

5<sup>2</sup> Ejemplar

**UNIVERSIDAD AUTONOMA DE GUADALAJARA**  
INCORPORADA A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



TESIS CON  
FALLA DE ORIGEN

“USO Y APLICACIONES DEL SISTEMA  
HP-41C, V, X, EN INGENIERIA”

**TESIS PROFESIONAL**

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

JAIME ALBERTO DAVILA MUNGUA

GUADALAJARA JALISCO. 1986



## **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.

Deseo agradecer a todas aquellas personas que directa o indirectamente me han brindado su apoyo durante mi vida de estudiante y ahora en mi desarrollo profesional. Particularmente deseo expresar mi cariño y gratitud a todos y cada uno de mis hermanos y sus respectivas familias.

Deseo agradecer especialmente al Ing. José Antonio Valencia G., por todas las incontables ocasiones en que desinteresadamente me hizo partícipe de sus conocimientos; quiero expresar que como profesionista y como amigo le tengo en la escala de más alto valor.

Deseo agradecer también al Ing. Juan Mauro Cárdenas Madero, por la amistad incondicional y el apoyo que me ha profesado a lo largo de todos estos años.

Por último, deseo hacer patente mi reconocimiento al profesor Ing. Carlos Cortés Buenrostro, M.E. por tan atinados comentarios que hizo durante la ejecución y la corrección de la presente tesis.

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

En ingeniería se presentan un sinnúmero de cálculos y procedimientos que sirven para definir dimensiones, características, tolerancias, etc. de: circuitos, elementos de construcción, diseño de estructuras, trazo de perfiles y otros, que pueden resultar tediosos por repetir un determinado número de operaciones y decisiones o bloques de ellas que, primeramente, ocasionan cansancio en la persona que los está realizando por su propia naturaleza repetitiva, segundo, involucra un alto grado de error producido por el factor humano y tercero y muy importante, consumen una gran cantidad de tiempo que se puede emplear en otras actividades como por ejemplo, evaluar los resultados obtenidos.

En la actualidad y gracias al desarrollo que se ha alcanzado en la electrónica, enfocada al área de computación, se cuenta con una herramienta de cálculo muy poderosa, de fácil uso, portátil, de bajo consumo de energía, sencilla programación y otras características (que son particulares de cada fabricante), que es la CALCULADORA PROGRAMABLE DE BOLSILLO.

Cuando hablamos de máquinas programables de bolsillo nos referimos generalmente a calculadoras o minicomputadoras en las cuales se ha logrado un gran avance tecnológico en los últimos años. El principal de estos avances es la aplicación de la integración a gran escala -- (Large Scale Integrating, LSI) que permite obtener un -- gran número de circuitos en un espacio muy reducido, resultando de esta manera máquinas totalmente portátiles.

Otra de las ventajas de la LSI es que con esta técnica de integración, se han logrado hacer circuitos de una demanda de potencia muy baja, siendo esta característica una de las principales causas que hicieron posible que se pudieran implementar las calculadoras con lo que se ha denominado "Memoria Constante".

Paralelo a estos avances, en terreno de las pantallas (displays) también se han hecho grandes logros con la ayuda del cristal líquido de cuarzo, obteniendo como resultado la elaboración de displays alfanuméricos de excelente definición y muy bajo consumo de energía.

Debido a todas las características antes mencionadas, este tipo de calculadoras se han convertido en una práctica herramienta de cálculo para cualquier profesionalista de las ciencias exactas y en especial para el ingeniero. Como ejemplos de estas máquinas podemos mencionar: la TRS-80 versión portátil de la firma Radio Shack, la calculadora modelo TI-59 de Texas Instruments y la HP-41C de Hewlett-Packard; siendo esta última en la que se desarrollarán los programas de la presente Tesis.

Con estas máquinas se ejecutan una gran cantidad de operaciones y decisiones a una velocidad mucho mayor que si se hiciera manualmente, dando por resultado un ahorro considerable de tiempo por cada cálculo que se efectúe, y también eliminan en su mayor parte los errores que pudieran presentarse por el factor de error en el hombre al desarrollarlos. Estas últimas características representan una gran utilidad al ingeniero, el cual debe tomar conciencia de la importancia y las ventajas que le proporcionan en el ejercicio de su profesión, el uso de -

uno de estos aparatos; ya que, aunque parezca increíble, existen hoy en día muchas personas que ignoran la gran cantidad de aplicaciones en las que se puede emplear una calculadora programable, siendo esto causado muchas veces por la falta de información acerca del tema.

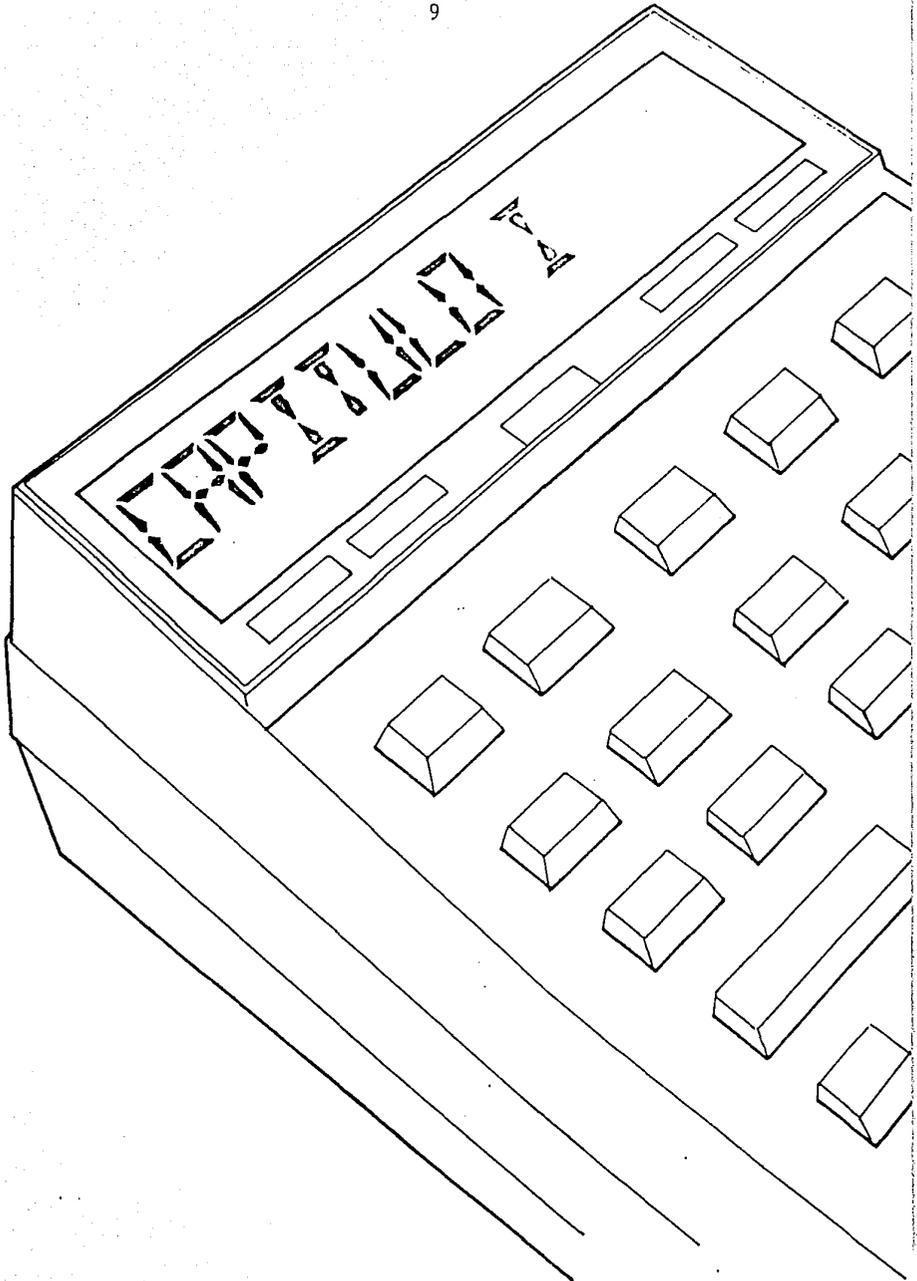
Por otra parte, aunque sabemos que es de considerarse como bien empleado el tiempo consumido en la capacitación de estas herramientas de cálculo, también sabemos que debido a las ocupaciones cotidianas de los usuarios, éstos carecen del tiempo suficiente como para dedicarse a estudiar la programación de la misma, además de que la razón principal de su adquisición es la utilización práctica de la máquina y no el realizar un estudio profundo sobre ella.

Para cubrir todas estas necesidades y con la finalidad de hacer una aportación para el impulso de las ramas ingenieriles por medio de la promoción para la mayor utilización de estas calculadoras, he decidido elaborar la presente TESIS sobre "Uso y Aplicaciones de la calculadora HP-41C en Ingeniería", en la cual trataré el tema de la forma más objetiva posible y se hará la presentación de programas de utilización práctica en el campo de trabajo. Así mismo, mostraré algunas técnicas de programación y algunas rutinas o "tips" con las cuales se pueden elaborar programas más eficientes.

Los programas que se tratan en la presente TESIS son sobre MATEMATICAS, INGENIERIA ELECTRONICA Y OTROS TEMAS.

En el capítulo I se sientan los precedentes para -

la mejor comprensión de los demás capítulos, así mismo - se hacen algunas observaciones sobre el uso de la calculadora; no pretendiendo de ninguna manera sustituir al manual, el cual encuentro sumamente completo para fines de enseñanza preliminar, ya que viene explicado en una forma clara, sencilla y es totalmente autodidáctico.



CONCEPTO  
GENERALES

## CONCEPTOS GENERALES

En el desempeño de su profesión, un Ingeniero debe ejecutar un sinnúmero de cálculos para fundamentar sus decisiones, mismos que generalmente se realizan en el transcurso de largas horas de trabajo manual, auxiliándose en el mejor de los casos de valores obtenidos en tablas de manuales específicos para cada tarea, ayuda que no representa gran reducción del tiempo total empleado para la solución del problema en cuestión. Así mismo en la realización de dichas operaciones es posible incurrir en errores que retardan aún más la obtención de la solución final.

Para visualizar claramente los conceptos anteriores, considere Ud. el problema de definir el valor de todas las variables en un circuito eléctrico, en el cual la solución más rápida puede ser mediante el planteamiento de ecuaciones que den lugar a una matriz de orden  $N \times N$ ; si  $N$  tiene el valor de 2 ó 3 es posible que Ud. solucione el problema sin mucha dificultad, pero si  $N$  es 10 o aún mayor, lo más seguro es que tarde todo un día cuando menos, eso sin contar la posibilidad de haber cometido un error numérico en la parte inicial de las operaciones por lo que el resultado estaría equivocado y tendría que hacer todo nuevamente.

Ahora bien, si Ud. soluciona este problema ayudado por una computadora o una calculadora programable de bolsillo, es factible que a lo mucho le tome 15 minutos llegar a la respuesta (tomando diez de ellos para ingresar, y después escribir los datos) y teniendo la certeza de que la máquina no se equivocó al efectuar las operaciones.

Bueno, pero quizás en este momento Ud. se pregunte:

- ¿Quién inventó las computadoras?

Estas maravillas de la técnica, que asombran al mundo entero, no nacieron de la noche a la mañana, sino que han sido el producto de multitud de investigaciones y experimentos de muchas personas a lo largo de la historia del hombre.

Para hacer una breve reseña recordemos que desde tiempos remotos el hombre ha buscado la manera de hacer cálculos de la manera más fácil y rápida posible, desarrollando diversos mecanismos y sistemas para conseguir ese objetivo; se podría iniciar la lista de ellos con el ábaco, que sirve para hacer operaciones elementales, y terminar con los modernos sistemas de computación con terminales remotas que procesan gigantescos volúmenes de información en pocos segundos.

Veamos algunos acontecimientos significativos que sucedieron entre estos dos extremos; para esto vayamos hasta el año de 1642 en el cual Blas Pascal diseñó y construyó lo que fue la primera máquina sumadora motivado por la necesidad de ayudar a su padre en su extenuante labor como recaudador de impuestos en Rouen, Francia; después de hacer algunas modificaciones al modelo original obtuvo al fin, hacia el año 1652, un modelo para producir en serie. Después de esto, en 1671 Leibniz construye una máquina que podía también multiplicar. Incidentalmente Leibniz a la vez introdujo en la terminología matemática la palabra "función" y fue el primero en reconocer la importancia del sistema numérico binario (aparte de, posi--

blemente los antiguos chinos).

El siguiente acontecimiento importante en nuestra historia sucede en 1812, cuando Charles Babbage diseña una máquina de diferencias cuyo propósito era ayudar a la construcción de tablas matemáticas. En 1833 él mismo propuso una máquina analítica que tenía un almacenaje mecánico para los datos e instrucciones, análogo al almacenaje electrónico de hoy en día; la falla en esta máquina como en la de "diferencias" se debió a que las herramientas y la tecnología de entonces no eran adecuadas para proporcionar los elementos de precisión requeridos.

Mientras que las computadoras analógicas se empezaron a diseñar y construir desde aquella época, las computadoras digitales no aparecen sino hasta 1940. En este año, la computadora COMPLEX de la Bell Telephone Laboratories se puso en operación. Era una máquina con una finalidad muy específica, diseñada para manejar los cálculos en que apareciesen números complejos con los que con tanta frecuencia se encuentran los Ingenieros Electricistas en su trabajo. La primera computadora digital de propósitos "generales", la HARVARD MARK I, entra en operación cuatro años más tarde; le sigue la ENIAC en 1946, hecha en la escuela Moore de Ingeniería Eléctrica de la Universidad de Pensilvania.

Desde 1952 han aparecido profusamente computadoras grandes y pequeñas, tanto en Estados Unidos como en otros países; a partir de entonces la arquitectura de estas máquinas quedó definida bajo una estructura que se puede delinear en forma general en cinco componentes básicos, los cuales se enlistan a continuación:

1. Mecanismos de entrada, por medio de los cuales se introducen datos e instrucciones.

2. Unidad de memoria, en la cual se almacenan los datos e instrucciones. La información puede retenerse permanentemente o borrarse y sustituirse por otra.

3. Unidad aritmético-lógica; este componente es el que ejecuta los cálculos internos tomando la información de la unidad de memoria de la computadora, la procesa y la regresa a la memoria. En esta unidad reside la lógica o "inteligencia" de la máquina.

4. Unidad de control, es la parte que sincroniza todas las operaciones de los componentes de la computadora.

5. Mecanismo de salida, que sirve para presentar al exterior los resultados intermedios y finales obtenidos por la máquina.

Actualmente las computadoras son utilizadas para un sinnúmero de fines, como: la reservación de boletos de avión en la mayoría de las líneas aéreas; el manejo de cuentas bancarias en gran número de instituciones de este tipo; la operación de inventarios en negocios de cualquier clase; también la elaboración de pólizas de diario, balances mensuales, nóminas, y la contabilidad en general de muchas empresas; la planeación, dirección y control de vuelos espaciales; la resolución de complicados cálculos físicos, químicos y matemáticos; el uso de robots industriales con una gran versatilidad en la utilización de éstos, y en fin, para no seguir con una lista interminable de usos, se puede decir que la única restricción para el empleo de las

computadoras es la que le ponga nuestra imaginación, pues se puede utilizar casi para cualquier aplicación.

Por sus altas velocidades de trabajo, gran capacidad de memoria y completos sistemas de monitores, las computadoras electrónicas se utilizan en todas las fases de la Ingeniería, desde el proyecto original hasta la construcción final. Pero los ingenieros deben seleccionar el método más económico, para lo cual necesitan comprender las características y principios básicos de la operación de una computadora.

Las computadoras satisfacen las expectativas de los usuarios en la medida en que se dan cuenta de que lo que está en juego no sólo es un equipo de cálculo, sino todo un nuevo concepto en el desarrollo de la Ingeniería. Una computadora electrónica, en manos de personal competente, puede llegar a competir con la producción de todo un equipo de diseñadores.

En cualquier proyecto, el trabajo se divide en dos etapas: pensamiento original y proyecto de detalle. En las principales áreas de Ingeniería, y gracias a los avances tecnológicos, actualmente la mayor parte del proyecto de detalle puede realizarse con una máquina.

Un examen de los beneficios que pueden derivarse del empleo de las computadoras indica que un Ingeniero debe aprender a trabajar con estas máquinas con objeto de estar en una posición competitiva, ya sea como individuo o como empresario. El uso correcto de las computadoras ahorra tiempo, mano de obra y dinero.

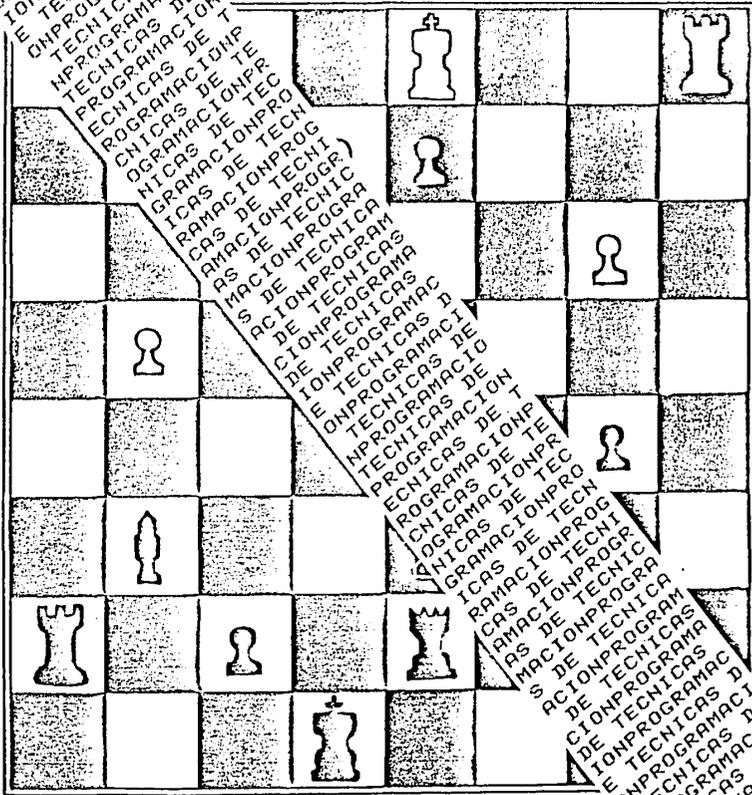
Paralelamente al crecimiento en potencia, rapidez - y capacidad de memoria de las computadoras, hay un desarrollo notablemente rápido en el empleo de calculadoras electrónicas de baja velocidad. Este auge en el uso de las mismas se debe al avance tecnológico que ha permitido aumentar considerablemente la capacidad y sofisticación de estos aparatos a la vez que se reduce su tamaño, precio y demanda de energía para trabajar, lográndose así obtener dispositivos tan pequeños como las calculadoras que vienen integradas en relojes de pulsera, hoy en día.

