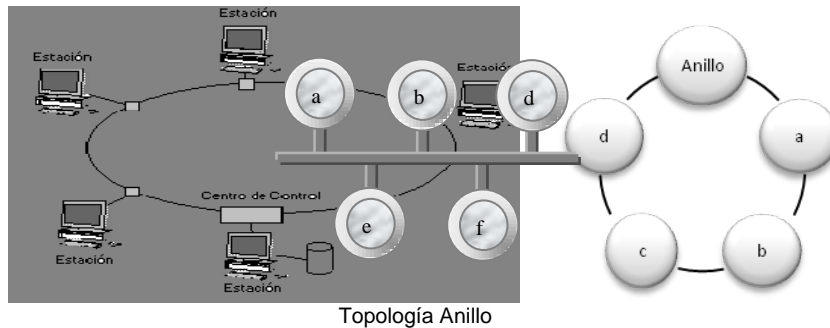
1. Red en bus

En esta topología, las computadoras y demás periféricos se conectan a través de una línea principal a la cual se adhieren todos los nodos de los dispositivos. (Un *nodo* es el punto de comunicación en un periférico con una línea principal de la red). Esta es una de las topologías más utilizadas, sin embargo, su gran problema son las colisiones de información debido a que todos los nodos envían y reciben simultáneamente datos por la misma línea.



2. Red en anillo

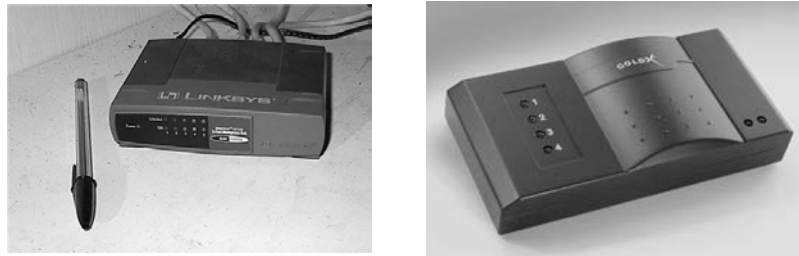
En una topología de red en anillo los dispositivos se conectan uno tras otro hasta cerrar la conexión. Aquí, los datos viajan en forma circular en un solo sentido, característica que evita la colisión de información y permite que solo el nodo destino capture los datos, los demás nodos actúan simplemente como conductores. Desventaja, si un nodo se daña o sufre problemas, la red se cae; por tanto todos los nodos deben estar funcionando correctamente siempre.



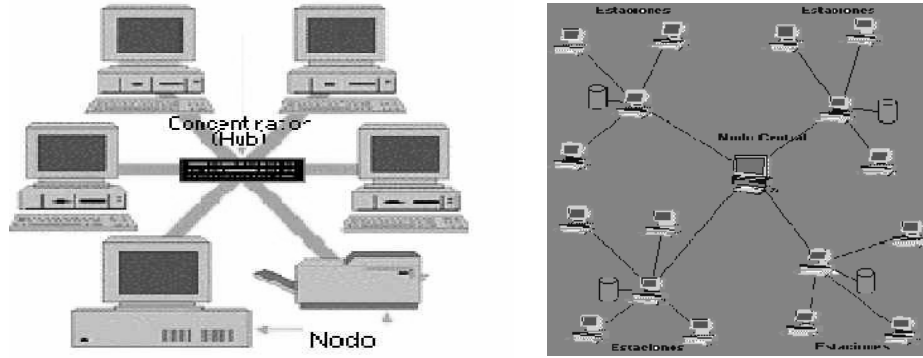
3. Red en estrella

En las redes de computadoras con topología en estrella los dispositivos (terminales, servidores, impresoras, etc.) se conectan por medio de un *hub* (dispositivo de conexión central) que actúa como un regulador de información para evitar la pérdida de datos por problemas de colisión.

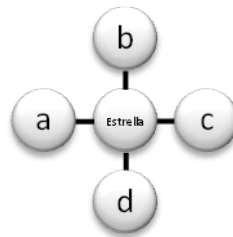
El *hub* se encarga de recibir los datos que están siendo transmitidos y los direcciona al nodo destino, el gran problema es que si el panel de control se daña, automáticamente se cae toda la red.



HUB comerciales



Topología tipo Estrella



V. TRANSMISIÓN DE DATOS

En los sistemas de comunicación en red intervienen varios elementos que hacen posible la intercomunicación de equipos.

