

28
25



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE FILOSOFIA Y LETRAS
COLEGIO DE PEDAGOGIA

"LA ENSEÑANZA DE LA PROGRAMACION EN NIÑOS
DE 5º Y 6º AÑO DE PRIMARIA COMO ESTIMULADORA
DEL DESARROLLO INTELECTUAL Y CREATIVO DEL
NIÑO.

PROPUESTA DE UN MANUAL DE BASIC."

*Vo Bo
on Gadsden*

T E S I S I N A

QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN PEDAGOGIA

P R E S E N T A :

MARIA DE LOS ANGELES GADSDEN CARRASCO

FACULTAD DE FILOSOFIA Y LETRAS



México, D. F.

COLEGIO DE PEDAGOGIA

1993

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

I N D I C E

Introducción	3
Antecedentes	6
1. Fundamentos Pedagógicos:	
Teoría de Jean Piaget. Desarrollo cognoscitivo del niño.	7
Transmitir o Descubrir. Dos clases de Aprendizaje.	20
Hablando un poco de programación.	23
2. Propuesta	
Objetivos.	25
Contenido.	27
3. Presentación del Manual de BASIC	
"Aprendiendo a programar computadoras"	32
Conclusiones y Recomendaciones.	136
Bibliografía.	138

"Descubriremos que el resultado más importante reside en la habilidad del niño para articular el trabajo de su propia mente y, en particular, la iteración entre él y la realidad a medida que aprende y piensa".

Jean Piaget.

I N T R O D U C C I O N

Las características y necesidades propias de la estructura socio-económica en los países industrializados, han dado origen a un amplio desarrollo tecnológico, especialmente en el área de la computación.

En nuestra sociedad, el uso de las computadoras ha invadido todos los campos de trabajo; actualmente en el ámbito educativo, se ha incluido el uso de la computación como auxiliar para analizar una diversidad de procesos educativos.

Sin embargo, la posibilidad de operar una computadora no la tienen únicamente los especialistas. Los niños también pueden manipularla y vincularse así, con uno de los avances tecnológicos más importantes de este siglo.

La computadora es un instrumento intelectual para el estudiante. Le permite explorar un microcosmos rico en experiencias lógicas; un mundo personal en que todas las

respuestas contribuyen a una profundización de sus experiencias.

Provoca un cambio en el proceso usual de aprendizaje, propiciando el razonamiento mediante la necesidad de pensar paso por paso, con una retroalimentación inmediata.

La enseñanza de la computación estimula el desarrollo intelectual. El niño, al aprender a programar, desarrollará una lógica que actualmente no se enseña en la escuela primaria, pero que es de gran ayuda para introducirlo en el mundo del razonamiento abstracto.

Los que participamos en la enseñanza de la computación enfrentamos el problema de la falta de buenos libros y apuntes escritos en español, debido a la dependencia tecnológica que países como el nuestro, en vías de desarrollo, tienen en éste y en otros muchos campos.

Es de suma importancia que a través de lo ameno en la forma de transmitir el conocimiento se dé la formalización del aprendizaje de tal manera que la relación, trabajo-juego-aprendizaje, se vuelva intrínsecamente una realidad en el niño.

Lo que se pretende con este trabajo es que el niño sea quien programe a las computadoras, desarrolle sus propias ideas, adquiriendo así un sentido de dominio sobre un elemento de la tecnología más moderna y poderosa y a la vez establezca un contacto en la construcción de modelos intelectuales.

Para poder llevar a cabo estas ideas, se ofrece un manual de computación para la enseñanza de la programación del lenguaje BASIC.

La base teórica está apoyada en las investigaciones de Jean Piaget, en lo que se refiere al desarrollo cognoscitivo del niño y su relación con el aprendizaje.

Bajo esta fundamentación, se propone la etapa de las operaciones concretas, específicamente entre los diez y doce años, factible para la enseñanza de la programación.

La primera parte de este trabajo está dedicada a la fundamentación teórica, posteriormente se expondrá la propuesta concreta y para finalizar se concluye con la presentación del manual "Aprendiendo a programar computadoras".

A N T E C E D E N T E S

Esta tesina formó parte de un proyecto de computación infantil de la Dirección General de Servicios de Cómputo Académico de la Universidad Nacional Autónoma de México.

Los conocimientos adquiridos a lo largo de la participación en este proyecto, fueron fundamentales para la elaboración del material que se expone.

Para su realización, se retomó la experiencia adquirida en los cursos de Introducción a la Programación por medio del lenguaje BASIC, para niños de ocho a trece años, impartidos en la U.N.A.M..

Todo lo que concierne a la descripción, elaboración y resultados obtenidos en dichos cursos, quedó plasmado en la tesina: "Técnicas de solución de problemas simulando una computadora", desarrollada por Sara Luz del Río Gómez, que también formó parte del proyecto y que tuvo estrecha relación en el desarrollo del manual, tema de la presente tesina. Por este motivo, sólo describo el manual y su fundamentación teórica.

FUNDAMENTOS PEDAGOGICOS.**Teoría de Jean Piaget. Desarrollo cognoscitivo del niño.**

El fundamento teórico de éste trabajo está basado en las investigaciones de Jean Piaget, en lo que se refiere al desarrollo cognoscitivo del niño y su relación con el aprendizaje.

La base fundamental de esta concepción es que toma en cuenta la naturaleza propia del niño, es decir, considera la estructura psicológica del pensamiento del niño y las leyes de su desarrollo para que éste conquiste por sí mismo, por su propia actividad, el conocimiento a través de investigaciones y descubrimientos libres.

Para Piaget, el desarrollo psíquico, que se inicia al nacer y concluye en la edad adulta se puede comparar al crecimiento orgánico, al igual que este último, consiste esencialmente en una marcha hacia el equilibrio.

El cuerpo evoluciona hasta alcanzar un nivel relativamente estable caracterizado por el final del crecimiento y la madurez de los órganos, así también, la vida mental puede concebirse como la evolución hacia una forma de equilibrio final.

"El desarrollo es, por lo tanto, en cierto modo una progresiva equilibración, un perpetuo pasar de un estado de menor equilibrio a un estado de equilibrio superior". (1)

Toda actividad, ya sea movimiento, pensamiento o sentimiento, es impulsada por una necesidad, y ésta es la manifestación de un desequilibrio. La actividad termina en cuanto la necesidad se satisface y se recupera el equilibrio. Por ejemplo, el encuentro con un objeto exterior desencadenará la necesidad de jugar, o suscitará una pregunta, y la acción terminará en cuanto su necesidad de respuesta esté satisfecha, provocando que el desequilibrio se restablezca.

En este mecanismo continuo y perpetuo de reajuste o equilibración consiste la acción humana, y por esta razón pueden considerarse las estructuras mentales sucesivas, que dan origen al desarrollo, cada una de las cuales representa un progreso con respecto al anterior.

Ahora bien, cada vez que un desequilibrio se presenta, por así decirlo, el niño se ve en la necesidad de "asimilar" aquella situación que produjo el cambio para poder "acomodar" sus estructuras cognoscitivas en forma cada vez más estable, y con esto hacer más sólido el equilibrio mental.

Para Piaget asimilar es: "... incorporar las cosas y las personas a la actividad propia del sujeto y, por consiguiente, ... 'asimilar' el mundo exterior a las estructuras ya construidas...".(2)

El concepto de acomodación funciona complementariamente al término de asimilación. Una vez que las experiencias han

(2) Ibidem, p. 18

sido incorporadas a las estructuras cognitivas del sujeto, es necesario "hacer" las modificaciones consecuentes en dichas estructuras, es decir, "... reajustar [las estructuras construidas] en función de las transformaciones sufridas, y, por consiguiente, a 'acomodarlas' a los objetos externos". (3)

De este modo la actividad cognoscitiva del sujeto es entendida como un constante reajuste ante situaciones nuevas, que le permiten lograr un mayor equilibrio mental.

De acuerdo con Richmond "... los procesos gemelos de asimilación y acomodación son rasgos permanentes del trabajo de la inteligencia, es decir, están presentes en todos los estados de desarrollo de la inteligencia. La adaptación al medio se produce tan solo cuando los dos procesos se hallan en equilibrio y entonces la inteligencia encuentra su equilibrio en el medio". (4)

Es decir, la percepción y los movimientos elementales, dan primero acceso a los objetos próximos en su estado momentáneo, luego la memoria y la inteligencia prácticas permiten a la vez reconstituir su estado inmediatamente anterior y anticipar sus próximas transformaciones. El pensamiento intuitivo viene luego a reforzar ambos poderes. La inteligencia lógica, en su forma de operaciones concretas y finalmente de deducción abstracta, termina esta evolución haciendo al sujeto dueño de los acontecimientos más lejanos, tanto en el espacio como en el tiempo.

El equilibrio se genera en el niño cuando ha logrado

(3) Idem, p. 18

(4) RICHMOND, P.G. Introducción a Piaget, p. 128

comprender un nuevo conocimiento y puede integrarlo a sus estructuras anteriores. El equilibrio es estable en la medida que el niño responde, conforme a sus estructuras internas, ya conformadas pero también es modificable ya que el niño está en constante desarrollo, y por tanto en constante asimilación de conocimientos.

Ahora bien, al asimilar de esta forma los objetos, la acción y el pensamiento se ven obligados a acomodarse a ellos.

Esto es lo que llama Piaget equilibrio psíquico y el desarrollo mental aparece finalmente, en su organización progresiva, como una adaptación cada vez más precisa a la realidad.

Cada uno de los estadios se caracteriza por la aparición de estructuras originales, cuya construcción le distingue de los estadios anteriores.

Cada estadio constituye una forma particular de equilibrio, y la evolución mental se efectúa en el sentido de una equilibración cada vez más avanzada.

La teoría piagetiana divide el desarrollo intelectual del niño en cuatro etapas principales.

1.- Período de la inteligencia sensorio-motora.

La evolución cognoscitiva en esta etapa abarca del nacimiento hasta los dos años aproximadamente. En este período el niño no presenta todavía pensamiento ni afectividad ligada a representaciones que permitan evocar las

personas o los objetos ausentes. Pero, pese a esas lagunas, el desarrollo mental es particularmente rápido y de importancia especial, porque el niño elabora a ese nivel el conjunto de las subestructuras cognoscitivas que servirán de punto de partida a sus construcciones perceptivas e intelectuales ulteriores.

Esta etapa se caracteriza por el desarrollo de los movimientos, que de reflejos innatos, pasan a ser movimientos voluntarios que le permiten al niño dirigir sus actividades hacia objetivos determinados.

Cuando nace, el primer movimiento que presenta es el reflejo de succión, el cual sufrirá un avance progresivo en los primeros días. Cuando la madre comienza a darle pecho, él presentará pequeños problemas para succionar, sin embargo poco a poco irá asimilando dicha acción.

Al llegar a los dos o tres meses el niño comenzará a presentar lo que Piaget llama "inteligencia práctica", que se basa exclusivamente en la manipulación de objetos. Esta manipulación le permitirá percibir movimientos, los que estarán organizados en esquemas de acción. Mientras el niño siga manejando los objetos irá experimentando diversas conductas que harán que se desarrollen y multipliquen los esquemas de acción, sin embargo no se debe perder de vista que esta asimilación está en un nivel sensorio-motor.

En el transcurso del primer año, el niño presentará un marcado egocentrismo, ésto provoca que la causalidad vaya implícita en la propia actividad del niño, no hay relación

entre un acontecimiento con otro, no obstante, con base en la experiencia, podrá comprobar que existe una causa para cada suceso. Hablando con respecto al nivel del niño, se da cuenta de que cuando tira de un mantel y se encuentra algún objeto encima de éste, el objeto caerá al suelo, o si jala un cordón cuyo extremo tiene una campana sabrá que la campana sonará. Por lo tanto "el niño reconoce las relaciones de causalidad de los objetos entre sí: objetiva y localiza, pues, las causas" (5).

Una vez que ha adquirido estas habilidades, aproximadamente a los dos años, se presenta la etapa preoperacional.

2.- Período del pensamiento preoperatorio.

Abarca un intervalo que comienza hacia los 2 años y que termina hasta los siete u ocho aproximadamente.

La adquisición del lenguaje es, quizá, el acontecimiento más importante de este período, ya que su desarrollo modifica sustancialmente tanto las estructuras mentales como su relación con las demás personas.

A los dos años aproximadamente, cuando el niño empieza a hablar, su mundo se amplía considerablemente, porque le permite evocar acciones pasadas o futuras. Es decir, anteriormente, el niño solo podía manifestar su situación presente a través de movimientos y algunas palabras o frases

(5) PIAGET, JEAN. Op. Cit., p. 27

aisladas, posteriormente utiliza sustantivos y verbos diferenciados y por último frases completas, permitiéndole un intercambio y una comunicación continua entre los individuos.

Es necesario aclarar que en este momento, el lenguaje sufre limitaciones análogas a los movimientos en el período sensorio-motor. Del mismo modo que el niño, al nacer refiere todos los acontecimientos a su propio cuerpo, así en esta etapa, refiere su conversación a su propio punto de vista, es decir no coordina su plática con la de otros niños.

Es muy difícil determinar el momento en el que aparece el pensamiento como tal, sin embargo el hecho de que el niño ya sea capaz de reconstruir situaciones sin necesidad de que estén presentes los objetos y/o personas, o bien que anticipe determinados acontecimientos hace evidente la aparición del pensamiento en el niño.

En el plano cognoscitivo tiene tres repercusiones principales:

Primera, permite mayor relación entre los individuos y el niño.

Segunda, aparece el pensamiento dicho.

Y tercera, estimula la formación del pensamiento intuitivo.

El pensamiento da un gran paso en el momento en que el niño, debido a que tiene más experiencias, intenta dar una explicación lógica a los fenómenos que ocurren. Solamente toma en cuenta algunas partes del acontecimiento y no logra ver el todo. Esto Piaget lo explica de la siguiente manera: si un niño ve una carrera de carritos, para él, el auto más

veloz es el que llegue primero a la meta, sin tomar en cuenta la distancia recorrida. O bien, si a un niño le dice que escoja entre dos vasos con agua; uno chico que está lleno y otro más grande no tan lleno, el escogerá el vaso chico porque tiene más agua, aunque haya visto que se vertió en ellos la misma cantidad de líquido.

El pensamiento intuitivo es en general, una: "...simple interiorización de las percepciones y los movimientos en forma de imágenes representativas y de experiencias mentales que prolongan por tanto los esquemas sensorio-motores sin coordinación propiamente racional." (6).

En suma, el pensamiento de la etapa preoperacional está limitado a la primacía de la percepción.

Es preciso decir que, de los dos a los siete años, se dan todas las transiciones entre dos formas extremas de pensamiento. La primera, es la del pensamiento por mera incorporación o asimilación, cuyo egocentrismo excluye por consiguiente toda objetividad. La segunda es la del pensamiento que se adapta a los demás y a la realidad, preparando así el pensamiento lógico.

"Hasta alrededor de los siete años, el niño sigue siendo prelógico y suple la lógica por el mecanismo de la intuición, simple interiorización de las percepciones y los movimientos en forma de imágenes representativas y de 'experiencias mentales', que prolongan por tanto los esquemas sensorio-motores sin coordinación propiamente racional." (7)

(6) Ibidem, p. 50

(7) Idem, p. 50

Para comprender mejor esta etapa, Piaget explica que si al niño se le presentan seis u ocho fichas azules, alineadas con pequeños intervalos de separación, y se le pide que encuentre otras tantas fichas rojas en un montón:

- Entre los 4 y 5 años, los niños construirán una hilera de fichas rojas exactamente de la misma longitud que la de las fichas azules, pero sin ocuparse del número de elementos, ni hacer corresponder una por una las fichas rojas y las azules. En este ejemplo se puede contemplar una forma primitiva de intuición, que consiste en valorar la cantidad sólo por el espacio ocupado, es decir, por las cualidades perceptivas globales de la colección tomada como modelo, sin preocuparse del análisis de las relaciones.

- Entre los 5 y los 7 años, en cambio, el niño pone una ficha roja delante de cada ficha azul y concluye la correspondencia de ambas colecciones en igualdad. Sin embargo al separar un poco las fichas de los extremos de la hilera de las rojas, de tal manera que no estén ya exactamente delante de las fichas azules, el niño, que ha visto que no se ha quitado ni añadido nada, afirma que las dos colecciones ya no son iguales y dice que la hilera más larga contiene más fichas.