Donde se ha avanzado grandemente es en la rama de las CALCULADORAS PROGRAMABLES DE BOLSILLO; éstas son de tamaño muy reducido lo que las hace totalmente portátiles; - no obstante esto, resultan tener gran capacidad de cálculo y su manejo es muy sencillo, en ellas se puede grabar un cierto procedimiento y repetirlo sistemáticamente. Esta característica es una gran ventaja en el caso de tener que realizar un mismo cálculo gran número de veces, sobre todo si es extenso y algo complejo, ya que de hacerse manualmente implicaría que las personas que los realizan pudieran llegar a un estado de cansancio y aburrimiento causados por la propia naturaleza repetitiva del cálculo que tiende a provocar errores, puesto que se pierde la concentración en el trabajo que se desarrolla; una máquina en cambio, siempre seguirá la secuencia que se le haya indicado paso a paso hasta llegar a la conclusión del problema para el cual fue programada. Otra de las características, muy importante, es la rapidez con la que se realiza el cálculo de principio a fin; el tiempo que se invertía en cálculos manuales ahora se puede emplear en efectuar otras actividades, como el análisis de la solución y la evaluación de otras alternativas.

A pesar de sus limitaciones en cuanto a velocidad, comparada con la de las computadoras, y de la incapacidad de aceptar programas en lenguajes de alto nivel como -- FORTRAN o COBOL, estas calculadoras programables de bolsi llo tienen una elevada razón beneficio-costos y una gran - aplicabilidad por parte del usuario. Además, con los - - avances logrados en el aspecto de la comunicación entre - la máquina y el usuario, ahora esta comunicación se puede realizar con letras y diversos caracteres alfanuméricos - en este tipo de calculadoras, siendo que antes esta carac - terística sólo se daba en computadoras grandes.

Dentro de la última generación de calculadoras se destaca el modelo HP-41C, de la compañía Hewlett-Packard; esta calculadora está fabricada con la más alta tecnolo-- gía del mundo, ha sido diseñada para usarse como una cal-- culadora portátil totalmente independiente o como la par-- te central de un sistema de cálculo muy completo.

Pero desafortunadamente se requiere invertir bas-- tante tiempo para conocerlas y poder emplearlas de la ma-- nera adecuada; esto no significa que sean difíciles de - comprender, mucho menos que sean aparatos para científi-- cos solamente o para personas dedicadas exclusivamente a la programación; no, estas máquinas son herramientas que pueden y deben ser usadas por cualquier profesionista que se preocupe por estar al día en las innovaciones de la - técnica, especialmente por los Ingenieros, quienes deben tener presente que para aprender a usar una nueva herra-- mienta siempre es necesario destinar a ello tiempo y es-- fuerzo, que sin lugar a dudas será un tiempo bien emplea-- do que a la larga se compensará con las ventajas descri-- tas anteriormente.



PROGRAMAS DE TECNICAS DE INSTRUCCION TECNICA

PROGRAMAS DE TECNICAS DE INSTRUCCION TECNICA

## TECNICAS DE PROGRAMACION

Las fases principales en la preparación y el desarrollo de los programas son las siguientes:

1. Determinación de la conveniencia de que un problema sea resuelto por medio de la calculadora.
2. Definición del problema en términos matemáticos.
3. Selección de un método de solución en la calculadora.
4. Análisis del problema para establecer el flujo lógico de las secuencias de programación.
5. Codificación del problema para traducirlo en un programa y grabarlo en la calculadora.
6. Prueba y corrección.
7. Depuración final del programa.

Veamos ahora el significado de cada uno de los apartados arriba mencionados.

1. La conveniencia de resolver un problema con la calculadora puede determinarse con la ayuda de los siguientes criterios:

¿Se repetirá el problema lo suficiente como para justificar la elaboración de un programa? ¿Es tan complejo el problema que resulta impráctico el cálculo manual? - El tiempo y el costo de crear un programa debe sopesarse contra los beneficios que puedan derivarse de él.

¿Está el programa dentro de las capacidades de la -

máquina? Si una solución completa del programa sobrepasa la capacidad de la calculadora o es un trabajo demasiado pesado para el usuario, tal vez sea más conveniente dividir el problema en partes y obtener una solución para cada uno.

La evaluación correcta de la posibilidad de que el problema se solucione mediante la calculadora requiere un buen criterio de ingeniería y un sólido conocimiento de programación.

2. La definición del problema en términos matemáticos es esencial para elaborar el programa. Para ello, en general, es necesario: el establecer los datos, determinar las respuestas requeridas y señalar la precisión deseada de los resultados que deben obtenerse con la calculadora.

3. En la selección de un método de solución se debe tomar en cuenta el tiempo en la elaboración del programa. Con frecuencia es posible resolver un problema por varios métodos. El usuario que conozca la capacidad de la calculadora y los antecedentes teóricos del problema optará por el enfoque óptimo. Sólo que, con la gran velocidad de las máquinas, la diferencia en el tiempo-máquina para la solución del problema por dos métodos distintos puede ser ínfima comparada con la diferencia en el tiempo de efectuar el programa.

Una vez seleccionado el método para la solución del problema en la calculadora, se definen por completo las fórmulas matemáticas necesarias.

4. El análisis del problema para establecer el flu-

jo lógico de las secuencias de programación sigue el objetivo de identificar las operaciones componentes del problema y sus interrelaciones. La profundidad del análisis de un problema dado varía en forma considerable y depende tanto del programador como del problema. El análisis puede consistir en una lista de pasos escritos en forma matemática para problemas sencillos; para problemas más complejos, una imagen del proceso (o diagrama de flujo), que establece los pasos del problema y su secuencia lógica.

Los diagramas de flujo pueden consistir en proposiciones generales que describen operaciones diferentes a realizarse o las secuencias detalladas de cada paso. Siempre es deseable tener una documentación completa de un problema para referencias futuras y como una ayuda invaluable para otros usuarios del programa.

5. La codificación del problema para traducirlo a un programa de calculadora consiste en pasar a instrucciones propias de la máquina todo el proceso lógico que se obtuvo en el diagrama de flujo.

Quizás este proceso sea el más largo, pero a la vez es el más interesante, ya que se pueden utilizar una multitud de formas de codificación para llegar al mismo resultado.

Es conveniente idear una forma de ataque que nos permita realizar la codificación por partes, ayudándonos para esto de mapas de registros, anotaciones referentes a los registros del STACK, y del estado operativo de la calculadora.

6. En la fase de prueba y corrección se introducen

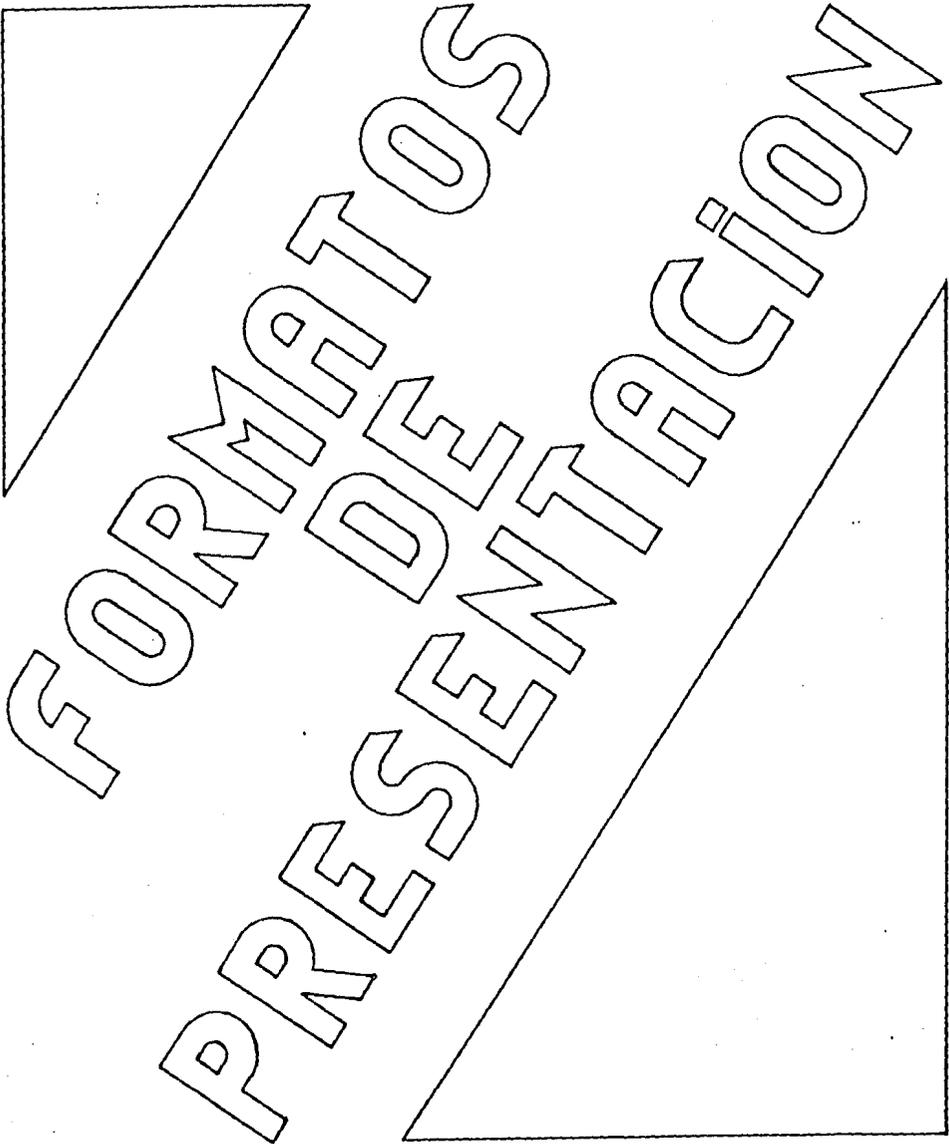
en la calculadora un conjunto de datos, por lo general - muy sencillos, con objeto de que puedan hacerse fácilmente cálculos manuales para comparar y verificar los resultados del programa. Si en el programa se hacen varias decisio-- nes, los datos de prueba deben cubrir todas y cada una de las posibilidades.

Deben corregirse todos los errores que se encuen- - tren al ejecutarse el programa, y una vez que se corrijan dichos pasos deberán volverse a correr una serie de datos con objeto de verificar si fueron atinadas las correccio-- nes realizadas.

7. La última fase en la elaboración de un programa es la denominada: depuración. En esta etapa el programa - es optimizado en lo referente a la técnica de programa-- ción, el menor número de pasos, la máxima confiabilidad y simplicidad de uso, etc.

La mayoría de las veces en que se realiza un progra ma, al completar la fase de prueba y corrección, o sea al dejarlo útil para que funcione por primera vez, el programa emplea una cantidad excesiva de pasos y memorias. Esto último lo podemos evitar analizando de nuevo todo el pro-- grama, desde el inicio hasta el paso final, tratando de en focar toda nuestra atención en detectar todas aquellas -- subrutinas o pasos que pueden ser sustituidos por otros - más simples o en función de los primeros.

Es muy común poder "compactar" un programa reducién do su longitud inicial en un buen porcentaje, obteniendo - con ésto, además de una cantidad menor de memoria utiliza-- da, una ejecución del programa más rápida y eficiente.



FORMATOS  
DE PRESENTACION

## FORMATOS DE PRESENTACION DE LOS PROGRAMAS

La presentación de los programas en los capítulos subsecuentes se hará de acuerdo a la siguiente estructura general:

- I Planteamiento del problema.
- II Ecuaciones y variables involucradas.
- III Uso del programa.
- IV Instructivo de uso.
- V Estado operativo de la calculadora.
- VI Diagrama de flujo.
- VII Ejemplos.
- VIII Listado del programa.
- IX Comentarios sobre el programa.

Veamos cual es la descripción de cada uno de los apartados:

### I PLANTEAMIENTO DEL PROBLEMA

Este punto se refiere al planteamiento global del problema a resolver, en el cual se incluirán los comentarios, observaciones y limitantes que sean pertinentes. Se añadirán dibujos y esquemas cuando así se requiera.

### II ECUACIONES Y VARIABLES INVOLUCRADAS

Tal como lo indica el título, se mencionarán las ecuaciones que se usan en la resolución analítica, sin llegar al detalle del manejo y sintetización de las mismas. Se definen en este punto todas las variables a utilizar.

### III USO DEL PROGRAMA

Esta sección fue especialmente redactada en cada uno de los programas para que el usuario logre captar totalmente el modo de empleo del mismo.

Para cumplir el objetivo de esta sección se ha procurado que la información contenida sea amplia y detallada, pero por esto mismo resulta impráctica para usarla como referencia continua, por esta razón, una vez comprendido claramente el uso del programa ya no será necesario apoyarse en ella, ya que para ese propósito se ha elaborado un resumen formatizado que se muestra en el siguiente apartado.

### IV INSTRUCTIVO DE USO

Se explica paso a paso el uso del programa. Para dicho efecto, se utilizan las hojas de "INSTRUCTIVO DE USO". En la columna de la izquierda se escribirá el número que corresponde a cada paso; a la derecha de ésta se describe la INSTRUCCION que se va a realizar; la columna de INGRESO indica las entradas que se realizan en el transcurso del programa, ya sea como un número en X o un caracter alfanumérico en ALPHA, cuando la pregunta mostrada en la pantalla termine con [ = ? ] se debe ingresar un número y si termina sólo con [ ? ] significa que el dato que se pide es alfanumérico.

Los nombres de las funciones, programas o subrutinas que se ejecuten, se escribirán en la columna para este propósito (FUNCIONES), anotando el nombre encerrado en un rectángulo.

En la columna de PANTALLA se anotará lo que se ve en ella al ejecutar un cierto paso, si ello lo requiere. Los resultados que vayan apareciendo se escribirán sin valor numérico; sólo aparecerá el nombre de la variable indicándose de esta manera que la cantidad está en función de los datos ingresados.

Cuando puedan aparecer dos o más letreros, resultados, funciones o entradas en un determinado punto de un programa, dependiendo de las propias características del programa y del problema, se pondrá la primera opción en su forma normal y las subsecuentes se escribirán entre paréntesis.

Para ilustrar todo lo expuesto anteriormente tengo por mostrar el siguiente ejemplo: (hoja siguiente)

## INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa		[XEQ] STCOM	LISTO ...
2	Ingresar datos en forma rectangular	Re IM	[ENTER] ✓ [RE]	LISTO ...
	En forma polar.	MAG θ	[ENTER] ✓ [PD]	LISTO ...
3	Se repite el paso 2 las veces que se necesite.			
4	Ya que se han ingresado los datos requeridos, se ejecutan las operaciones deseadas (operaciones complejas).		[-] [+] [*] [/]	
5	Para ver el resultado, que quedó en el "X" del STCOM, se oprime [A] o [a].		[A] [R/S] * [a] [R/S] *	Re= IM= MAG= ∠ =
6	Si se desea ejecutar el programa de nuevo.		[STCOM]	LISTO...
7	Para "borrar" el último ingreso o resultado.		[CLXCOM]	LISTO...

## V ESTADO OPERATIVO DE LA CALCULADORA.

Para presentar el estado operativo de la calculadora durante la ejecución del programa se hace uso del formato que se muestra en la siguiente página; los números - marcados en él tienen el siguiente significado:

### 1. Registros.

Esta columna contiene el número del registro a la izquierda y su contenido a la derecha. Dicho contenido - puede ser un dato específico (como una altura o un peso), un número de control para gobernar una bifurcación o un - salto y por último, también puede ser un dato alfanumérico. Pueden usarse uno o varios renglones para definir - cada registro.

### 2. Estatus.

En esta zona se pondrá lo siguiente: el número de bytes usados por el programa (que al dividir por siete nos da su equivalente en cantidad de registros); el SIZE o - número de registros necesarios para el almacenamiento de datos (si es variable se pondrá la fórmula correspondiente en el espacio bajo este renglón); la suma de los dos - conceptos anteriores da el total de registros usados en - el programa.

La forma de PRESENTACION NUMERICA se señalará escribiendo el número de dígitos requeridos a la derecha - del letrero correspondiente (ENG, SCI o FIX), si no se - usa un solo formato se marcarán los diferentes dígitos en un mismo renglón, por ejemplo : FIX 0,3.



Si la modalidad USER se enciende en el transcurso del programa, se marcará una X enseguida del letrero ON.

Se marcará con una X la MODALIDAD TRIGONOMETRICA empleada en el programa. Cuando sea indiferente una u otra opción no se marcará ninguna, tanto para presentación numérica, USER como para modalidad trigonométrica.

### 3. Banderas.

En la columna de la izquierda se escribirá el número de la bandera en cuestión; en la columna debajo de "SET INDICA" se pondrá el significado que tiene esa bandera o la decisión que se tomará a causa de esa condición.

Lo mismo se aplica a la columna de "CLEAR INDICA" excepto en el caso de que CLEAR indique exactamente lo contrario y por lo mismo no se especificará su efecto, por ejemplo:

BANDERAS		
Nº.	SET INDICA :	CLEAR INDICA :
00	Se ingresó X	<del>No se ingresó X</del>
01	Número base 2	Número base 10

### 4. Asignaciones.

Aquí se escribirán las funciones que estén asignadas al teclado; esto incluye a las funciones locales (de la [A] a la [J] y de la [a] hasta la [e]), a las funciones de la máquina que no se encuentran disponibles directamente en

una tecla, a las rutinas del mismo programa o fuera de él, y a los programas asignados para operarse en la modalidad USER, incluyendo la asignación del mismo programa en que se esté trabajando.

La tecla respectiva se señala a la derecha de la función de acuerdo a la convención normalmente usada de renglón-columna, con el signo negativo para la función secundaria o SHIFT respectiva.

#### 5. Esquema de Asignaciones.

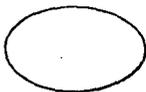
Para visualizar de manera más efectiva las funciones o subrutinas que se describan en el cuadro de asignaciones, se añade un dibujo del teclado donde se pondrán las funciones correspondientes a cada tecla.

Cuando alguna de estas secciones (1 al 5) no se use, se indicará a la derecha del nombre de la sección de la siguiente manera:

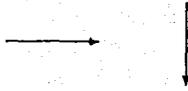
R E G I S T R O S - N O

#### VI DIAGRAMA DE FLUJO

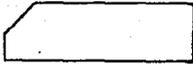
Se dibujará el diagrama de flujo que indique el método de resolución adoptado en el programa; la simbología usada en estos dibujos será la siguiente:



Una elipse indica el inicio del programa; contendrá en su interior el nombre del mismo. También sirve para señalar el final del programa.



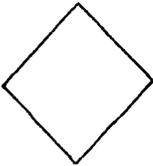
Las flechas marcan el sentido de la ejecución del programa. De la misma manera señalan la forma en que se ejecutan los saltos y los lazos.



Un rectángulo con una esquina recortada nos señala el ingreso de datos.



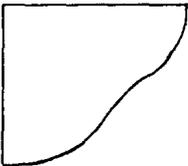
Un rectángulo representa una instrucción o un bloque de ellas, que pueden ejecutar ya sea una o varias funciones.



El rombo nos indica que se ha planteado una pregunta de la cual se tomará una decisión para definir el camino a seguir.

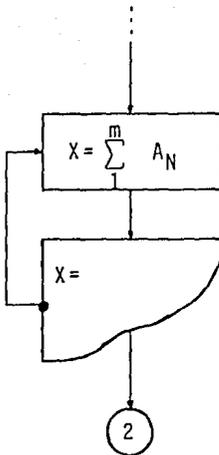


El círculo puede tener un número o un nombre y nos indica que hay un salto hacia un lugar en otro punto de la diagramación marcado con el mismo número o letrero.