Uno de tales elementos son los cables que facilitan la comunicación entre computadoras, entre los tipos más importantes está el cable coaxial, el cable par trenzado, el cable de fibra óptica y los enlaces inalámbricos.

- Cable coaxial.

Cable utilizado en redes para la transmisión de información en forma digital, consta de un núcleo o salida de cobre cubierto por una capa de plástico y en la parte exterior está rodeado por una lámina protectora. Entre sus principales características están: inmunidad al ruido e interferencias, no necesita modulación ni energía, se utiliza principalmente en redes con topologías en anillo, estrella y bus lineal y se emplea para comunicaciones en redes de banda base, banda ancha y televisión por cable.



Cable coaxial

- Par trenzado.

El cable de par trenzado es muy similar al cable coaxial, pero a diferencia de este consta de dos o más pares de cables aislados y entrelazados, aproximadamente seis trenzas en una pulgada. La razón para utilizar dos cables en par trenzado es que uno de ellos es un polo a tierra y el otro transporta la señal.

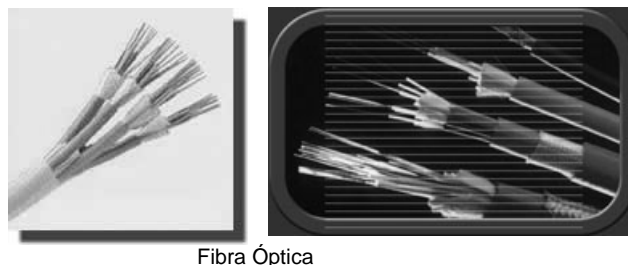
El cable de par trenzado transporta señales digitales a una velocidad de transmisión de 2 a 4 Mbits/seg. Se utiliza en topologías como bus, estrella y anillo; es inmune al ruido electromagnético.



- Fibra óptica.

El cable de fibra óptica utiliza el fenómeno físico de *refracción* para transmitir un haz de luz a lo largo de un cable de vidrio especialmente diseñado para transportar el rayo generado por un láser. Un cable de fibra óptica consta de un conductor central recubierto por una capa protectora que impide la interferencia electromagnética.

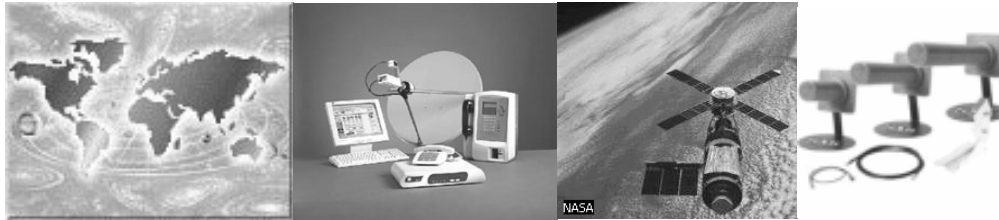
Este tipo de cables se utiliza para redes de computadoras en anillo y estrella; su principal característica es que transmite datos, sonido y video en tiempo real a una velocidad alta.



- Enlaces inalámbricos.

Otra forma de transmitir datos a grandes distancias es a través de enlaces inalámbricos, como los establecidos mediante las ondas de radio, las señales de microondas y los satélites.

Las ondas de radio son muy utilizadas para enlazar computadoras portátiles. Las microondas son un tipo de ondas de radio que transmiten punto a punto por medio de antenas de comunicación, y los satélites se utilizan para cubrir millones de kilómetros de distancia, es decir, cuando se comunican personas alrededor del mundo, un satélite en órbita geoestacionaria recibe las señales de radio emitidas desde un amplificador y las envía a su destino en la Tierra.

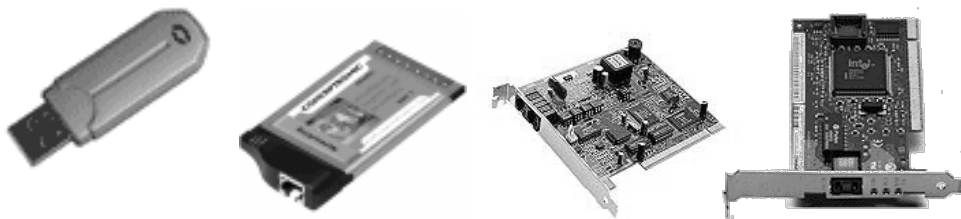


Enlace satelital.

VI. PROTOCOLOS DE RED

Además del cable, la topología y el tipo de red que establece su estructura, cada una de las computadoras necesita de *software* y *hardware* especial que permite el acceso del periférico de la red.