En resumen, hay equivalencia mientras hay correspondencia visual, pero la igualdad no se conserva por correspondencia lógica. No hay operación racional alguna, sino simplemente intuición que está sometida por la percepción. Comparada con la lógica, la intuición es un equilibrio menos estable por

falta de reversibilidad, que es, la posibilidad permanente de regresar al punto de partida de la operación dada.

La principal actividad de los niños en esta edad es jugar. El juego constituye la forma de actividad inicial de casi toda tendencia, o por lo menos un ejercicio funcional de esa tendencia que lo activa al margen de su aprendizaje propiamente dicho y reacciona sobre éste reforzándolo.

3.- Período de las operaciones concretas.

Comienza hacia los siete u ocho años y termina a los doce aproximadamente.

Si bien es cierto que en la etapa preoperacional el pensamiento avanza a pasos agigantados, también es cierto que en esta edad se logra la formación de operaciones, aunque éstas se limiten a situaciones concretas.

Resulta necesario definir el elemento que permite al niño llegar a formar operaciones concretas: la reversibilidad, que es por lo tanto la característica principal de este período.

La reversibilidad es como ya se dijo, la capacidad que tiene el niño para analizar una situación desde el principio al fin y regresar al punto de partida, o bien para analizar un acontecimiento desde diferentes puntos de vista y volver al original.

La forma de pensamiento que esta nueva habilidad hace posible, es algo más organizado, toma en cuenta todas las

partes de una experiencia y las relaciona entre si como un todo organizado.

Ahora el niño puede clasificar y seriar, pero solo cuando tiene los objetos presentes para manipularlos, de ahí el nombre de operaciones concretas.

La clasificación consiste basicamente en concebir un objeto con relación a un conjunto más amplio. Es decir, al mismo tiempo que los objetos tienen diferencias, existen características que los vinculan por medio de su similitud.

La seriación es la relación que se establece entre varios objetos en el momento de hacer comparaciones y establecer un criterio de jerarquía, es decir, en ordenar los elementos según sus dimensiones crecientes o decrecientes.

La reversibilidad, por medio de la cual se hacen posibles estos avances en el pensamiento, presupone el concepto de permanencia.

Por ejemplo, ya en esta etapa el niño se dará cuenta que si el contenido de un vaso chico lleno de agua se vierte en un vaso más grande, sigue siendo la misma cantidad de líquido, aunque se vea menos lleno, es decir la cantidad permanece. Esto se debe a que ahora el niño piensa en la situación inicial; no puede haber más líquido porque no se ha aumentado nada.

Más tarde, el niño podrá realizar la misma operación con relación al peso y dimensión, y no será sino hasta el final de esta etapa, que obtendrá la capacidad para hacerlo con respecto al volumen de un objeto o líquido.

No obstante que surgen una variedad muy rica de operaciones en esta etapa, no se debe perder de vista que el niño aún se encuentra en la etapa concreta, es decir que el campo de acción del niño aún es limitado puesto que sólo actuará sobre los objetos y no sobre hipótesis o enunciados verbales. Sin embargo al realizar una serie de ejercicios presentados en forma concreta, el niño podrá ejercitar su pensamiento para poder llegar a otro modo de razonamiento con bases cada vez más firmes.

A manera de conclusión: El pensamiento del niño se convierte en lógico únicamente por la organización de sistemas de operaciones que obedecen a leyes de conjunto comunes.

El paso de la intuición a la lógica o a las operaciones matemáticas se efectúa en esta etapa por la construcción de agrupamientos y grupos, es decir que las nociones y relaciones no pueden construirse aisladamente, sino que son organizaciones de conjunto en las cuales todos los elementos son solidarios y se equilibran entre sí. Esta estructura propia de la asimilación mental de orden operatorio asegura un equilibrio superior que el de la asimilación intuitiva o egocéntrica, ya que la reversibilidad adquirida permite alcanzar un estado de coherencia y de no-contradicción desde el momento que se libera de su punto de vista inmediato para "agrupar" las relaciones.

4.- Período de las operaciones formales.

El cuarto y último período aparece cuando el sujeto cuenta con once o doce años.

Es característica general de esta etapa la formación de un nuevo tipo de razonamiento más adaptado, que se refiere no solamente a objetos y realidades directamente manipulables, sino también a hipótesis y a enunciados verbales, es decir proposiciones, de las que se pueden extraer consecuencias. Es así como las operaciones empiezan a ser transferidas del nivel de las acciones concretas, al nivel de las solas ideas, expresadas en algún tipo de lenguaje.

La reflexión operatoria obra sobre sus propios actos intelectuales. Esta nueva orientación del proceso cognoscitivo da por resultado una adaptación muy eficiente en la solución de diversos tipos de problemas, pues su movilidad es creciente así como la libertad y la destreza para manejar los más complejos sistemas de representaciones conceptuales.

Estos logros se consolidan hasta que el sujeto cuenta con 14 o 15 años aproximadamente.

Transmitir o Descubrir. Dos clases de Aprendizaje.

El desarrollo de la civilización tecnológica, masificada y despersonalizante, ha contribuido en gran parte a la negación casi total de la expresión y la creatividad. La vida moderna, tan rápida y mecanizada, ha disminuido estas capacidades. Existen programas educativos que se proponen, como objetivo, crear "hombres producto" para la sociedad, pero que pierden de vista al ser en su totalidad, en sus deseos y aptitudes.

En el primer supuesto, el de transmitir, se encuentran frecuentemente educadores, padres y maestros, para quienes lo importante es la cantidad de conocimientos que el niño "debe saber"; el maestro realiza el proceso de enseñanza mecánicamente, perdiendo iniciativa y el espíritu de descubrimiento, dentro de su labor docente. El alumno no necesita desarrollar una capacidad creativa para cumplir con lo que se le encomienda, únicamente debe memorizar el material y repetirlo al final del curso, de la manera más exacta posible.

Los métodos conducen a la memorización, inhibiendo la posibilidad de construir los propios conocimientos a través de la razón. El alumno se enfrenta a afirmaciones prefabricadas que deberá adquirir sin usar su pensamiento.

Con esta disciplina se adquiere un aprendizaje mecánico y una memorización anecdótica que matiza su desarrollo intelectual, de modo que para el niño se torna cotidiano

copiar patrones organizados de conocimiento, que no puede descomponer y adaptar a situaciones de aprendizaje diferentes o novedosos.

Si de lo que se trata es de un aprendizaje por descubrimiento, Piaget dice que la educación no debe formar un beneficiario pasivo de lo que sucede en su entorno, el aprendizaje viene a ser la modificación de la experiencia, resultado de las acciones que lleva a cabo el niño para transformar la naturaleza de su experiencia.

Enseñar no significa transmitir estructuras que sólo pueden ser asimiladas a nivel verbal. Enseñar significa crear situaciones donde las estructuras puedan ser descubiertas.

Liberar la curiosidad, permitir que las personas evolucionen según sus propios intereses, desatar el sentido de indagación, abrir todo a la pregunta y a la exploración, permitirle estar en contacto con múltiples y variadas oportunidades de acción directa sobre objetos.

El hecho de dejar más libre al niño para pensar y reflexionar sobre su aprendizaje, no quiere decir que el profesor pierda su importancia como docente, ya que el niño por sí mismo no es capaz de ser consciente del desarrollo de su proceso de aprendizaje.

El maestro debe convertirse en un investigador cotidiano del proceso enseñanza-aprendizaje y no en un transmisor de experiencias.

Piaget, al mencionar los objetivos que debe plantearse la escuela de hoy, y la educación en general, se refiere al

desarrollo del potencial del niño, potencial que está esperando crecer y desarrollarse, como base fundamental para el entendimiento lógico de las materias científicas tales como matemáticas, física, o biología y como requisito indispensable para la formación de la personalidad.

"Entender es inventar o reconstruir por invención, y no habrá más remedio que doblegarse a este tipo de necesidades si se pretende, de cara al futuro, modelar individuos capaces de producir o de crear y no tan solo de repetir." (8)

(8) SEFCHOVICH, GALIA. Hacia una pedagogía de la Creatividad.
p. 19

HABLANDO UN POCO ACERCA DE PROGRAMACION

Al enseñar a los niños un lenguaje de programación, es necesario explicarles que la computadora funciona a través de una serie de circuitos lógicos, los cuales interpretarán los datos siguiendo un orden rígido de sucesión.

Para que se ejecuten las órdenes, es necesario entablar una comunicación con la máquina a través de un lenguaje.

Las órdenes a efectuar no pueden ir expresadas en un lenguaje corriente. Es necesario un lenguaje de programación como BASIC, LOGO, PASCAL, FORTRAN, COBOL, etc.

Muchos lenguajes incluyen palabras en inglés, esto se debe a que fueron creados en países de habla inglesa.

BASIC significa Beginners All-purpose Symbolic Instruction Code, creado en 1965 por Kemeny y Kurtz en los E.U.

Es un sencillo lenguaje de uso general, el más difundido entre las microcomputadoras. Su principal vocabulario comprende una veintena de palabras en inglés y otros signos como el punto, la coma, el asterisco, etc.. El BASIC es fácil de aprender y de utilizar, se pueden escribir programas sencillos con menos de 10 instrucciones. Es el lenguaje ideal para programas de talla y complejidad media.

BASIC se recomienda como primer lenguaje de programación.

Ahora bien, la computadora no sabe efectuar más que un número muy limitado de operaciones elementales, es necesario

transcribir en una serie de instrucciones simples la tarea compleja que se quiere ejecutar. El conjunto de estas operaciones simples o instrucciones constituye el programa.

Antes de iniciar la programación propiamente dicha, es indispensable describir la serie de instrucciones elementales en que es necesario descomponerlo para lograr su resolución. Esta es una tarea puramente de análisis.

A continuación hay que expresar las instrucciones en un lenguaje de programación, que en este caso será BASIC.

Aprender a programar no es solamente aprender las instrucciones del lenguaje, sino que la parte más importante, y es ahí donde se basa este trabajo, radica en la búsqueda de la resolución de problemas, descomponiendo el problema en partes pensando en la forma lógica de solucionarlo.

P R O P U E S T A

Tomando en cuenta los fundamentos pedagógicos expuestos anteriormente y la idea de un aprendizaje por descubrimiento, se propone un Manual de Computación para la enseñanza de la programación del Lenguaje BASIC, en donde el niño sea quien programe a las computadoras, desarrolle sus propias ideas, adquiriendo así un sentido de dominio sobre un elemento de la tecnología más moderna y poderosa y a la vez le permita explorar un microcosmos rico en experiencias lógicas, propiciando el razonamiento mediante la necesidad de pensar paso por paso, buscando las posibilidades y realizando las combinaciones necesarias para la resolución de un problema determinado, estimulando así el desarrollo intelectual y creativo del niño.

Es importante mencionar que el manual está dirigido a niños que se encuentran en el transcurso de la etapa de las operaciones concretas, específicamente entre diez y doce años, esto se debe a que, la enseñanza de la programación del lenguaje Basic, requiere de ciertos elementos de maduración tanto físicos como mentales. Aspectos como coordinación motriz fina, clasificar, seriar, igualar, establecer correspondencia, etc.; los cuales se presentan en el momento en que el pensamiento puede deducir el punto de partida de una acción.

Con la adquisición de la "reversibilidad", como Piaget lo estipula en esta etapa, el niño tiene la posibilidad

permanente de reflexionar y regresar al punto de partida de la operación dada, y esto le permite al niño ordenar y relacionar la experiencia como un todo organizado.

No obstante que surgen una variedad muy rica de operaciones en esta etapa, es importante aclarar, que el niño aun se encuentra en la etapa concreta, es decir, que su campo de acción aun es limitado puesto que solo actuará sobre objetos y no sobre hipótesis o enunciados verbales. Sin embargo al realizar una serie de ejercicios presentados en forma concreta, el niño podrá ejercitar su pensamiento para llegar a otro modo de razonamiento con bases cada vez más firmes. Sobre este punto Piaget nos dice:

"Desde el punto de vista de la escuela, esto significa por una parte que hay que reconocer la existencia de una evaluación mental; que todo alimento intelectual no es bueno indiferentemente para todas las edades; que deben tenerse en cuenta los intereses y necesidades de cada período. Esto significa también, por otra parte, que el medio puede jugar un papel decisivo en el desarrollo del espíritu; que la evaluación de las etapas no está determinada de una vez para siempre en lo que se refiere a las edades y a los contenidos del pensamiento; que por tanto, los métodos sanos pueden aumentar el rendimiento de los alumnos e incluso acelerar su crecimiento espiritual sin perjudicar su solidez". (9)

Para poder llevar a cabo estas ideas, se elaboraron los objetivos principales, que a continuación se exponen, sirviendo de base para la elaboración del contenido de dicho manual.

(9) PIAGET, JEAN. Psicología y Pedagogía, p. 174

OBJETIVOS GENERALES

- + Difundir la computación en el ámbito educativo.
- + Conformar en el niño una mentalidad crítica y analítica frente a la computadora.
- + Elaborar el manual de acuerdo a la edad evolutiva del niño.
- + Utilizar a la computadora como una herramienta de aprendizaje, en donde el niño pueda desarrollar sus propias ideas.
- + Enseñar al niño a programar, utilizando los elementos fundamentales del lenguaje BASIC.
- + Conducir a los alumnos hacia una nueva forma de resolver problemas basada en el razonamiento lógico.
- + Estimular el desarrollo cognoscitivo y creativo del niño.

C O N T E N I D O

Unidad 1

Objetivo:

Conocerá los elementos básicos para manipular la microcomputadora visualizando la forma lógica de establecer los pasos necesarios para la resolución de problemas.

1.- INTRODUCCION A LA COMPUTACION

1.1 Funcionamiento de una computadora.

1.1.1 Entrada, Procesamiento y Salida.

1.2 Tipos de computadoras.

1.2.1 Partes principales de una computadora.

1.3 Teclado (como unidad de entrada).

1.3.1 Manejo del teclado.

1.4 Tipos de lenguajes de computación

1.5 Concepto de programa.

Unidad 2

Objetivo:

Aprenderá el formato de las proposiciones en BASIC, así como las palabras claves para la elaboración de un programa.

2.- INSTRUCCIONES PARA EMPEZAR

2.1 Estructura de una proposición.

2.1.1 Formato de una proposición.

2.2 Número de línea.

2.3 Instruccion PRINT.

2.3.1 Impresión de textos.

2.3.2 Impresion de números.

2.3.3 Uso del "punto y coma" y la "coma".

2.4 Instrucción END.

Unidad 3

Objetivo:

Diferenciará los tipos de variables que existen asignándoles un nombre de variable correcto.

3.- VARIABLES EN BASIC

- 3.1 Variables numéricas.
- 3.2 Variables alfanuméricas.
- 3.3 Instrucción REM
- 3.4 Instrucción LET.
 - 3.4.1 Concepto de asignación.

Unidad 4

Objetivo:

Aplicará las formas de lectura que se pueden utilizar en BASIC.

4.- INSTRUCCIONES

- 4.1 Instrucción IMPUT.
 - 4.1.1 Para variables numéricas.
 - 4.1.2 Para variables alfanuméricas.
- 4.2 Instrucción READ-DATA.
- 4.3 Diferencia entre READ-DATA e IMPUT.

Unidad 5

Objetivo:

Comprenderá los diferentes operadores aritméticos, la jerarquía y el uso de parentesis así como el concepto de igualdad, dentro de las operaciones aritméticas.

5.- OPERADORES ARITMETICOS

- 5.1 Operadores +, -, *, /, ^.
- 5.2 Jerarquía de los operadores.
- 5.3 Utilización de paréntesis.

Unidad 6**Objetivo:**

Aplicará las formas de transferencia de control tanto condicionadas (IF - THEN) como incondicional (GOTO), utilizando en la transferencia condicionada los diferentes signos de comparación usados en BASIC =, <, >, <=, >=, <>.

6.- TOMA DE DECISIONES**6.1 Instrucciones GOTO.****6.2 Instrucción IF - THEN.**

6.2.1 Operadores lógicos o de realción.

6.2.2. Uso de IF - THEN combinado con GOTO.

Unidad 7**Objetivo:**

Practicará el ciclo iterativo que forman las instrucciones FOR y NEXT, aumentando la utilidad de la computadora como instrumento.

7.- CICLOS**7.1. Instrucciones FOR - NEXT.****7.2 Diferencia entre GOTO, IF - THEN, FOR - NEXT.****Unidad 8****Objetivo:**

Elaborará sus propios programas, practicando todas las instrucciones del lenguaje BASIC.

