

4
EJ 2

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



FACULTAD DE INGENIERIA



AUTOMATIZACION DEL CALCULO DE UNA POLIGONAL TOPOGRAFICA CERRADA EN GW - BASIC

TESIS PROFESIONAL

QUE, PARA OBTENER EL TITULO DE
INGENIERO TOPOGRAFO Y GEODESTA

P R E S E N T A :

MARIA DE LA PAZ GARCIA MEDINA

MEXICO, D. F.

1993

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

<i>Capítulo</i>	<i>Página</i>
I. INTRODUCCION.	1
I.1. Definición del problema.	2
I.2. Objetivo	3
II. METODO DE PROGRAMACION.	4
II.1. Fases del desarrollo de un programa.	4
II.1.1. Definición del problema	4
II.1.2. Algoritmo	4
II.1.3. Codificación.	4
II.1.4. Depuración.	4
II.1.5. Documentación del programa.	5
II.1.5.1. La documentación del personal informático.	5
II.1.5.2. La documentación del usuario	5
II.2. Método de programación	6
II.2.1. Programación modular.	6
II.2.2. Programación estructurada	7
II.2.2.1. Estructuras básicas de control	8
II.2.2.2. Recursos abstractos.	12
II.2.2.3. Diseño descendente "arriba-abajo" (top-down).	12
II.2.3. Método Jackson.	13
II.2.4. Método Bertini.	14
II.2.5. Método Warnier.	14
II.3. Lenguajes de programación.	17
III. PROGRAMA PARA COMPROBAR EL CIERRE ANGULAR	22
III.1. Generalidades.	22
III.2. Algoritmo.	22
III.3. Límites del programa	23
III.4. Fórmulas matemáticas	23
III.4.1. Fórmulas generales y básicas.	23
III.4.2. Fórmulas usadas en el programa.	24
III.5. Ejemplo.	25
III.6. Variables utilizadas	26
III.7. Diagrama de flujo.	27
III.8. Codificación	32
III.9. Prueba de escritorio	34
III.10. Impresión del programa	36
III.11. Impresión de datos y resultados.	38

<i>Capítulo</i>	<i>Página</i>
IV. PROGRAMA PARA CALCULAR RUMBOS	39
IV.1. Generalidades.	39
IV.2. Algoritmo.	39
IV.3. Límites del programa	41
IV.4. Fórmulas matemáticas	42
IV.4.1. Fórmulas generales y básicas.	43
IV.4.1.1. Origen de la cantidad C.	43
IV.4.1.2. Interpretación de la cantidad C.	44
IV.4.1.3. Síntesis	44
IV.4.2. Fórmulas usadas en el programa.	45
IV.4.2.1. Reglas para el programa.	45
IV.5. Ejemplo.	46
IV.6. Variables utilizadas	47
IV.7. Diagrama de flujo.	48
IV.8. Codificación	72
IV.9. Prueba de escritorio	76
IV.10. Impresión del programa	79
IV.11. Impresión de datos y resultados.	83
V. PROGRAMA PARA CALCULAR AZIMUTES	84
V.1. Generalidades.	84
V.2. Algoritmo.	85
V.3. Límites del programa	86
V.4. Fórmulas matemáticas	87
V.4.1. Fórmulas generales y básicas.	87
V.4.2. Fórmulas usadas en el programa.	88
V.5. Ejemplo.	89
V.6. Variables utilizadas	90
V.7. Diagrama de flujo.	91
V.8. Codificación	94
V.9. Prueba de escritorio	95
V.10. Impresión del programa	97
V.11. Impresión de datos y resultados.	98
VI. PROGRAMA PARA COMPENSAR LA POLIGONAL.	99
VI.1. Generalidades	99
VI.2. Algoritmo	99
VI.3. Límites del programa.	101
VI.4. Fórmulas matemáticas.	102
VI.4.1. Fórmulas generales y básicas	103
VI.4.2. Fórmulas usadas en el programa	107
VI.5. Ejemplo	110

<i>Capítulo</i>	<i>Página</i>
VI.6.	Variables utilizadas. 115
VI.7.	Diagrama de flujo 117
VI.8.	Codificación. 122
VI.9.	Prueba de escritorio. 126
VI.10.	Impresión del programa. 140
VI.11.	Impresión de datos y resultados 144
VII.	PROGRAMA PARA CALCULAR RUMBOS Y DISTANCIAS. 145
VII.1.	Generalidades. 145
VII.2.	Algoritmo. 145
VII.3.	Límites del programa 145
VII.4.	Fórmulas matemáticas 146
VII.4.1.	Fórmulas generales y básicas. 146
VII.4.2.	Fórmulas usadas en el programa. 146
VII.5.	Ejemplo. 147
VII.6.	Variables utilizadas 149
VII.7.	Diagrama de flujo. 150
VII.8.	Codificación 155
VII.9.	Prueba de escritorio 156
VII.10.	Impresión del programa 158
VII.11.	Impresión de datos y resultados. 159
VIII.	PROGRAMA PARA CALCULAR EL AREA. 160
VIII.1.	Generalidades 160
VIII.2.	Algoritmo 160
VIII.3.	Límites del programa. 161
VIII.4.	Fórmulas matemáticas. 162
VIII.4.1.	Fórmulas generales y básicas 162
VIII.4.2.	Fórmulas usadas en el programa 164
VIII.5.	Ejemplo 165
VIII.6.	Variables utilizadas. 166
VIII.7.	Diagrama de flujo 167
VIII.8.	Codificación. 171
VIII.9.	Prueba de escritorio. 172
VIII.10.	Impresión del programa. 173
VIII.11.	Impresión de datos y resultados 174
IX.	RESULTADOS. 175
IX.1.	Programa para comprobar el cierre angular. 176
IX.2.	Programa para calcular rumbos. 177
IX.3.	Programa para calcular azimutes. 178

<i>Capítulo</i>	<i>Página</i>
IX.4. Programa para compensar la poligonal	179
IX.5. Programa para calcular rumbos y distancias	181
IX.6. Programa para calcular el área	182
X. DISCUSION DE RESULTADOS	183
XI. CONCLUSIONES.	189
BIBLIOGRAFIA.	191

INDICE DE FIGURAS

<i>Número</i>	<i>Descripción</i>	<i>Página</i>
1	Elementos de una poligonal topográfica cerrada.	2
2	Estructura de control	11
3	Estructuras básicas del método de Jackson	13
4	Estructuras básicas del método de Bertini	14
5	Estructuras básicas del método de Warnier	16
6	Rumbos.	39
7	Cuadrantes conocidos.	42
8	Vértices de una poligonal	42
9	Croquis de una poligonal.	46
10	Rumbo del lado 1-2.	46
11	Rumbo del lado 2-3.	46
12	Rumbo del lado 3-4.	46
13	Rumbo del lado 4-5.	46
14	Rumbo del lado 5-6.	46
15	Rumbo del lado 6-1.	46
16	Azimuthes.	84
17	Condición del cierre lineal	103
18	Levantamiento hecho en la Colonia "El Molino". Delegación Iztapalapa. México, D.F. 1993. Escala=1:5000.	175

CAPITULO I

INTRODUCCION

Antaño, todos los cálculos matemáticos, se elaboraban usando primeramente : el ábaco; después, las tablas de logaritmo; así, las reglas de cálculo y más adelante con las calculadoras. Pero, las computadoras han venido a innovar y a hacer más fáciles los mecanismos de cálculo.

He aquí, una cita de G.W.Leibniz, donde queda sintetizada la importancia de la Informática :

"No es admisible que los estudiosos y científicos en lugar de elaborar y confrontar nuevas técnicas, pierdan su tiempo como esclavos en las fatigas del cálculo, que podía ser confiado a cualquiera si se pudiera utilizar máquinas para ello".

El éxito de la computadora, se basa en su capacidad de almacenar y procesar cantidades muy extensas de información. Del mismo modo, puede agregar más datos a la información ya existente, la actualiza, la recupera y la transmite de un continente a otro a través de satélites o de líneas telefónicas. También, puede efectuar cálculos, establecer comparaciones, simular hechos y controlar operaciones científicas e industriales que están ocurriendo en la realidad.

Actualmente, es factible emplear la computadora en aplicaciones que, hasta hace tiempo habrían sido impracticables, desde el punto de vista económico ; ya que los costos de computación, han descendido gracias a las innovaciones tecnológicas que han ocurrido las tres últimas décadas.

1.1. DEFINICION DEL PROBLEMA.

La creciente demanda de estudios topográficos para la elaboración de proyectos y construcción de infraestructura, plantea la necesidad de sistematizar los cálculos correspondientes.