El hardware lo constituye una tarjeta de red, la cual es conectada a una de las ranuras de expansión de la tarjeta principal y también ofrece un puerto para conectar el cable de la red.



Tarjetas de red

El software está compuesto por los programas que permiten la instalación y el buen funcionamiento de la tarjeta de red. Tanto la tarjeta como el software deben cumplir con ciertas reglas y parámetros estandarizados denominados **protocolo de red**.

Existen varios tipos de protocolos; entre los más utilizados están **Ethernet**, **Token Ring**, **ARCnet** y uno de los más conocidos **TCP/IP**.

Un protocolo busca lograr mayor eficiencia mediante la reducción de la cantidad de información transmitida prefijando ciertas condiciones; por ejemplo, técnica de revisión de errores. Además, posee reglas que permiten a una terminal receptora diferenciar entre información de control y de datos.

Los protocolos de comunicaciones establecen una conexión, verifican identificaciones de terminales, manejan el secuenciamiento de datos, permiten la interrupción o suspensión de la transmisión y manejan el control de errores.

1. Ethernet

Es el protocolo de red más utilizado en redes con topología en bus lineal. Transmite datos a una velocidad de 10 Mbts/seg y los nodos se conectan por cable coaxial, fibra óptica o par trenzado. Ethernet utiliza **CSMA/CD** (Carrier Sense Múltiple Access / Collision Detection) este se utiliza para prevenir fallos en la red por colisiones cuando dos dispositivos intentan transmitir al mismo tiempo, así como el estándar **IEE** (Institute, of Electrical and Electronics Engineers) 10 base-T.

2. Token Ring

Protocolos de redes de IBM que usa el paso de señales para regular el tráfico de información y evitar colisiones. Transmite de 4 a 16 Mbts/seg; se basa en el estándar IEEE 802.5 y permite conectar computadoras personales, mini computadoras y mainframes. Una red Token Ring se basa en la topología en anillo y tiene la ventaja de que la información viaja de una manera controlada en una sola dirección.

3. ARCnet

El protocolo de Datapoint Corporation, ARCnet (Attached Resource Computing Network) permite conectar varias computadoras y estaciones de trabajo por medio de cable coaxial, par trenzado o fibra óptica en topología de estrella.

ARCnet utiliza el método de acceso propio **token passing** (paso de señales) a velocidad de hasta 2.5 Mbts/seg. Es muy fácil de instalar y sus componentes son de bajo costo.

4. TCP / IP

Se han desarrollado diferentes familias de protocolos para comunicación por red de datos para los sistemas **UNIX**. El más ampliamente utilizado es el Internet **Protocol Suite**, comúnmente conocido como **TCP / IP**.

Es un protocolo DARPA que proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP / IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto.

El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PCs, minicomputadoras y computadoras centrales sobre redes de área local y área extensa. TCP / IP fue desarrollado y mostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET una red de área extensa del departamento de defensa.



VII. ¿QUÉ ES INTERNET?

Internet ha producido un gran cambio en el uso de la tecnología en casi todos los ámbitos, pero esencialmente en las comunicaciones.

Internet es una red mundial de computadoras que permite que los usuarios conectados a ellas, puedan compartir recursos e información. Las computadoras se interconectan usando la red telefónica, ya sea conectándose a ella mediante módems o directamente a enlaces digitales.

Internet es un gran banco de datos, donde puede encontrarse una gran cantidad de información acerca de cualquier tema, lo que la hace de gran utilidad para maestros, estudiantes, científicos, músicos, niños y, en general, para cualquier persona.

1. Características de Internet

El término Internet corresponde a la expresión **Internetwork System** o sistema de intercomunicación de redes.

Internet es la red de área amplia (WAN) más grande del mundo, cuyos elementos (computadoras, redes y usuarios) se encuentran distribuidos por todo el planeta.

Al trabajar en Internet es posible efectuar una infinidad de operaciones relacionadas con el manejo de la información, es decir, existe la posibilidad de establecer comunicación con cualquier lugar del mundo en donde exista una computadora conectada a Internet.