8.- ANEXO.**8.1 Visualización de programas utilizando todas las instrucciones aprendidas.**

A continuación se presentara el manual de BASIC, llamado "Aprendiendo a Programar Computadoras".

Este manual es una guía que llevará paso por paso al niño, para que pueda aprender por sí mismo a programar computadoras por medio del lenguaje BASIC.

El material incluido solo le dará las primeras bases de programación, posteriormente estará preparado para aprender más acerca de éste lenguaje.

A través del manual, el niño encontrará la explicación de cada paso o instrucción, acompañada de ejemplos para que pueda comprender mejor.

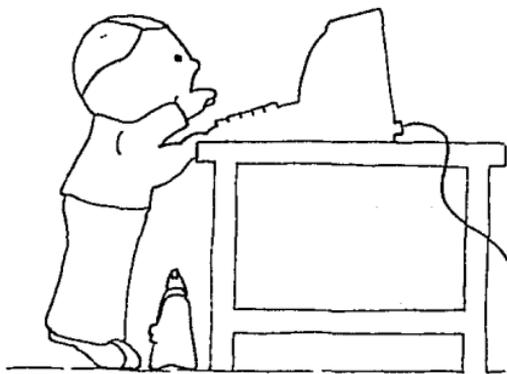
El primer capítulo es una introducción a la computación. Aquí se encontrará la manera como funcionan, de que partes constan y como se puede comunicar con ellas.

El capítulo 2 le introducirá en la manera de como estructurar un programa. Es decir, pensara y planeara los pasos a seguir dependiendo de lo que quiera hacer para que su computadora lo ejecute.

En los capítulos siguientes 4, 5, 6 y 7 aprendera a comunicarse con la computadora por medio de las instrucciones en BASIC. Poco a poco ira descubriendo y experimentando las muchas cosas interesantes que puede hacer con esta.

Al final del manual se encontrara un anexo de programas en donde se utilizan la mayoría de las instrucciones.

"PROPUESTA DE UN MANUAL DE BASIC"



APRENDIENDO A PROGRAMAR COMPUTADORAS

I N D I C E

Introducción	3
Unidad 1	4
Introducción a la computación	
Unidad 2	19
Proposición	
Número de línea	
PRINT	
END	
RUN	
LIST	
Unidad 3	37
Variables	
REM	
LET	
Unidad 4	47
INPUT	
READ-DATA	
Diferencia entre:	
READ-DATA e INPUT	
Unidad 5	58
Operadores aritméticos	
Uso de paréntesis	
Unidad 6	70
GOTO	
IF - THEN	
Unidad 7	84
FOR - NEXT	
Diferencia entre:	
GOTO, IF - THEN y FOR - NEXT	
Unidad 8	98
Anexo:	
Programas para practicar	
Recomendaciones para padres y maestros	103

I N T R O D U C C I O N

Este manual está pensado, escrito e ilustrado para niños como tú, que están interesados en conocer que son las computadoras, como funcionan y como programarlas.

¿Qué es programar una computadora? Programar es aprender a comunicarnos con ella a través de algún lenguaje entendible para la computadora.

En esta ocasión conoceremos el lenguaje BASIC, el cual fué desarrollado por John Kemeny y Thomas Kurtz durante los años 60's. Es uno de los más populares, muy fácil de aprender y simple de usar. Una vez que domines este lenguaje podrás ordenarle a la computadora las cosas que quieras que haga.

El material incluido a continuación, está acompañado de ejemplos que tu puedes practicar, pero te recomendamos que experimentes con otros programas creados por ti.

Es importante que consideres a la computadora como una herramienta dentro de tus actividades. Al igual que una pluma, debes llevarla de la mano por todo su recorrido. La computadora no posee ninguna capacidad para tomar decisiones por sí misma, puede hacer muchas cosas, siempre y cuando tu se lo indiques.

Todo lo que tienes que hacer debe estar especificado de modo preciso en las instrucciones.

Una de las cosas que es importante que siempre tengas en mente cuando quieras programar es:

- 1.- Pensar que es lo que quieres hacer.
- 2.- Como quieres que lo haga la computadora.
- 3.- Y cuando debe de hacerlo.

¡Ahora ya puedes continuar!

UNIDAD 1

Introducción a la Computación:

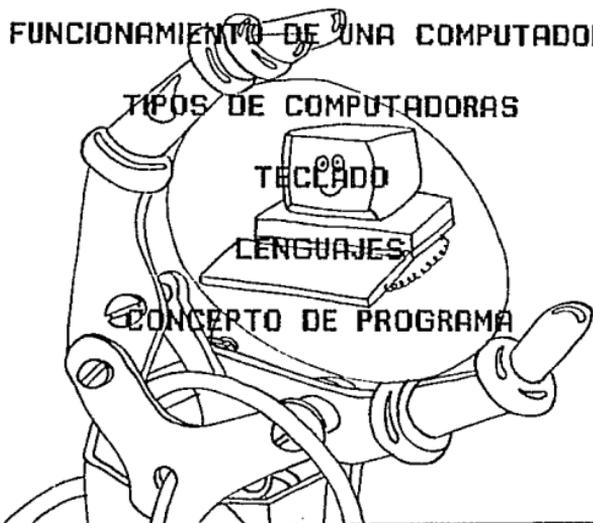
FUNCIONAMIENTO DE UNA COMPUTADORA

TIPOS DE COMPUTADORAS

TECLADO

LENGUAJES

CONCEPTO DE PROGRAMA



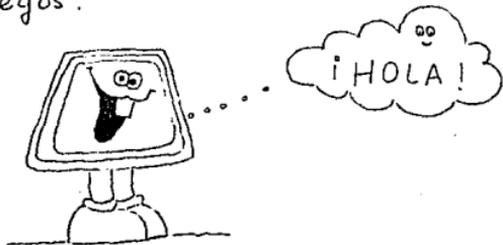
Funcionamiento de una Computadora³⁶

Seguramente has visto y oído hablar alguna vez de las computadoras. ¿Te has preguntado para qué sirven? ¿Cómo funcionan?

Bueno, aquí te hablaremos en general de lo que es una computadora, cómo funciona y cómo podemos trabajar con ella.

Una computadora es una máquina electrónica, como un radio o una t.v., que necesita de electricidad para trabajar.

Las computadoras sirven para hacer operaciones con mucha rapidez, pueden guardar grandes cantidades de información y encontrar datos cuando tú los necesitas; La podemos utilizar para jugar o hacer nuestros propios juegos.



Los barcos y aviones usan computadoras³⁷ para desempeñar bien su trabajo. Estas les ayudan para indicarles donde se encuentran, en que dirección pueden viajar, las condiciones climatológicas, etc.

También las podemos encontrar en tiendas, bancos, oficinas, hospitales, escuelas, fábricas ... ¡UUF! en muchos lugares.



Como ves, son muchas las cosas que puede hacer una computadora, pero también son muchas las que no puede hacer; por ejemplo: no puede tener frío o calor, o tener miedo o sentir algo. Además, la computadora no puede pensar, sólo recibe órdenes y las obedece.

Las computadoras son algo así como un lápiz. Fijate cuantas cosas puedes hacer con él:



HOLA, SOY
JUAN 

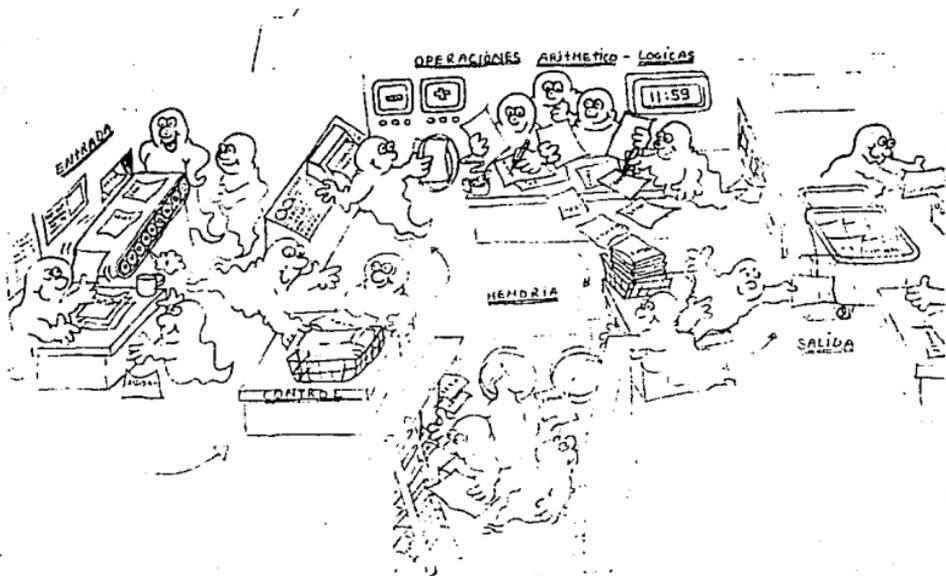
$$\frac{4+4 \times 5}{6} = $$

Pero el lápiz no puede hacer nada solo. Tú tienes que hacer las cosas, y lo usas nada más como una herramienta, es decir, como

algo que te ayuda a describir lo que tú quieras. Puedes escribir tus pensamientos, razonamientos e ideas, dibujos, ... etc.

Lo mismo pasa con las computadoras. Son herramientas que sirven para ayudarnos a resolver problemas o para hacer algún trabajo. Pero recuerda que solamente cumplen las órdenes que les damos.

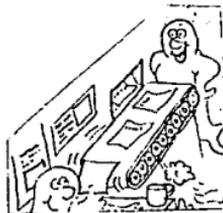
¿Sabes cómo funciona una computadora? Fíjate en estos dibujos:



Describe brevemente que hacen los señores de la caricatura:

Tu respuesta es correcta si se parece a ésta: en la sección de ENTRADA, el señor recibe una hoja con operaciones aritméticas, en seguida pasa a la sección CONTROL para que sea clasificada y enviada al departamento de Operaciones Aritmético-Lógicas, donde son ejecutadas las operaciones; Los resultados son guardados en un archivo y finalmente se entrega la respuesta en la sección SAUDA. Lo mismo hacen las computadoras. Reciben órdenes, las clasifican, las ejecutan, guardan los resultados y entregan las respuestas.

Para que te quede más claro cómo es que la computadora funciona, te lo mostraremos paso a paso.



Primero tenemos que darle las instrucciones. A esto le llamamos

ENTRADA



Antes de resolver las ⁴⁰ operaciones, la computadora las clasificará. Esto se llama:

CONTROL .



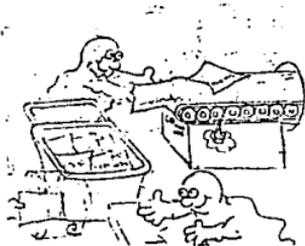
Después realiza las operaciones. A esto le diremos:

SOLUCIÓN DE OPERACIONES ARITMÉTICAS Y LÓGICAS .



Guarda las respuestas en un lugar especial, que se llama:

MEMORIA .



Entrega los resultados. A esto le diremos:

SALIDA .

Hay muchos tipos de computadoras, de las cuales se pueden clasificar en tres grupos: las grandes, las medianas y las pequeñas.

Las más grandes se llaman:

MACRO COMPUTADORAS

Las medianas se llaman:

MINI COMPUTADORAS

y las pequeñas:

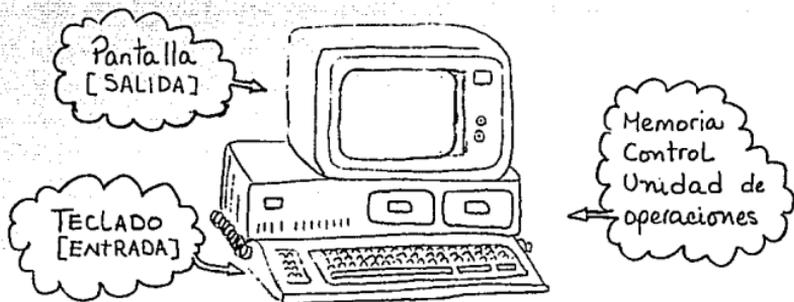
MICRO COMPUTADORAS

La diferencia entre ellas es que las macrocomputadoras y las minicomputadoras pueden atender a varias personas al mismo tiempo. Claro que las grandes atienden a más gente que las medianas. Por eso les dicen también:

COMPUTADORAS DE TIEMPO COMPARTIDO

Las microcomputadoras sólo pueden atender a una persona cada vez que se prenden.

OBSERVA cómo es una microcomputadora.



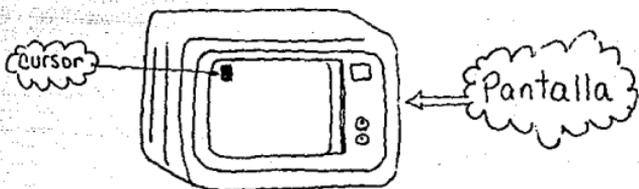
No todas las microcomputadoras son iguales. Hay muchas marcas diferentes, y cada fabricante les hace pequeñas modificaciones, pero todas deben tener las cinco partes principales:

ENTRADA, SALIDA, MEMORIA, CONTROL Y
UNIDAD DE OPERACIONES

Observa la tuya para que la conozcas bien antes de comenzar a trabajar con ella.

Si tienes alguna duda, consulta el manual.

PANTALLA: Todo lo que escribas a través del teclado aparecerá aquí.



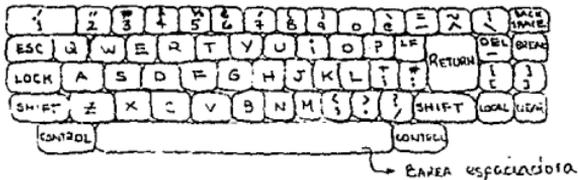
Cuando enciendes la pantalla aparece un cuadrado que se prende y se apaga llamado: CURSOR "■"

TECLADO:

Es la parte de la computadora que sirve para escribir. Todo lo que tú quieras escribir en la pantalla debe ser por medio del teclado.

Se parece al de las máquinas de escribir.

Fíjate en el siguiente dibujo:



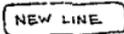
Este es un tipo de teclado. Acuérdate⁴⁴ que el tuyo puede ser diferente.

Para empezar a trabajar, tienes que localizar tres teclas, que son:

la de CORREGIR, la que da las MAYUSCULAS y la que da una **LÍNEA NUEVA**.

En el teclado de la figura, la tecla de CORREGIR es la . En otras máquinas la puedes encontrar como: . Esta la usas cuando te equivocas al escribir. Si la presionas, el cursor se regresa borrando lo que habías escrito. En tu computadora ¿cómo es la tecla de corregir? _____.

La tecla que da las MAYUSCULAS sirve también para escribir los signos que en algunas teclas aparecen arriba de los números o letras. En este ejemplo es . ¿En la tuya cuál es? _____.

La que sirve para saltar un renglón, o dar una línea nueva, es:  También se puede llamar:  o . ¿Cómo se llama en tu computadora?

La BARRA ESPACIADORA es igual en todos los teclados, te sirve para mover el cursor un lugar hacia adelante, sin escribir ningún carácter.

Ahora sí, como ya conoces tu computadora, ya puedes empezar a usarla.



PERO, ¿Cómo nos vamos a comunicar con ella?

Para comunicarnos con alguien, primero necesitamos conocer su idioma.

Los hombres tenemos muchos lenguajes para comunicarnos unos con otros.

¿Puedes enumerar algunos? _____

Lo mismo sucede con las computadoras. Son muchos los lenguajes que ellas pueden entender. Algunos son:

FORTRAN PASCAL

ALGOL LOGO

Aquí, vamos a estudiar:

BASIC

Como todos los lenguajes de las computadoras, el **BASIC** es muy sencillo. Tendrás que aprender unas cuantas palabras, que sirven como **INSTRUCCIONES** ¡Y sólo con escribirlas mediante el teclado la computadora entenderá la orden!

Todas las INSTRUCCIONES deben ir dentro de un:

PROGRAMA.

FIJATE EN EL SIGUIENTE EJEMPLO

Rosita quiere hacer un pastel, pero no sabe cómo, y le pide ayuda a su mamá.

- Mira Rosita, los ingredientes que necesitas son:



100 grs. de mantquilla.

150 grs. de azúcar.

5 huevos.

1 taza de leche

350 grs. de harina

1 cucharadita de polvo de hornear.

Y los pasos para hacerlo son:



- 1.- Batir la manteguilla y el azúcar
- 2.- Agregar los huevos y seguir batiendo
- 3.- Añadir la leche, harina y polvo de hornear.