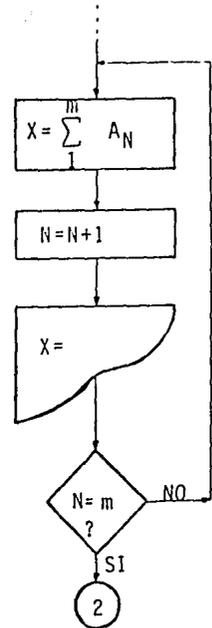


Este es el símbolo que se utiliza para la salida de datos, que puede ser mediante la pantalla o la impresora; se pueden obtener resultados intermedios y/o finales.

Cuando uno o más bloques estén comprendidos en un lazo donde haya una variable que toma N valores, se



indicará como se ve en la figura de la izquierda. - Significa que los bloques comprendidos en el lazo se repetirán hasta que  $N$  tome todos sus valores. Los bloques incluidos pueden ser de cualquier tipo. Nótese que el inicio de la ramificación se marca con un pequeño círculo (repleto,  $\bullet$ ) y que principia en medio del bloque que es el último en el lazo. En la figura de la derecha se muestra un equivalente, en simbología estándar, del efecto que se describe.

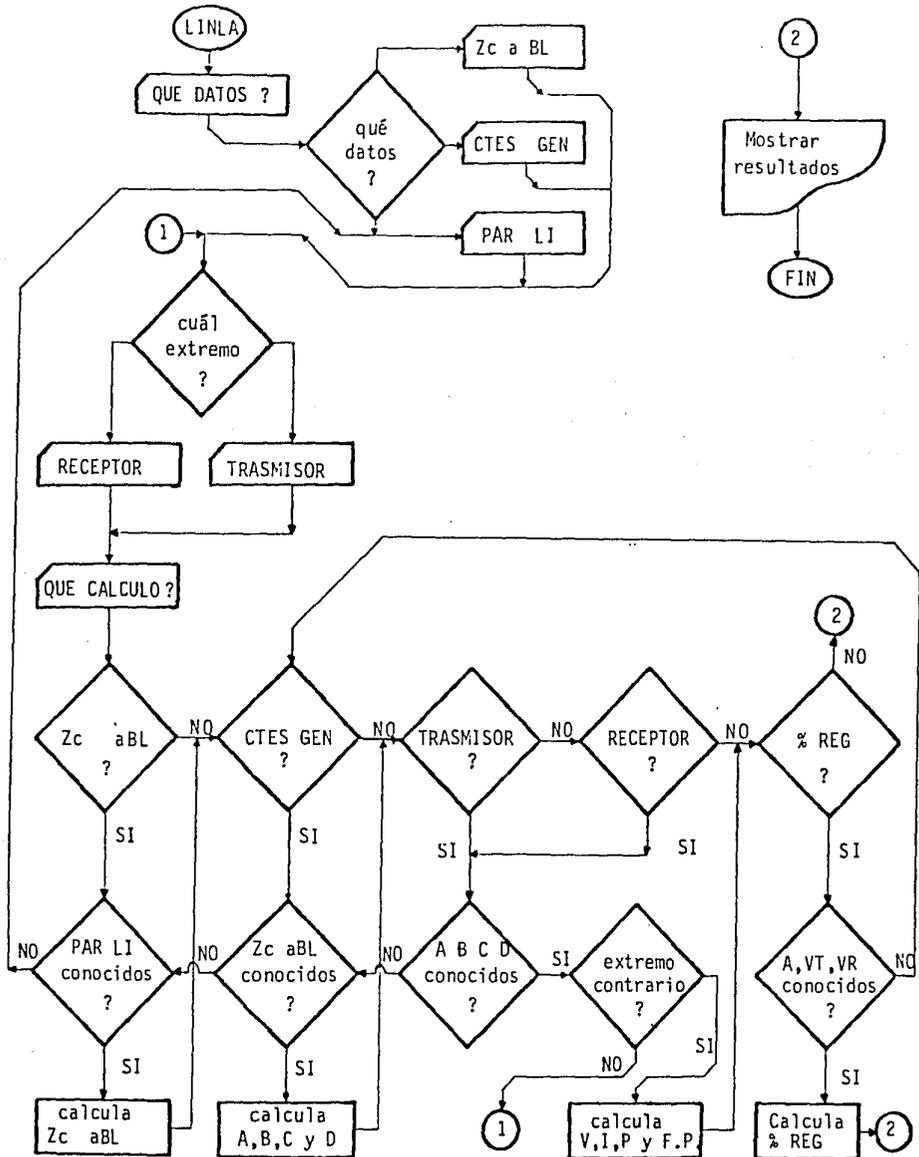


Un cuadro nos indica la posibilidad de acceso inmediato y directo a un punto del programa, en cualquier momento de la ejecución, - por medio de una función local o una subrutina asignada en alguna tecla.



Ahora ejemplificaremos todas las reglas anteriormente descritas con el diagrama de flujo que se muestra a continuación.

# DIAGRAMA DE FLUJO



## VII EJEMPLOS

Cada programa se acompañará con uno o más ejemplos que permitan la comprensión total de su empleo.

Los problemas-ejemplo se definirán acompañándolos de dibujos complementarios cuando sea oportuno.

A la izquierda se pondrán las funciones y entradas conforme al proceso de ejecución del programa, y a la derecha lo referente a lo que aparece en la pantalla.

Cuando sea necesario se escribirán notas y comentarios, haciendo énfasis en detalles que se consideren importantes.

## VIII LISTADO DEL PROGRAMA

En una hoja especialmente diseñada para este propósito, se enlistarán todas las instrucciones del programa.

Los listados fueron escritos por la impresora - - HP 182A, utilizando el formato NORMAL de impresión con - ancho sencillo.

En la siguiente hoja se puede observar un ejemplo de estos listados.

## LISTADO DEL PROGRAMA

551 XEQ *-1*		606 CTO 0*	
552 F-P		607*LBL 1*	
553 STO 29		608 *FIX...*	
554 PIn		609 BEEP	
555 STO 30		610 PPOMPT	
556 RCL 26		611 RCL 37	
557 -		612 STOPLAG	
558 SLOW		613 CTO *111L*	
559 LAST*		614*LBL *67*	
560 COS		615 FIX 9	
561 *		616 *Re A=*	
562 STO 38		617 ARCL 10	
563 ABS		618 RVIEW	
564 RCL 19		619 *IM A=*	
565 RCL 29		620 ARCL 11	
566 *		621 RVIEW	
567 *		622 *Fe B=*	
568 ?		623 ARCL 12	
569 *		624 RVIEW	
570 STO 18		625 *IM B=*	
571 SF 03		626 APCL 13	
572 PTH		627 RVIEW	
573*LBL 37		628 *Re C=*	
574 FC? 04		629 ARCL 14	
575 XEQ 04		630 RVIEW	
576 FC? 07		631 *IM C=*	
577 XEQ 02		632 APCL 15	
578 FC? 02		633 RVIEW	
579 XEQ 67		634 *Re D=*	
580 RCL 11		635 ARCL 16	
581 RCL 10		636 RVIEW	
582 P-P		637 *IM D=*	
583 RCL 22		638 ARCL 17	
584 /		639 RVIEW	
585 1/X		640 PTH	
586 RCL 19		641*LBL *82*	
587 X<>Y		642 FIX 1	
588 XCH		643 *V REC.=*	
589 STO 31		644 RCL 19	
590 RTN		645 J	
591*LBL 09		646 SORT	
592 TONE 5		647 *	
593 FIX 6		648 ARCL X	
594 CF 29		649 *F V*	
595 SF 21		650 RVIEW	
596 1 E2		651 FIX 2	
597 ST* 24		652 *AV R.=*	
598 RCL 24		653 RCL 26	
599 INT		654 R-D	
600 ST- 24		655 ARCL X	
601 CLR		656 RVIEW	
602 APCL X		657 FIX 1	
603 ASTO X		658 *I REC.=*	
604 SF 29		659 ARCL 29	
605 XEQ INH X		660 *F A*	

## IX. COMENTARIOS SOBRE EL PROGRAMA.

En esta sección de cada programa, se pretende ofrecer al lector una serie de TIPS que seguramente encontrará interesantes y le servirán para mejorar, en lo posible, su técnica de programación.

Al escribir estas "lecciones", se trata de cubrir el objetivo de enseñar algo de "técnicas avanzadas" en el uso de la calculadora para lograr su utilización óptima.

## CONVENCIONES UTILIZADAS

Con el fin de simplificar el uso de los programas que se presentan a lo largo del texto se han tomado las siguientes CONVENCIONES:

-En el transcurso de la tesis al referirnos a las funciones, avisos, y procedimientos de la calculadora, he preferido hacerlo en los términos originales en inglés, para evitar confusiones, ya que esa nomenclatura está integrada a la máquina y no se puede alterar. Además siempre se escribirán con letras mayúsculas, por ejemplo la función trigonométrica SENO aparecerá con su abreviatura inglesa SIN, la raíz cuadrada de un número como SQRT, etc.

-Las funciones se representan por su nombre encerrado en un rectángulo como: **SIN**, **SQRT**, **LN**, **STO**.

-Existen funciones llamadas secundarias, que se activan presionando la tecla amarilla (SHIFT) antes de oprimir la tecla correspondiente, por ejemplo la función %.- Cuando el manual de operación (editado por la compañía HP) se refiere a ella lo hace mencionando las dos teclas: **SHIFT** **%**, en este texto se suprime la primera poniéndose sólo: **%**, esto se aplica en general a todas las funciones secundarias.

-Hay funciones que aun cuando no están en el teclado sí se encuentran presentes en la máquina; para ejecutarlas se sigue este procedimiento: se oprime la función **XEQ**, luego se ingresa el nombre de la función deletreando caracter por caracter, de la siguiente manera: **ALPHA**

F A C T **ALPHA** (se tomó como ejemplo la función factorial), al oprimir la tecla **ALPHA** por segunda vez se da por terminado el ingreso del nombre y automáticamente la máquina efectúa la función; en esta Tesis se ha elegido poner únicamente el nombre como si se tratara de una función ordinaria, así pues quedará como **FACT** .

-Al ejecutar un programa, se comienza oprimiendo la tecla **XEQ** y luego se deletrea el nombre correspondiente con caracteres alfanuméricos, por ejemplo: **XEQ ALPHA A B C ALPHA** . Este procedimiento se efectuará siempre y cuando no se tenga el programa en cuestión asignado a una tecla. Por convención para esta Tesis se ha decidido iniciar dichas subrutinas o programas como **XEQ A B C**, excluyendo la tecla **ALPHA** .

-Cuando se activa la modalidad USER, se habilita la ejecución de funciones asignadas en el teclado; ellas aparecerán designadas con su nombre correspondiente, así, pues, las funciones locales, que son desde la **A** a la **J** y de la **a** hasta **e**; las funciones que no están presentes en el teclado normalmente como **FACT** , **MEAN** ; y las de asignación global (programas) por ejemplo **ABC** , se presentarán como si fuesen teclas de función cuando estén asignadas.

-Si se requiere mostrar en el texto los mensajes que aparecen en la pantalla en algún momento, se pondrá lo correspondiente enmarcado por paréntesis rectangulares, por ejemplo: **[COORDENADA =?]**.

-Para diferenciar la letra O del cero, éste apare-

cerá con una diagonal, Ø. Por ejemplo [COORDENADA=1Ø.Ø8Ø].

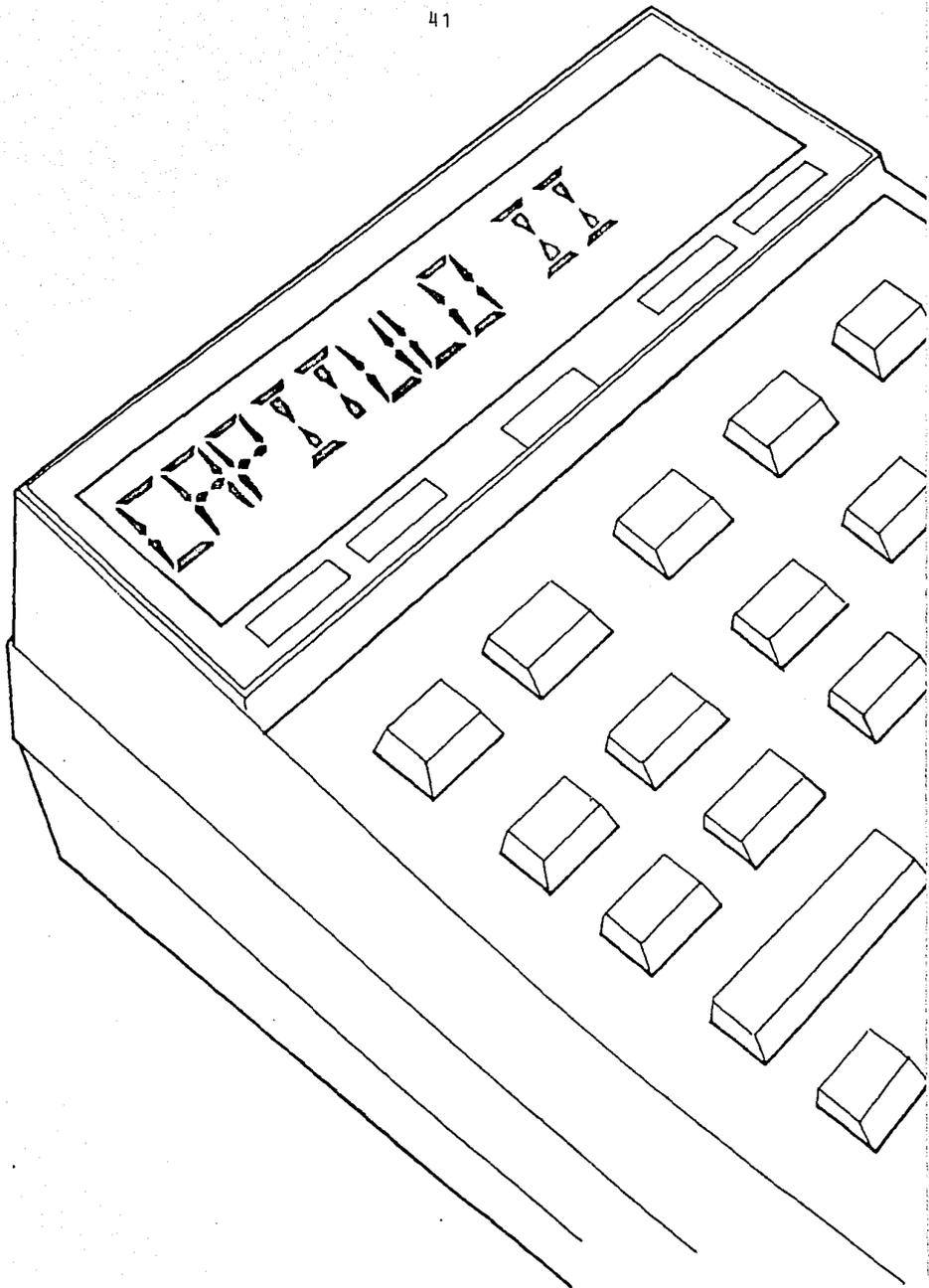
-El exponente de los números con formato SCI o ENG se presentarán de la siguiente manera [123.45E6] en lugar de [123.45 Ø6] que es como se ve realmente en la pantalla, para evitar confusiones por los espacios entre el número y el exponente.

-Para designar los registros que se usen se sigue la siguiente nomenclatura: R ab indica un registro de la memoria principal como R ØØ , R Ø1, (ocasionalmente ab - podrá ser un número de tres dígitos 1mn). Los registros del STACK son referidos como X,Y,Z,T y L sin anteponerles la letra R; el registro alfanumérico será: ALPHA.

-En las hojas de INSTRUCTIVO DE USO que acompañan a cada programa, frecuentemente se verá R/S \*, este asterisco quiere decir que si la impresora se encuentra conectada NO se deberá oprimir R/S para que continúe la ejecución del programa, sino que la máquina lo hará automáticamente.

-Todos los programas presentados en esta Tesis son aplicables en las calculadoras de esta serie o sea, la HP41CX, la HP41CV junto con el módulo de XFUNCTIONS y la HP41C con el mismo módulo y uno más de memoria cuádruple, QUAD MEMORY MODULE, o sus equivalentes en módulos de memoria sencilla.

-Todos los programas se pueden utilizar con cualquier conjunto congruente de unidades, por lo que no se especifica nada al respecto en el contexto del mismo.



## CAMBASE

En algunas ocasiones es necesario calcular el -  
equivalente numérico en base 10 (también conocido como -  
sistema decimal, o arábigo) de una cantidad expresada en  
una base diferente a 10, o viceversa. Estos cambios se  
realizan comúnmente al sistema binario (B2), octal (B8)  
o hexadecimal (B16).

El presente programa está diseñado para efectuar  
todo tipo de cambios de BASE X a BASE Y, donde:  $X \leq 36 \geq Y$ ,  
tenemos por ejemplo los siguientes cambios:

B2 a B16	B10 a B36
B8 a B10	B16 a B20
B16 a B4	B3 a B10

Al elaborar este programa, se incluyeron algunas  
funciones de la calculadora que son pobremente usadas ya  
que no son accesibles directamente en el teclado (excep-  
to cuando se asignan a alguna tecla, obviamente). Sin -  
embargo, debemos tener siempre presentes a todas esas --  
funciones "ocultas" de la máquina, ya que de no hacerlo  
nos veremos involucrados, en muchas ocasiones, en compli-  
cados problemas de programación que se solucionarían en -  
forma práctica y rápida con la ayuda de estas herramien-  
tas. Algunas de estas funciones son: MOD, ATOX, AROT, -  
etc.

## VARIABLES Y ECUACIONES.

Las variables que se necesitan ingresar son:

DE QUE BASE se va a efectuar el cambio (base original, BX).

A QUE BASE se quiere cambiar (base final, BY).  
 NUMERO que se desea transformar en base X, (NX).  
 NY es la cantidad expresada en base Y.

El programa consta, esencialmente, de dos transformaciones: de base X a base 10 y de base 10 a base Y.

$$BX \rightarrow B10: \quad N10 = (D_n * BX^n) + (D_{n-1} * BX^{n-1}) + \dots + (D_1 * BX) + D_0$$

donde  $D_n$  es el dígito enésimo de la cantidad dada en BX, comenzando con  $D_0$  en la extrema derecha (y aumentando de derecha a izquierda).

$$B10 \rightarrow BY: \quad D_n NY = \text{MOD}_n ((N10/BY) \rightarrow BY)$$

donde:  $D_n NY$  es el dígito enésimo del número en base Y, a partir de  $D_0$  (el de extrema derecha).

$N10$  es el número original expresado en base 10.

$\text{MOD}_n (\rightarrow)$  significa el módulo (residuo) enésimo calculado entre el cociente de  $N10$  sobre BY y, BY, hasta que el resultado sea cero.