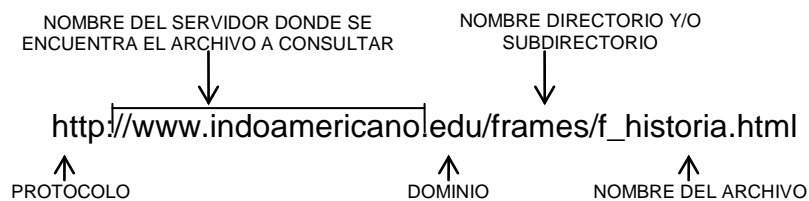
A diferencia de otras redes, Internet no posee un administrador central que tenga control sobre su totalidad, en lugar de esto, existe un grupo de usuarios identificados como “la sociedad de Internet” **ISOC** (Internet Society) que cumple las funciones de promover el intercambio de información utilizando Internet, así como de identificar estándares tecnológicos en el *hardware* y *software*.

Entre los servicios más importantes que pueden usarse a través de Internet están el correo electrónico, Finger, Usenet, FTP, Chat, Gopher, Wais, Navegación, Telnet, entre otros.

2. Direcciones en Internet

La estructura de una dirección en Internet se revisa, para su comprensión, de derecha a izquierda, yendo, en ese sentido, de lo más general a lo más particular.

La estructura de una dirección en Internet es la siguiente:



Donde se tienen diferentes dominios así como sigue:

Categorías de los dominios en INTERNET para Estados Unidos		
Cat.	Dominio	Ejemplo
.com	Comerciales	http://www.kodak.com
.edu	Área educacional	http://www.indoamericano.edu
.gov	Gobierno	http://www.nih.gov
.org	Organizaciones sin fines de lucro	http://info.soc.org
.net	Organización administrativa para una red	http://cool.infi.net
.mil	Área militar	http://www.nic.mil/
.int	Organismos internacionales	http://www.itu.int/
.biz	Área de los negocios	http://edomain.biz/
.coop	Negocios y empresas que actúan como cooperativas	http://www.ica.coop/
.name	Páginas personales	http://john.brier.name/
.museum	Museos	http://www.york.srt.museum/

Dominios para países (algunos ejemplos)		
Dominio	País	Ejemplo
.ar	Argentina	www.cn.rffdc.edu.ar
.br	Brasil	www.ctaa.embrapa.br
.fr	Francia	mistral.culture.fr
.mx	México	www.yahoo.com.mx

Adicionalmente a las direcciones de Internet, cada persona puede tener direcciones únicas en Internet, con la finalidad de intercambiar correo electrónico y hacer uso de algunos otros servicios.

Nombre_usuario@dominio

En donde *nombre_usuario* corresponde a la identificación del usuario, que es la manera de reconocer en forma única a entidades, computadoras y usuarios en Internet. El *dominio* está dividido por puntos y cada una de estas subdivisiones corresponde a niveles de organización, grupos de computadoras y usuarios individuales. Las subdivisiones van de lo específico a lo general (de izquierda a derecha); esta clasificación recibe el nombre de **DNS** (Domain Name System, sistema de nomenclatura de dominios), por ejemplo una dirección podría ser la siguiente:

Juan@yahoo.com

3. Servicios de Internet

Internet es una red de redes que une a millones de personas y organizaciones de diferentes partes del mundo, en la cual básicamente se realizan cuatro operaciones: manejo de correo electrónico, navegación en páginas web, transferencia de archivos y comunicaciones en tiempo real.

- Correo electrónico.

Este servicio también conocido como *e-mail*, permite enviar mensajes electrónicos que pueden incluir archivos con textos, imágenes, sonidos, datos numéricos, etc.

Para transferir un mensaje por correo electrónico se requiere de lo siguiente:

- a) Tener acceso a Internet y contar con el servicio de correo, pues el proveedor es el que administra el correo entrante y saliente.
- b) Poseer un programa para enviar y recibir cartas digitales.
- c) Conocer la dirección electrónica del destinatario del mensaje.

- Chat.

Otro de los servicios que ofrece Internet es la posibilidad de conversar de manera interactiva en la red. Microsoft Chat es uno de estos programas que permite transferir información de tipo textual, voz y video. Opera como un gran salón de conversación donde los usuarios se conectan e intercambian mensajes.

- Teleconferencia.

Las teleconferencias son otro tipo de comunicación en tiempo real, donde uno o más usuarios pueden hablar y observar a su interlocutor.

- Transferencia de archivos.

Consiste en tomar información (textos, fotografías, dibujos, etc.) en forma de archivos, desde un sitio de Internet y almacenarla en la computadora local. Un servicio específico de esta clase lo ofrecen los servidores FTP, que son grandes bases de datos debidamente organizadas para este propósito.

- Navegación.

Este es uno de los servicios más populares de Internet, donde los usuarios exploran una cantidad enorme de información que se encuentra almacenada en páginas web, es decir, van de una página web a otra; para ello, emplean programas como Internet Explorer o Netscape Navigator.