Uno de los problemas que se presentan con frecuencia ante el Ingeniero Topógrafo y Geodesta, es el de calcular todos los elementos que intervienen al llevar a cabo una poligonal topográfica cerrada. Dichos elementos, son: las distancias, los ángulos respectivos, las direcciones, las coordenadas de todos los vértices y el área total. Como se muestra en la siguiente figura:

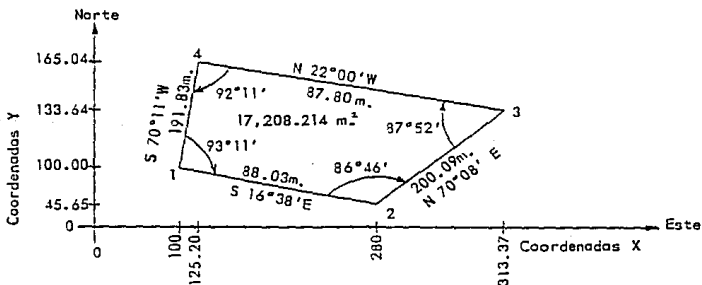



FIGURA (1)

ELEMENTOS DE UNA POLIGONAL TOPOGRAFICA CERRADA

Descripción de la figura anterior:

Vértice 1
 Distancia 88.03 m.
 Angulo  93° 11'
 Dirección S 16° 38' E
 Coordenadas 100.00, 100
 Área total 17,208.214 m²

1.2. OBJETIVO.

Por lo tanto, el objetivo del presente trabajo, será que cualquier computadora, equipada con un lenguaje BASIC, puede almacenar los programas descritos y tratados más adelante, para el procesamiento del cálculo de una poligonal topográfica cerrada y de igual manera, puede facilitar los cálculos preliminares de la misma.

CAPITULO II

METODO DE PROGRAMACION

II.1. FASES DEL DESARROLLO DE UN PROGRAMA.

Las etapas que integran el diseño de un programa y las empleadas en la preparación de esta tesis, son las siguientes :

II.1.1. DEFINICION DEL PROBLEMA.

Se tendrá una descripción clara y detallada de la aplicación que se va a desarrollar, o bien, del problema a resolver.

II.1.2. ALGORITMO.

Es el desarrollo de la secuencia lógica de pasos para la solución de un problema determinado.

II.1.3. CODIFICACION.

Se le denomina al proceso de convertir el algoritmo, en un lenguaje de programación, como el Basic, el Pascal, el Algol, el Fortran, etc.

II.1.4. DEPURACION.

Es la operación de detectar y corregir los errores existentes en el programa.

Una vez hecha las pruebas suficientes con otros datos y

verificando que no existan errores al momento de ejecutar el programa, ésta se considerará terminado.

II.1.5. DOCUMENTACION DEL PROGRAMA.

Todo material escrito sobre el programa, es la documentación. La documentación deberá ser adecuada para el uso, el mantenimiento y puesta al día del programa. Esta puede ser de dos tipos:

II.1.5.1. La documentación del personal informático.

Aquí se incluye la información técnica del programa o del conjunto de programas que integran una aplicación informática. Se divide en dos partes :

a).- Documentación interna.

Son los comentarios en los listados del programa, para permitir futuros cambios en el mismo.

b).- Documentación externa.

Son los comentarios ajenos al listado del programa, como son los algoritmos, los diagramas de flujo, la definición de variables, etc.

II.1.5.2. La documentación del usuario.

Es la descripción general de las tareas que ejecuta el programa, las instrucciones para su instalación y funcionamiento; la entrada y salida de datos, menús de opciones, las recomendaciones, etc.

II.2. METGDO DE PROGRAMACION.

La elaboración de un programa, sin seguir un método de programación, aunque éste funcione, a la larga, no será más que un conjunto de instrucciones. El desarrollo de sus etapas, generalmente son descontinuas, deficientes e indefinidas. Su documentación es incompleta: escasa en diagramas, en su descripción y actualización.

El método de programación, es la técnica empleada para que los programas sean claros, entendibles, concisos y que a la vez permitan modificaciones en su estructura, para mantenerlos al día.

Las técnicas de programación más empleadas, para seguir un método de programación, son las siguientes:

II.2.1. Programación modular.

II.2.2. Programación estructurada.

En este apartado, se tratarán las dos técnicas mencionadas, ya que éstas se utilizaron para el desarrollo del presente trabajo y porque ambas son complementarias: primero, el problema se dividió en varias partes (en módulos) y segundo, en cada uno de ellos, se utilizó un método estructurado.

II.2.1. PROGRAMACION MODULAR.

La programación modular, es la división de un programa en módulos. Un módulo, puede ser : un programa, una subrutina (o

procedimiento), una función. Serán desarrollados de forma independiente, se harán pruebas parciales de cada uno de los módulos componentes y luego se enlazarán. Dependiendo de las funciones que realizan los módulos, se clasifican en:

- Módulos tipo raíz, director o principal.
- Módulos tipo subraíz.
- Módulos de entrada (captura de datos).
- Módulos de variación de entradas.
- Módulos de proceso.
- Módulos de creación y formatos de salida.

Hay tres aspectos básicos que deben cumplirse al momento de crear los módulos, éstos son: descripción, rendimiento y diseño.

En la descripción, se definen las funciones y objetivos del programa.

En el rendimiento, se comprueba que el programa ejecute el proceso, aprovechando al máximo todos los recursos de los que dispone.

En el diseño, se comprueba la estructura que sigue el módulo, en cuanto a la estructura de los datos y la relación entre los demás módulos.

II.2.2. PROGRAMACION ESTRUCTURADA.

La programación estructurada, es una técnica de construcción de programas, que utilizan al máximo los recursos

del lenguaje, limita el conjunto de estructuras a leer y presenta una serie de reglas que coordinan el desarrollo de las distintas etapas de la programación. Utiliza en su diseño, los siguientes principios:

11.2.2.1. ESTRUCTURAS BÁSICAS DE CONTROL.

Se dividen en tres tipos:

a).- Estructura Secuencial.

Ejecuta acciones sucesivamente, sin omitir ninguna y sin bifurcaciones.

b).- Estructura Alternativa o Condicional.

Realiza una alternativa, dependiendo del valor de una determinada condición. Son de tres tipos :

b1).- Estructura alternativa simple.

Aquella en el que el cumplimiento de la condición, implica la ruptura de la secuencia y la ejecución de una acción especificada.

b2).- Estructura alternativa doble.

Permite la elección entre dos acciones en función de que se cumpla o no la condición determinada.

b3).- Estructura alternativa múltiple.

La condición, puede tomar "n" valores enteros distintos: 1,2,3,....,n. Según se elija uno de estos valores, en la condición se realizará una de las "n" acciones.

d).- Estructura Repetitiva o Iterativa.

Aquella en que las acciones se ejecutan un número determinado de veces y dependen de un valor predefinido o el cumplimiento de una determinada condición. Los tipos de estructuras más usuales y las indicadas, con mayor frecuencia, en este trabajo, son :

c1).- Estructura Mientras (WHILE).

Determina la repetición de un grupo de instrucciones, mientras la condición inicial se cumpla.

Fases de que constan:

- * Entrada de datos.
- * Condición o condiciones.
- * Instrucciones finales o resto del proceso.
- * Salida de resultados.

c2).- Estructura Repetir-Hasta (REPEAT-UNTIL).

El número de repeticiones del grupo de instrucciones, se ejecuta hasta que la condición final deje de cumplirse.

Fases de que constan:

- * Entrada de datos.
- * Instrucciones previas y proceso.
- * Condición o condiciones.
- * Salida de resultados.

c) .- Estructura Para "Desde-Hasta" (FOR-NEXT).

Repite "n" veces el grupo de instrucciones, dependiendo del valor final. Puede incrementarse o decrementarse de 1 en 1, de 2 en 2, etc. y este puede ser positivo o negativo.

Fases de que constan:

- * Entrada de datos e instrucciones previas.
- * Lazo o bucle (conjunto de instrucciones que se repiten y ejecutan un número determinado de veces).
- * Instrucciones finales o resto del proceso.
- * Salida de resultados.