Ya batido todo:

- 4.- Engrasar y enharinar el molde.
- 5.- Prender el horno.
- 6.- Vaciar la mezcla en el molde y meterlo al horno durante 25 minutos.

Esta receta consta de una serie de pasos que se ejecutan en orden para hacer el pastel. Pero también la receta tiene una ventaja: si alguna vez Rosita tiene más invitados tendrá que hacer un pastel de doble tamaño, y la receta le sirve en sólo poner el doble de cada ingrediente y seguir el mismo procedimiento.

Esto sucede porque la receta indica la MANERA como se hacen TODOS los pasteles de ese tipo, los grandes y los pequeños.

Para que a Rosita le salga bien el pastel, tiene que seguir todos los pasos EN ESE ORDEN. No puede hacer el paso 6 antes que el 2.

Ahora observa cómo se escriben estas instrucciones en BASIC:

DATOS

```

10 REM Este programa es para una
    receta.
20 LET A = 100
30 LET B = 150
  
```

DATOS

INSTRUCCIONES

```

40 LET C=5
50 LET D=1
60 LET E=350
70 LET F=1
80 PRINT "Batir"; A; "grs. de mantequilla"
90 PRINT "Agregar"; B; "grs. de azucar"
100 PRINT "Agregar"; C; "huevos"
110 PRINT "Añadir"; D; "tazas de leche"
120 PRINT "Batir"; E; "grs. de harina"
130 PRINT "Poner"; F; "cucharadas de polvo"
140 PRINT "de hornear"
150 PRINT "Engrasar y enharinar el molde"
160 PRINT "Atender el horno"
170 PRINT "Hornear 25 minutos"
180 END
  
```

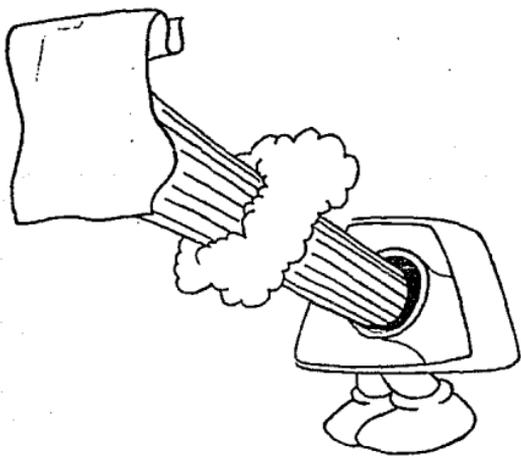
Este es un programa para hacer un pastel. Observalo con cuidado y trata de responder la siguiente pregunta: ¿Puedes explicar con tus palabras lo que es un programa?

Tu respuesta está bien si se parece a ésta: Un

programa es una serie de instrucciones numeradas y ordenadas.

Estas instrucciones le indicarán a la computadora lo que quieras que haga.

En la siguiente unidad, estudiarás todos los elementos de una instrucción en BASIC.



UNIDAD 2

Contenido:

Proposición
Número de línea
Print
End
Run
List



Como ya sabemos lo que es un PROGRAMA, ahora estudiaremos lo que es una **PROPOSICIÓN**.

Se llama proposición a cada una de las líneas que componen un programa.

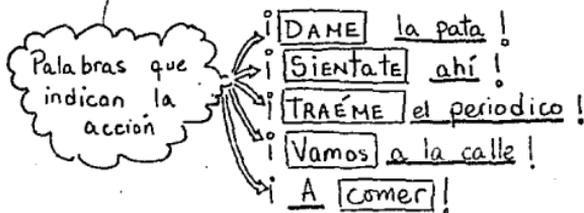
IO PRINT "HOLA"

Esta es una proposición.

Cada proposición es una orden que le damos a la computadora. Todas las órdenes tienen una palabra principal, que indica qué es lo que se debe hacer.

EJEMPLO :

Fíjate en las órdenes que le das a tu perro :



Las palabras que están encerradas en un cuadro indican la acción que quieres que haga el perro. Las palabras que están subrayadas sólo explican la orden.

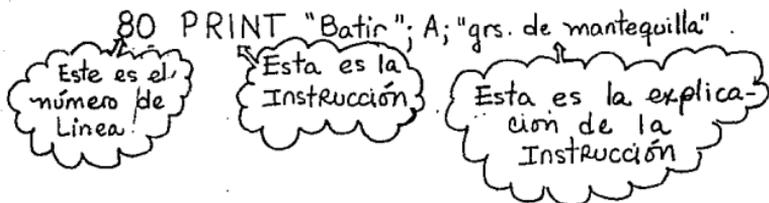
En las proposiciones siguientes di cuáles son las palabras principales? Enciérralas en

un cuadro 8

Vámos al campo
Llévate la canasta
Cómete la torta

trae la pelota
Pon el mantel allá
Lleva los refrescos al río

Lo mismo sucede en BASIC. Las órdenes que la computadora recibe en cada proposición tienen tres partes:



La palabra que da la orden en esta proposición es PRINT. Por eso la llamamos **INSTRUCCIÓN**

Esta instrucción indica a la computadora que ESCRIBA, y ella va a escribir todo lo que se encuentre entre comillas (").

Aquí siempre escribiremos las instrucciones usando letras **MAYÚSCULAS**.

Observa atentamente estas proposiciones. Subraya con rojo la palabra que corresponda a La INSTRUCCIÓN, con azul las que correspondan a La explicación de la orden y con verde el número de línea.

Ejemplo: 40 LET A=8

- a) 110 PRINT "El resultado es"; X
- b) 48 REM Este es un programa en BASIC
- c) 60 LET X=A+B
- d) 100 READ A,B,C
- e) 70 GOTO 230
- f) 10 LET L=0
- g) 5 PRINT 8+15
- h) 20 REM Vamos a calcular el área
- i) 140 DATA 4,8,12
- j) 65 END

En el siguiente programa aparecerán varias flechas marcando las distintas partes que constituyen una proposición.

ESCRIBE los nombres de cada una de ellas.

```

10 PRINT "La suma de 15+24 es"
20 LET S=15+24
30 PRINT S
40 END
  
```

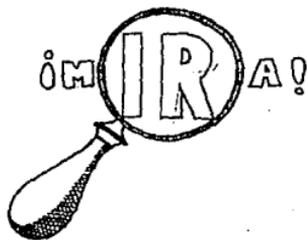
Diagrama de anotaciones: Una nube roja rodea la cadena de texto "La suma de 15+24 es" en la línea 10. Flechas azules indican: una desde la nube hacia la línea 20; una desde la línea 30 hacia la línea 40; y una desde la línea 40 hacia la línea 30. Líneas horizontales azules subrayan la línea 30 y la línea 40.

NÚMERO DE LÍNEA :

54

Para que la computadora sepa el orden en el que debe ejecutar el programa, es necesario numerar cada una de las proposiciones.

Además el número de línea sirve también para otras cosas: Corregir una proposición, añadir una nueva y borrar una que ya no sirve.



Cada proposición
lleva su número
de línea

Recordemos el programa que hizo Rosita para hacer el pastel:

```
10 REM Programa para una receta
20 LET A=100
30 LET B=150
40 LET C=5
50 LET D=1
60 LET E=350
70 LET F=1
80 PRINT "Batir"; A; "grs. de mantequilla"
90 PRINT "Agregar"; B; "grs. de azúcar"
100 PRINT "Agregar"; C; "huevos"
110 PRINT "Añadir"; D; "tazas de leche"
120 PRINT "Batir"; E; "grs. de harina"
130 PRINT "Poner"; F; "cucharaditas de
    polvo de hornear"
140 PRINT "Engrasar y enharinar el molde"
```

CONTINÚA 

150 PRINT "prender el horno"
 160 PRINT "hornear 25 minutos"
 170 END.

Este programa para pastel sale muy bien. Pero, un día Rosita Conoció a un gran cocinero que le dijo que hiciera unas modificaciones a su receta para que el pastel quedara mejor. Las indicaciones del cocinero fueron:

En vez de leche, añade 3 huevos y agrégale una cucharada de esencia de vainilla después del polvo de hornear

Rosita tiene que hacer tres modificaciones en su programa:

- 1.- Quitar las instrucciones que se refieran a la leche.
- 2.- Añadir una instrucción nueva para la esencia de vainilla
- 3.- Cambiar el número de huevos.



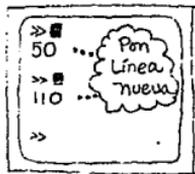
Primero tiene que localizar los números de las proposiciones que quiera modificar. ¿Le puedes ayudar?

Las que se refieren a la leche son la _____ y la _____.

Las que se refieren a los huevos son la _____ y la _____.

La que incluye la vainilla debe ir entre la _____ y la _____.

Rosita ya sabe que tiene que borrar las proposiciones 50 y 110. Para lograrlo, simplemente deberá escribir los números de línea y presionar la tecla que da una línea nueva. Así, desaparecerán las líneas que no quiera:



El programa ahora aparecerá así:

```
10 REM Programa para una receta
20 LET A=100
30 LET B=150
40 LET C=5
60 LET E=350
```

¿Desapareció la Instrucción 50!

```

70 LET F=1
80 PRINT "Batir"; A; "qs. de mantequilla"
90 PRINT "Agregar"; B; "grs. de azúcar"
100 PRINT "Agregar"; C; "huevos"
120 PRINT "Batir"; E; "grs. de harina"
130 PRINT "Poner"; F; "cucharadas de polvo
    de hornear"
140 PRINT "Engrasar y enharinar el molde"
150 PRINT "Hornear 25 minutos"
160 PRINT "Prender el horno"
170 END

```

¡TAMBIÉN
desapareció
la 110!

Ahora tiene que cambiar la proposición en la que está el número de huevos, que es la 40. Para conseguirlo, basta con que escriba la proposición nuevamente.

La proposición decía:

```
40 LET C=5
```

ahora debe decir:

```
40 LET C=8
```

o también:

```
40 LET C=5+3
```

Al terminar de corregir la proposición debe presionar la tecla de línea nueva. Fíjate como escribió la línea corregida con el mismo número que tenía la anterior.

Solo le falta añadir una nueva proposición entre la

130 y la 140. Afortunadamente la numeración va de 10 en 10, así que es fácil aumentar la línea. La puede poner con el número 135, o con cualquier otro que esté entre 130 y 140.

```
135 PRINT "Añadir una cucharada de
        esencia de vainilla"
```

Después, se presiona la tecla de la línea nueva.

¿Qué hubiera pasado si la numeración del programa original hubiera ido de 1 en 1? _____

Finalmente el programa queda así:

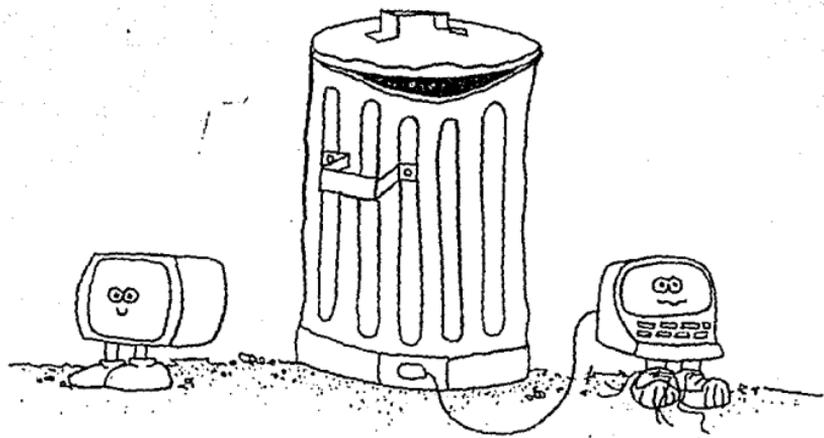
Esta línea
es diferente

Desaparecieron
dos instruc-
ciones

```
10 REM Programa para una receta
20 LET A = 100
30 LET B = 150
40 LET C = B
60 LET E = 350
70 LET F = 1
80 PRINT "Batir"; A; "grs. de mantequilla"
90 PRINT "Agregar"; B; "grs. de azúcar"
100 PRINT "Agregar"; C; "huevos"
120 PRINT "Batir"; E; "grs. de harina"
130 PRINT "Poner"; F; "cucharadas de polvo de
        hornear"
```

Esta instrucc-
ción no
estaba

135 PRINT "Añadir una cucharada de ⁵⁹
 esencia de vainilla"
140 PRINT "Engrasar y enharinar el molde"
150 PRINT "Prender el horno"
160 PRINT " horrear 25 minutos"
170 END



PRINT

OBSERVA cuidadosamente



ENTRADA

Es cuando tú introduces el programa a la computadora.

```
10 PRINT "Hola, soy la computadora"  
20 PRINT "Este es un programa en BASIC"  
30 END
```

SALIDA

Son los resultados de tu programa.

```
RUN  
Hola, soy la computadora  
Este es un programa en BASIC
```

PRINT

Esta instrucción es utilizada para imprimir algo. Si te fijas, hemos puesto unas comillas "" tanto al principio como al final de la frase. Esto le indica a la computadora que la escriba tal y como está.

END

Cuando tu programa esté terminado debes utilizar END, para indicarle a la computadora que has llegado al FIN. La instrucción END siempre debe ser la última de tu programa.

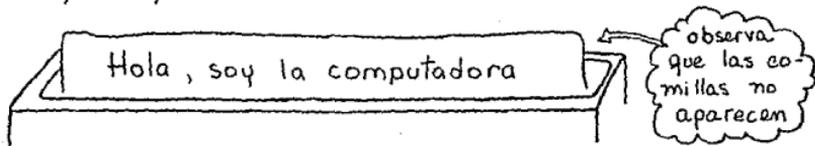
La instrucción END siempre debe ser la última de tu programa.



Esta instrucción va fuera del programa y no tiene número de línea. Se usa para que la computadora realice las instrucciones del programa. Es decir, al teclear la palabra RUN, leerá la primera línea, que en este caso es:

```
10 PRINT "Hola, soy la computadora"
```

Y como tú le estas indicando con el PRINT que imprima, ella escribirá:



Al haber ejecutado esta proposición, seguirá con la segunda, en este caso:

```
20 PRINT "Este es un programa en BASIC"
```

e imprimirá:

62

Este es un programa en BASIC

La tercera proposición es:

30 END

En este momento, la computadora se para porque le has indicado que tu programa ha terminado.

```
10 PRINT "Hola, soy la computadora."
20 PRINT "Este es un programa en BASIC."
30 END

>> RUN

Hola, soy la computadora.
Este es un programa en BASIC

*** END ***

>> █
```

Si quieres ver nuevamente tu programa, lo que necesitas hacer es teclear la palabra **LIST** y presionar la tecla que da nueva línea.

no lleva
número de
líneas porque
no forma parte
del programa.

```

10 PRINT "Hola, soy la computadora."
20 PRINT "Este es un programa en BASIC."
30 END
>> RUN
Hola, Soy la Computadora
Este es un programa en BASIC
*** END ***
>> LIST
10 PRINT "Hola, soy la computadora."
20 PRINT "Este es un programa en BASIC."
30 END
>> █

```

Si has hecho algún cambio en tu programa y quieres ver como ha quedado, usa la instrucción LIST



LIST → Listar o enlistar un programa.

Antes de hacer un programa nuevo, tienes que borrar de la memoria el anterior. Si no lo haces, ambos programas se mezclarán y obtendrás resultados incorrectos.

Hay varias maneras para borrar un programa. la forma que uses dependerá del tipo de computadora que tengas. En algunas se usa **S** **C** **R**, en otras , o también

¿En tu computadora cuál es? _____

PRINT :

64

La instrucción PRINT sirve para escribir cualquier letra. Para que la computadora sepa qué es lo que debe escribir, debes marcárselo con comillas.

Por ejemplo, si quieres escribir :

Voy a ir a Cuernavaca

La instrucción correcta es :

```
10 PRINT "Voy a ir a Cuernavaca"
```

y si lo que quieres escribir es :

La Suma de $4+8$ es 12

La instrucción debe ser :

```
40 PRINT "La suma de  $4+8$  es 12"
```

Ahora fijate en esta instrucción

```
30 PRINT "Mi perro se llama 'Tikal'"
```

¿Puedes adivinar cómo será la salida ?

Las proposiciones que llevan la instrucción `PRINT` también se pueden usar de otra manera.