Para que esto último quede claro, observe el siguiente ejemplo de la función MOD; calcule el módulo (residuo) de 8 con 3, MOD (8  $\rightarrow$  3)

ENTRADAS / FUNCIONES

PANTALLA

8  3

2.0000

## USO DEL PROGRAMA.

El programa cuenta con dos alternativas en la forma del ingreso de datos. La primera es en forma consecutiva, ingresando los datos con **R/S** a medida que se van solicitando, al hacerlo de esta manera, si no se efectúa un ingreso, la variable cuestionada quedará con el mismo valor que tenía anteriormente.

La segunda opción consiste en usar las funciones locales **A** y **B** en modo de usuario. Al oprimir la primera de éstas (**A**), el programa hará que la calculadora vuelva a pedir los datos de las bases no importando en qué punto del programa se encontraba el proceso. De la misma manera, cuando se oprime la tecla **B**, la calculadora pedirá otro número para efectuar el cambio de base.

El ingreso de la cantidad NX puede hacerse indistintamente en el registro X o en ALPHA. Por ejemplo el número 1011 en B2 puede señalarse en X o en ALPHA; aunque claro, el número 5AF (que contiene letras) sólo puede ingresarse en ALPHA. Cuando la máquina solicita el NX, aparece el letrero: [NUMERO=? :BX:], señalando la base original, como recordatorio.

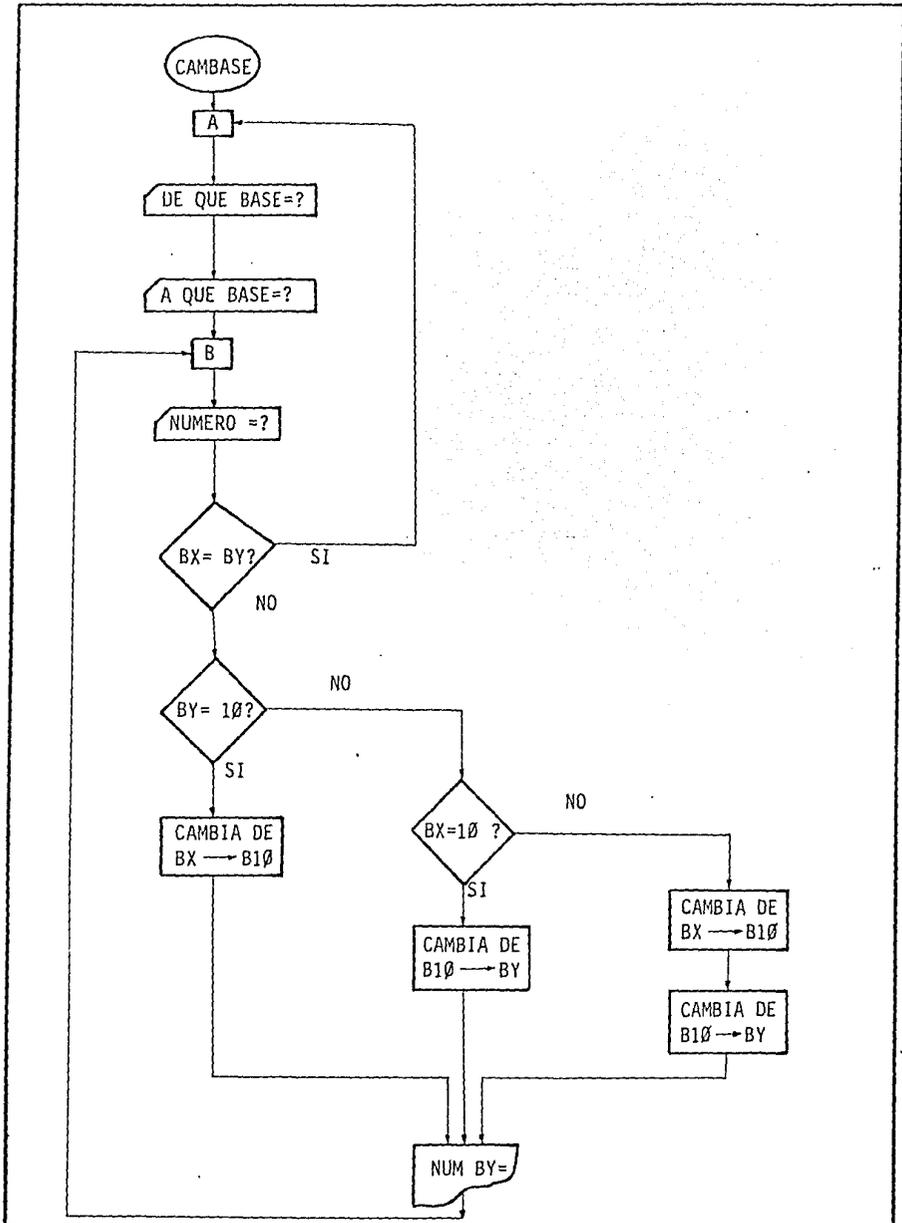
Después de ingresar la cantidad NX, al oprimir la tecla **R/S** se ejecutará el proceso y aparecerá en la pantalla el resultado de la siguiente manera: [NUM.BY= ].

## INSTRUCTIVO DE USO

PASO Nc	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa.		<input type="checkbox"/> XEQ CAMBASE	DE Q.BASE=?
2	Se ingresa la base original (si no hay ingreso queda vigente el dato anterior).	BX	<input type="checkbox"/> R/S	A QUE BASE=?
3	Se ingresa la base final.	BY	R/S	NUMERO=? :BX:
4	Se ingresa el número en la base original (en ALPHA si es necesario).	NX  (NX)	<input type="checkbox"/> R/S <input type="checkbox"/> (ALPHA) <input type="checkbox"/> R/S )	
5	Aparece el resultado...			NUM. BY=
6	Al oprimir <input type="checkbox"/> R/S se regresa al punto 4.		<input type="checkbox"/> R/S	NUMERO=? :BX:
7	Si desea cambiar alguna base se oprime <input type="checkbox"/> A , y sigue el paso 2.		<input type="checkbox"/> A	DE Q. BASE=?
8	Para acceder al paso 4 desde cualquier punto del programa oprimir <input type="checkbox"/> B.		<input type="checkbox"/> B	NUMERO=? :BX:



## DIAGRAMA DE FLUJO



# LISTADO DEL PROGRAMA

```

01 *
02 *
03 *
04 *DE 0. BASE=0*
05 CF 22
06 PROMPT
07 FS? 22
08 STO A1
09 *DE 0. BASE=0*
10 PROMPT
11 FS? 22
12 STO 02
13 *LBL P
14 FIX 0
15 CF 00
16 SF 21
17 CF 22
18 CF 23
19 CF 29
20 *NUMERO? *
21 ARCL 01
22 *
23 PROMPT
24 OFFF
25 FS? 22
26 STO 03
27 RCL 01
28 RCL 02
29 X=Y?
30 GTD P
31 10
32 X=Y?
33 GTD 01
34 RCL 01
35 X=Y?
36 GTD 04
37 *LBL 01
38 SF 02
39 *LBL 01
40 SF 01
41 RCL 01
42 STO Y
43 FS? 23
44 GTD 02
45 CLR
46 ARCL 03
47 *LBL 02
48 CF 02
49 ATOM
50 X=0?
51 GTD 02
52 60
53 X=Y?
54 SF 02

```

```

55 *
56 *
57 48
58 FC? 02
59 55
60 -
61 FC? 01
62 *
63 *
64 STO 02
65 *LBL 02
66 RDN
67 *
68 *
69 STO 06
70 ARCL 03
71 FC? 03
72 STO 06
73 *LBL 04
74 FS? 23
75 GTD 06
76 CLR
77 *LBL 05
78 *
79 RCL 03
80 INT
81 Y=0?
82 GTD 06
83 RCL 02
84 ST/ 06
85 RDN
86 48
87 -
88 59
89 *
90 SF 02
91 RDN
92 FS? 02
93 +
94 XTOP
95 -1
96 ARCT
97 GTD 05
98 *LBL 06
99 RCL 02
100 1/X
101 LOG
102 *NUM. P*
103 ARCL 02
104 *
105 8
106 -
107 ARCT
108 VIEW

```

LISTADO DEL PROGRAMA

--	--	--	--

## EJEMPLO 1.

Calcular los siguientes cambios:

(1001)<sub>2</sub> a B10  
 (A4F)<sub>16</sub> a B 8  
 (825022)<sub>10</sub> a B36  
 (17426908)<sub>10</sub> a B36

## ENTRADAS/FUNCIONES

XEQ C A M B A S E  
 2  R/S  
 10  R/S  
 1001  R/S  
 R/S \*  
 A  
 16  R/S 8  R/S  
 ALPHA A 4 F  R/S  
 A 10  R/S 36  R/S  
 825 022  R/S  
 R/S  
 17426908  R/S

## PANTALLA

DE Q. BASE=?  
 A QUE BASE=?  
 NUMERO=? :2:  
 NUM. B 10 = 9  
 NUMERO=? :2:  
 DE QUE BASE=?  
 NUMERO=? :16:  
 NUM. B8 = 5117  
 NUMERO=? :10:  
 NUM.B36= HOLA  
 NUMERO=? :10:  
 NUM.B36= ADIOS

## EJEMPLO 2.

¿Cuál es el direccionamiento absoluto (en B16) de la localidad número 50,000 (en B10) en una computadora - de 64K bytes de memoria? ¿Cuál es su capacidad total de memoria? (pasar  $(FFFF)_{16}$  a B10).

## ENTRADAS/FUNCIONES

[A] 10 [R/S] 16 [R/S]

50000 [R/S]

[A] 16 [R/S] 10 [R/S] [ALPHA] FFFF [R/S]

## PANTALLA

NUMERO=? :10:

NUM. B16=C350

NUM. B10 = 65535

## COMENTARIOS SOBRE EL PROGRAMA.

En este programa se emplea el registro ALPHA para lograr el cambio de bases; para efectuar la transformación de BX a base 10 se logra por medio de un algoritmo, éste consiste en ingresar la BX en el STACK y luego formar un lazo donde se decodifica el contenido del ALPHA - y se suma y multiplica cada dígito del número NX, por la base BX. Este lazo es el marcado como LBL 02.

Para lograr el cambio de B10 a BY, se hace en forma inversa, ejecutando divisiones consecutivas y tomando el residuo resultante en cada una para luego ingresarlo al ALPHA, formando el letrero final: [NUM. BY= ]. Es preciso señalar en este procedimiento la manera en que se forma este letrero. Comienza "escribiendo" el número resultante de derecha a izquierda y luego lo coloca en su posición correcta por medio de una rotación, con el lazo rotulado como LBL 05; para después agregar el letrero propiamente dicho (NUM.BY=) y pasarlo a la izquierda de la pantalla, también con una rotación.

Hay que recalcar el hecho de que la transformación de una BX a una BY, en las cuales ninguna es B10, tiene que hacerse pasando primeramente de la base original a la base 10 y después de base 10 a la base final.

## STCOM

En los cálculos de redes de potencia, es común encontrar números complejos. Estos sirven para expresar como fasores la corriente y el voltaje de un circuito. Un fador es una representación en el dominio de la frecuencia de una cantidad expresada originalmente como una función del tiempo; por ejemplo:  $i(t) = 5 \cos(300t + 150^\circ)$ , - está en el dominio del tiempo y se transforma en el fador siguiente:  $I = 5 \angle 150^\circ$  .

Los valores de inmitancia y de potencia, dentro de un circuito, también se pueden representar como cantidades complejas. Todas estas representaciones (o transformaciones) simplifican los cálculos de redes con fuentes sinusoidales, ya que eliminan la necesidad de usar ecuaciones integro-diferenciales que describan al circuito, - para conocer su comportamiento.

El programa STCOM (STACK COMPLEJO) es una simulación del stack estándar de la calculadora, y nos servirá para ejecutar operaciones aritméticas con cantidades complejas como las mencionadas anteriormente, haciéndolas - con suma facilidad y sencillez.

## VARIABLES Y ECUACIONES.

Las variables utilizadas se enlistan a continuación:

MAG = magnitud de un número complejo.

$\angle$  = ángulo ( $\theta$ ) de un número complejo respecto al semieje positivo de X, en grados sexagesimales.

Re = componente en X de un número complejo (real).

IM = componente en Y de un número complejo (imaginario).

Los números complejos se escriben generalmente en dos formas: POLAR o RECTANGULAR. En el siguiente dibujo se presentan estas dos formas; más adelante se relacionan los términos que aparecen en la figura.

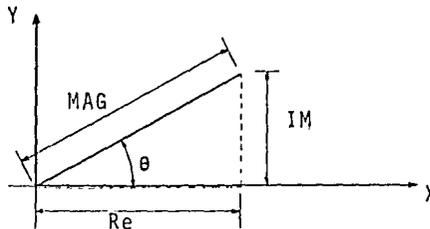


FIG. 1. Formatos rectangular y polar de los números complejos.

Del esquema podemos partir para ver la relación que guardan entre sí los dos sistemas, que se anotan en seguida:

$$\text{FORMA POLAR: } \text{MAG} \angle \theta \quad \left\{ \begin{array}{l} \theta = \text{ATAN} (IM/Re) \\ \text{MAG} = \sqrt{Re^2 + IM^2} \end{array} \right.$$

$$\text{FORMA RECTANGULAR: } Re + j IM \quad \left\{ \begin{array}{l} Re = \text{MAG} \cos (\theta) \\ IM = \text{MAG} \text{ sen } (\theta) \end{array} \right.$$

Para efectuar operaciones aritméticas con números complejos es preferible usar la forma polar en las multiplicaciones y divisiones y la forma rectangular para sumas y restas. Estas operaciones se ejecutan siguiendo las siguientes reglas:

$$\text{Sean : } A = \text{MAG1} \angle \theta_1 = \text{Re1} + j \text{IM1} \quad \text{y}$$

$$B = \text{MAG2} \angle \theta_2 = \text{Re2} + j \text{IM2}$$

$$\text{SUMA: } A+B = (\text{Re1} + \text{Re2}) + j (\text{IM1}+\text{IM2})$$

$$\text{RESTA: } A-B = (\text{Re1} - \text{Re2}) + j (\text{IM1}-\text{IM2})$$

$$\text{MULTIPLICACION: } AB = \text{MAG1} \cdot \text{MAG2} \angle \theta_1+\theta_2$$

$$\text{DIVISION: } A/B = \text{MAG1}/\text{MAG2} \angle \theta_1-\theta_2$$

### USO DEL PROGRAMA

En el programa STCOM se pueden ingresar datos - en ambas formas: polar o rectangular, y con él podemos - ejecutar las cuatro operaciones descritas arriba; estas operaciones trabajan en la forma usual con que se maneja el stack estándar de la calculadora, que es el método - RPN; es decir para cada operación que se va a a efectuar se emplean los dos últimos números complejos que se ha-- yan ingresado en el stack, y el resultado queda listo para usarse con el valor inmediato superior.

Los números en forma polar se ingresan de la manera siguiente: primero la magnitud (MAG) y después el ángulo ( $\angle$ ), o sea que el ángulo queda en el registro X y la magnitud en Y. Para almacenar estos datos el programa los cambia automáticamente a forma rectangular. Al - ingresar cantidades en forma rectangular se procede primeramente a señalar la parte real (Re) y luego la imaginaria (IM), es decir la parte real queda en Y y la imaginaria en X.

Para efectuar esos ingresos se tienen las funciones **PO** y **RE** asignadas a la función secundaria -- (shift) de las teclas **P** y **R** respectivamente.

Los operadores aritméticos complejos se encuentran asignados al shift de las funciones estándar.

Si se quieren ver los resultados de cada operación o el último dato ingresado al STCOM se debe oprimir la tecla **A** que mostrará el "X" del STCOM en forma rectangular o, **a** para ver el "X" en forma polar.

La función **CLXCOM** nos sirve para eliminar del STCOM el último número complejo que esté almacenado en el mismo; esto se aplica indistintamente al último ingreso o a un resultado intermedio. No significa que ponga en cero al "X", sino que simplemente el programa lo "ignora" por medio de reducir el tamaño del STCOM que se esté empleando en ese momento (decrementando el registro de control).

El número de cantidades complejas pendientes en el STCOM no tiene más límite que el SIZE que se use, a diferencia del stack normal de la calculadora, en el cual el límite es de cuatro.

Para comenzar de nuevo el programa, sólo es necesario oprimir **STCOM**. Esto se emplearía en el caso de tener un SIZE pequeño, o de no querer emplear muchos registros de la memoria en el almacenamiento de los datos del STCOM.

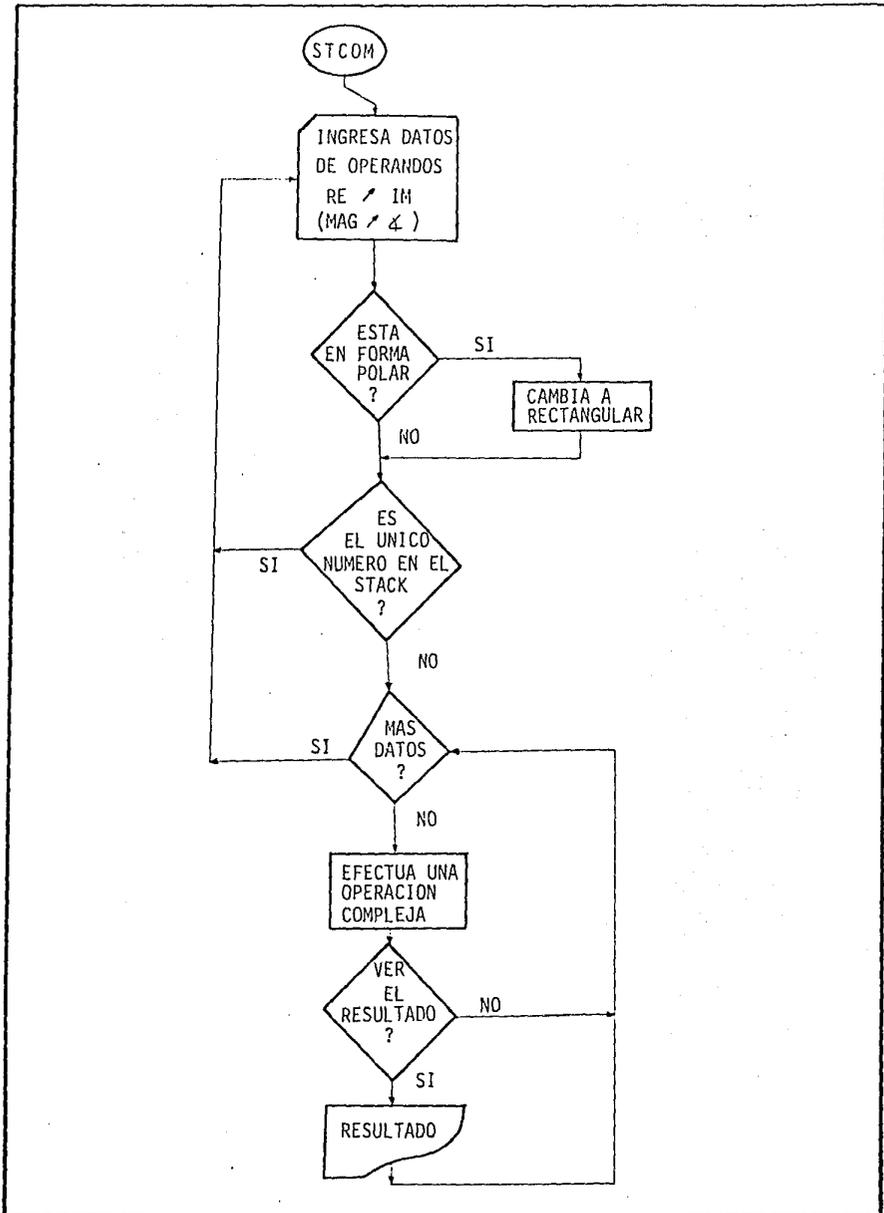
Si se cuenta con el módulo MATH 1, pueden usarse las subrutinas 'c\*', 'c/', 'c+' y 'c-', y eliminar los pasos 111 a 139 del programa para economizar memoria.