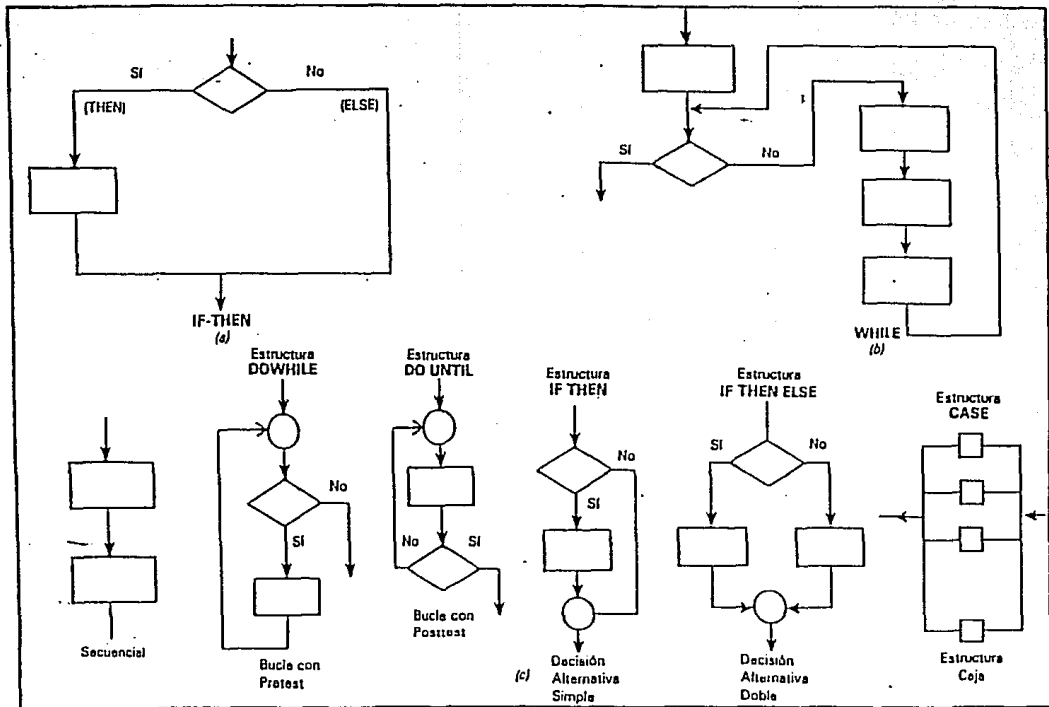


Figura (2) Estructura de control: (a) IF-THEN; (b) WHILE; (c) estructuras de control básicas en programación estructurada.

11.2.2.2. RECURSOS ABSTRACTOS.

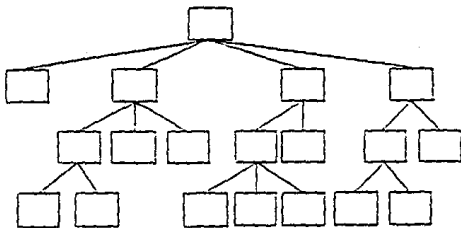
Un recurso abstracto es el proceso de la realización de diferentes pasos, hasta encontrar la solución de un problema.

Esto implica la comprobación de las distintas acciones para ser ejecutadas por la máquina. Tal y como han sido concebidas; si esto no fuese posible, será necesario descomponer las acciones en otras subacciones más elementales, continuando así el proceso hasta obtener el resultado buscado.

11.2.2.3. DISEÑO DESCENDENTE "ARRIBA-ABAJO" (top-down).

Consiste en establecer una serie de niveles de menor a mayor complejidad (arriba-abajo), que den solución al problema.

Se representa en forma de árbol.



El diseño se basa en la elaboración de distintos niveles. El primer nivel, resuelve el problema en su totalidad, el segundo, y demás niveles, son refinamientos sucesivos del primero y se sigue la metodología de recursos abstractos.

Existen diferentes métodos de programación estructurada. Los más conocidos, son :

11.2.3. METODO JACKSON.

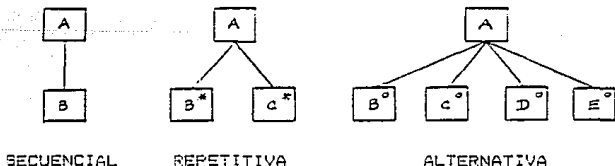
Basado en que la estructura de un programa, está en función de la estructura de los datos que manipula.

Emplea módulos, según un orden jerárquico, dentro de los distintos niveles donde se encuentra.

Cada módulo es un dato o un conjunto de datos.

Utiliza las estructuras: secuencial (sin indicación), repetitiva (se indica con un asterisco) y alternativa (se indica con la letra "O").

FIG. (3) ESTRUCTURAS BASICAS DEL METODO DE JACKSON.



Los pasos a seguir, para la aplicación de este método, son:

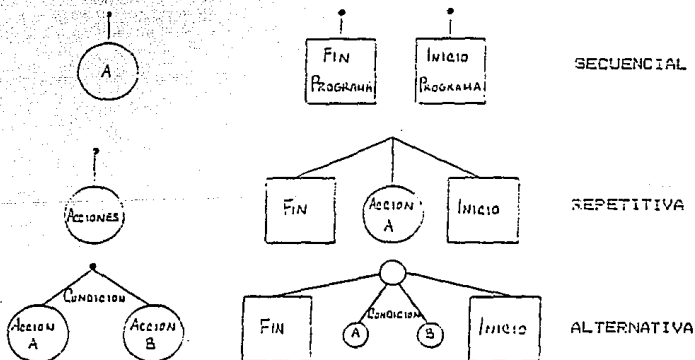
- * Establecer los datos de entrada y salida (E/S).
- * Creación de la estructura del programa, a partir de las diferentes estructuras de los datos.
- * Utilizar los recursos que posee el método, para conseguir los resultados.
- * Escribir el pseudocódigo y codificar.

II.2.4. METODO SERTINI.

Consiste en la descomposición de un problema en niveles, teniendo cada uno de ellos un principio, un conjunto de procesos y un fin.

Aquí se presenta la estructura de los programas y no las operaciones del tratamiento.

FIG. (4) ESTRUCTURAS BASICAS DEL METODO DE SERTINI. .



II.2.5. METODO WARNIER.

Este método es el empleado en la elaboración de este trabajo.

Basado en una metodología matemática que establece un único lenguaje de comunicación entre usuarios, analistas y programadores, lo que permite la comprensión de forma sencilla por cualquier programador.

El método consiste en la descomposición del problema, por niveles.

En cada nivel, se detallan los tratamientos que permiten la resolución del problema planteado.

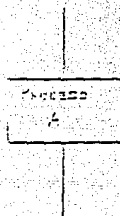
Pasos a seguir, utilizando este método:

- * Estudio de los datos de salida.
- * Estudio de los datos de entrada, teniendo en cuenta la organización de los datos en el fichero lógico de salida y analizando las posibles fases de tratamiento.
- * Cuadro sinóptico o de descomposición, formado a partir de la estructura de los datos de entrada, salida (E/S) y del tratamiento.
- * Diagrama de flujo y lista de instrucciones. El diagrama, se obtiene de manera automática del cuadro sinóptico y su uso, puede no ser imprescindible. La lista de instrucciones, su orden de ejecución, produce en casi su totalidad la escritura del programa, independientemente del lenguaje a utilizar.

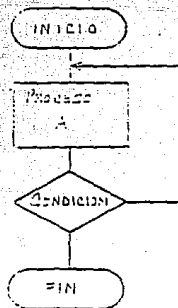
Utiliza las mismas estructuras que las de Jackson, salvo que su representación tiene variantes.

FIG. 1.5 ESTRUCTURAS BASICAS DEL METODO DE WARNIER.

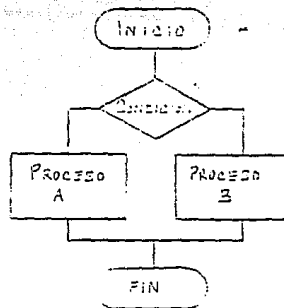
SECUENCIAL



REPETITIVA



ALTERNATIVA



A continuación, se describen los motivos por los que se utilizaron los procedimientos expuestos anteriormente:

- * Para aumentar la eficacia de la programación, su fiabilidad y su mantenimiento.

- * Asegurar que los programas sean claros en su desarrollo, adaptables, comprensibles, simples y transportables.

- * Prever: el aumento de datos y estructuras; los cambios en la organización de la información; cambios en sus formatos; etc. para la vida útil del programa.

II.3. LENGUAJES DE PROGRAMACION.

Existen algunos lenguajes por los que se puede procesar el cálculo de una Poligonal Topográfica, a saber, el PASCAL, el FORTRAN, EL LOTUS 1-2-3, los CAD/CAM/CAE, el BASIC, etc.

El lenguaje PASCAL, fue diseñado por Niklaus Wirth. Es aplicable en las áreas científicas y administrativas.