Observa con atención el cuadro siguiente:

ENTRADA	SALIDA
<code>PRINT 5</code>	5
<code>PRINT 7+4</code>	11
<code>PRINT "7+4"</code>	7+4
<code>PRINT "Hola"</code>	Hola
<code>PRINT 1,2,3</code>	1 2 3
<code>PRINT 1;2;3</code>	123
<code>PRINT "X=";3</code>	X=3

Cuando quieras imprimir números no es necesario poner comillas.

Si quieres el resultado de una operación no debes usar comillas.

Pero si quieres que aparezca la operación, debes usarlas.

Con las palabras sucede lo mismo.

Con el símbolo `,` (coma), indica a la computadora que debe separar los números.

El símbolo `;` (punto y coma) indica a la computadora que no debe dejar espacios entre los elementos.

trata de crear un programa
como el que sigue a
continuación.

```
10 PRINT " Programa que hace suma "
```

```
20 PRINT " La suma es: "; 3+7 "
```

```
30 PRINT " El resultado es: ", 3+7 "
```

```
>> RUN
```

```
Programa que hace suma
```

```
La suma es: 3+7
```

```
El resultado es:      10
```

```
*** END ***
```

```
>> █
```

Con la Instrucción PRINT se pueden hacer dibujos como Este .

```

10 PRINT "
20 PRINT "
30 PRINT "
40 PRINT "
50 PRINT "
60 PRINT "
70 PRINT "
80 PRINT "
90 PRINT "
100 PRINT "
110 PRINT "
120 PRINT "
130 PRINT "
140 PRINT "
150 PRINT "
160 PRINT "
170 PRINT "
180 PRINT "
190 PRINT "
200 PRINT "
210 PRINT "
220 PRINT "
230 EN

```

EJERCICIOS

67

Relaciona la columna de la izquierda con la de la derecha, de acuerdo con la salida de la instrucción PRINT.

- | | |
|-------------------------------|--------------------------|
| a) PRINT 8; 4 | () 4 |
| b) PRINT "El resultado es", 4 | () 8-4 |
| c) PRINT "8-4" | () 8 4 |
| d) PRINT 8 | () 84 |
| e) PRINT "suma = 8" | () El resultado es 4 |
| f) PRINT 8-4 | () El resultado es 4 |
| g) PRINT "x="; 8 | () 8 |
| h) PRINT "x=", 8 | () x=8 |
| i) PRINT 8, 4 | () suma = 8 |
| j) PRINT "El resultado es"; 4 | () x= 8 |

c° Cuál instrucción debes poner para que el programa se ejecute? _____

c° Para que la computadora sepa que el programa ha terminado, la última instrucción debe ser: _____

c° Para listar un programa, cuál es la instrucción que se necesita? _____

UNIDAD 3

Variables NUMERICAS Y ALFANUMERICAS



**REM
LET**



VARIABLES

Una variable es algo que puede cambiar de valor.
Las variables pueden ser de dos tipos:

VARIABLES

numéricas (números)

alfabéticas (palabras)

Observa el resultado de una encuesta que se realizó en una escuela:

Estos son
Los valores
que toma
la variable
NOMBRE

NOMBRE	ESTATURA
Pedro	1.34
Rosa	1.29
Lupe	1.36
Juan	1.40
Pancho	1.35
María	1.28
Carlos	1.42

Estos son los
valores que to-
ma la variable
ESTATURA

En esta encuesta se están manejando dos variables: el NOMBRE y la ESTATURA. Todos los niños tienen nombres y estaturas diferentes.

La variable NOMBRE es alfabética porque los valores que pueden tomar son PALABRAS.

La variable ESTATURA es numérica porque los valores que toma son NÚMEROS.

La computadora también puede trabajar con estos dos tipos de variables. Pero tenemos que decirle con mucha claridad cuándo vamos a usar una variable, para que la guarde en un lugar especial de su memoria.

Las variables se pueden nombrar con las letras del alfabeto, desde la A hasta la Z.

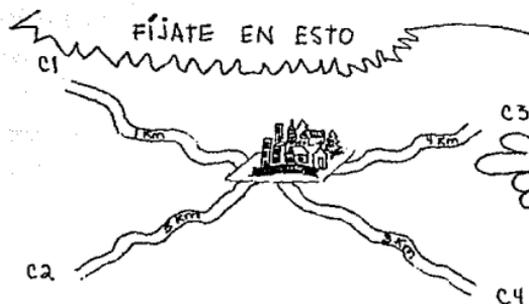
Imagínate que adentro de la memoria hay muchas cajas para guardar valores:

A	E	I	N	S	W
B	F	J	O	T	X
C	G	K	P	U	Y
D	H	L	Q	V	Z
	M	R			

Tú puedes guardar cualquier número en estas cajas. Por ejemplo, si quieres guardar un 5, lo puedes poner en la caja H o en cualquier otra.

También puedes usar como variables la combinación de letras con números de la siguiente forma: H1, A3, R9

La combinación de letras con números te pueden ⁷² servir para referirte a variables de la misma naturaleza pero con distinto valor.



tenemos diferentes caminos para llegar al centro de la ed. Cada uno va a ser representado por una variable.

En este caso, utilizaremos la letra "c" de la palabra

CAMINO, para que

nos acordemos que las variables se refieren a la longitud de éstos.

Para distinguir un camino de otro, usaremos la combinación de letras con números.

Así tendremos:

El camino 1 tiene 1 Km.

El camino 2 tiene 3 Km.

El camino 3 tiene 4 Km.

El camino 4 tiene 5 Km.

C1	1
----	---

C2	3
----	---

C3	4
----	---

C4	5
----	---

Estos son los valores de cada variable

Aquí usamos las combinaciones de las letras con los números.

Cuando quieras guardar en la cajita de las variables una palabra en lugar de un número, lo único que necesitas es indicarle a la computadora que vas a utilizar una palabra, y se hace por medio de este signo: $\$$

A\$ CASTILLO
 B\$ SI
 C\$ PERRO

NOTA:
 Las palabras no pueden ser mayores de ocho letras.

Aquí también puedes utilizar la combinación de letras con números.

Fíjate que es muy cómodo ponerle nombre a las variables, porque así no tenemos que escribirlas muchas veces, sobre todo cuando hacemos un programa muy largo.

POR EJEMPLO:

A 5
 B 8

C A+B

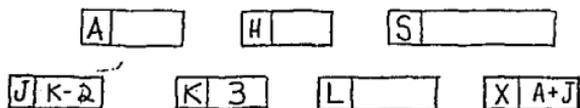
En esta caja se guarda la suma de $A+B$. Es lo mismo que guardar un 13 en la caja C.

73

Esto es, cuando ya tenemos algunos valores guardados en la memoria, podemos referirnos a ellos con la letra de la caja que los contiene. Es lo mismo que volver a poner el número.

EJERCICIOS

Observa estas variables:



- 1.- Escribe tu nombre en la caja que corresponda.
 - 2.- Pon un 7 en la caja H, un 4 en la L y un 6 en la A.
 - 3.- ¿Qué valor está almacenado en la caja J?
-

4.- ¿Cuál número quedó en la caja X? _____

REM

Esta instrucción la puedes usar para identificar tu programa o ponerle algún comentario. La forma correcta de emplearla es:

5 REM Este es un programa en BASIC

Puedes poner cualquier comentario que sirva para aclarar el programa.

10 REM Programa para sumar.
20 Read A,B
30

Estas instrucciones no aparecen en la corrida.

```
10 REM **Programa en BASIC **
20 PRINT "Ejemplo de REM"
30 REM Este programa hace una suma
40 PRINT "La suma de 3+8 es"; 3+8
50 END
>> RUN
Ejemplo de REM
La suma de 3+8 es 11
** END **
```

Cuando pongas la instrucción RUN, la proposición que lleva la instrucción REM será ignorada por la computadora.

```
10 REM *** tiempo ***
20 REM
30 REM Programa en BASIC
40 REM ---+---+---+---+
50 DATA 24, 60, 60
60 REM Horas, Minutos, Segundos
:
```

REM para
Lineas en blanco.

REM para
adornar.

REM para
claridad.

LET

76

La instrucción LET es usada para asignarle un valor a una variable.

La palabra LET es seguida por el nombre de una variable (ya sea numérica o alfabética); después, continúa el signo igual (=), y finalmente el valor que tome la variable.

VARIABLE

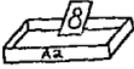
10 LET A=5
20 LET F\$="ROSA"
30 LET B=A+7

observa que si el valor es alfabético debes ponerlo entre comillas

deja un espacio

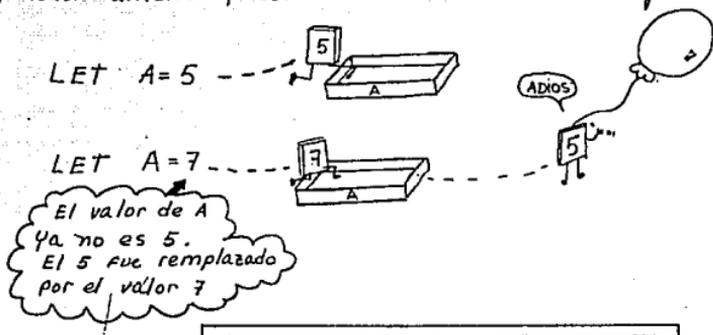


Para que te quede claro el concepto de VARIABLE y la función que desempeña la instrucción LET, imagínate a las variables como pequeñas cajas que guardan cada una, un valor.

LET A=5	 A=5
LET A2=8	 A2=8
LET B=12	 B=12

77

Cada caja puede contener solamente un valor en un determinado momento, pero tú puedes cambiarlo fácilmente. Esto se logra asignándole otro valor a la misma variable, la computadora borra automáticamente el valor anterior, colocando el nuevo en su lugar.



NOTA:

La palabra LET puede ser omitida.

40 LET X=24

40 X=24

Puedes usar las dos opciones.

```
10 REM *** INSTRUCCION LET ***
```

```
20 REM
```

```
30 PRINT "Programa en BASIC"
```

```
40 LET X=24
```

```
50 LET Y=13
```

```
60 LET Z=8
```

```
70 PRINT "El valor de X es: "; X
```

```
80 PRINT "El valor de Y es: "; Y
```

```
90 PRINT "El valor de Z es: "; Z
```

```
100 END
```

```
>> RUN
```

```
Programa en BASIC
```

```
El valor de X es:24
```

```
El valor de Y es:13
```

```
El valor de Z es: 8
```

```
>> █
```

UNIDAD 4



**INPUT
READ-DATA**

**Diferencia entre
READ-DATA e
INPUT**



INPUT

79

Tú puedes hacer un programa en BASIC que te pregunte por uno o varios datos:



```
10 PRINT "Ejemplo de INPUT"  
20 PRINT "Dame un número"  
30 INPUT X  
40 PRINT "El número que teclaste es: "; X  
50 END  
>> RUN  
Ejemplo de INPUT  
Dame un número  
?
```

Cuando la computadora se encuentre con la instrucción 30, escribirá en la pantalla un signo de interrogación para indicarte que está esperando un número. Ella podrá esperarte indefinidamente, hasta que escribas la respuesta y presiones la tecla de la línea nueva.

```
>> RUN  
Ejemplo de INPUT  
Dame un número  
? 325  
El número que teclaste es: 325
```

La instrucción INPUT puede necesitar más de un número:

```
30 INPUT X, Y, Z  =>  ? 325, 8, 2.5
```

También puede utilizar variables alfabéticas:

```
40 INPUT P1$, P2$  =>  ? Mercurio, Venus
```

Y tú puedes mezclar tanto variables numéricas como alfabéticas:

```
50 INPUT H$, A  =>  ? Independencia, 1810
```

La computadora pide, los datos separados por comas.

Por tanto tú debes apartarlos con una coma.

Para que no se te olvide el tipo de datos que debes escribir, cuando te pregunte la computadora, puedes usar la instrucción PRINT:

```
25 PRINT "Escribe una palabra y un número"  
30 INPUT T$, C
```



Que pasa si tecleas más datos de los que necesita la computadora:

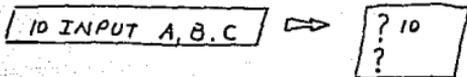
```
10 INPUT A, B, C  =>
```

```
? 10, 20, 30, 40  
ERROR  
?
```

¡CUIDADO!
la computadora sólo te pide 3 números.

Debes escribirlos correctamente.

Si la máquina necesita 3 números y tú sólo teclas-⁸¹te 1, ésta te volverá a poner el signo de Interrogación.



EJEMPLO:

```
10 PRINT "Programa que pregunta"  
20 PRINT "¿ cómo te llamas?"  
30 INPUT A$  
40 PRINT "¿ Cuántos años tienes?"  
50 INPUT N  
60 PRINT " me llamo: "; A$  
70 PRINT " tengo "; N; " años "  
80 END  
  
>> RUN  
Programa que pregunta  
¿ cómo te llamas?  
? Juan  
¿ Cuántos años tienes?  
? 11  
 me llamo: Juan  
 tengo 11 años.  
>> □
```

INPUT

82

¿Sabes cuántos años vas a tener en el año 2000?
Es muy fácil calcularlo si sigues estos pasos:

- 1.- Saber en qué año estamos
- 2.- Restar ese número a 2000
- 3.- Sumar el resultado con tu edad.

Vamos a calcularlo con un programa:



```
10 REM Programa que calcula la edad
20 REM que tenemos en el año 2000
30 PRINT "¿En qué año estamos?"
40 INPUT A
50 LET D=2000-A
60 REM En la caja D tenemos guardado
70 REM el número de años que faltan
80 REM para el año 2000
90 PRINT "¿Cuántos años tienes?"
100 INPUT E
110 LET F=D+E
120 REM La caja F contiene la edad
130 REM que tendrás en el año 2000
140 PRINT "¿Cómo te llamas?"
150 INPUT N$
160 PRINT N$;"tendrá";F;"años en
    el año 2000"
170 END
```

```

>> RUN
¿ cómo te llamas ?
? Pedro
¿ En qué año estamos ?
? 1984
¿ Cuántos años tienes ?
? 11
Pedro tendrá 27 años en el
año 2000

```

Estos son los datos que tú le dice a la computadora.

Este programa tiene la ventaja de que puede calcular también la edad que tendrán en el año 2000 tus amigos o tus familiares. Si lo vuelves a correr puedes calcular la edad que tendrá tu hermana pequeña:

```

>> RUN
¿ cómo te llamas ?
? Ana
¿ En qué año estamos ?
? 1983
¿ Cuántos años tienes ?
? 7
Ana tendrá 24 años en el
año 2000

```

EJERCICIO:

Haz un programa que calcule el año en el que naciste, pero que sirva también para calcular el año en el que nació cualquier persona. Acuérdate que debes saber el año en el que estamos y la edad de la persona.

DATA

84

Además de la instrucción INPUT, hay otra manera de proporcionarle datos a la computadora.

Esta nueva forma es mediante la Instrucción DATA.



DATA en latín significa DATOS

La instrucción DATA almacena los valores de las variables, números o palabras. Puede colocarse en cualquier lugar del programa, siempre y cuando sea antes del END.

```
20 DATA 10, 13 "SI", "NO", 4, 8, 6
```

Cada valor está separado del anterior por una coma.

Para que la computadora pueda guardar estos valores en la memoria, es necesario darle nombre a las variables.

Esto se hace mediante la instrucción READ :

Es decir, hay que abrir 7 cajitas para guardar esos valores.



```
30 READ X, Y, R1$, R2$, Z, A1, C2
```

los nombres también se separan por comas.

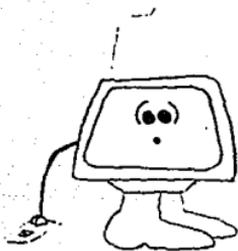
Si en un programa pones la instrucción DATA, tienes que poner la instrucción READ, y si usas READ, necesariamente tienes que poner DATA.

Observa cómo funciona esta pareja de instrucciones:

```
20 DATA 10, 13, "SI", "NO", 4, 8, 6
```

```
30 READ X, Y, R1$, R2$, Z, A1, C2
```

En este ejemplo la computadora lee la primera variable contenida en READ, y le asigna el primer valor que encuentra en DATA. Así tenemos que:



X → 10
Y → 13
R1\$ → SI
R2\$ → NO
Z → 4
A1 → 8
C2 → 6

Recuerda que cuando utilizas valores alfabéticos (letras) debes usar este tipo de variables.

Recuerda que cuando uses valores alfabéticos, debes poner \$. Observa que estos valores van entre comillas en la instrucción DATA.