## INSTRUCTIVO DE USO

PASO No	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa		STCOM	LISTO ...
2	Ingresar datos en forma rectangular	Re IM		LISTO ...
	En forma polar.	MAG $\theta$		LISTO ...
3	Se repite el paso 2 las veces que se necesite.			
4	Ya que se han ingresado los datos requeridos, se ejecutan las operaciones deseadas (operaciones complejas).			
5	Para ver el resultado, que quedó en el "X" del STCOM, se oprime  o .		* *	Re= IM= MAG= $\angle$ =
6	Si se desea ejecutar el programa de nuevo.			LISTO...
7	Para "borrar" el último ingreso o resultado.			LISTO...



## DIAGRAMA DE FLUJO



## EJEMPLO 1

60

Sean :  $A=30.48/26.3^\circ$  y  $B=8.6 - j\sqrt{11.5}$ , calcule la suma de ambos. ¿Cuál es el resultado en forma rectangular?

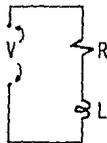
ENTRADAS/FUNCIONES

PANTALLA

<code>XEQ</code> STCOM	LISTO
30.48 <code>ENTER</code> 26.3 <code>PD</code>	LISTO
8.6 <code>ENTER</code> 11.5 <code>SQRT</code> <code>CHS</code> <code>RE</code>	LISTO
<code>+</code> **	LISTO
<code>A</code>	Re= 35.92
<code>R/S</code> *	IM= 10.11
<code>R/S</code> *	LISTO

## EJEMPLO 2

En el siguiente diagrama, ¿cuál es la corriente a través de la impedancia? La frecuencia es de 60 Hz.



$$V = \frac{440}{\sqrt{3}} / 120^\circ \text{ V}$$

$$R = 8 \text{ ohms}$$

$$L = 30 \text{ mH} \quad (X_L = 2\pi fL)$$

ENTRADAS/FUNCIONES

PANTALLA

<code>STCOM</code>	LISTO
440 <code>ENTER</code> 3 <code>SQRT</code> / 120 <code>PD</code>	LISTO
8 <code>ENTER</code> 30 <code>EEX</code> 3 <code>CHS</code> <code>PI</code> * 120 * <code>RE</code>	LISTO
<code>/</code> **	LISTO
<code>a</code>	MAG= 18.34
<code>R/S</code> *	$\angle = 65.27$
<code>R/S</code> *	LISTO

\*\* Estas son funciones del STCOM

## EJEMPLO 3.

Tenemos una línea de transmisión de potencia de - 175 mi de longitud. Sabemos que el voltaje en el extremo transmisor es  $V_T = 225.4381 \angle 6.4931^\circ$  KV línea-línea, y la corriente es  $I_T = 120.5641 \angle 35.3095^\circ$  A, ¿cuál es el -- voltaje en el extremo receptor (la carga)? Utilizaremos los siguientes datos:

$$D = 0.935559703 + j 0.015924077$$

$$B = 33.49546647 + j 137.1679057$$

$$V_R = D \cdot V_T - B \cdot I_T \quad (V_T \text{ en volts línea-neutro})$$

ENTRADAS/FUNCIONES	PANTALLA
<input type="checkbox"/> XEQ STCOM	LISTO...
.935559703 <input type="checkbox"/> ENTER <input type="checkbox"/> .015924077 <input type="checkbox"/> RE	LISTO...
225.4381 <input type="checkbox"/> EEX 3 <input type="checkbox"/> ENTER 3 <input type="checkbox"/> SQRT / 6.4931 <input type="checkbox"/> PO	LISTO...
<input type="checkbox"/> * **	LISTO...
33.49546647 <input type="checkbox"/> ENTER 137.1679057 <input type="checkbox"/> RE	LISTO...
120.5641 <input type="checkbox"/> ENTER 35.3095 <input type="checkbox"/> PO	LISTO...
<input type="checkbox"/> * **	LISTO...
<input type="checkbox"/> - **	LISTO...
<input type="checkbox"/> a	MAG=127,017.0459
<input type="checkbox"/> R/S *	$Z = 0.000000068$
<input type="checkbox"/> R/S *	LISTO ...
<input type="checkbox"/> ←	127,017.0459
3 <input type="checkbox"/> SQRT <input type="checkbox"/> *	219,999.9770

Por lo tanto el voltaje en el receptor es  $220 \angle 0^\circ$  KV (véase el ejemplo 3 del programa LINLA).

\*\* Estas son funciones del STCOM.

# LISTADO DEL PROGRAMA

02 -51  
 03 -51  
 04 FASH  
 05 \*\*  
 06 -61  
 07 FASH  
 08 \*\*  
 09 -71  
 10 FASH  
 11 \*\*  
 12 -51  
 13 FASH  
 14 \*PQ\*  
 15 -43  
 16 FASH  
 17 \*RE\*  
 18 -52  
 19 FASH  
 20 \*CLXCOM\*  
 21 -44  
 22 FASH  
 23 \*STCOM\*  
 24 -41  
 25 FASH  
 26 IEG  
 27 CLRG  
 28 SF 21  
 29 SF 27  
 30 \*LBL 01  
 31 4.02  
 32 ST 07  
 33 RDN  
 34 SF 00  
 35 OF 01  
 36 GTO 1-  
 37 \*LBL 1-  
 38 \*-0\*  
 39 GTO 02  
 40 \*LBL 14\*  
 41 \*-0\*  
 42 GTO 02  
 43 \*LBL 12\*  
 44 \*-0\*  
 45 GTO 02  
 46 \*LBL 12\*  
 47 \*-0\*  
 48 \*LBL 02  
 49 FS2 00  
 50 GTO 04  
 51 ASTO L  
 52 NEG INF L  
 53 4  
 54 ST- 02

55 \*LBL 100\*  
 56 \*LBL 100\*  
 57 \*LBL 100\*  
 58 \*LBL 100\*  
 59 \*LBL 100\*  
 60 \*LBL 100\*  
 61 \*LBL 100\*  
 62 \*LBL 100\*  
 63 \*LBL 100\*  
 64 \*LBL 100\*  
 65 \*LBL 100\*  
 66 \*LBL 100\*  
 67 \*LBL 100\*  
 68 \*LBL 100\*  
 69 4  
 70 \*LBL 02  
 71 ST- 02  
 72 RDN  
 73 SF INF 02  
 74 FS2 02  
 75 GTO 01  
 76 OF INF 02  
 77 RCL INF 02  
 78 ISG 02  
 79 RCL INF 02  
 80 ISG 02  
 81 RCL INF 02  
 82 ISG 02  
 83 RCL INF 02  
 84 ISG 02  
 85 \*LBL 04  
 86 \*LISTO...  
 87 FROM\*  
 88 GTO 0-  
 89 \*LBL 1  
 90 F-P  
 91 \*NEG\*  
 92 ARDL Y  
 93 AVIE\*  
 94 \*-0\*  
 95 ARDL Y  
 96 AVIE\*  
 97 F-R  
 98 GTO 0-  
 99 \*LBL 1  
 100 \*-0\*  
 101 ARDL Y  
 102 AVIE\*  
 103 \*-0\*  
 104 ARDL Y  
 105 AVIE\*  
 106 GTO 04  
 107 \*LBL \*CLXCOM\*  
 108 FS2 00

## LISTADO DEL PROGRAMA

107 GTO 05		
108 G		
111 GTO 05		
112*LBL *-0*		
113 CHS		
114 X<Y		
115 CHS		
116 X<Y		
117*LBL *-0*		
118 ST+ Z		
119 FDN		
120 ST+ Z		
121 FDN		
122 RTH		
123*LBL *-0*		
124 R-P		
125 1/X		
126 FDN		
127 CHS		
128 GTO 05		
129*LBL *-0*		
130 R-P		
131 FDN		
132*LBL 05		
133 FDN		
134 F-P		
135 ST+ Z		
136 FDN		
137 ST+ Z		
138 FDN		
139 F-R		
140 END		

## COMENTARIOS SOBRE EL PROGRAMA

En este programa quiero hacer notar la facilidad con que se puede disponer de "nuevas" funciones en la máquina por medio de la ASIGNACION de cualquier función, subrutina o programa (inclusive) a alguna tecla de la calculadora.

Con esto se logra transformar el teclado a una versión "especializada" para cada tipo de programa o problema, que resulta de mucha utilidad. Esto se ve claramente en este programa STCOM, ya que, con este sistema - disponemos de funciones especiales que podemos localizar en la tecla que más nos convenga para su mejor utilización y fácil identificación, por ejemplo: el ingreso de datos en forma rectangular mediante la subrutina RE, queda localizada en la tecla que tiene la letra R

Además, por supuesto, se dispone de los LBL's locales, en las diez teclas superiores, los cuales también nos sirven para definir funciones especiales dentro de cada programa.

También se distingue en este programa el empleo de SUBRUTINAS que vienen grabadas en los MODULOS DE APLICACIONES, en este caso el MATH 1, que pueden resultar una gran ayuda, ya que nos ahorra el tiempo y trabajo que invertiríamos al elaborar esas subrutinas y también nos reduce los requerimientos de memoria principal de la calculadora.

## R A I C E S

Este programa está diseñado para calcular las raíces de una ecuación de segundo grado con una sola variable. Las raíces pueden resultar reales o complejas.

Además el programa sirve para ilustrar una forma de optimizar el uso de la calculadora mediante el manejo adecuado del STACK, ya que, con toda intención, en este programa no se utiliza ningún registro de la memoria principal. Se anexa una hoja de REGISTROS DEL STACK donde se aprecian los movimientos que suceden en el transcurso del algoritmo de resolución.

## VARIABLES Y ECUACIONES.

Las variables que se utilizan son las siguientes:

a = coeficiente del término de segundo grado.

b = coeficiente del término de primer grado.

c = coeficiente del término independiente.

La fórmula general de las cuadráticas es:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Manipulando algebraicamente a ésta, llegamos a la expresión más adecuada para su utilización en la calculadora, que es:

$$x_{1,2} = \frac{-b}{2a} \pm \sqrt{\left(\frac{-b}{2a}\right)^2 - \frac{c}{a}}$$

Las raíces complejas se producen cuando el discri-

minante es menor que cero, de otra manera las raíces son reales.

Los datos que se necesitan ingresar son los coeficientes "a", "b" y "c" de la ecuación, pero ordenados en la forma siguiente:

$$aX^2 + bX + c = 0$$

Cualquiera de los coeficientes puede ser cero, y se debe ingresar como tal.

#### USO DEL PROGRAMA.

La utilización del programa es muy sencilla, sólo se deben ingresar los datos con la secuencia:

a  b  c

Una vez ingresados estos datos, para obtener los resultados se debe de oprimir la tecla de  .

Si las raíces son reales aparecerán los resultados como: [X1= ], y [X2= ]; pero si son complejas se verá primero la parte real en la forma siguiente: [Re= ], y la parte imaginaria de las dos raíces aparecerá como: - [IM=+/- ], esto es porque siempre son raíces conjugadas, es decir del mismo valor aunque con diferente signo en la parte imaginaria.

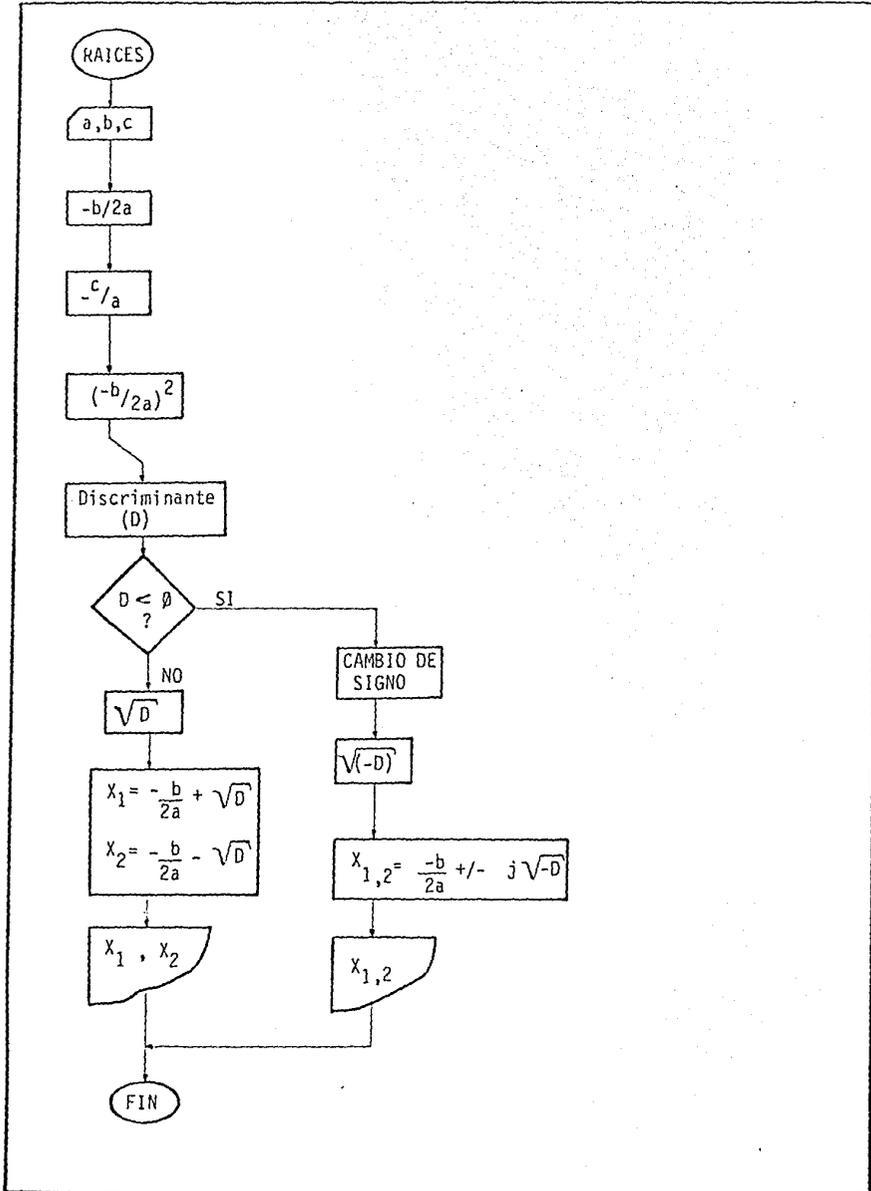
Después de mostrar los resultados el programa regresa automáticamente al inicio y solicita nuevos datos para otro problema.

## INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa		<input type="checkbox"/> XEQ RAICES	DATOS = ? a,b,c
2	Se ingresan los datos	a	<input type="checkbox"/> ENTER	
		b	<input type="checkbox"/> ENTER	
		c	<input type="checkbox"/> R/S	
3	Inmediatamente aparecerán las respuestas, - reales - o complejas		<input type="checkbox"/> R/S  ( <input type="checkbox"/> R/S )	X1= X2= ( Re=     ) ( IM=+/- )
4	Para comenzar otro cálculo, basta con oprimir - R/S.  Sigue el paso 2.		<input type="checkbox"/> R/S	DATOS=? a,b,c



# DIAGRAMA DE FLUJO



## EJEMPLO 1

Calcule las raíces de la siguiente ecuación:

$$3x^2 = 6x - 7$$

Primeramente se debe pasar la ecuación a su forma canónica Esto es:

$$3x^2 - 6x + 7 = 0$$

ENTRADAS/FUNCIONES

PANTALLA

[XEQ] R A I C E S

DATOS =?

3 [ENTER] 6 [CHS] [ENTER] 7 [R/S]

Re= 1.000

[R/S]

IM=+/- 0.577

De esto tenemos que la solución es:  $1 \pm j 0.577$

## EJEMPLO 2.

¿Cuáles son los polos de la función de transferencia que se muestra?

$$T(s) = \frac{8}{(s^2 - 3s + 2)s}$$

Claramente observamos que la ecuación a analizar es:  $s^2 - 3s + 2 = 0$

ENTRADAS/FUNCIONES

PANTALLA

[XEQ] RAICES

DATOS=?

1 [ENTER] 3 [CHS] [ENTER] 2 [R/S]

X1 = 2

[R/S]

X2 = 1

Entonces, los polos son:

$$S_1 = 1$$

$$S_2 = 2$$

$$S_3 = \emptyset$$

Este último se obtiene directamente por inspección de la función.

## LISTADO DEL PROGRAMA

```
01 *LBL "FINCES"  
02 *LB: 0  
03 *DPTC:--" a b c"  
04 PROMPT  
05 *X1: 2  
06 *S1: 2  
07 ,  
08 -2  
09 ,  
10 ENTER  
11 ENTER  
12 *X2:  
13 *R1  
14 -  
15 *X:0?  
16 GTO 01  
17 SORT  
18 *S1- 2  
19 +  
20 *X1=  
21 ARCL 2  
22 PROMPT  
23 *X2=  
24 ARCL 1  
25 PROMPT  
26 GTO 06  
27 *LBL 01  
28 CHG  
29 SORT  
30 *Re=  
31 ARCL Y  
32 PROMPT  
33 *IM+/-  
34 ARCL X  
35 PROMPT  
36 GTO 06  
37 END
```

## COMENTARIOS SOBRE EL PROGRAMA.

Como se mencionó anteriormente, la hoja de REGISTROS DEL STACK fue ideada con la finalidad específica de mostrar en forma sencilla cuáles son los movimientos que ocurren con los números contenidos en el STACK en el transcurso de cualquier programa.