En 1957, un grupo de expertos de la IBM (International Business Machines), crea un lenguaje simbólico avanzado llamado FORTRAN (FORMula TRANslation), "traductor de fórmulas". Este lenguaje, muy parecido al lenguaje normal utilizado por el hombre, es particularmente útil para impartir instrucciones que resuelven problemas matemáticos, técnicos o científicos.

Con el FORTRAN, el hombre escribe un programa, con palabras tan comprensibles como "multiplica", "suma", etc., las cuales son transformadas automáticamente por el procesador, con el auxilio de un programa particular, al lenguaje de la máquina.

El LOTUS 1-2-3, fue diseñado por Mitchell Kapor. Es una Hoja de Cálculo que, consiste en una matriz compuesta de hileras y columnas. Tiene 3 partes lógicas totalmente integradas: hoja de cálculo, manejo de datos y gráficas. Sus aplicaciones son para las áreas científicas y administrativas.

En Ingeniería, la computadora puede ser un auxiliar para ejecutar diseños, dibujos, proyectos, etc. Para esto, se crearon

Los sistemas : CAD/CAM/CAE (Computer-Aided Design / Computer-Aided Manufacturing / Computer-Aided Engineering) , " Dibujo Asistido por Computadora / Fabricación Asistida por Computadora / Ingeniería Asistida por Computadora ". Los sistemas mencionados, son clasificados como : Dibujo Mecánico, Arquitectónico y Civil ; Ingeniería Estructural, Electrónica y Cartográfica.

El nombre : " BASIC " , proviene de las siglas que componen su significado : Beginner's All-purpose Symbolic Instruction Code (Código de instrucción simbólica para todo uso para principiantes). Util para resolver problemas científicos y administrativos.

El lenguaje que se tratará aquí, será el GW-BASIC. Las razones por las que se adoptó este lenguaje de programación de computadoras, son las indicadas a continuación:

El lenguaje para computador : GW-BASIC , (el cual es la forma genérica de BASIC de Microsoft) es el más popular y al que fácilmente se puede tener acceso, ya que casi todas las computadoras personales "PC", traen integrado este paquete. Al igual que es el lenguaje primordial y esencial para comenzar a introducirse en el mundo de la Computación.

Hoy en día, los paquetes para computadoras, como el "WORDSTAR" , el "WORDPERFECT" , el "VENTURA" , el "AUTOCAD" , el "LOTUS" , etc. , se venden por separado.

Se ha dicho que el BASIC es obsoleto, pero por lo indicado

y expuesto anteriormente, se puede utilizar lo que se tiene a la mano, aprovechando así, los programas referidos más adelante en esta tesis, mientras se adquieren los paquetes actuales y modernos.

De igual modo, será más sencillo familiarizarse con los programas, para después transferirlos a otros lenguajes.

El 1 de Mayo de 1964, a las 4:00 horas, el profesor John Kemeny y Thomas E. Kurtz del Dartmouth College, desarrollaron el lenguaje BASIC original, como una herramienta instructora para programadores principiantes. El propósito de los profesores, era diseñar un lenguaje que fuese fácil de aprender y a la vez, útil para cualquier tarea de programación. El éxito de BASIC y su extendido uso son debidos a su simplicidad, facilidad de uso y potencia de computación de propósito general.

El BASIC original de Dartmouth College, fue diseñado para lo siguiente :

- a).- Tener naturaleza de propósito general y así ser útil para escribir cualquier tipo de programa.
- b).- Permitir añadir características avanzadas más tarde.
- c).- Proporcionar interacción usuario/computadora.
- d).- Proporcionar mensajes de error claros y agradables.
- e).- Dar respuesta rápida para programas pequeños.
- f).- Hacer innecesario el conocimiento de hardware.
- g).- Proteger al usuario contra el sistema operativo de la computadora.

Alguna forma de BASIC de Microsoft está incorporada con la mayoría de los computadores usados en las casas, escuelas y negocios de hoy en día. Es conocido por varios nombres, incluyendo los siguientes: GW-BASIC, BASICA, Tandy BASIC.

GW-BASIC, funciona bajo el Sistema Operativo de Disco de Microsoft (MS-DOS). Los requisitos del sistema serán utilizar un intérprete de GW-BASIC de versión 3.22, con un sistema operativo MS-DOS de versión 3.2 ó posterior. Dichas versiones, serán las tratadas en este trabajo.

Los Comandos, las Funciones Especiales y algunas Instrucciones utilizados en el desarrollo de los programas, con el lenguaje GW-BASIC, son los descritos a continuación:

** COMANDOS **

- AUTO : Genera e incrementa automáticamente los números de línea.
- LIST : Lista todo, o una parte de, un programa en la pantalla.
- RUN : Ejecuta el programa determinado.
- EDIT : Muestra una línea específica. Coloca el cursor debajo del primer dígito del número de línea, de manera que la línea pueda ser modificada.
- SAVE : Guarda un archivo de programa en el disco.
- LOAD : Carga un archivo del disco a la memoria.
- LLIST : Lista todo, o parte del programa, que está en la memoria, a la impresora de líneas.
- LPRINT : Imprime datos en la impresora de líneas.
- KILL : Elimina un archivo de un disco.
- RENUM : Vuelve a numerar las líneas del programa.

**** FUNCIONES ESPECIALES ****

SQR : Da la raíz cuadrada de un valor.

TAB : Avanza a la posición "n" en la pantalla.

ATN : Obtiene la arcotangente de un valor, el cual está expresado en radianes.

ABS : Obtiene el valor absoluto de un número.

**** INSTRUCCIONES ****

WIDTH : Asigna la amplitud de la línea impresa, en número de caracteres, para la pantalla y la impresora de líneas.

OPEN : Establece la entrada/salida a un archivo o a un dispositivo.

CAPITULO III

PROGRAMA PARA COMPROBAR EL CIERRE ANGULAR

III.1. GENERALIDADES.

Para que una poligonal topográfica, quede como el de una figura geométrica perfecta, se comprobará el cierre angular. Motivo por el cual, se tendrá en cuenta lo siguiente:

a) Si el error está dentro de la tolerancia, el trabajo se hizo correctamente, los ángulos serán aceptados y el error se compensará para que el polígono cumpla la condición de cierre.

b) Si el error es mayor que la tolerancia, el trabajo será incorrecto. En este caso, las medidas angulares se rectificaran o repetirán.

III.2. ALGORITMO.

Como dato inicial, el programa pide el número de vértices del polígono y la aproximación del aparato topográfico utilizado. De acuerdo a lo anterior, calcula la tolerancia angular permisible.

Después, pide los ángulos (en grados decimales). Calcula la suma de los ángulos y compara tal suma con la tolerancia para saber si los ángulos son aceptados o rechazados.

Finalmente, muestra un mensaje para compensar los ángulos o seguir adelante para calcular los rumbos.

III.3. LIMITES DEL PROGRAMA.

- a) No debe usarse para poligonales menores a 3 ni mayores a 100 vértices.
- b) Los ángulos deberán ser internos y medidos a la derecha. Además, deberán calcularse en grados decimales.

III.4. FORMULAS MATEMATICAS.

III.4.1. FORMULAS GENERALES Y BASICAS.

La condición de cierre angular , para un polígono topográfico cerrado , es :

$$\sum \alpha = 180^\circ (n-2)$$

Donde: \sum = Sumatoria

α = Angulos internos

n = Número de ángulos medidos

La tolerancia angular , es :

$$T = \pm a \sqrt{n}$$

Donde: T = Tolerancia

a = Aproximación del aparato

n = Número de ángulos medidos

III.4.2. FORMULAS USADAS EN EL PROGRAMA.

La condición de cierre angular, es :

$$\text{SUMA} = 180 (NA - 2)$$

Donde: SUMA = Sumatoria de los ángulos internos

NA = Número de ángulos medidos

La tolerancia, es :

$$TA = \left(\frac{1}{AP} \right) \sqrt{NA}$$

Donde: TA = Tolerancia angular (puede ser + ó -)

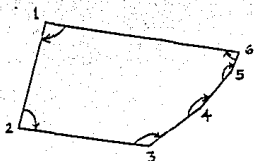
AP = Aproximación del aparato

NA = Número de ángulos medidos

La fórmula anterior, es válida para el programa y difiere de : $T = a \sqrt{n}$, ya que se divide entre 60 para expresarse directamente en grados decimales y así facilita la conversión a grados-minutos-segundos.

III.3. EJEMPLO.