Este símbolo no se encuentra en tu teclado; sirve para indicar que hay un espacio, y esto se logra poniendo .

ESPACIO 

```

10 PRINT "Número telefónico"
20 DATA "Mi", "teléfono"
30 READ M$, T$, E$, N1, N2, N3, N4
40 DATA "ES", 5, -34
50 PRINT M$; " "; T$; " "; E$; " ";
60 PRINT N1; " "; N2; " "; N3; " "; N4;
70 PRINT
80 DATA -25, -83
90 END
>> RUN
Número telefónico
Mi teléfono es:
5-34-25-83

```

En este programa hay una sola instrucción READ y tres instrucciones DATA. Esto se vale sólo cuando hay el mismo número de datos que de variables.

Los valores se van almacenando de acuerdo con la instrucción READ: La instrucción READ pregunta por los valores de M\$, T\$, E\$, N1, N2, N3, y N4.

Las instrucciones con DATA son:

```

20 DATA "Mi", "teléfono"
40 DATA "ES", 5, -34
70 DATA -25, -83

```

Por lo tanto, Los valores de las variables son:

M\$ = Mi	N2 = -34
T\$ = Teléfono	N3 = -25
E\$ = ES	N4 = -83
N1 = 5	

EJERCICIOS

87

Observa cuidadosamente este programa:

```
10 READ A, Z$
20 DATA 3, "pesos", "valen"
30 READ N$, F$, B
40 DATA "peras", 40
50 PRINT A; F$; N$; B; Z$
60 END
```

¿Cuáles son los valores que se almacenan en estas variables?

A = _____
Z\$ = _____
N\$ = _____
F\$ = _____
B = _____

¿Cuál es la salida del programa?

DIFERENCIA ENTRE DATA-READ E INPUT. ⁸⁸

READ - DATA {
READ \Rightarrow Lee los datos de la instrucción DATA
DATA \Rightarrow Esta instrucción almacena los datos.

INPUT \Rightarrow Es otra forma que puedes usar para darle datos a la computadora. Con esta instrucción no necesitas almacenar los datos dentro del programa como lo hace el READ-DATA

```
10 REM ** EJEMPLO DE DATA-READ **
20 PRINT "Programa en BASIC"
30 DATA 2,4,6,10
40 PRINT "Los Números que tenemos en DATA son : "
50 READ A,B,C,D
60 PRINT A,B,C,D
70 END
>> RUN
Programa en BASIC
Los Números que tenemos en DATA son :
2      4      6      10
>> ■
```

Los Datos están dentro del programa (Memoria), y siempre serán los mismos.

```
10 REM ** EJEMPLO DE INPUT **
20 PRINT "Programa en BASIC"
30 PRINT "Dame 4 Números"
40 INPUT A,B,C,D
50 PRINT "Los Números que teclaste son : "
60 PRINT A,B,C,D
70 END
>> RUN
Programa en BASIC
Dame 4 Números
? 2, 4, 8, 10
Los Números que tenemos en DATA son :
2      4      6      10
>> ■
```

El número de datos que necesita la computadora para ejecutar el programa, debe ser el mismo que el número de variables que tiene la instrucción INPUT.

Los datos te los pregunta la computadora y tú los puedes cambiar cada vez que corras el programa.

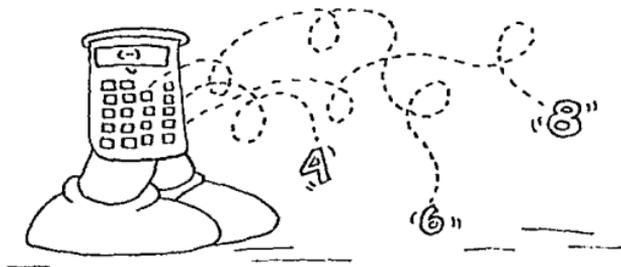
UNIDAD 5

OPERADORES ARITMETICOS

$+$, $-$, $*$, $/$

JERARQUIA DE LOS OPERADORES

PARENTESIS



OPERADORES ARITMÉTICOS

Hasta ahora sólo hemos visto cómo ordenarle a la computadora que haga sumas o restas. Pero ella sabe hacer más operaciones aritméticas:

Las operaciones aritméticas que puede hacer son:

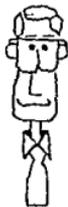
- 1.- sumas y restas
- 2.- multiplicaciones y divisiones
- 3.- Elevar un número a una potencia.

Para indicarle la operación que queremos que realice usaremos los OPERADORES ARITMÉTICOS

Estos operadores no son los mismos que tú conoces. Los que has usado hasta ahora son:

- + Para la suma
- Para la resta
- X Para la multiplicación
- ÷ Para la división

Ninguna para las potencias, porque las escribimos así: 2^3



Se llaman operadores porque, indican la operación que se va a realizar

En el lenguaje BASIC los operadores aritméticos son estos:

Estos dos operadores son los mismos que en aritmética.

+ para la suma. Ejemplo: $5+4$

- para la resta. Ejemplo: $5-4$

* para la multiplicación. Ejemplo: $5*4$

/ para la división. Ejemplo: $5/4$

^ para las potencias. Ejemplo: 5^4

Estos tres son diferentes:

$5*4$ es 5×4

$5/4$ es $5 \div 4$

4^5 es 5^4

Tú puedes usar uno o varios operadores en cada proposición. Si usas varios debes tener cuidado con dos cosas:

- 1.- Nunca pongas dos operadores juntos. La computadora no lo entenderá y te marcará un error:

$3* / 4$ ERROR

- 2.- Tampoco debes empezar a escribir una operación poniendo un operador. Si lo haces, también te marcará un error:

* 5 - 3 ERROR

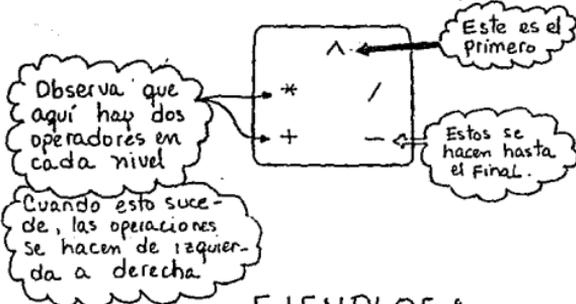
MIRA ESTOS EJEMPLOS:

Expresión aritmética	Expresión en BASIC
$3+4 \times 6 \div 2$	$3+4*6/2$
$16 \div 8 \times 5^2 - 3 + 2 * 3$	$16/8*5^2-3+2*3$
$6 \div 3 \times 4^2$	$6/3*4^2$

La computadora no puede realizar todas las operaciones al mismo tiempo. Por eso escoge las que va a hacer primero. El orden en que hace las operaciones es:

- 1.- Primero hace todas las potencias
- 2.- Después, hace las multiplicaciones y las divisiones.
- 3.- Finalmente, hace las sumas y las restas.

Entonces el orden de los operadores es:



EJEMPLOS:

$2 * 3 / 4 \Rightarrow$ Como tenemos dos operadores del mismo nivel se hacen de izquierda a derecha: primero se hace la multiplicación.

$$\boxed{2 * 3} / 4 = 6 / 4 \quad \text{y al final la división}$$

$$\boxed{6 / 4} = 1.5$$

$5 \wedge 2 + 7 * 4 - 6/3 \Rightarrow$ El operador que se ejecuta primero es: \wedge

$$\boxed{5 \wedge 2} + 7 * 4 - 6/3 = 25 + 7 * 4 - 6/3$$

Después $*$ y $/$, de izquierda a derecha:
 $25 + \boxed{7 * 4} - \boxed{6/3} = 25 + 28 - 2$

Y hasta el final $+$ y $-$, también de izquierda a derecha: $\boxed{25 + 28} - 2 = 53 - 2 = 51$

$4 + 6 \wedge 2 * 3/2 \Rightarrow$ Primero se hace la potencia:

$$4 + \boxed{6 \wedge 2} * 3/2 = 4 + 36 * 3/2$$

después $*$ y $/$ de izquierda a derecha

$$4 + \boxed{36 * 3} / 2 = 4 + \boxed{108 / 2} = 4 + 54$$

y al final la suma:

$$4 + 54 = 58$$

PARENTESIS

Si quieres alterar el orden en que se realizan las operaciones, utiliza paréntesis: la computadora hará primero las operaciones que estén encerradas en ellos:

Observa:

$$4 + 3 * 2 = 4 + 6 = 10$$

Primero se hace la multiplicación.

Si usamos paréntesis en la misma expresión, el resultado será diferente:

$$(4 + 3) * 2 = 7 * 2 = 14$$

Primero se efectúa la operación que está entre paréntesis.

Mira otro ejemplo:

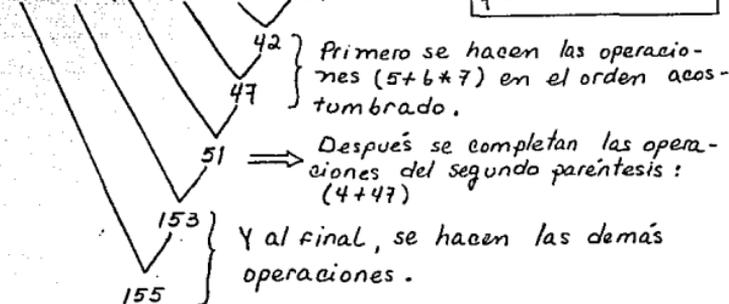
$$\begin{aligned} \text{a) } 2 + 3 * (4 \wedge 2) + 8 / 4 &= 2 + 3 * 16 + 8 / 4 \leftarrow \text{Primero } \wedge \\ &= 2 + 48 + 2 \leftarrow \text{Después } * \text{ y } / \\ &= 52 \leftarrow \text{Al Final } + \end{aligned}$$

$$\begin{aligned} \text{b) } 2 + 3 * 4 \wedge 2 + 8 / 4 &= 2 + 12 \wedge 2 + 8 / 4 \leftarrow \text{Primero } () \\ &= 2 + 144 + 8 / 4 \leftarrow \text{Después } \wedge \\ &= 2 + 144 + 2 \leftarrow \text{sigue } / \\ &= 148 \leftarrow \text{y al final } + \end{aligned}$$

También puedes usar tantos
paréntesis como necesites

NOTA:
Siempre se comienza
con las operaciones
de los paréntesis que
queden más al centro.

$$2 + 3 * (4 + (5 + 6 * 7)) = 155$$



EJERCICIOS:

Escribe una expresión en BASIC para cada una de las siguientes expresiones aritméticas:

a) $(4^2 + 13) \times 18$ _____

b) $(6 + (4 - 2)^2)$ _____

c) $\frac{4 + 3}{7}$ _____

d) $\frac{38}{6} \times 4 - 5^3$ _____

Relaciona la columna de la izquierda con la de la derecha:

$$1.- A = \frac{B \times h}{2}$$

$$2.- F = \frac{C+D}{2}$$

$$3.- F = \frac{C+D}{2}$$

$$4.- H = (5+4)^3$$

$$5.- H = 5+4^3$$

$$6.- X = 2 \times 3.14 \times R$$

$$7.- X = 2 \times 3.14 \times R^2$$

$$8.- A = \frac{B+h}{2}$$

$$(\quad) F = C+D/2$$

$$(\quad) H = (5+4)^3$$

$$(\quad) X = 2 \times 3.14 \times R^2$$

$$(\quad) A = (B+h)/2$$

$$(\quad) A = B \times h/2$$

$$(\quad) X = 2 \times 3.14 \times R$$

$$(\quad) H = 5+4^3$$

$$(\quad) F = (C+D)/2$$

Encierra en un cuadro las respuestas correctas:

1.- El resultado de la expresión $4 \times 5^2 / 20$ es:

20

5

1

2.- La expresión $(4+5)^2 \times 2 - 4$ tiene como resultado:

54

-162

158

3.- El resultado de $4+5^2 \times 2 - 4$ es:

50

-162

158

OBSERVA LOS SIGUIENTES PROGRAMAS

```

10 PRINT "SUMA"
20 PRINT "Cuántos centímetros salta la rana?"
30 INPUT A
40 PRINT "En el siguiente salto, ¿cuánto brinca?"
50 INPUT B
60 LET C = A + B
70 PRINT "La rana saltó "; C; "centímetros"
80 END
>> RUN
SUMA
Cuántos centímetros salta la rana?
? 50
En el siguiente salto, ¿cuánto brinca?
? 30
La rana saltó 80 centímetros
>> ■

```



```

1 REM  Combinación de operadores aritméticos
5 PRINT "Programa ABACO"
10 PRINT "Piensa en un número de tres cifras"
20 PRINT "¿Cuántas unidades tiene tu número?"
30 INPUT U
40 PRINT "¿Cuántas decenas tiene?"
50 INPUT D
60 PRINT "¿Y cuántas centenas?"
70 INPUT C
80 LET N=U*1 + D*10 + C*100
90 PRINT "El número que pensaste es: "; N
100 END
>> RUN

```

Programa ABACO

Piensa en un número de tres cifras

¿Cuántas unidades tiene tu número?

? 1

¿Cuántas decenas tiene?

? 7

¿Y cuántas centenas?

? 3

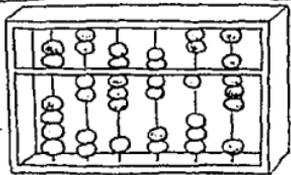
El número que pensaste es: 371

>> ■

Nosotros Los
Chinos fuimos los
que inventamos
el ábaco.



人中國創造
算盤





```

10 REM Programa para calcular la
15 REM edad promedio de una familia.
20 PRINT "¿Cuál es la edad del Padre?"
30 INPUT A
40 PRINT "¿Cuál es la edad de la Madre?"
50 INPUT B
60 PRINT "¿Cuántos años tiene el hijo
    mayor?"
70 INPUT C
80 PRINT "¿Cuántos años tiene el hijo
    menor?"
90 INPUT D
100 LET S = A + B + C + D
110 LET P = S / 4
120 PRINT "El promedio de edad"
130 PRINT "de la familia es"; P;
    años "
140 END
  
```

>> RUN

¿Cuál es la edad del Padre?
? 45

¿Cuál es la edad de la Madre?
? 40

¿Cuántos años tiene el hijo
mayor?

? 12

¿Cuántos años tiene el hijo
menor?

? 8

El promedio de edad
de la familia es 26.25
años

>> ■

Ejercicio:

Modifica este programa
para que calcule el promedio
de edad de tu salón de clases.

```

10 REM Programa para sumar fracciones
20 PRINT "SUMA DE FRACCIONES"
30 PRINT "Dame la primera Fracción"
40 INPUT "Numerador": N1
50 INPUT "Denominador": D1
60 PRINT "Dame la segunda Fracción"
70 INPUT "Numerador": N2
80 INPUT "Denominador": D2
90 REM El denominador de la suma
100 REM es el producto de los dos
110 REM denominadores.
120 LET DE = D1 * D2
130 LET NU = DE/D1 * N1 + DE/D2 * N2
140 PRINT
150 PRINT N1; " / "; D1; " + "; N2; " / "; D2; " = "; NU
160 PRINT "-----"
170 PRINT D1; " / "; D1; " + "; D2; " / "; D2; " = DE
180 END
>> RUN

```

SUMA DE FRACCIONES
 Dame la primera Fracción
 Numerador ? 2
 Denominador ? 3
 Dame la segunda Fracción
 Numerador ? 3
 Denominador ? 2

$$\frac{2}{3} + \frac{3}{2} = \frac{13}{6}$$

con estas instrucciones imprimimos la salida.

UNIDAD 6

TOMA DE DECISIONES



**GOTO
IF - THEN**



GO TO

102

Observa este programa :

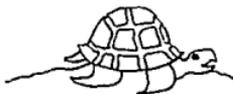
```
10 PRINT "Programa que sirve para calcular áreas"  
20 PRINT "Vamos a calcular el área de un cuadrado"  
30 PRINT "¿Cuanto mide el lado?"  
40 INPUT L  
50 LET A=L^2  
60 PRINT "El área del cuadrado es "; A  
70 END
```

¿Qué pasa cuando damos la instrucción RUN?

La computadora irá ejecutando cada proposición del programa siguiendo el orden que le marca el número de línea. Primero hará la proposición 10, y luego la 20 y así seguirá hasta llegar a la 70, que es la que le indica que el programa ha terminado:

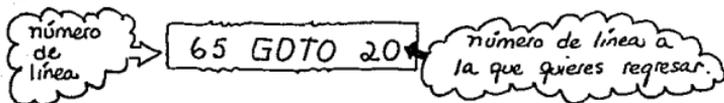
>>RUN

```
Programa que sirve para calcular áreas  
Vamos a calcular el área de un cuadrado  
¿Cuanto mide el lado?  
? 12  
El área del cuadrado es 144
```



Si quieres calcular el área de otro cuadrado, tendrás¹⁰³ que volver a correr el programa, dándole un nuevo valor al lado.