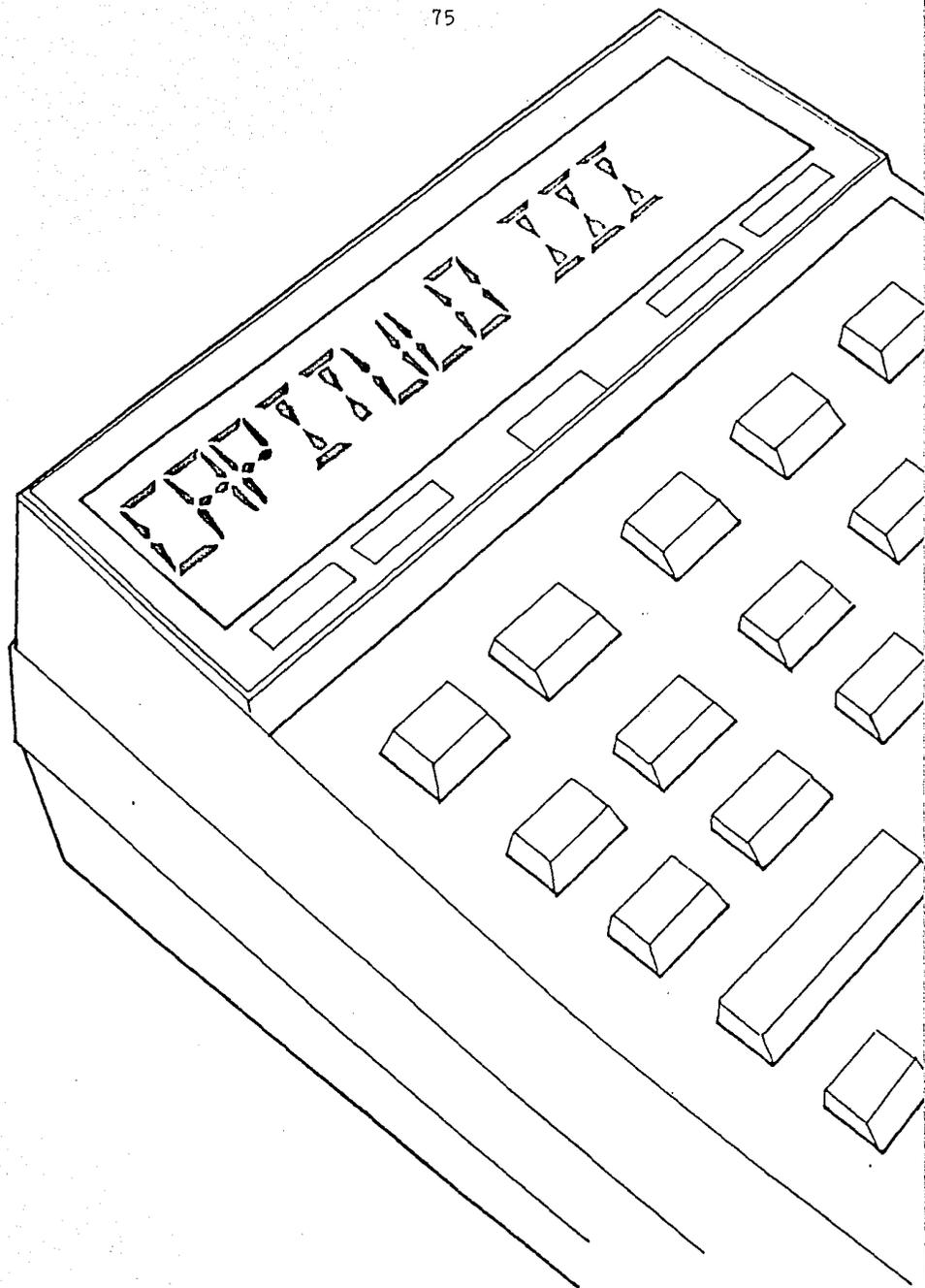
Particularmente, en este caso, veremos qué sucede en dichos registros a lo largo de la ejecución del programa RAICES, hasta llegar a la obtención de los resultados.

Los movimientos anotados en esta hoja son consecuencia directa de las instrucciones que aparecen en el listado del programa.

La hoja de REGISTROS DEL STACK que se muestra a continuación representa el desarrollo del programa en términos de las variables de la ecuación cuadrática general. El proceso de la ejecución se puede seguir en el listado o en el diagrama de flujo, siendo más recomendable en el primero.

Los números que se encuentran debajo de cada casilla indican la línea de la instrucción que originó ese estado (de la casilla).



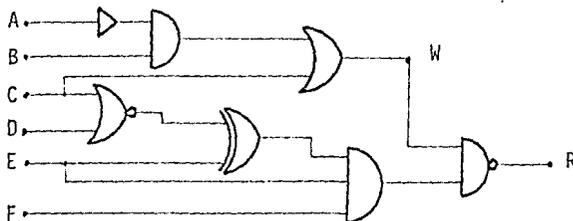


## BOOLE

El álgebra booleana es un área de las matemáticas que ha pasado a ocupar un lugar prominente con el advenimiento de la computadora digital.

Si tenemos un grupo de compuertas lógicas conectadas entre sí formando un circuito complejo y es necesario analizarlo para cada una de las posibles entradas, nos encontramos que esto es una tarea larga y cansada cuando el número de entradas ( $N$ ) es grande, digamos 5 ó más, ya que la cantidad de posibilidades es igual a  $2^N$ .

Por ejemplo, en el circuito que se muestra a continuación, se buscan las condiciones de salida final y también en el punto marcado con  $W$ , para todas las combinaciones de entrada.



Analizar manualmente los dos puntos para todas las opciones (64+8 variantes) tomaría un tiempo considerable.

Con el programa BOOLE es posible efectuar este proceso fácilmente, el único requisito necesario es plantear una ecuación que describa adecuadamente la conexión de los circuitos lógicos. No es indispensable que esta función sea la más simplificada.

## VARIABLES Y ECUACIONES.

Para pasar del circuito lógico a la ecuación booleana se relacionan las variables por medio de los operadores lógicos siguientes:

Y (AND), O (OR), OEX Y NEGACION (NOT), este último también es conocido como INVERSOR o COMPLEMENTO.

Su equivalencia con los símbolos en el circuito lógico se muestra a continuación.

OPERADOR	NOMBRE	SIMBOLO
.	Y (AND)	
+	O (OR)	
$\oplus$	OEX	
$\bar{A}$	INVERSOR	

En este texto el inversor se designará con una raya sobre la(s) variable(s) afectada(s), de esta forma:  $\bar{A}$ . -- Usualmente al escribir una ecuación, se suprime el punto, que es el signo del operador Y (AND), y la operación queda definida como AB en lugar de A·B. Es común juntar los símbolos de los operadores Y, O y OEX con el del inversor, en estos casos éste se representa con un pequeño círculo (o) ya sea en las entradas o en la salida de cada compuerta.

En el programa se pueden usar hasta 10 variables -- (entradas), que son: A,B,C..., I y J; la respuesta final es R.

Con base a lo anterior las funciones equivalentes al circuito de ejemplo; en los puntos W y R, serían :

$$W = \overline{AB} + C$$

$$R = \overline{(\overline{AB} + C) ((C+D)+E)EF}$$

### USO DEL PROGRAMA.

Para que la calculadora analice la función, es necesario que ésta se grabe con una subrutina en la memoria de la máquina, el nombre (LBL) que se le ponga puede ser cualquiera, pero con una condición: no debe tener más de seis caracteres alfanuméricos.

El programa está diseñado para manejar hasta diez variables en una sola función, las variables A,B,C,...y J corresponden a los registros 01, 02, 03,... y 10 respectivamente.

La ejecución de las operaciones se hace siguiendo el método RPN. Para llevar al STACK el valor de A, llamamos el contenido del registro 01, con la función **RCL** ; para B el registro 02; etc. Al dejar resultados pendientes debe tenerse cuidado en no sobrepasar el cupo del - - STACK, en caso necesario se pueden usar registros auxiliares.

Para que los operadores lógicos se puedan usar fácilmente, de forma similar a una operación aritmética, - quedarán asignados automáticamente a las siguientes teclas.

OPERADOR	CLAVE DE TECLA	LETRA	OBSERVACIONES
Y	7 1	Y	Este operador equivale a la multiplicación común, no es asignación real
OEX	6 4	X	
O	4 2	O	
NEG.	4 1	N	

De este punto en adelante al referirnos a los operadores lógicos en el teclado, lo haremos por su letra correspondiente.

Después que se ha grabado la función como una rutina, para regresar al programa BOOLE, se debe oprimir **[R/S]**, en donde se habrá realizado automáticamente la asignación del propio programa, con ese fin.

Para que lo anterior quede totalmente comprendido, procedamos a definir como subrutina la función del punto W.

#### ENTRADAS/FUNCIONES

**[GTO]** **[.]** **[.]** **[PRGM]**  
**[LBL]** **[ALPHA]** ABC **[ALPHA]**  
**[RCL]**  $\emptyset 1$   
**[N]**  
**[RCL]**  $\emptyset 2$   
**[Y]**  
**[RCL]**  $\emptyset 3$   
**[O]**  
**[PRGM]**

#### COMENTARIOS.

Primero debemos ir al final del archivo de programas, luego se activa el modo PRGM; el nombre de la subrutina es cualquiera, (ABC). La variable A se representa por el R  $\emptyset 1$ . N es el inversor. El R  $\emptyset 2$  -- equivale a B, luego se efectúa la operación  $\bar{A}B$ . Llamamos C y se opera con el resultado parcial pendiente para tener el resultado final en el registro X.

El programa evalúa automáticamente todas las posibilidades de la "tabla de verdad" de la función pero, si se desea, es posible analizar sólo una combinación específica, para esto se debe oprimir la tecla **[E]**.

ESTE LIBRO DEBE  
 SER DEVOLUIDO  
 A LA BIBLIOTECA  
 DESPUES DE  
 HABER SALIDO

# INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa.		<input type="checkbox"/> XEQ BOOLE	DEF. SBR.,GTO..
2	Definir la función como subrutina, si ya está grabada siga en el paso 7 .		<input type="checkbox"/> GTO <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> PRGM	ØØ REG 1min
3	Se graba la etiqueta de la subrutina.	ABC	<input type="checkbox"/> LBL <input type="checkbox"/> ALPHA <input type="checkbox"/> ALPHA	Ø1 LBL - Ø1 LBL <sup>T</sup> ABC
4	Las variables A,B,C,...,se toman de los registros Ø1, Ø2,Ø3,..., respectivamente.	Ø1  Ø2	<input type="checkbox"/> RCL  <input type="checkbox"/> RCL	Ø2 RCL -- Ø2 RCL Ø1  Ø3 RCL Ø2
5	Se relacionan las variables con los operadores lógicos. (La función Y es equivalente a la multiplicación).		<input type="checkbox"/> N <input type="checkbox"/> O <input type="checkbox"/> X <input type="checkbox"/> Y	Ø4 XEQ <sup>T</sup> N Ø5 XEQ <sup>T</sup> O Ø6 XEQ <sup>T</sup> X Ø7 * . .
6	Terminamos la subrutina y desactivamos la modalidad PRGM .		<input type="checkbox"/> RTN <input type="checkbox"/> PRGM	
7	Se oprime <input type="checkbox"/> R/S para volver al programa principal.		<input type="checkbox"/> BOOLE	FUNCION ?
8	Ingresamos el nombre de la función. No más de seis - caracteres.	ABC	<input type="checkbox"/> R/S	NO. de VAR=?
9	Ingresamos el número de variables que aparecen en la función. Máximo diez.	n	<input type="checkbox"/> R/S	
10	Si está conectada la impresora aparecerá el letrero.			EVALUACION DE FUNCION "ABC"

## INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
11	Aparecen los nombres de las columnas: las variables y - el resultado.			ABCD... R
12	Van apareciendo todas las - combinaciones de la tabla - de verdad, con su respues- - ta, hasta completarse.		<input type="checkbox"/> R/S * . . <input type="checkbox"/> R/S *	0000... R . . 1111... R
13	Si se desea se evalúa una combinación específica, al terminar la evaluación automática o interrumpiendo ésta.		( <input type="checkbox"/> E )	ENTRADA=?
14	Se ingresa la combinación requerida y se obtiene el resultado.	1010...	<input type="checkbox"/> R/S	1010... R





## EJEMPLO 1

Obtener la tabla de verdad del circuito de ejemplo en el punto R. Nombraremos la subrutina como "EJ 1". Su ponemos que ya está grabada la función "ABC".

ENTRADAS /FUNCIONES	P A N T A L L A
<input type="checkbox"/> XEQ B O O L E	DEF. SBR., GTO . .
<input type="checkbox"/> GTO <input type="checkbox"/> . <input type="checkbox"/> . <input type="checkbox"/> PRGM <input type="checkbox"/> LBL <input type="checkbox"/> ALPHA E J 1 <input type="checkbox"/> ALPHA	Ø1 LBL T EJ1
<input type="checkbox"/> XEQ <input type="checkbox"/> ALPHA A B C <input type="checkbox"/> ALPHA	Ø2 XEQ T ABC
<input type="checkbox"/> RCL Ø3 <input type="checkbox"/> RCL Ø4 <input type="checkbox"/> D <input type="checkbox"/> N <input type="checkbox"/> RCL Ø5 <input type="checkbox"/> X	Ø8 XEQ T X
<input type="checkbox"/> RCL Ø5 <input type="checkbox"/> RCL Ø6 <input type="checkbox"/> Y <input type="checkbox"/> Y <input type="checkbox"/> Y <input type="checkbox"/> N <input type="checkbox"/> RTN	15 RTN
<input type="checkbox"/> PRGM <input type="checkbox"/> BOOLE	FUNCION ?
EJ1 <input type="checkbox"/> R/S	No. de VAR.= ?
6 <input type="checkbox"/> R/S	*EVALUACION DE LA
	*FUNCION E J 1
<input type="checkbox"/> R/S *	A B C D E F R
<input type="checkbox"/> R/S *	Ø Ø Ø Ø Ø Ø 1
<input type="checkbox"/> R/S *	Ø Ø Ø Ø Ø 1 1
⋮	⋮
<input type="checkbox"/> R/S *	1 1 1 1 1 1 Ø
<input type="checkbox"/> R/S *	ENTRADA = ?

Después que se terminó la tabla, verificamos la tercera combinación.

Ø Ø Ø Ø 1 Ø <input type="checkbox"/> R/S	Ø Ø Ø Ø 1 Ø 1
<input type="checkbox"/> R/S *	ENTRADA = ?

\* Sólo si está conectada la impresora.

## LISTADO DEL PROGRAMA

01*LBL 10001	50 STO 1
02 SIZE	51 VIEW
03 16	52 STO 12
04 X/Y?	53 ADV
05 SIZE	54*LBL 11
06 FSTC 00	55 PCL 11
07 GTO 18	56 64
08 *-0*	57 *
09 42	58 *
10 PASH	59 *
11 *-H*	60 65
12 41	61 *
13 PASH	62 CLA
14 *-X*	63*LBL 12
15 64	64 X/Y?
16 PASH	65 32
17 *BOOLE*	66 XTOP
18 64	67 RDN
19 PASH	68 150 2
20 SF 00	69 GTO 12
21 SF 27	70 *-F*
22 *-DEF. SBR.,GTO.,*	71 VIEW
23 PROMPT	72 ADV
24*LBL 18	73 *
25 CLA	74 STO 11
26 42	75 2
27 PASH	76 RCL 11
28 41	77 Y/X
29 PASH	78 1
30 64	79 -
31 PASH	80 11
32 64	81 *
33 PASH	82 STO 15
34 CF 27	83 GTO 15
35*LBL 00	84*LBL 13
36 FIX 8	85 4
37 CF 29	86 STO 13
38 RDN	87 2
39 *FUNCTION ?*	88 STO 14
40 TONE 9	89 RCL 15
41 STOP	90 INT
42 ASTO 00	91 STO 12
43 OFF	92*LBL 14
44 SF 21	93 RCL 1
45 *-NO. de VAR.=?*	94 RCL 14
46 PROMPT	95 ADV
47 STO 11	96 -
48 11	97 X=0?
49 X<Y?	98 GTO 15
50 GTO 00	99 STO 1
51 FC? 55	100 PCL 13
52 GTO 11	101 *
53 *EVALUACION de L*	102 ST+ 12
54 *-R FUNCTION*	103 2
55 XEQ *PKA*	104 STO 14

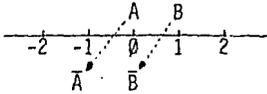
## LISTADO DEL PROGRAMA

111 S	166 GTO
112 STG 13	167*LBL
113 F1	168
114 STG 14	169
115*LBL 15	170
116 PCL 11	171 SIGA
117 STG 15	172 RTN
118 .1	173*LBL
119 .	174
120 .	175
121 .	176
122 STG 14	177*LBL
123 PCL 12	178 RTN
124*LBL 16	179
125 16	180
126 .	181
127 STG Y	182
128 INT	183*LBL
129 X:Y	184
130 FRC	185
131 10	186
132 .	187
133 STG IND 13	188
134 RDH	189
135 DSE 17	190
136 GTO 16	191
137 XEQ IND 06	192
138 CLA	193
139*LBL 17	194
140 ARCL IND 14	195
141 "F"	196
142 ISG 14	197
143 GTO 17	198
144 "F"	199
145 ARCL X	200
146 AVIEW	201
147 FS?C 61	202
148 RTN	203
149 ISG 15	204
150 GTO 17	205
151 SF 12	206
152 -----	207
153 FS? 55	208
154 AVIEW	209
155 CF 13	210
156 TONF ?	211
157*LBL 5	212
158 CF 27	213
159 *ENTRAD=1*	214
160 PROMPT	215
161 SF 01	216
162 ADV	217
163 STG 12	218
164 XCO 15	219
165 CF 27	220

## COMENTARIOS SOBRE EL PROGRAMA.

Como nota interesante de este programa veremos la forma en que quedaron implementadas las funciones lógicas AND, OR, OEX y NOT.

Estas funciones lógicas actúan sobre los registros Y y X (en el caso de la función NOT, sólo sobre X), que contienen el valor de las variables con las que se esté operando, es decir su contenido es 0 'cero' o 1 'uno' (nótese que son números, no caracteres alfanuméricos). Con base en ésto, las funciones quedan como sigue:

<u>FUNCION LOGICA</u>	<u>INSTRUCCIONES</u>	<u>COMENTARIOS</u>
AND	LBL <sup>+</sup> Y * RTN	Aprovechando que son números, esta función queda - como una multiplicación.
OR	LBL <sup>+</sup> 0 + X=0 ? RTN SIGN RTN	Se ejecuta una suma, si el resultado es cero no se modifica. Si el resultado - de la suma es diferente - de cero (1 ó 2), entonces se transforma a 1.
NOT	LBL <sup>+</sup> N DSE X ABS ABS RTN	Para cambiar de 0 a 1 y viceversa, sólo es necesario recorrer en uno (decrementar) la recta numérica. 
OEXC	LBL <sup>+</sup> X + 2 MOD RTN	Se ejecuta una suma y al - calcular el residuo respecto a 2, queda un 0 si la suma resultó 0 ó 2, y un 1 de otra manera.

El circuito integrado 555 es un dispositivo de - tecnología bipolar, aunque no es TTL. Opera entre los - valores de 4.5 a 15 volts de C.D.; este rango de capaci- dades en fuentes de alimentación lo hace fácilmente compa- tible con circuitos TTL, CMOS y circuitos discretos.

La salida de este circuito puede drenar hasta 200 mA de corriente, por lo que se pueden conectar a ella fo- cos indicadores, bobinas de relevadores, pequeñas bocinas y cualquier otra carga que opere dentro de estos rangos - de amperaje.

La siguiente figura muestra el circuito interno y la forma de conectarse externamente como CIRCUITO MONOSTA- BLE (o de disparo). Para activar este circuito es neces- ario inyectar un pulso con valor menor a 1/3 de  $V_{cc}$  a la - patita 2.