Sea la siguiente figura:



Sus ángulos internos y medidos a la derecha son:

VERTICE	ANGULO
1	92° 20'
2	86 46
3	104 54
4	104 11
5	160 57
6	81 12
Σ 720° 00'	

DATOS: $NA = 6$
 $AP = 60$

FORMULAS:

$$SUMA = 180 (NA - 2)$$

$$TA = (1 / AP) \sqrt{NA}$$

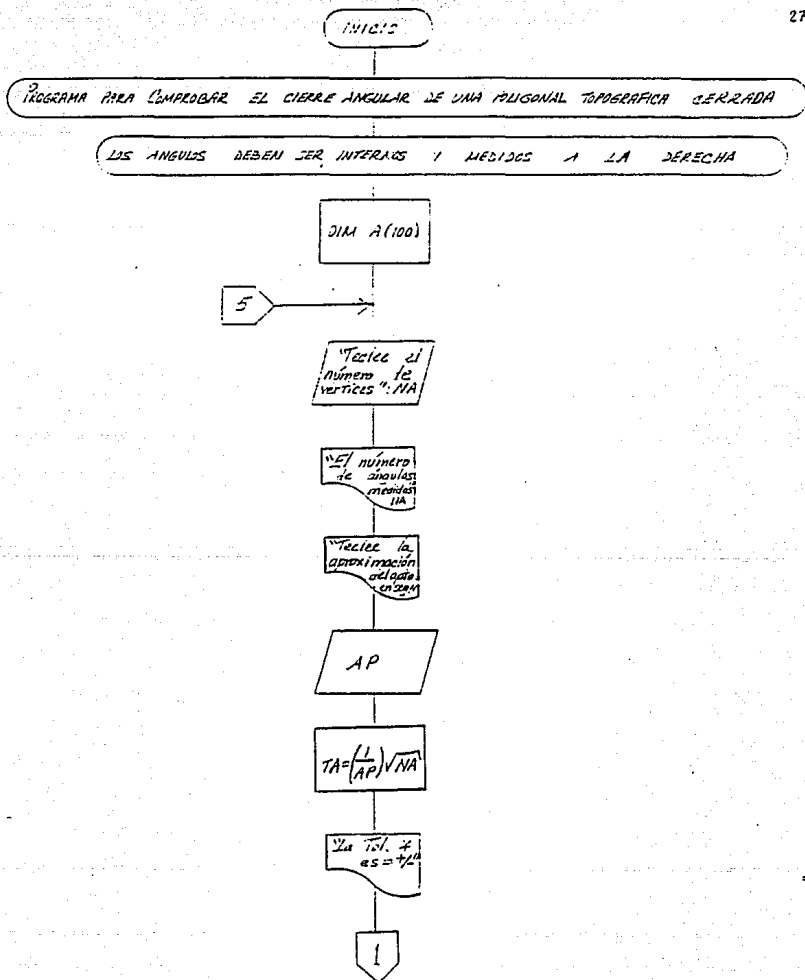
SUSTITUCIONES:

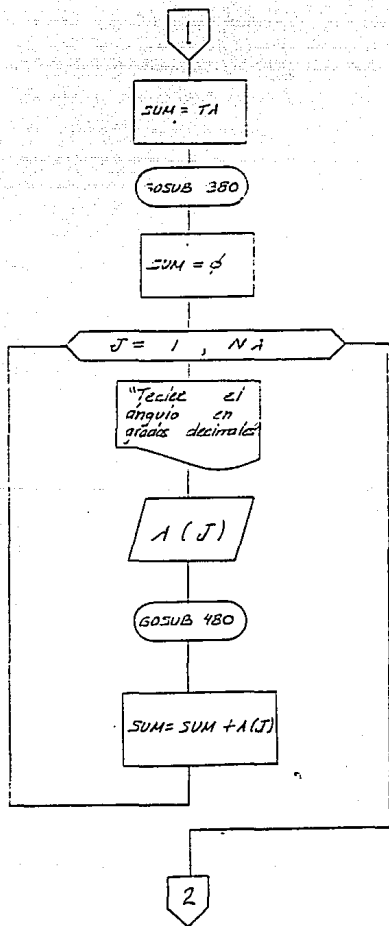
$$\begin{aligned} SUMA &= 180 (6 - 2) \\ &= 720^\circ \end{aligned}$$

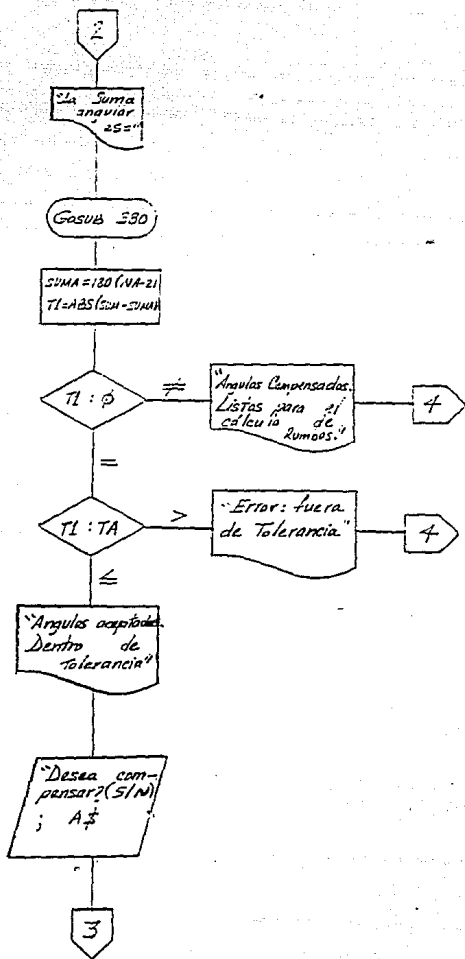
$$\begin{aligned} TA &= (1 / 60) \sqrt{6} \\ &= 0^\circ 04' 08.25'' \\ &= 0^\circ 02' 26'' \end{aligned}$$

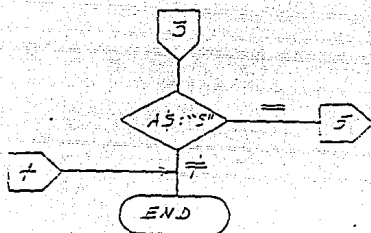
III.3. VARIABLES UTILIZADAS.

VARIABLE	DEFINICION
NA	Número de ángulos medidos a la derecha.
AP	Aproximación del aparato dado en segundos.
TA	Tolerancia angular.
SUM	Sumador de ángulos internos.
J	Contador.
A(J)	Angulo en grados decimales.
SUMA	Sumatoria de todos los ángulos internos (CONDICION DE CIERRE ANGULAR).
T1	Diferencia del sumador de ángulos internos, con el cierre angular. Es decir: $T1 = SUM - SUMA$
AS	Opción para compensar ("S") o para finalizar ("N"). Es una variable de "cadena".
	Conversión a grados, minutos y segundos: (SUBROUTINA)
H, H3, H6, H7, H10	Enteros.
H1, H4, H8, H11	Diferencias.
H2, H5, H9, H12	Multiplicados por 60.









SUB 380

SUBROUTINA PARA CONVERTIR A GRADOS-MINUTOS-SEGUNDOS

```

H = INT (SUM)
H1 = SUM - H
H2 = H1 (60)
H3 = INT (H2)
H4 = H2 - H3
H5 = H4 (60)
H6 = INT (H5)
  
```

Grados, minutos,
segundos"; H,
H3, H6

RETURN

CODSUB 780

SUBROUTINA PAKH CONVERTEIR DE SEGUNDOS-MINUTOS-SEGUNDOS

```
H7 = INT (A(J))  
HB = A(J) - H7  
H9 = HB * 60  
H10 = INT (H9)  
H11 = H9 - H10  
H12 = H11 * 60
```

```
"EL RESULTADO"  
A(J), H7,  
H10, H12
```