Existe una manera de indicarle a la computadora que regrese a la parte del programa que queremos repetir. Esto se logra con la instrucción GOTO.



Ahora el programa queda así:

>>LIST

```
10 PRINT "Programa que sirve para calcular áreas
20 PRINT "Vamos a calcular el área de un cuadrado"
30 PRINT "¿Cuánto mide el lado?"
40 INPUT L
50 LET A = L^2
60 PRINT "El área del cuadrado es"; A
65 GOTO 20
70 END
```

>>RUN

Programa que sirve para calcular áreas
Vamos a calcular el área de un cuadrado
¿Cuánto mide el lado?

? 12

El área del cuadrado es 144

Programa que sirve para calcular áreas
Vamos a calcular el área de un cuadrado
¿Cuánto mide el lado?

? 10

El área del cuadrado es 100

Programa que sirve para calcular áreas

Vamos a calcular el área de un cuadrado
¿Cuánto mide el lado?

? 8

El área del cuadrado es 64

Programa que sirve para calcular áreas

Vamos a calcular el área de un cuadrado
¿Cuánto mide el lado?

?

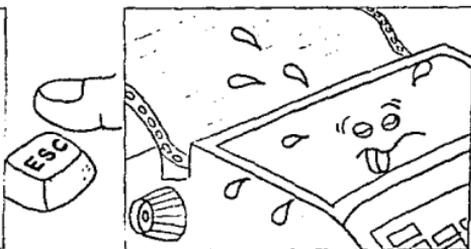
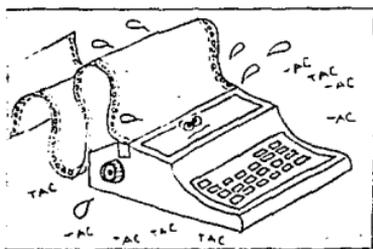
Al llegar a la línea 65, la computadora recibe la orden de regresar a la 20. Desde ahí volverá a ejecutar las instrucciones 20, 30, 40, 50 y 60. En la 65 encontrará nuevamente la orden de ir a la 20, y así se quedará formando un ciclo interminable, pues nunca va a llegar a la instrucción 70, que es la que le indica el final del programa.

Para detener este programa tienes que presionar la tecla  En algunas computadoras se llama



y en otras puede llamarse de otra manera.

¿Cómo se llama en tu computadora? _____



Con la Instrucción GOTO también podemos saltar instrucciones que no queremos que se ejecuten.

Observa:

```

10 PRINT "Hola"
20 GOTO 40
30 PRINT "Este lettero nunca se va a escribir"
40 END
>> RUN
Hola
** END **

```

no hizo la
instrucción

30

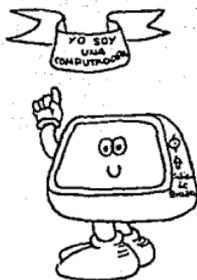
Con la Instrucción GOTO también podemos hacer que el programa no se ejecute en el orden acostumbrado:

```

10 REM Explicación de GOTO
20 PRINT "¿Cómo te llamas?"
30 INPUT N$
40 PRINT "¿Cuál es tu edad?"
50 INPUT E
60 GOTO 100
70 PRINT N$; "tiene"; E; "años. Va en la escuela";
80 PRINT E$; "y está aprendiendo BASIC"
90 GOTO 130
100 PRINT "¿En qué escuela estás?"
110 INPUT E$
120 GOTO 70
130 END

```

CONTINÚA →



>> RUN
 ¿Cómo te llamas?
 ? Diego
 ¿Cuál es tu edad?
 ? 11
 ¿En qué escuela estás?
 ? México
 Diego tiene 11 años. Va en la escuela
 México y está aprendiendo BASIC.
 ** END **

EJERCICIOS:

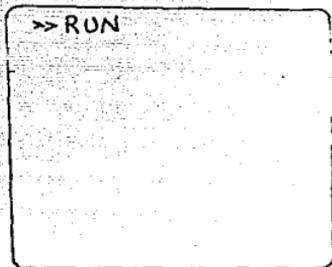
1.- observa atentamente este programa:

```
10 REM Ejemplo de GOTO
20 GOTO 60
30 LET F=N^2/4
40 PRINT "tienes";F;"días de no bañarte"
50 GOTO 90
60 PRINT "Dime un número par"
70 INPUT N
80 GOTO 30
90 END
```



Escribe los números de instrucción en el orden en que se ejecutan: _____

Escribe una corrida posible de este programa.



2.- Haz un programa que no tenga fin, que escriba tu nombre en tres columnas

3.- Observa esta corrida.

```
>> RUN
+++++
  +++++
++ +++
  +++++
+++ ++
  +++++
```

trata de escribir el programa.

IF - THEN

Estas dos palabras forman una sola instrucción. La instrucción **IF-THEN** sirve para poner condiciones

Por Ejemplo:

- SI Tomo el camino equivocado ENTONCES me pierdo.
- SI Pedro y María tienen los mismos padres ENTONCES son hermanos
- ⇒ SI llueve ENTONCES habrá buenas cosechas

Estas tres, son frases condicionales. La condición está escrita entre las palabras **SI - ENTONCES**

Observa que si la condición no es cierta, no se cumple la conclusión. La conclusión sólo se alcanza si la condición se cumple.

En el lenguaje BASIC también podemos poner este tipo de condiciones. Para eso se usa la instrucción **IF-THEN**

Para que puedas usar esta instrucción, debes escribirla de la siguiente forma:

IF Condición THEN Número de línea.

=

SI Se cumple la condición ENTONCES vete a la línea tal.

Las condiciones se ponen mediante las operaciones de **RELACION**.

OPERADORES DE RELACION		
≡	IGUAL A	$A \equiv 8$
➤	MAYOR QUE	$A \gt 5$
➤	MENOR QUE	$A \lt 20$
➤≡	MAYOR • IGUAL	$A \gt \equiv 5$
≡➤	MENOR • IGUAL	$A \lt \equiv 20$
↔	DIFERENTE A	$A \lt \gt B$

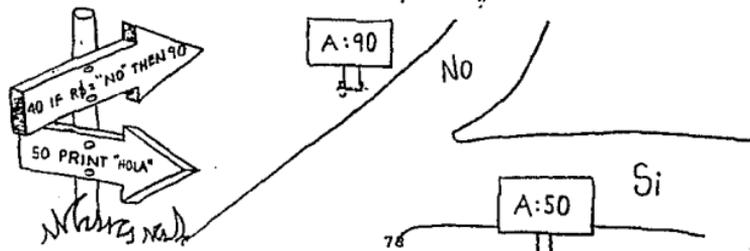
Ejemplo:

40 IF R\$ = "NO" THEN 90

operador de Relación.

si la condición se cumple, se irá a la línea 90 y continuará la ejecución a partir de ahí.

si la condición no se cumple, la ejecución continuará en la línea que sigue a la 40.



Observa este ejemplo :



```

10 PRINT "¿Quieres saber un secreto?"
20 PRINT "Si/No"
30 INPUT R$
40 IF R$ = "No" THEN 90
50 PRINT "El cometa HALLEY"
60 PRINT "pasará por la tierra"
70 PRINT "en 1986"
80 PRINT
90 PRINT "Hasta luego"
100 END
  
```

```

>> RUN
¿Quieres saber un secreto?
Si/No
? No
Hasta luego
  
```

Al correr
tu programa
pueden su-
ceder dos
cosas.

```

>> RUN
¿Quieres saber un secreto?
Si/No
? Si
El cometa HALLEY
pasará por la tierra
en 1986
Hasta luego
  
```

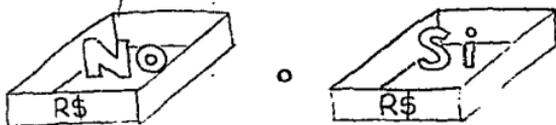
Vamos a ver cómo funciona este programa:

```
10 PRINT "¿Quieres saber un secreto?"
20 PRINT "Si/No."
30 INPUT R$
```

En esta parte del programa, lo que hacemos es imprimin

```
C'Quieres saber un secreto?
Si/No
?
```

Y hacer una pregunta, la respuesta se guardará en la caja R\$. Puede guardar dos valores distintos, dependiendo de la respuesta que des:



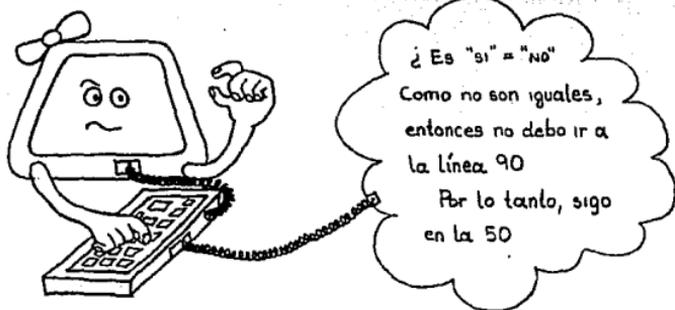
```
40 IF R$ = "NO" then 90
```

Si tu respuesta fue NO entonces la ejecución del programa continúa en la línea 90

```
90 PRINT "Hasta Luego"
100 END
```

Y la computadora nunca te dirá su secreto.

Si por el contrario, tu respuesta fue "SI", la computadora se pregunta:



```
50 PRINT "El cometa HALLEY"
60 PRINT "pasará por la tierra"
70 PRINT "en 1986"
80 PRINT
90 PRINT "Hasta luego"
100 END
```

¡Y te dirá su secreto!

Ahora mira este programa:

```

10 PRINT "¿Qué hora es?"
20 INPUT H
30 PRINT "A.M. o P.M. (A o P)"
40 INPUT Q$
50 IF Q$="A" THEN 90
60 IF H < 6 THEN 110
70 PRINT "Buenas Noches"
80 GOTO 120
90 PRINT "Buenos Días"
100 GOTO 120
110 PRINT "Buenas tardes"
120 END

```

Es importante que aprendas a combinar las instrucciones GOTO e IF-THEN. Corre este programa sin las instrucciones 80 y 100 para que notes la diferencia.

```

>> RUN
¿Qué hora es?
? B
A.M o P.M (A o P)
? A
Buenos Días
*** END ***

```

>> □

Observa que si las condiciones se cumplen, la ejecución del programa continúa en la 90 o en la 110, y si no se cumplen, continúa en la línea de abajo.

EJERCICIOS:

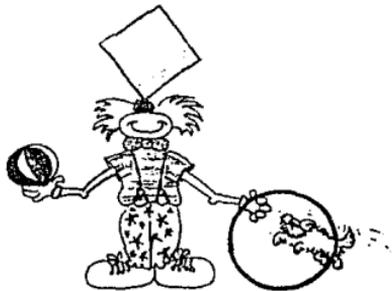
1.- Observa con atención este programa y escribe las dos corridas posibles :

```

10 PRINT "Te invito a una fiesta"
20 PRINT "¿Quieres ir? (Si o No)"
30 INPUT R$
40 IF R$="si" THEN 70
50 PRINT "Bueno, pues prende la t.v."
60 GOTO 90
70 PRINT "tienes que llevar un pastel"
80 PRINT "te espero el Jueves a las 6"
90 END

```

2.- Haz un programa que te sirva para calcular áreas de círculos y de cuadrados. Utiliza la instrucción condicional IF - THEN



UNIDAD 7

FOR - NEXT

Diferencia entre



GOTO



IF - THEN

FOR - NEXT

FOR-NEXT

Puedes utilizar esta instrucción para repetir cuantas veces quieras una parte de tu programa.

```

5 PRINT "Programa que cuenta del 1 al 3"
10 FOR N=1 to 3
20 PRINT "Hola! En este momento el valor de "
30 PRINT "N es : "; N
40 NEXT N
50 END
  
```

CICLO

Instrucciones que quieres repetir

La instrucción se forma de dos partes:

① FOR variable - valor inicial TO valor final

```
10 FOR N=1 TO 3
```

Instrucciones: ciclo que se repite hasta llegar al valor final de N.

```

20 PRINT "Hola! En este momento el valor de "
30 PRINT "N es : "; N
  
```

② NEXT variable

```
40 NEXT N
```

Las palabras FOR y NEXT siempre aparecen ¹¹⁷ en renglones separados.

Con la palabra FOR se inicia el ciclo. Esta parte de la instrucción le indica a la computadora cuántas veces debe ejecutar las instrucciones comprendidas entre FOR y NEXT.

Cuando escribimos:

```
FOR N=1 TO 3
```

Le estamos indicando que repita las instrucciones 3 veces: para $N=1$, para $N=2$ y para $N=3$

La palabra NEXT es la última del ciclo. Esta indica que ya se terminó una parte y que tiene que volver a empezar tomando el siguiente valor de N .

Veamos cómo funciona este programa:

```
10 FOR N=1 TO 3
20 PRINT "Hola! En este momento el valor de"
30 PRINT "N es"; N
40 NEXT N
50 END
```

Si nosotros fuésemos la computadora ejecutaríamos el programa de la siguiente forma:

1ª VUELTA

1.- Leeríamos :

FOR PARA la variable N, sus valores irán cambiando de 1 HASTA 3 por lo tanto $N = 1$

2.- Escribiríamos :

HOLA! En este momento el valor de N es: 1

3.- Regresaríamos por el siguiente valor de N (línea 10)

2ª VUELTA

1.- Ahora N vale 2

 $N = 2$

2.- Escribiríamos :

HOLA! En este momento el valor de N es: 2

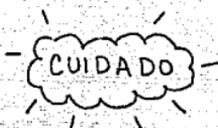
3.- Regresaríamos por el siguiente valor de N (línea 10)

1.- Ahora N vale 3

 $N = 3$

2.- Escribiríamos :

HOLA! En este momento el valor de N es: 3



Al regresar al siguiente valor de N nos damos cuenta de que el último valor de N fue 3. Por lo tanto continuamos con la instrucción que sigue al NEXT N .

En este caso la instrucción siguiente es:

40 END

y en este momento nuestro trabajo quedaría terminado.

```
5 PRINT "Programa que cuenta del 1 al 3"
10 FOR A=1 TO 3
20 PRINT "HOLA! En este momento el valor de ";
30 PRINT "N es : "; N
40 NEXT N
50 END
```

>> RUN

```
Programa que cuenta del 1 al 3
HOLA! En este momento el valor de N es : 1
HOLA! En este momento el valor de N es : 2
HOLA! En este momento el valor de N es : 3
```

* * * END * * *

¿Quieres saber la tabla de multiplicar del 17?
Vamos a calcularla con un programa.

```

10 PRINT "tabla del 17"
20 PRINT
30 FOR I=1 to 10
40 LET R = 17*I
50 PRINT "17 * "; I; " = "; R
60 NEXT I
70 END

```

>> RUN

tabla del 17

$17 * 1 = 17$
 $17 * 2 = 34$
 $17 * 3 = 51$
 $17 * 4 = 68$
 $17 * 5 = 85$
 $17 * 6 = 102$
 $17 * 7 = 119$
 $17 * 8 = 136$
 $17 * 9 = 153$
 $17 * 10 = 170$

Observa que el nombre de la variable puede ser cualquier letra

Los valores de I van cambiando desde 1 hasta 10



Si quieres calcular otra tabla, puedes modificar tu programa de la siguiente forma:

```

10 PRINT "Tablas de multiplicar"
20 PRINT
30 PRINT "¿Cuál tabla quieres?"
40 INPUT N
50 FOR J=1 TO 10
60 LET R=N*J
70 PRINT N; "*"; J; "="; R
80 NEXT J
90 END

```

>> RUN

Tablas de multiplicar
¿Cuál tabla quieres?