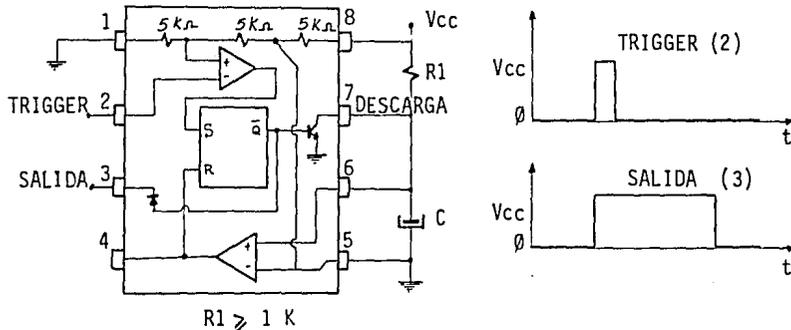


Figura 1. Circuito monostable.

El circuito 555 también puede ser usado como oscilador de onda cuadrada (multivibrador). La forma más -

sencilla de lograr esto es conectando la patita 2 a la número 6, y conectar este punto común al lado positivo del condensador, obteniéndose con esto que el voltaje del condensador esté en los dos comparadores internos. A este circuito también se le conoce como CIRCUITO ASTABLE.

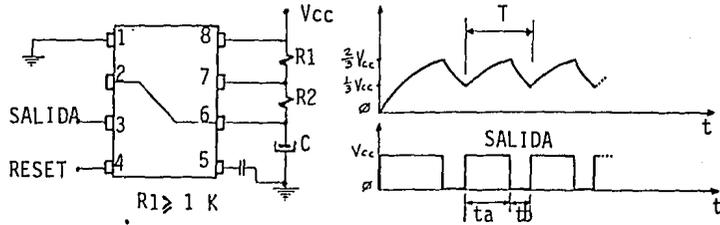


Figura 2. Circuito astable.

Podemos deducir que en el circuito astable,  $t_a$  será siempre mayor que  $t_b$ , ya que el capacitor se carga a través de  $R1+R2$  y se descarga solamente a través de  $R2$ .

También es posible usar este circuito como multivibrador con tiempos parciales independientes. En pocas palabras, se puede decir que al circuito astable se le agrega un diodo en paralelo a  $R2$ , con lo cual se logra que esta resistencia quede nulificada durante el periodo de carga del condensador, obteniendo de ese modo, que el tiempo de carga sea función de  $R1$  y el de descarga de  $R2$ . Por este motivo, al conectarse así se le conoce como CIRCUITO ASTABLE CON DIODO.

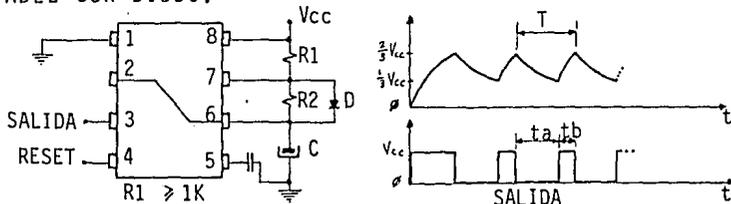


Figura 3. Circuito astable con diodo.

## VARIABLES Y ECUACIONES

Las variables que se manejan son las siguientes:

- T = período total. (segundos)  
 F = frecuencia. (hertz)  
 C = capacitancia. (faradios)  
 R1 = resistencia # 1. (ohmios)  
 (\*) R2 = resistencia # 2. (ohmios)  
 (\*) ta = tiempo parcial a. (segundos)  
 (\*) tb = tiempo parcial b. (segundos)  
 (\*) d = relación de tiempos parciales , 'duty time'.

(\*)= NO SE APLICA EN EL CIRCUITO MONOSTABLE.

Estas variables están relacionadas con las siguientes ecuaciones.

- T = ta + tb  
 T = (1.1) C R1 C. MONOSTABLE  
 T = (0.694)C(R1 + 2R2) C. ASTABLE  
 T = (0.694)C(R1 + R2 ) C. ASTABLE CON DIODO
- F = 1/T
- ta = (0.694)C(R1 + R2) C. ASTABLE  
 ta = (0.694)C R1 C. ASTABLE CON DIODO
- tb = (0.694)C R2
- d = tb/T  
 d = R2/(R1+2R2) C. ASTABLE  
 d = R2/(R1+R2 ) C. ASTABLE CON DIODO

Dependiendo de los datos que se tengan, es posible emplear una u otra y además despejar alguna de las variables que se encuentran del lado derecho de la igualdad.

#### USO DEL PROGRAMA

Este programa está elaborado para poder realizar el cálculo de una variable (cualquiera) en función de los datos disponibles. Por ejemplo si queremos que un circuito astable con diodo tenga un período  $X$ , una relación de tiempos parciales ( $d$ )  $Y$ , y conectado un capacitor de  $Z$  faradios, ¿cuáles son los valores de:  $t_a$ ,  $t_b$ ,  $R_1$ ,  $R_2$  y  $F$  ?

Al comenzar la ejecución se debe elegir con qué tipo de circuito se va a trabajar: monostable, astable o astable con diodo; para efectuar la elección basta con oprimir una tecla:  $M$ ,  $A$  o  $D$  respectivamente, en la pantalla aparecerá el tipo de circuito elegido (como confirmación) y luego aparecerá el letrero:  $[T-F C R_1 R_2]$  como recordatorio de las variables que están asignadas a las cinco teclas superiores. Aunque no aparecen en este letrero también se cuentan con las 'funciones'  $t_a$ ,  $t_b$ ,  $c$ ,  $d$  e  $I$  (vea el dibujo correspondiente en la hoja del ESTADO OPERATIVO DE LA CALCULADORA).

La 'función'  $c$  nos sirve para comenzar un nuevo cálculo dentro del mismo tipo de circuito, es decir no es necesario iniciar el programa. La secuencia de instrucciones de  $LbLc$  nos sirve para apagar (clear) todas las banderas que nos señalan si un dato es conocido.

La 'función'  $I$  nos sirve para "imprimir" TODOS los valores de las variables del circuito con que se es-

té trabajando (obviamente si está conectada la impresora, de otra manera sólo aparecerán en la pantalla).

Cuando el circuito elegido es monostable, R2 queda excluída del letrero, es decir aparece [ T-F C R1 ]. Después de que apareció este letrero se pueden ingresar todos los datos conocidos; para efectuarlos sólo se necesita marcar el valor de la variable y luego oprimir su tecla correspondiente; al hacer esto se verá en la pantalla la confirmación de ese ingreso. Tras lo anterior vuelve a aparecer el letrero de las variables/teclas, para que se ingresen uno por uno los valores conocidos.

Para que se realice el cálculo de alguna incógnita sólo es necesario oprimir la tecla deseada (sin ningún ingreso) y se ejecutará ese cálculo, apareciendo luego el resultado. Si se oprime R/S en este momento, vuelve a verse el letrero de variables/teclas; pero si se desea, se puede oprimir otra tecla cuyo valor se requiere conocer, sin necesidad de pasar por el letrero de recordatorio.

Si al estar trabajando con un circuito monostable, se intenta ingresar o calcular un dato que no es aplicable a éste, como son: ta, tb, d y R2, aparecerá un letrero como éste: [R2 = NO HAY]. (Se tomó R2 como ejemplo).

## INSTRUCTIVO DE USO

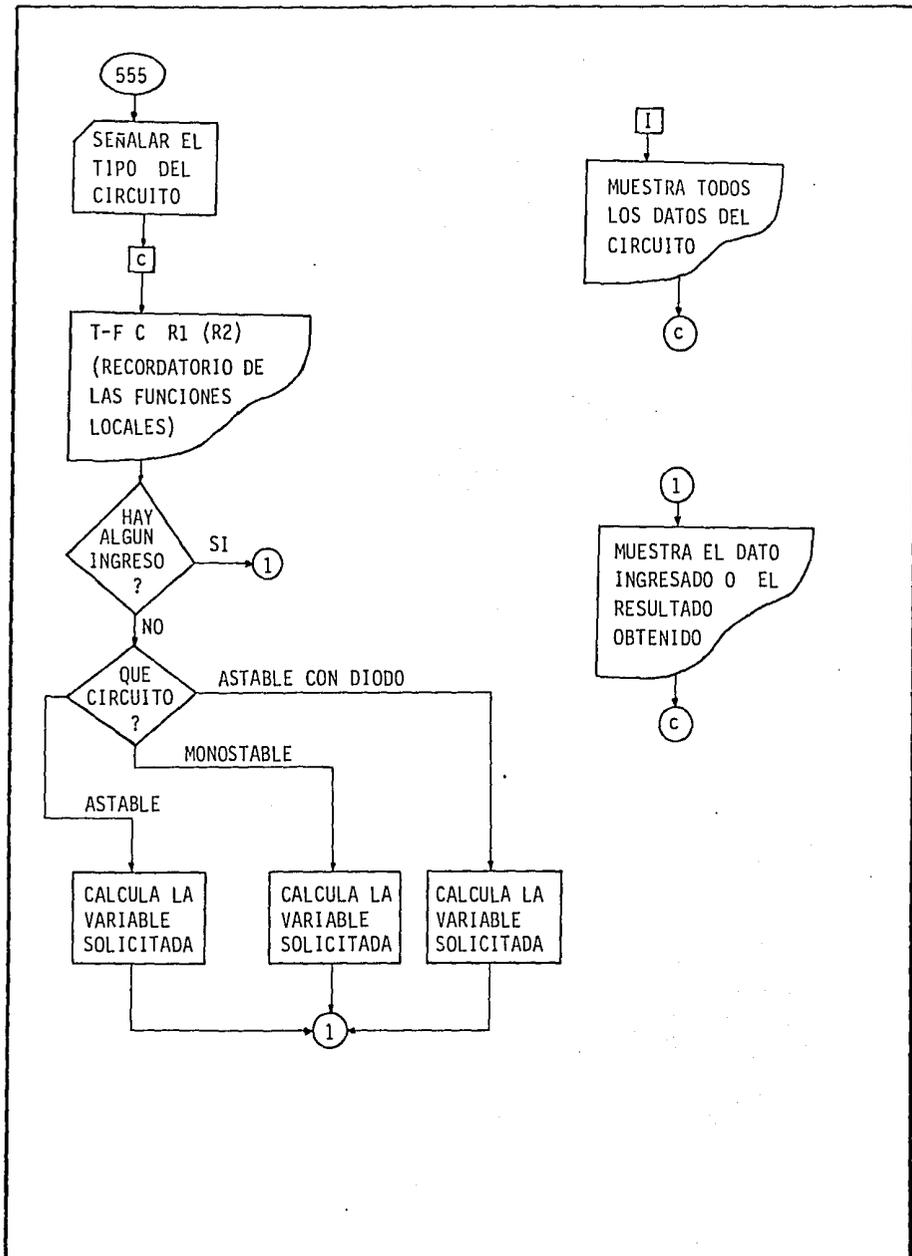
PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa.		<input type="checkbox"/> XEQ 5 5 5	:M:ON :A:S A:D:?
2	Seleccionar el tipo de circuito.	M (A) (D)		MONOSTABLE (ASTABLE) (AS.CON DIODO)
3	Aparece el letrero de recordatorio de los variables/ <u>tē</u> clas.			T-F C R1 (T-F C R1 R2)
4	Se ingresan los valores conocidos. Momentáneamente - aparece la confirmación del ingreso.  Luego sigue el paso 3, hasta completar los ingresos. Después sigue el paso 5.	ta (tb) (d)  (T) (F) (C) (R1) (R2)	<input type="checkbox"/> ta ( <input type="checkbox"/> tb ) ( <input type="checkbox"/> d )  ( <input type="checkbox"/> T ) ( <input type="checkbox"/> F ) ( <input type="checkbox"/> C ) ( <input type="checkbox"/> R1 ) ( <input type="checkbox"/> R2 )	Ta= (Tb= ) ( d= )  (PRIOD.= ) (FREC.= ) (CAP.= ) (R1= ) (R2= )
	NOTA: Si el circuito es monostable, al oprimir una de las funciones que no son - aplicables a él, aparecerá:		<input type="checkbox"/> ta <input type="checkbox"/> tb <input type="checkbox"/> d <input type="checkbox"/> R2	Ta = NO HAY Tb = NO HAY d = NO HAY R2 = NO HAY
5	Para que se realice el cálculo de alguna variable. (Véase la nota del punto 4).		<input type="checkbox"/> ta ( <input type="checkbox"/> tb ) ( <input type="checkbox"/> d ) ( <input type="checkbox"/> T ) ( <input type="checkbox"/> F ) ( <input type="checkbox"/> C ) ( <input type="checkbox"/> R1 ) ( <input type="checkbox"/> R2 )	Ta= (Tb= ) ( d= ) (PRIOD.= ) (FREC.= ) (CAP.= ) (R1= ) (R2= )

## INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
6	Si se oprime <input type="button" value="R/S"/> se vuelve al paso 3. De lo contrario puede continuar en el 5.		<input type="button" value="R/S"/>	T-F C R1 (T-F C R1 R2)
7	Si se desean ver todos los valores involucrados en el circuito...		<input type="button" value="I"/> <input type="button" value="R/S"/> * <input type="button" value="R/S"/> *	PRIO.D.= FREC.= CAP.= R1 = R2 = Ta = Tb = d = T-F C R1 (T-F C R1 R2)
8	Quando se desee comenzar un cálculo nuevo del mismo tipo de circuito. Sigue el paso 3		<input type="button" value="C"/>	T-F C R1 (T-F C R1 R2)
	NOTA: Como las funciones locales nos proporcionan acceso directo desde cualquier punto del programa, no es necesario seguir estrictamente toda la secuencia de instrucciones.			



# DIAGRAMA DE FLUJO



## EJEMPLO 1.

Tenemos un circuito integrado 555 configurado como monostable; con él debemos obtener un pulso que tenga un período de  $0.55 \pm 10\%$  segundos. ¿Qué valor de resistencia (R1) debemos usar, si se cuenta con un capacitor de  $10 \mu F$  ?

## ENTRADAS/FUNCIONES

XEQ 5 5 5  
M  
.55  T  
10  EEX 6  CHS  C  
 R1

## PANTALLA

:M:ON :A:S A:D:?  
T-F C R1  
PRIOD.= 550.0E-3  
T-F C R1  
CAP.= 10.00E-6  
T-F C R1  
R1=50.06E3

El resultado es  $R1=50 \text{ k}\Omega$  , pero si sólo tenemos resistencias de  $47 \text{ k}\Omega$  , ¿qué duración tendrá el pulso? ¿Es aceptable ese valor?

C  
47  EEX 3  R2 (¡UNA EQUIVOCACION!)  
47  EEX 3  R1  
10  EEX 6  CHS  C  
T  
550  ENTER 516.3  %CH

T-F C R1  
R2=NO HAY  
R1=47.00E3  
CAP.=10.00E-6  
PRIOD.= 516.3E-3  
-6.127 00

El período resultante es de  $0.517$  segundos y es  $6\%$  menor que el deseado; por lo tanto ésta es una solución aceptable.

## EJEMPLO 2.

Suponga que se necesita un generador de pulsos que produzca pulsos de  $100 \mu s$  con una frecuencia de  $1000$  pulsos/seg. El multivibrador astable es el generador adecuado (pongamos el valor de  $C$  como  $0.1 \mu F$ ).

## ENTRADAS/FUNCIONES

XEQ 5 5 5  
 A  
 100 EEX 6 CHS tb  
 1000 F  
 .1 EEX 6 CHS C  
 ta  
 R2  
 R1

## PANTALLA

:M:ON :A:S A:D:?  
 ASTABLE  
 T-F C R1 R2  
 Tb= 100.0E-6  
 FREC.= 1.000E 3  
 CAP.= 100.0E-9  
 Ta= 900.0E-6  
 R2= 1.443E 3  
 R1= 11.54E 3

## EJEMPLO 3.

Queremos diseñar un oscilador de tiempos parciales independientes, con  $d=0.60$  y un período total de  $0.555$  seg. El valor de  $C$  lo pondremos de  $4.7 \mu F$ .

## ENTRADAS/FUNCIONES

XEQ 555 0  
 .555 T  
 .6 d  
 4.7 EEX 6 CHS C  
 ta  
 tb  
 R1  
 R2

## PANTALLA

AS. CON DIODO  
 T-F C R1 R2  
 PRIOD.= 555.0E-3  
 d= 0.6000  
 CAP.= 4.700E-6  
 Ta= 222.0E-3  
 Tb= 333.0E-3  
 R1= 68.14E3  
 R2= 102.2E3

## LISTADO DEL PROGRAMA

00	00
01	01
02	02
03	03
04	04
05	05
06	06
07	07
08	08
09	09
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99

## LISTADO DEL PROGRAMA

109 GTO 55	164 FTO 01
110 DL 55	165 *
111 FSP 06	166 RTN
112 GTO 66	167 LBL 16
114 MEP 27	168 FHO MAR
115 FOL 05	169 GTO 30
117 *	170 LBL 27
117 RTN	171 FSP 01
118 LBL 69	172 1.029812209
119 RCL 06	173 FSP 01
120 FOL 07	174 .6931471886
121 *	175 RCL 04
122 RTN	176 RCL 05
123 LBL 65	177 FSP 02
124 PCL 07	178 ST X
125 RCL 02	179 FSP 01
126 *	180 CLX
127 RTN	181 *
128 LBL 66	182 *
129 RCL 02	183 RTN
130 1/X	184 LBL 26
131 RTN	185 RCL 01
132 LBL 05	186 FSP 01
133 MEP 05	187 1.898612269
134 SF 25	188 FOL 01
135 1/X	189 .6931471886
136 RTN	190 RCL 03
137 LBL 07	191 *
138 RCL 01	192 *
139 MEP 27	193 RTN
140 *	194 LBL 1
141 RTN	195 SF 00
142 LBL 06	196 MEP 6
143 FSP 06	197 MEP 8
144 GTO 57	198 MEP 0
145 FSP 04	199 MEP 1
146 GTO 58	200 FOL 01
147 MEP 28	201 MEP 8
148 RCL 05	202 FOL 01
149 FSP 02	203 MEP 8
150 ST Y	204 FOL 01
151 FSP 01	205 MEP 8
152 CLX	206 FOL 01
153 *	207 MEP 6
154 RTN	208 GTO 12
155 LBL 09	209 LBL 8
156 FSP 06	210 "Tan"
157 GTO 56	211 FSP 01
158 FSP 05	212 GTO 10
159 GTO 55	213 FSP 22
160 J5	214 SF 04
161 MEP 23	215 FOL 22
162 RCL 04	216 MEP 36