RETURN

III.3. CODIFICACION.

```

10 REM PROGRAMA PARA COMPROBAR EL CIERRE ANGULAR DE UNA
20 REM POLIGONAL TOPOGRAFICA CERRADA
30 REM LOS ANGULOS DEBEN SER INTERNOS Y MEDIDOS A LA DERECHA
40 DIM A(100) 'LIMITE DE 100 VERTICES
50 PRINT "TECLEE EL NUMERO DE VERTICES"
60 INPUT NA
70 PRINT "EL NUMERO DE ANGULOS MEDIDOS ES=";NA
80 PRINT "TECLEE LA APROXIMACION DEL APARATO, EN SEGUNDOS"
90 INPUT AP
100 TA=(1/AP)*SQR(NA)
110 PRINT "LA TOLERANCIA ANGULAR ES= +/- "
120 SUM=TA
130 GOSUB 380
140 SUM=0
150 FOR J=1 TO NA
160 PRINT "TECLEE EL ANGULO";J,"EN GRADOS DECIMALES"
170 INPUT A(J)
180 GOSUB 480
190 SUM=SUM+A(J)
200 NEXT J
210 PRINT "LA SUMA ANGULAR ES="
220 GOSUB 380
230 SUMA=180*(NA-2)
240 T1=ABS(SUM-SUMA)
250 IF T1=0 GOTO 290
260 IF T1<=TA THEN 340
270 PRINT "ERROR:FUERA DE TOLERANCIA"
280 GOTO 370
290 PRINT "*****"
300 PRINT "      Angulos compensados      "
310 PRINT "LISTOS para el CALCULO de RUMBOS"
320 PRINT "*****"
330 GOTO 370
340 PRINT "ANGULOS ACEPTADOS.DENTRO DE TOLERANCIA"
350 LINE INPUT "DESEA COMPENSAR?(S/N):";A$
360 IF A$="S" THEN GOTO 50
370 END
380 REM SUBROUTINA PARA CONVERTIR A GRADOS-MINUTOS-SEGUNDOS
390 H=INT(SUM)
400 H1=SUM-H
410 H2=H1*60
420 H3=INT(H2)
430 H4=H2-H3
440 H5=H4*60
450 H6=INT(H5)
460 PRINT H;"GRADOS",H3;"MINUTOS",H6;"SEGUNDOS"

```

```
470 RETURN
480 REM SUBROUTINA PARA CONVERTIR A GRADOS-MINUTOS-SEGUNDOS
490 H7=INT (A(J))
500 H8=A(J)-H7
510 H9=H8*60
520 H10=INT(H9)
530 H11=H9-H10
540 H12=H11*60
550 PRINT "EL ANGULO";J,"=";A(J);TAB(35)H7 TAB(40)H10 TAB(45);
      USING "##.##":H12
560 RETURN
```

11.9. PRUEBA DE ESCRITORIO

PROYECTO	Tesis	ARCHIVO	LANG.8	FECHA	1993
PROGRAMA	Basic	ELABORO	MPGM	H01A	34 DE 38

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
NA	6					
AP	60°					
TA	0.04082483					
SUM	0.04082483					
H	0					
H1	0.04082483					
H2	2.44949					
H3	2					
H4	0.44949					
H5	26.96939					
H6	26					
H, H3, H6	0° 02' 26"					
SUM	0					
J	1	2	3	4	5	6
A(J)	92.33334	86.76667	134.5667	164.1833	160.95	81.2
H7	92	86	134	164	160	81
H8	0.33334	0.76667	0.5667	0.1833	0.95	0.20
H9	20.0004	46.0002	34.002	10.998	57.0	12.0
H10	20	46	34	10	57	12
H11	0.0004	0.0002	0.002	0.998	0	0
H12	0.024	0.012	0.12	59.88	0	0

III.10. IMPRESION DEL PROGRAMA.

```

10 REM PROGRAMA PARA COMPROBAR EL CIERRE ANGULAR DE UNA
20 REM POLIGONAL TOPOGRAFICA CERRADA
30 REM LOS ANGULOS DEBEN SER INTERNOS Y MEDIDOS A LA DERECHA
40 DIM A(100) 'LIMITE DE 100 VERTICES
50 PRINT "TECLEE EL NUMERO DE VERTICES"
60 INPUT NA
70 LPRINT "EL NUMERO DE ANGULOS MEDIDOS ES=":NA
80 PRINT "TECLEE LA APROXIMACION DEL APARATO, EN SEGUNDOS"
90 INPUT AP
100 TA=(1/AP)*SGR(NA)
110 LPRINT "LA TOLERANCIA ANGULAR ES= +/- "
120 SUM=TA
130 GOSUB 380
140 SUM=0
150 FOR J=1 TO NA
160 PRINT "TECLEE EL ANGULO";J,"EN GRADOS DECIMALES"
170 INPUT A(J)
180 GOSUB 480
190 SUM=SUM+A(J)
200 NEXT J
210 LPRINT "LA SUMA ANGULAR ES="
220 GOSUB 380
230 SUMA=180*(NA-2)
240 T1=ABS(SUM-SUMA)
250 IF T1=0 GOTO 290
260 IF T1<=TA THEN 340
270 LPRINT "ERROR: FUERA DE TOLERANCIA"
280 GOTO 370
290 LPRINT "*****"
300 LPRINT "      Angulos compensados"
310 LPRINT "LISTOS para el CALCULO de RUMBOS"
320 LPRINT "*****"
330 GOTO 370
340 LPRINT "ANGULOS ACEPTADOS. DENTRO DE TOLERANCIA"
350 LINE INPUT "DESEA COMPENSAR?(S/N)";A$
360 IF A$="S" THEN GOTO 50
370 END
380 REM SUBROUTINA PARA CONVERTIR A GRADOS-MINUTOS-SEGUNDOS
390 H=INT(SUM)
400 H1=SUM-H
410 H2=H1*60
420 H3=INT(H2)
430 H4=H2-H3
440 H5=H4*60
450 H6=INT(H5)
460 LPRINT H;"GRADOS",H3;"MINUTOS",H6;"SEGUNDOS"

```

```
470 RETURN
480 REM SUBROUTINA PARA CONVERTIR A GRADOS-MINUTOS-SEGUNDOS
490 H7=INT (A(J))
500 H8=A(J)-H7
510 H9=H8*60
520 H10=INT(H9)
530 H11=H9-H10
540 H12=H11*60
550 LPRINT "EL ANGULO";J,"=";A(J);TAB(35)H7 TAB(40)H10 TAB(45);
    USING "##.##";H12
560 RETURN
```

III.11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE ANGULOS MEDIDOS ES= 6
 LA TOLERANCIA ANGULAR ES= +/- 1

	0 GRADOS	2 MINUTOS	26 SEGUNDOS			
EL ANGULO 1	= 92.55554		92	20	0.04	
EL ANGULO 2	= 36.76667		36	46	0.01	
EL ANGULO 3	= 134.36667		134	24	0.11	
EL ANGULO 4	= 164.18333		164	10	59.89	
EL ANGULO 5	= 160.75		160	56	59.99	
EL ANGULO 6	= 81.2		81	11	59.99	

LA SUMA ANGULAR ES=

	720 GRADOS	0 MINUTOS	0 SEGUNDOS

Angulos compensados			
LISTOS para el CALCULO de RUMBOS			

CAPITULO IV

PROGRAMA PARA CALCULAR RUMBOS

IV.1. GENERALIDADES.

La dirección de una línea, puede definirse por su RUMBO, el cual puede ser magnético o astronómico.

Un rumbo, es el ángulo que forma una línea con el Eje Norte-Sur, contado de cero a noventa grados, a partir del Norte (N) o a partir del Sur (S), hacia el Este (E) o hacia el Oeste (W).

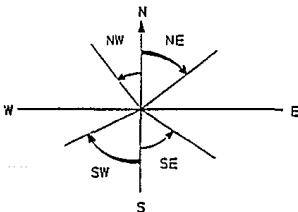


FIG. (6) RUMBOS.

El cálculo de los rumbos de los lados de una poligonal topográfica, cuando los ángulos de ésta, se miden a la derecha, puede hacerse por el Método de Transporte de Rumbos, el cual se tratará aquí.

IV.2. ALGORITMO.

Inicialmente, el programa pide el número de vértices, luego, pregunta si el rumbo observado fue medido al final o al principio de la poligonal.

Después, pide dicho rumbo (en grados decimales) y el cuadrante, inicial o final, de acuerdo a como se haya indicado.

A continuación, pide el ángulo (medido hacia la derecha y en grados decimales).

Una vez, teniendo los datos, compara el cuadrante proporcionado y calcula los rumbos. Para su cálculo, el programa está dividido en las cuatro partes siguientes :

Cuando el origen de C es al :

- a).- SUR y POSITIVO.
- b).- NORTE y POSITIVO.
- c).- SUR Y NEGATIVO.
- d).- NORTE Y NEGATIVO.

De acuerdo a lo anterior, será el cuadrante final asignado.

Pregunta en cada iteración si, se desea continuar o interrumpir. Esto puede ser útil para evitar equivocaciones al proporcionar los datos.

Al llegar al último dato, pregunta si se desea comprobar. Si la respuesta es afirmativa, hace una iteración más, preguntando nuevamente si se desea continuar. En caso de que la respuesta sea negativa, imprime el siguiente mensaje: "FIN DEL CALCULO".

Las subrutinas que contiene el programa, son para calcular la conversión de grados decimales a grados, minutos y segundos de arco.