VIII. SISTEMAS OPERATIVOS DE RED

El sistema operativo de una red (NOS) es la parte fundamental de la misma. Como todo sistema operativo, es responsable de gestionar las operaciones entre las estaciones de trabajo y los servidores. El NOS determina también la seguridad de la red, por ejemplo, controlando claves de acceso o autorizaciones para que algún usuario pueda o no utilizar alguna aplicación o dato.

Los sistemas operativos de red se forman por muchos módulos de software que administran los recursos de la red, la transferencia de archivos, accesos, asignación de direcciones a las computadoras conectadas, uso de menú, etc.

Los principales sistemas operativos de red, para la industria de las redes que utilizan PC, son Novell, Microsoft, IBM y Banyan. Los distintos NOS tienen operaciones comunes para la gestión de directorios de archivos y operaciones de los servidores, de manera que hay cierta compatibilidad entre ellos. Sin embargo, una persona que pretende instalar o administrar una red, debe analizar con cuidado las características de los fabricantes, para seleccionar el NOS que mejor se adapte a sus necesidades particulares, ya que la forma específica en que cada sistema operativo funciona, es completamente distinta.

Para evaluar un NOS se debe tener en cuenta:

- Facilidad de uso.
- Eficiencia para las tareas.
- Compatibilidad en las computadoras de la red.
- Precio.

IX. CUESTIONARIO

1. ¿Qué es una red de computadoras?
2. ¿Qué ventajas tienen las redes de computadoras?
3. ¿Cuáles son las reglas y patrones que se deben considerar para conectar una red de computadoras?
4. ¿Cuáles son los tipos de redes y en qué consiste cada uno de ellos?
5. Investiga el o los tipos de redes que existen en tu colegio.
6. ¿Qué es un módem?
7. ¿Cuál es la finalidad de un módem y cómo funciona?
8. Realiza fichas técnicas de cinco marcas diferentes de módems.
9. ¿Qué es un servidor de red?
10. Menciona algunos servidores.
11. Investiga qué tipos de servidores tienen en tu colegio.
12. ¿Qué es una topología de red? Describe las más comunes.
13. Investiga la(s) topología(s) de red(es) en tu colegio.
14. Realiza fichas técnicas de cinco HUB con marcas comerciales.
15. En tu cuaderno realiza una tabla comparativa de los diferentes modos de transmisión de datos.
16. Realiza fichas técnicas de tarjetas inalámbricas de red.
17. ¿Qué es un protocolo de red?
18. ¿Cuáles son los protocolos de red más comunes?
19. ¿En qué consiste el TCP/IP?
20. ¿Qué es el Internet?
21. Menciona tres características del Internet.
22. Explica en qué consiste la estructura de una dirección en Internet.
23. Investiga otros servicios de Internet que no se hayan mencionado en el texto.

24. Explica cuál es la finalidad de los NOS (Sistemas Operativos de Red).
25. Realiza fichas técnicas sobre algunos sistemas operativos de red.

BIBLIOGRAFIA

TÍTULO: Enciclopedia de Microsoft Visual Basic 6.

AUTOR: Fco. Javier Ceballos.

EDITOR: Alfaomega.

TÍTULO: Metodología de la Programación.

Algoritmos, Diagramas de Flujo y Programas. Tomo 1.

AUTOR: Osvaldo Cairó.

EDITOR: Alfaomega.

TÍTULO: Metodología de la Programación.

Algoritmos, Diagramas de Flujo y Programas. Tomo 2.

AUTOR: Osvaldo Cairó.

EDITOR: Alfaomega.

TÍTULO: Algebra Recreativa.

AUTOR: Y. Perelman.

EDITOR: Ediciones Quinto Sol.

TÍTULO: Desarrollo de Habilidades del Pensamiento.

Pensamiento Verbal y Solución de Problemas.

AUTOR: Margarita A. de Sánchez.

EDITOR: Editorial Trillas.

TÍTULO: Introducción a la Computación. Bachillerato.

AUTOR: Jorge Vasconcelos Santillán.

EDITOR: Publicaciones Cultural.

TÍTULO: Informática Para Cursos de Bachillerato.

AUTOR: Gonzalo Ferreira Cortés.

EDITOR: Alfaomega.

TÍTULO: Aprendiendo Delphi 2 en 21 días.

AUTOR: Dan Osier, Steve Grobman y Steve Batson.

EDITOR: Prentice Hall Hispanoamericana, S. A.