## LISTADO DEL PROGRAMA

217	STO 18	271	STO 45
218	SF 04	272	STO 46
220	GTO 19	274	FS? 04
221	*LBL b	275	GTO 45
222	*Tb=	276	.6931471866
223	FS? 01	277	RCL 03
224	GTO 18	278	RCL 05
225	FS? 01	279	*
226	SF 05	280	*
227	FC? 22	281	RTN
228	NEO 35	282	*LBL 40
229	STO 07	283	RCL 08
230	FS? 06	284	RCL 01
231	SF 05	285	*
232	GTO 19	286	RTN
233	*LBL d	287	*LBL 45
234	*d=	288	RCL 01
235	FS? 01	289	RCL 06
236	GTO 10	290	-
237	FS? 22	291	RTN
239	SF 06	292	*LbL 46
239	FC? 22	293	FS? 04
240	NEO 46	294	GTO 47
241	STO 08	295	FS? 05
242	FIX 4	296	GTO 48
243	GTO 19	297	RCL 05
244	*LBL 36	298	RCL 04
245	FS? 06	299	RCL Y
246	GTO 37	300	FS? 02
247	FS? 05	301	ST+ 7
248	GTO 38	302	*
249	.6931471866	303	/
250	RCL 03	304	RTN
251	*	305	*LbL 47
252	RCL 04	306	FS? 05
253	RCL 05	307	GTO 49
254	FC? 62	308	RCL 01
255	CLX	309	RCL 06
256	*	310	-
257	*	311	STO 07
258	RTN	312	GTO 49
259	*LbL 37	313	*LbL 48
260	RCL 01	314	RCL 01
261	STO Y	315	RCL 07
262	RCL 05	316	-
263	*	317	STO 09
264	-	318	*LBL 49
265	RTN	319	RCL 07
266	*LBL 38	320	STO Y
267	RCL 01	321	RCL 05
268	RCL 07	322	*
269	-	323	/
270	RTN	324	RTN

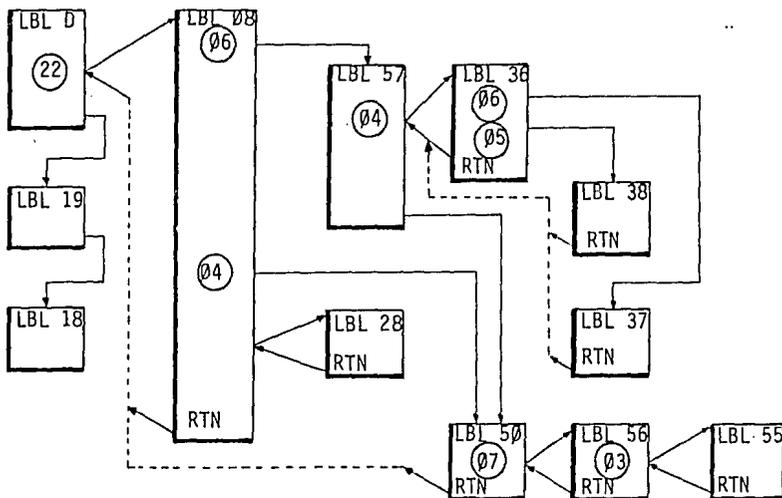
## LISTADO DEL PROGRAMA

325	FCP 07		
327	MEC 56		
328	PCL 06		
329	.6931471586		
330	RCL 03		
331	*		
332	/		
333	PCL 05		
334	FSQ 03		
335	CLX		
336	-		
337	FTN		
338	LBL 55		
339	SF 07		
340	RCL 07		
341	.6931471806		
342	RCL 03		
343	*		
344	/		
345	RTN		
346	LBL 56		
347	FSQ 03		
348	FTN		
349	MEC 56		
350	STO 05		
351	RTN		
352	LBL 57		
353	FCP 04		
354	MEC 36		
355	GTO 50		
356	LBL 58		
357	FCP 05		
358	MEC 36		
359	GTO 50		
360	END		

## COMENTARIOS SOBRE EL PROGRAMA.

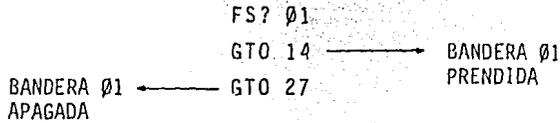
En este programa se debe hacer notar el estilo de ejecución, que hace mucho uso de subrutinas y emplea las banderas como control para poder determinar hacia qué subrutina se dirige la ejecución.

Para ejemplificar veamos el LBL D, que se refiere a R1; en el siguiente dibujo se muestran las subrutinas - en un rectángulo, con su número en la parte superior y en un círculo del número de la bandera que sirve para definir el camino en cada bifurcación. Las líneas que inician horizontalmente parten de una instrucción GTO y - las inclinadas de un XEQ .



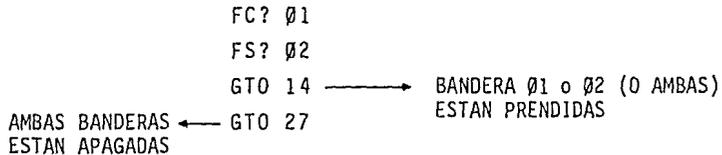
También debe mencionarse de manera especial la forma en que se diseñaron las decisiones que incluyen dos banderas a la vez. Ordinariamente una bifurcación que in

volucra a una bandera se controla con una sola pregunta, por ejemplo:



Cuando es necesario establecer dos condiciones a la vez (o una de las dos, mínimo); es decir un OR lógico, se requiere hacer lo siguiente: la primera pregunta debe estar invertida (negada) y enseguida la segunda sin modificación alguna. Veamos el siguiente ejemplo:

Si la bandera 01 o la 02 (o ambas) están prendidas, que continúe la ejecución en 14, de lo contrario en 27. Las instrucciones son:



Para ejecutar la función AND lógica con dos banderas, debemos tener presente que:

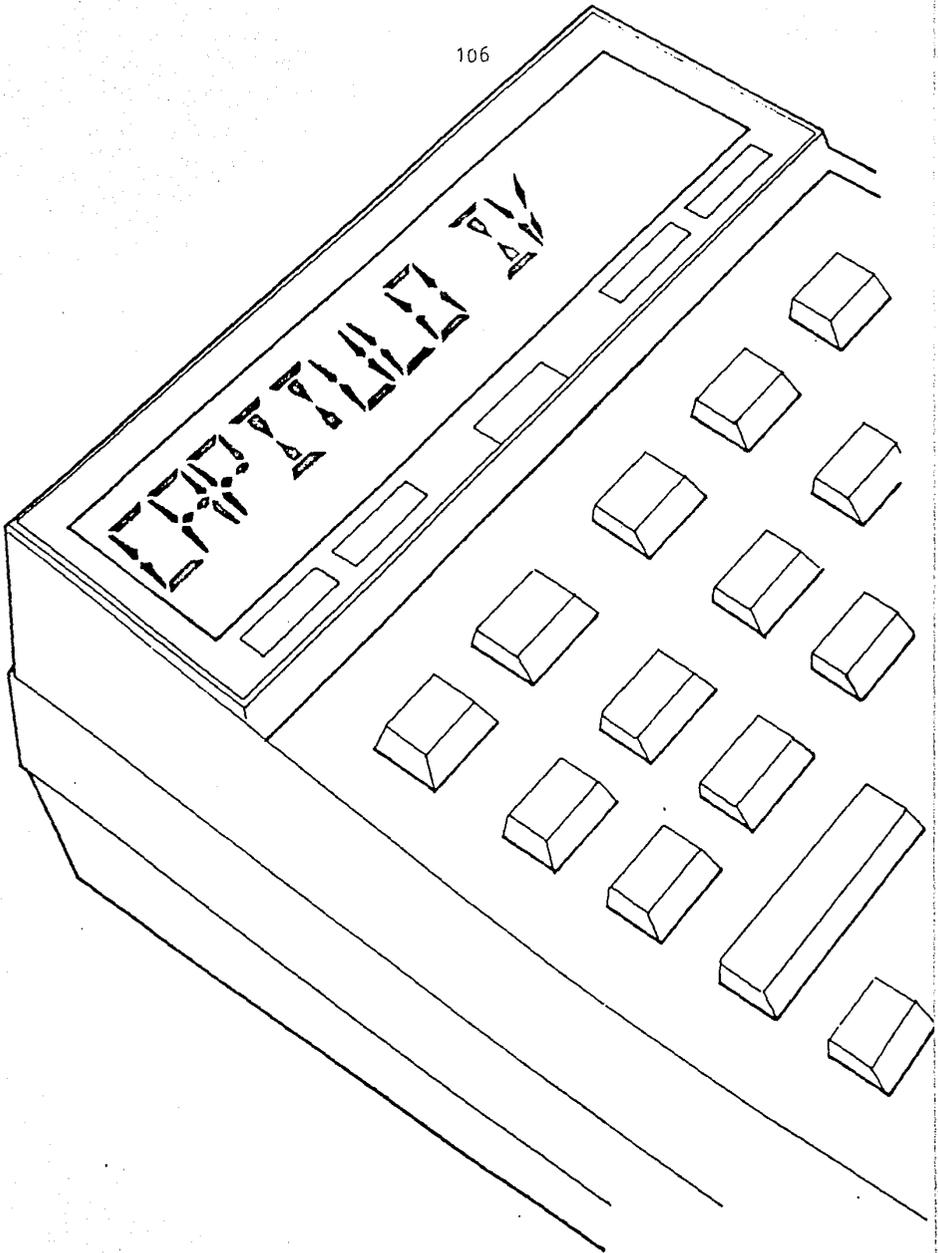
$$AB = \overline{\overline{A} + \overline{B}}$$

por lo tanto quedamos en una situación parecida a la anterior, sólo que con valores invertidos.

Ejemplo: si la bandera 01 y la 02 (ambas) están

prendidas, sigue la ejecución en 14, de lo contrario en 27. El listado es:

		FS? Ø1	
		FC? Ø2	
UNA O AMBAS ESTAN	←	GTO 27	
APAGADAS		GTO 14	→ BANDERAS Ø1 y Ø2 ESTAN PRENDIDAS



## NUMERO?

Este programa es un juego que consiste en encontrar un número comprendido entre el cero y el mil, basándose sólo en tanteos y la suerte del operador para hallar tal número.

## USO DEL PROGRAMA.

Sólo se necesita iniciar el programa y la calculadora automáticamente "esconde" un número, luego hace la pregunta [NUMERO=?] tras lo cual el operador debe ingresar una cantidad (entre 0 y 1,000). Cada vez que se mete un número y no se acierta, se indica si es alto o bajo, respecto al buscado, con un letrero y con un tono.

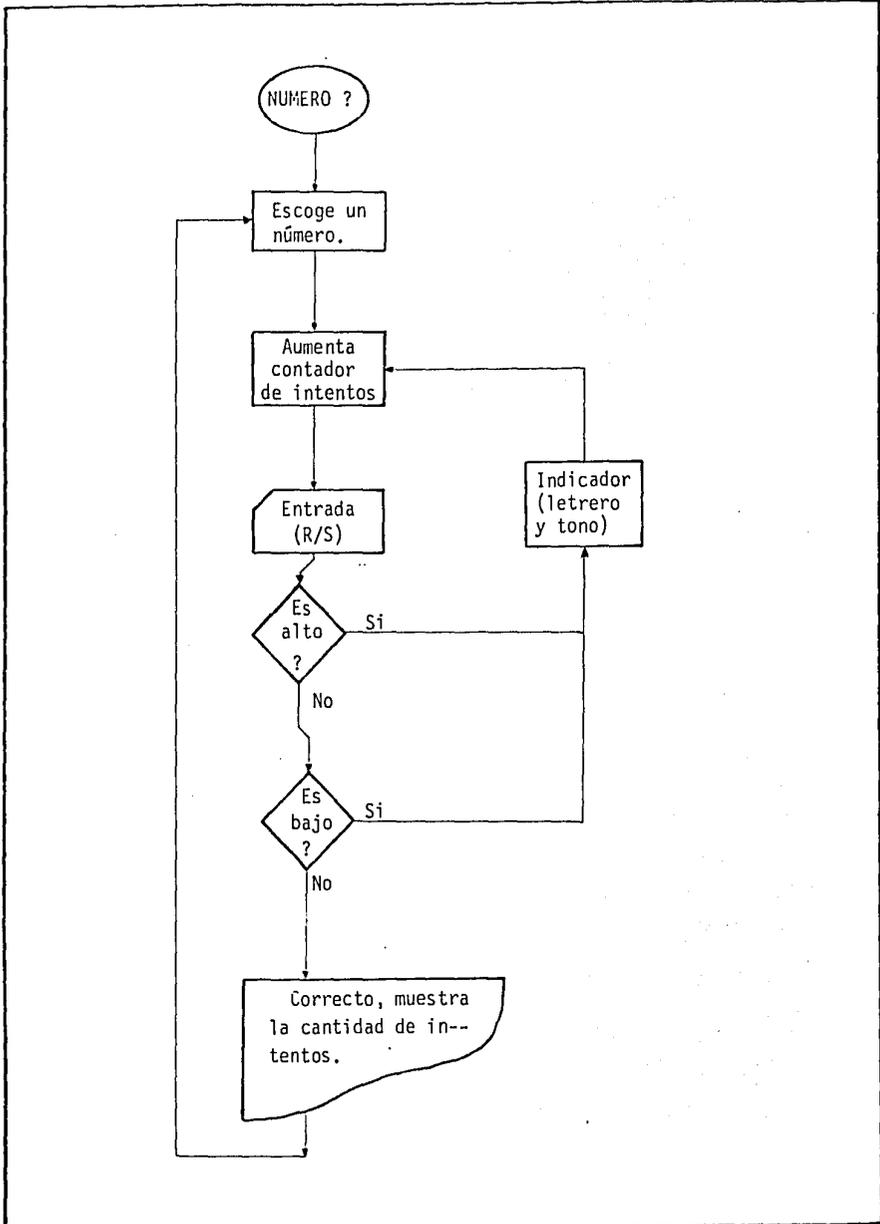
Una vez que se encuentra el número "escondido", se ve en la pantalla la cantidad de intentos que se necesitaron para lograrlo.

## INSTRUCTIVO DE USO

PASO Nº	INSTRUCCION	ENTRADA	FUNCION	PANTALLA
1	Iniciar el programa.		[XEQ] NUMERO?	NUMERO=?
2	Ingresar una cantidad.	n	[R/S]	ES ALTO (ES BAJO)
3	Se repite el paso 2 hasta hallar el número "escondido".			CORRECTO n
4	Luego se ve cuántos intentos se requirieron.		[R/S]	INTENTOS=
5	Comienza el juego de nuevo; la calculadora "esconde" otro número. Sigue el paso 2.		[R/S]	



## DIAGRAMA DE FLUJO



## EJEMPLO 1

Hallar el número "escondido".

ENTRADAS/FUNCIONES

PANTALLA

EQ NUMERO?

NUMERO=?

512 R/S

ES ALTO

230 R/S

ES BAJO

360 R/S

ES ALTO

256 R/S

ES BAJO

285 R/S

ES ALTO

267 R/S

ES BAJO

274 R/S

ES BAJO

278 R/S

ES ALTO

276 R/S

CORRECTO 276

R/S

INTENTOS=9

R/S

(comienza otro juego)

NUMERO=?

## LISTADO DEL PROGRAMA

01\*LBL "NUMERO?"  
02 F1: 0  
03 .  
04 LAST:  
05 ABS  
06 .7654321  
07 ST+ Y  
08 \*  
09 FRC  
10 I E3  
11 \*  
12 INT  
13\*LBL 0:  
14 I  
15 ST+ Z  
16 F0n  
17 "NUMERO?"  
18 PROMPT  
19 X=Y?  
20 GT0 02  
21 X>Y?  
22 "ES ALTO"  
23 X<Y?  
24 "ES BAJO"  
25 RVIEW  
26 X>Y?  
27 TONE 9  
28 X<Y?  
29 TONE 0  
30 F0n  
31 GT0 01  
32\*LBL 0:  
33 "CORRECTO "  
34 ARCL X  
35 RVIEW  
36 BEEP  
37 "INTENTOS="  
38 ARCL Z  
39 RVIEW  
40 TONE 5  
41 TONE 7  
42 GT0 "NUMERO?"  
43 END

## COMENTARIOS SOBRE EL PROGRAMA.

El único comentario que quiero hacer en este caso es respecto a que este instrumento llamado calculadora - programable (o una computadora, en su caso) es realmente capaz de realizar cualquier cosa que nos imaginemos, incluyendo juegos de diversos tipos, con el único requisito de que sepamos indicarle a la máquina, qué es lo que queremos que haga (con todas las variaciones que pueden presentarse) en forma detallada y de acuerdo al lenguaje propio de la máquina.

## CONCLUSIONES

Los programas que se presentaron en esta tesis tuvieron dos objetivos principales: el primero ofrecer un grupo de ellos, que se puede utilizar directamente en la solución de problemas de ingeniería; para esto he procurado presentarlos en la forma más sencilla posible (aunque totalmente documentados) para que se comprendan fácilmente.

El segundo objetivo que trata de cumplir es que estos programas sirven como ejemplo de las muchas aplicaciones que tiene una máquina de este tipo y, tomando en cuenta sus alcances, se refuerce su estudio en la etapa de formación de los ingenieros, como antes eran objeto de estudio las tablas de logaritmos y la regla de cálculo, o sea: debe considerársele una herramienta de trabajo muy importante por su rapidez y eficiencia al desarrollar cualquier trabajo.

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

- SYNTHETIC PROGRAMMING ON THE HP-41C  
William C. Wickes  
Larken Publications. Corvallis, Or., U.S.A. 1980.
  
- CALCULATOR TIPS AND ROUTINES ESPECIALLY FOR THE HP-41C  
John Dearing  
Corvallis Software Inc., Corvallis, Or., U.S.A. 1981
  
- OWNER'S HANDBOOK AND PROGRAMMING GUIDE FOR THE HP-41C/CV  
y STANDARD APPLICATIONS HANDBOOK  
Hewlett-Packard Co., Corvallis, Or., U.S.A. 1980
  
- HANDBOOK OF ELECTRONIC DESIGN AND ANALYSIS PROCEDURES  
USING PROGRAMMABLE CALCULATORS  
Bruce K. Murdock  
Van Nostrand Reinhold Co., New York, N.Y., U.S.A. 1979
  
- 555 COOKBOOK  
Roland S. Thelts  
Tab Book Inc., Blue Ridge Summit, U.S.A. 1983
  
- KEY NOTES  
Hewlett-Packard Company, Corvallis, Or., U.S.A. 1982
  
- INGENIERIA ELECTRONICA  
Charles L. Alley y Kenneth W. Atwood  
Ed. Limusa, S.A., México, D.F. 1979
  
- INTRODUCCIÓN A LA INGENIERIA Y AL DISEÑO EN LA INGENIERIA  
Edward V. Krick  
Ed. Limusa, S.A., México, D.F. 1973

## I N D I C E

AGRADECIMIENTOS . . . . .	2
INTRODUCCION . . . . .	5
CAPITULO I . . . . .	9
-Conceptos Generales . . . . .	.10
-Técnicas de programación. . . . .	.18
-Formatos de Presentación. . . . .	.23
-Convenciones Utilizadas . . . . .	.38
CAPITULO II. PROGRAMAS SOBRE MATEMATICAS . . . . .	.41
-CAMBASE . . . . .	.42
-STCOM . . . . .	.53
-RAICES. . . . .	.65
CAPITULO III. PROGRAMAS SOBRE ELECTRONICA. . . . .	.75
-BOOLE . . . . .	.76
-555 . . . . .	.88
CAPITULO IV. OTROS PROGRAMAS . . . . .	106
-NUMERO. . . . .	107
CONCLUSIONES . . . . .	114
BIBLIOGRAFIA . . . . .	115