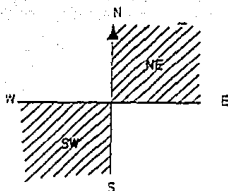
Finalmente, el programa imprime los resultados en forma de tabla, mostrando: el número de vértices; sus lados respectivos; el cuadrante y el Rumbo Magnético Calculado (R. M. C.).

IV.3. LIMITES DEL PROGRAMA.

- a) Los ángulos deberán medirse a la derecha.
- b) Como máximo, la poligonal, deberá tener 100 vértices.
- c) Los rumbos deberán tomarse: al principio o al final de la poligonal.
- d) El programa pondrá un letrero de: "Rumbo Franco", cuando éste lo sea; sólo y exclusivamente por CALCULO.
- e) Al principio, es decir, como DATO INICIAL, no admite un RUMBO FRANCO.

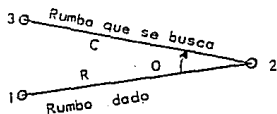
IV.1. FORMULAS MATEMATICAS.

FIG. (7) Sean los cuadrantes conocidos:



Las partes sombreadas estarán relacionadas, asimismo, las que no lo están. Es decir, el primer con el tercer cuadrante y el segundo con el cuarto cuadrante.

FIG. (8) Sean 1, 2 y 3 los vértices de una poligonal, C el Rumbo que se Busca, R el Rumbo Dado y O el ángulo horizontal medido hacia la derecha:



IV.4.1. FORMULAS GENERALES Y BASICAS.

El planteamiento del Método de Transporte de Rumbos , puede hacerse por la manera que se indica en seguida:

a) Si el rumbo está dado en los cuadrantes : NE ó SW ; al ángulo horizontal (O) , se le suma el rumbo dado (R) :

$$C = O + R$$

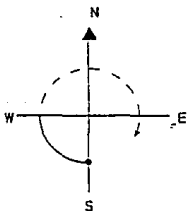
b) Si el rumbo está dado en los cuadrantes : NW ó SE ; al ángulo horizontal (O) , se le resta el rumbo dado (R) :

$$C = O - R$$

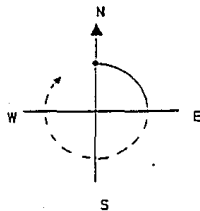
Ver Figuras (7) y (8).

IV.4.1.1. ORIGEN DE LA CANTIDAD C.

La cantidad C es un ángulo contado en el mismo sentido en el que fueron medidos los ángulos horizontales y a partir del extremo opuesto de donde se mide el rumbo dado (R).



NW ó NE



SW ó SE

IV.4.1.2. INTERPRETACION DE LA CANTIDAD C.

- a) Es un AZIMUT.
- b) Si $C < 90^\circ$ el rumbo sale directo.
- c) La cantidad C se resta de 100° o 130° .
- d) Si C es negativo, se interpreta en sentido contrario al medido.

IV.4.1.3. SINTESIS.

El procedimiento, queda sintetizado en el cuadro que a continuación se indica :

CUADRANTE DEL RUMBO DADO (R)	OPERACION NUMERICA	ORIGEN DE C
NE SW	$C = 0 + R$	S N
NW SE	$C = 0 - R$	S N
	Si C = Negativa (- C)	Cambia de sentido al medido.

IV.4.2. FORMULAS USADAS EN EL PROGRAMA.

$$C(K) = D(K) - R$$

$$C(K) = D(K) - R$$

Donde :

C (K) = Rumbo buscado.

D (K) = Angulo horizontal medido a la derecha.

R = Rumbo dado (inicial o final).

IV.4.2.1. REGLAS PARA EL PROGRAMA.

CUADRANTE FINAL				RESTAR	INTERVALO
A	B	C	D		
SW	NE	SE	NW	-	$0 < C(K) < 90$
S	N	S	N	-	$C(K) = 0$
W	E	E	W	-	$C(K) = 90$
NW	SE	NE	SW	180	$90 < C(K) < 180$
N	S	N	S	180	$C(K) = 180$
NE	SW	NW	SE	180	$180 < C(K) < 270$
E	W	W	E	180	$C(K) = 270$
SE	NW	SW	NE	360	$270 < C(K) < 360$
S	N	S	N	360	$C(K) = 360$

NOTAS :

A = Cuando el origen de C es al SUR y es POSITIVO (+).

B = Cuando el origen de C es al NORTE y es POSITIVO (+).

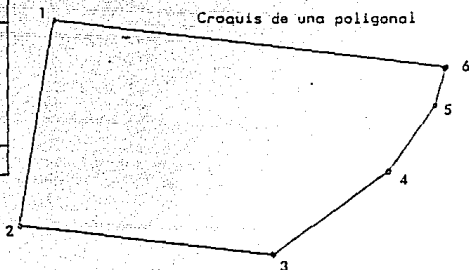
C = Cuando el origen de C es al SUR y es NEGATIVO (-).

D = Cuando el origen de C es al NORTE y es NEGATIVO (-).

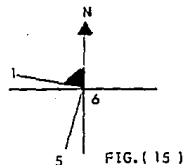
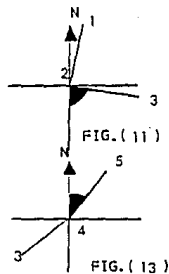
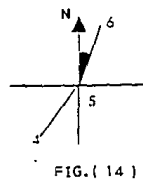
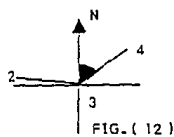
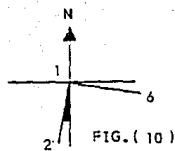
IV.5. EJEMPLO.

DATO = Rumbo dado : 1-6 = S 77° 01' E

VERTICE	ANGULO HORIZONTAL (INTERNO)
1	92° 20'
2	86° 46'
3	102° 08'
4	164° 11'
5	160° 57'
6	81° 12'
SUMATORIA = 720° 00'	

FIG. (9) :
Croquis de una poligonal

-	6-1 NW	77° 01'	
-	1	92° 20'	
Rbo. 1-2=SW	15° 19'	FIG. (10)	
+	2	86° 46'	
-		102° 08'	
-		180° 00'	
Rbo. 2-3=SE	77° 55'	FIG. (11)	
-	3	102° 08'	
Rbo. 3-4=NE	56° 39'	FIG. (12)	
+	4	164° 11'	
-		220° 50'	
-		180° 00'	
Rbo. 4-5=NE	40° 50'	FIG. (13)	
+	5	160° 57'	
-		201° 47'	
-		180° 00'	
Rbo. 5-6=NE	21° 47'	FIG. (14)	
COMPROBACION :			
-	5-6 NE	21° 47'	
+	6	81° 12'	
-		102° 59'	
-		180° 00'	
Rbo. 6-1=NW	77° 01'	FIG. (15)	



IV.3. VARIABLES UTILIZADAS.

VARIABLE:	DEFINICION:
NV	Número de vértices de la poligonal.
K	Contador numérico.
CS, CI, CC	Respuesta SI ó NO (S/N).
R	Valor inicial del rumbo (en grados decimales).
RB	Valor inicial del rumbo o rumbo buscado (en grados decimales).
CS	Cuadrante de los rumbos : NE , NW , SW ó SE .
B(K)	Angulos de la poligonal , medidos a la derecha.
C(K)	Rumbo buscado.
CC , CC	Valores absolutos del rumbo buscado : / C(K) /
SUBROUTINAS :	
(CALCULO DE : Grados-Minutos-Segundos)	
R1, R4, R7, R10	Números enteros.
R2, R5, R8, R11	Diferencias.
R3, R6, R9, R12	Multiplicados por 60.