? 8

$$8 * 1 = 8$$

$$8 * 2 = 16$$

$$8 * 3 = 24$$

$$8 * 4 = 32$$

$$8 * 5 = 40$$

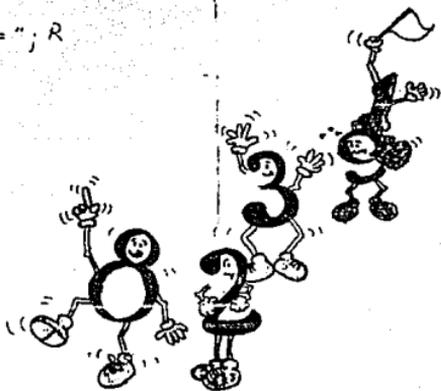
$$8 * 6 = 48$$

$$8 * 7 = 56$$

$$8 * 8 = 64$$

$$8 * 9 = 72$$

$$8 * 10 = 80$$



¿Cuál es tu promedio de Calificaciones? Cálzalo con este programa :

```

10 REM ** INSTRUCCION FOR-NEXT **
20 PRINT "Programa que saca promedio"
30 PRINT "Cuántas calificaciones quieres promediar"
40 INPUT N
50 REM --- Nes el número de calificaciones que vamos
60 REM a promediar
70 PRINT "Dame la primera calificación"
80 INPUT C
90 REM Se utiliza el FOR para repetir el ciclo, que te
100 REM pedirá desde la segunda calificación hasta
110 REM N (que va a ser la última calificación)
120 FOR I = 2 TO N
130 PRINT "Dame la siguiente calificación"
140 INPUT C1
150 LET C = C + C1
160 REM --- En la primera vuelta C = a la suma de la
170 REM primera calificación con la segunda cali-
180 REM ficación; En la segunda vuelta, cuando I = 3,
190 REM C = a la suma de la primera y segunda más
200 REM la tercera calificación, y así hasta llegar a
210 REM la suma de todas las calificaciones.
220 NEXT I
230 LET P = C/N
240 REM --- P (promedio) = C (la suma de todas las cali-
250 REM ficaciones) ENTIC (el número total de califi-
260 REM caciones).
270 PRINT "El promedio es : "; P
280 END

```

Continuación del programa :

123

>> RUN

Programa que saca promedio
¿ Cuántas calificaciones quieres promediar ?

? 3

Dame la primera calificación

? 10

Dame la siguiente calificación

? 8

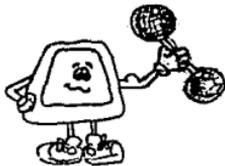
Dame la siguiente calificación

? 6

El promedio es : 8

>> □

EJERCICIOS:



1.- Encierra en un círculo las instrucciones que forman el ciclo en este programa :

```
10 FOR K=3 TO 6
20 LET S=10-K
30 LET D=10*K
40 LET P=K^2
50 PRINT K,S,D,P
60 NEXT K
70 END
```

¿ Cuántas veces se va a repetir? _____

Escribe la corrida del programa

- 2.- Elabora un programa que sirva para sacar la distancia promedio de todos los planetas al sol. Utiliza la instrucción FOR - NEXT
- 3.- Trata de hacer un dibujo utilizando la instrucción FOR - NEXT

A continuación te presentamos tres programas iguales, con el objeto de que compares la diferencia que existe entre: **IF THEN, GOTO Y FOR-NEXT**



```

10 REM EJEMPLO DE IF-THEN
20 PRINT "Programa que suma y multiplica
30 LET I=1
40 IF I>3 THEN 100
50 PRINT "      (+)  (*)"
60 PRINT "I="; I, I+1, I*1
70 PRINT
80 LET I=I+1
90 GOTO 40
100 END
>> RUN

```

Programa que suma y multiplica

	(+)	(*)
I=1	2	1
I=2	4	4
I=3	6	9

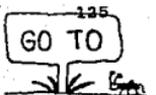


```

10 REM EJEMPLO DE GOTO
20 PRINT "Programa que suma y multiplica"
30 LET I=1
40 PRINT "      (+)  (*)"
50 PRINT "I="; I, I+1, I*I
60 PRINT
70 LET I=I+1
80 GOTO 50
90 END
>> RUN
Programa que suma y multiplica

```

	(+)	(*)
I= 1	2	1
I= 2	4	4
I= 3	6	9
I= 4	8	16
⋮		
I= 10	20	100
⋮		



```

10 REM EJEMPLO DE FOR-NEXT
20 PRINT "Programa que suma y multiplica"
30 PRINT "      (+)  (*)"
40 FOR I=1 TO 3
50 PRINT "I="; I, I+1, I*I
60 PRINT
70 NEXT I
80 END

```



>> RUN		
Programa que suma y multiplica		
	(+)	(*)
I = 1	2	1
I = 2	4	4
I = 3	6	9

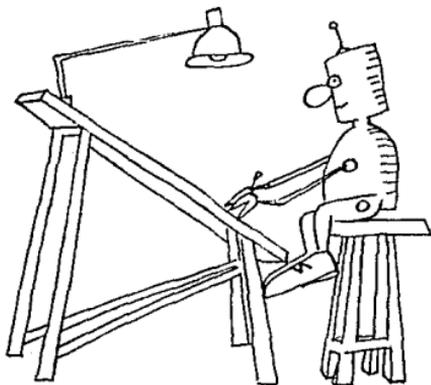
Con esto, queremos recordarte que un mismo programa lo puedes hacer de diferentes formas.

DIFERENCIA :

IF THEN	Instrucción condicional
GOTO	Ciclo sin Fin
FOR-NEXT	Ciclo con Fin

Puedes combinar estas tres instrucciones.

ANEXO



AHORA que ya has
aprendido a programar
en BASIC, estudia los
siguientes programas y
trata de hacer algunos;
utiliza las instrucciones
que ya conoces para
que adquieras mayor
experiencia.

```

5 REM * PROGRAMA ADIVINADOR *
10 PRINT "Piensa un número"
20 PRINT "¿Estás Listo? S./No."
30 INPUT R$
40 IF R$ = "No" THEN 20
50 PRINT "Al número que pensaste"
60 PRINT "súmalo 10"
70 PRINT "réstale 5"
80 PRINT "súmalo 3"
90 PRINT "quítale el número que pensaste"
100 PRINT "divídelo entre 2"
110 PRINT "multiplícalo por 7"
120 PRINT "súmalo 5"
130 LET N = (10 - 5 + 3) / 2 * 7 + 5
140 PRINT "agregale el número que pensaste"
150 PRINT "¿Cuál es el resultado?"
160 INPUT R
170 LET X = R - N
180 PRINT "El número que pensaste es "; X
190 END

```



```

>> RUN
Piensa un número
¿Estás Listo? S./No
? S.
Al número que pensaste
súmalo 10
réstale 5
súmalo 3
quítale el número que pensaste
divídelo entre 2
multiplícalo por 7
súmalo 5
agregale al número que pensaste
¿Cuál es el resultado?
? 53
El número que pensaste es : 20

```

```
10 REM Programa que convierte de Km a M
15 REM o de M a Km
20 PRINT "Quieres convertir: "
30 PRINT "De Km a M (Km)"
40 PRINT "De M a Km (M)"
50 PRINT "Dame las unidades (Km/M)"
60 INPUT N$
70 IF N$="Km" THEN 90
80 IF N$="M" THEN 140
90 PRINT "Cuál es el número"
100 INPUT K
110 LET M=K*1000
120 PRINT K;" Kilómetros son"; M;" metros"
130 GOTO 180
140 PRINT "e cuál es el número"
150 INPUT M
160 LET K=M/1000
170 PRINT M;" metros son"; K;" kilómetros"
180 PRINT "Quieres saber otra conversión (Si o No)"
190 INPUT R$
200 IF R$="si" THEN 20
210 PRINT "Adios"
230 END
```

>> RUN

Quieres convertir:

de Km a M (Km)

de M a Km (M)

Dame las unidades (Km/M)

? Km

¿Cuál es el número?

? 8

8 Kilómetros son 8000 metros

Quieres saber otra conversión (Si/No)

? No

Adios

>> ■



```

10 REM Ejemplo de GOTO e IF-THEN
20 PRINT "Cálculo de áreas"
30 PRINT "¿Quieres saber el área de"
40 PRINT "algunas de las siguientes"
45 PRINT "figuras?"
50 PRINT "Rectángulo, Cuadrado, triángulo"
55 PRINT "Si o No"
60 INPUT C$
70 IF C$ = "No" THEN 280
80 PRINT "Escribe el nombre de la figura"
90 INPUT F$
100 IF F$ = "Rectángulo" THEN 150
110 IF F$ = "Cuadrado" THEN 190
120 IF F$ = "triángulo" THEN 230
130 PRINT "ERROR"
140 GOTO 30
150 PRINT "Dame la base y la altura"
160 INPUT B, A
170 LET R = B * A
180 GOTO 260
190 PRINT "Cuanto mide un lado"
200 INPUT L
210 LET R = L * L
220 GOTO 260
230 PRINT "Dame la base y la altura"
240 INPUT B1, A1
250 LET R = B1 * A1 / 2
260 PRINT "El área es: "; R
270 GOTO 30
280 PRINT "Adios"
290 END

```



>> RUN

Cálculo de áreas

¿Quieres saber el área de algunas de las siguientes

Figuras?

Rectángulo, cuadrado, triángulo

Si o No

? Si:

Escribe el nombre de la Figura

? Cuadrado

¿Cuánto mide un lado?

? 6

El área es : 36

¿Quieres saber el área de algunas de las siguientes

Figuras?

Rectángulo, cuadrado, triángulo

Si o No

? No

Adiós

RECOMENDACIONES PARA PADRES Y MAESTROS

El presente material, diseñado para enseñar a programar computadoras por medio del lenguaje BASIC a niños entre diez y doce años, es un manual autodidacta, que lo llevará de la mano y lo introducirá en el campo de la programación.

Su principal sentido es estimular el desarrollo intelectual, propiciando el razonamiento mediante la necesidad de pensar paso por paso la resolución del problema planteado; así como proporcionarle los elementos del lenguaje BASIC que le permitan incorporarse a la tecnología más moderna y poderosa de la actualidad.

Ante este objetivo es necesario que el niño descubra y adquiera sus propias ideas aunque sean erróneas, es importante que él experimente y analice el porque del error y de manera individual vea sus limitaciones y busque la manera de superarlas, así evolucionará según sus intereses y de acuerdo con su etapa cognoscitiva correspondiente.

FIN

C O N C L U S I O N E S

A manera de conclusión es importante tomar en cuenta, como pedagogos que la computadora es una herramienta involucrada en el campo de la educación en México y en los últimos años, muchas escuelas han adquirido equipo de computación para ser usadas en el salón de clases, sin saber cómo utilizarlas y en que aplicarlas. De la misma forma a nivel personal, muchos padres de familia han obtenido equipo para enseñarles computación a sus hijos en sus casas, sin contar con asesoría educativa.

El auge que ha tenido la computadora y la influencia que puede tener ésta en el desarrollo lógico del niño y el desconocimiento por parte de educadores de esta poderosa herramienta, es importante para que los pedagogos, así como todos los profesionales que tengan que ver con la educación, se preparen en este nuevo campo y aporten nuevos estudios, para que el niño cada día se vea más beneficiado.

R E C O M E N D A C I O N E S

+ El maestro debe conocer perfectamente lo que este enseñando al niño. Se recomienda previamente probar las actividades y ejercicios que se proponen en el manual.

+ Debe conocer la herramienta de trabajo (computadora), manipularla y manejarla antes de comenzar. (Para posibles problemas que se presenten).

+ Tener en cuenta las particularidades individuales de cada alumno. Se recomienda leer a Jean Piaget y su teoría de desarrollo cognoscitivo del niño, para no exigir más de lo que él pueda dar.

+ Este manual pretende que el niño exprese su creatividad y no la repetición del ejemplo del maestro.

+ Se pretende que el aprendizaje de BASIC sea teórico-práctico.

+ Dejar que el niño descubra por sí mismo los pasos necesarios que requiere la computadora para llevar a cabo la resolución del problema planteado.

+ Dejar que el niño explore e invente sus programas, lo importante es que entienda el proceso y lo sepa llevar a otras situaciones similares.

+ Dejar que cada niño descubra sus errores y trate de aprender de ellos.

B I B L I O G R A F I A

- AIKEN, H. et al. Perspectivas de la revolución de las computadoras. Madrid, Alianza editorial, 1975. 695 p.
- ALBRECHT, ROBERT. et al. Lenguaje BASIC. Aprenda con rapidez y facilidad en esta serie de textos programados. México, Limusa, 1982. 284 p.
- COUFFIGNAL, LOUIS et al. La cibernética en la enseñanza. Tr. por Eli de Gortari. México, Grijalvo, 1968. 218 p. (colección Diana).
- FARINA, MARIO. Diagramas de flujo. México, Diana, 1984. 142 p.
- FRANK HELMAR, G. y B. S. MEDER. Introducción a la pedagogía cibernética. Tr. por Julio Balderrama. Buenos Aires, Troquel, 1976. 171 p.
- FUCHS WALTER, ROBERT. El libro de los nuevos métodos de enseñanza. Barcelona, Omega, 1973. 349 p.
- HINGUE, FRANCOIS. La enseñanza programada: Hacia una pedagogía cibernética. Buenos Aires, Kapelusz, 1969. 172 p.
- KEMENY, JOHN y KURTZ. Programación BASIC. México, Limusa, 1982. 284 p.
- LANDA. Cibernética y Aprendizaje. Tr. por Juan Judgloben. Buenos Aires, Paidós, 1977. 234 p.
- LINKER, JERRY. Diseño de material visual didáctico. Teoría: composición: ejecución. México, Centro Regional de ayuda técnica, 1971. 43 p.
- LIVAS GONZALEZ, IRENE. Análisis e Interpretación de los resultados de la evaluación educativa. México, Trillas, 1980. 151 p. (colección ANUIES).
- PAPERT, SEYMOUR. Mindstorms: Children, Computers and Powerful Ideas. Nueva York, Basic Books, 1980. 189 p.
- PIAGET, JEAN. La Formación del símbolo en el niño. Tr. por José Gutierrez. México, F.C.E., 1961. 401 p.

- PIAGET, JEAN. El mecanismo del desarrollo mental. Tr. por Juan A. Delval. Madrid, Editora Nacional, 1976. 177 p. (Colección Psicología y Educación).
- _____. Psicología y Educación. México, SEP/ARIEL, 1981. 208 p.
- _____. Seis estudios de Psicología. Madrid, Seix Barral, 1982. 199 p.
- RICHMOND, P.G. Introducción a Piaget. Madrid, Fundamentos, 1980. 152 p.
- ROGERS, CARLOS. Libertad y creatividad en la educación. Buenos Aires, Paidós, 1983. 256 p.
- ROSSI, N. La Revolución Cibernética. México, F.C.E., 1979.
- SEFCHOVICH, GALIA Y GILDA WAISBURD. Hacia una pedagogía de la creatividad. Expresión plástica. México, Trillas, 1985. 115 p.

R E V I S T A S

- ACHON, CEMELI y UTGES. "Educación Informática y Ordenadores en la EGB". En Cuadernos Pedagógicos, Nº 107, Año IX, Noviembre de 1983:11-14.
- DROSCHA, HELMUT. "Computadora y Escuela. Posibilidades y Limitantes". En Educación. Vol. 4, 1971:49
- EISELE, JAMES. "Classroom use of Microcomputers Educational Technology". 19(10), Oct. 1979:13-15
- ELLIS, BARRI. "Are Teachers ready for the computer age?". Education Canada, 18(1), 1978:4-8
- HILLEBRAND. "Aspectos fundamentales de la enseñanza programada". En Educación, Vol. 4, pag. 84-85.
- LABRODA, JAVIER. "Postmodernidad y educación: el microordenador como herramienta metodológica". En Cuadernos Pedagógicos, Nº 107, Año IX, Noviembre de 1983:15-21
- LANZILLOTA, MATILDE. "Un pequeño paso concreto. La Cibernética en la enseñanza primaria". En Educación y Cultura, (1), Ene-Mar. 1977:227-286.

PERFILES EDUCATIVOS. "Semiología y Educación". Nº 9, CISE, 1980. UNAM.

RIBERA, PERE y KIRCHNER, XAVIER. "Una experiencia de introducción de la informática en la enseñanza". En Cuadernos Pedagógicos, Nº 107, Año IX, Noviembre de 1983:15-21.

O T R A S P U B L I C A C I O N E S

DEL RIO G., SARA LUZ. Técnica de solución de problemas. (tesina). México, U.N.A.M., 1988.

RIEGO GAONA, M. ALEJANDRA. La computadora: Un juguete didáctico usado en la enseñanza. (tesina). México, U.N.A.M., 1986.

SIMPOSIO INTERNACIONAL La computación en la educación Infantil. (memoria). 24-28 de Octubre de 1984. Programa Universitario de Computo. México.