INICIO

PROGRAMA PARA EL CALCULO DE RUMBOS
 METODO = TRANSPORTE DE RUMBOS
 CONDICION: ANGULOS MEDIDOS A LA DERECHA

DIM 2 (100)
 DIM 3 (100)
 Para 100 vértices

NV

El N°
 de vértices
 NV

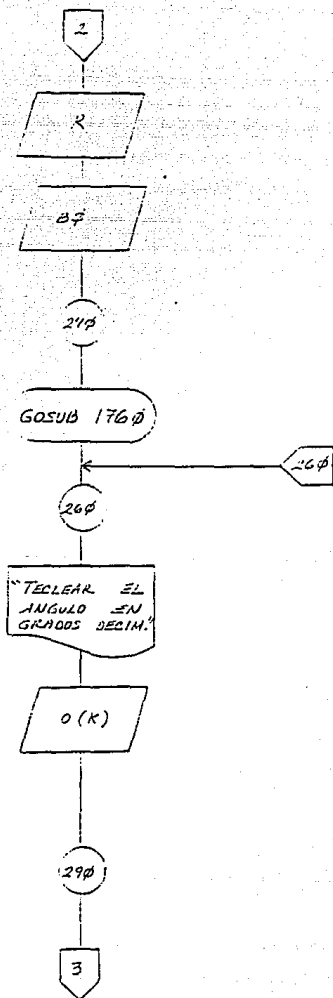
K=1

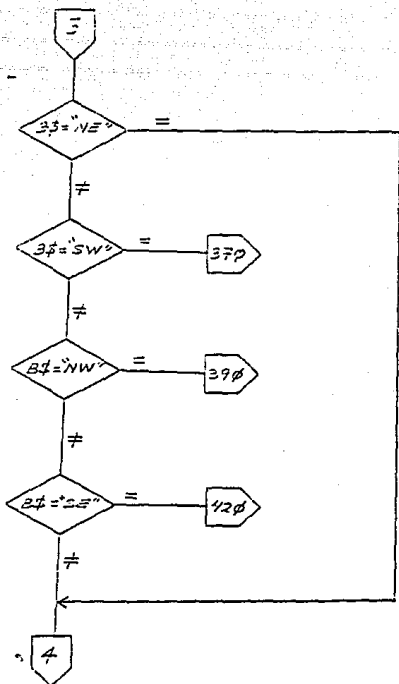
Se toma
 al rumbo
 al final? S/N
 c.s.

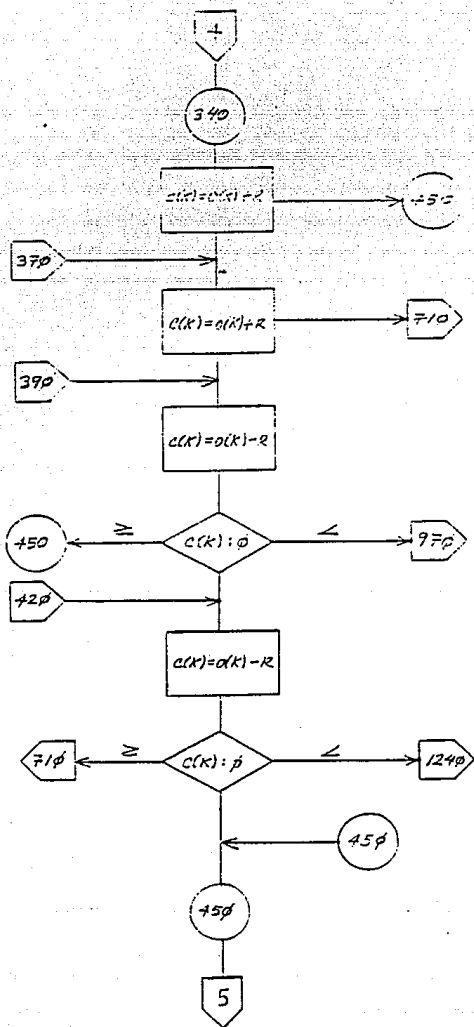
c.s. "S"

210

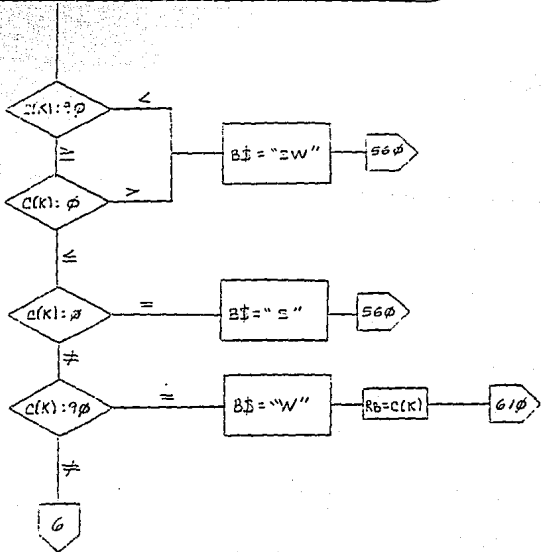
1

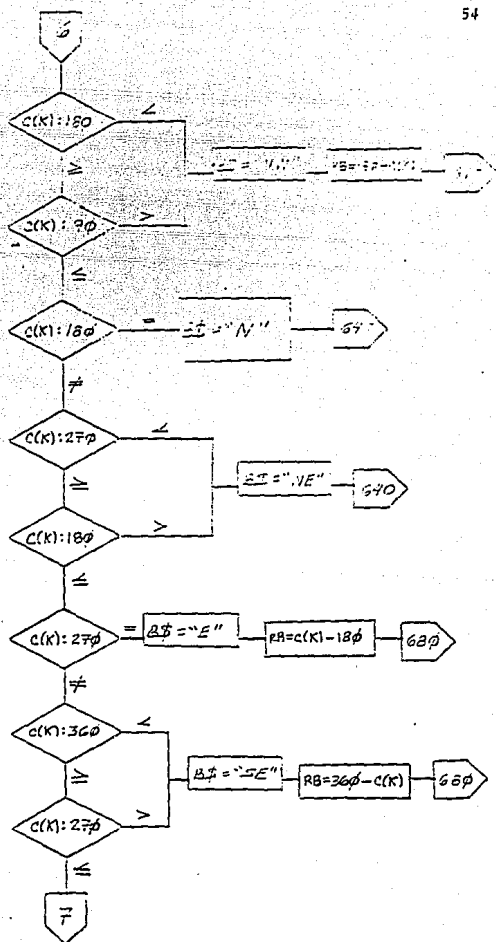


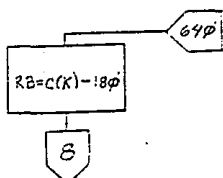
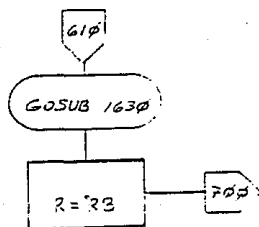
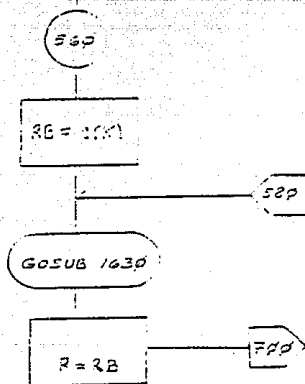
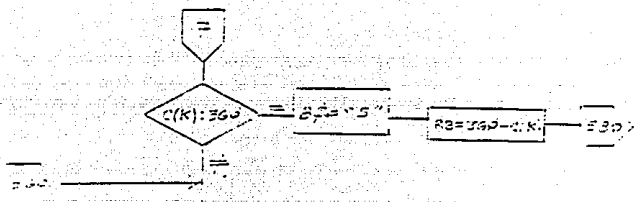


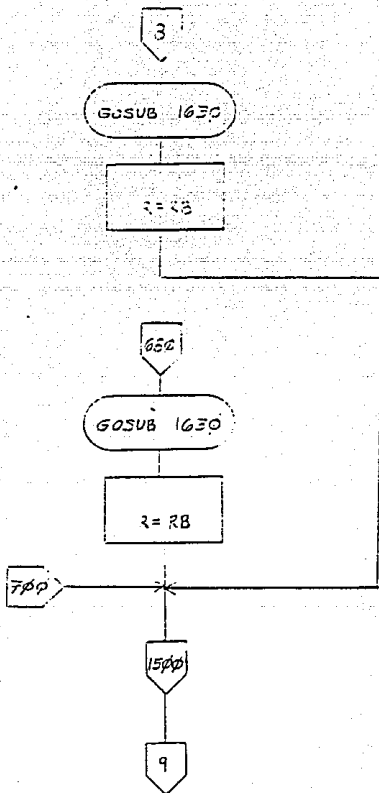


VALOR CUMULO E DIFERENÇA DE "C" E
AL JOR "Y" E POSITIVO.





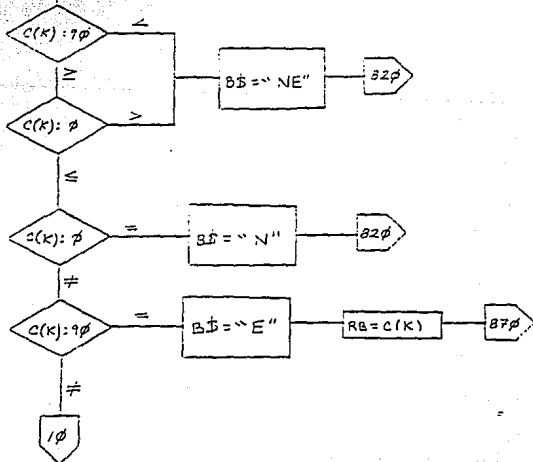


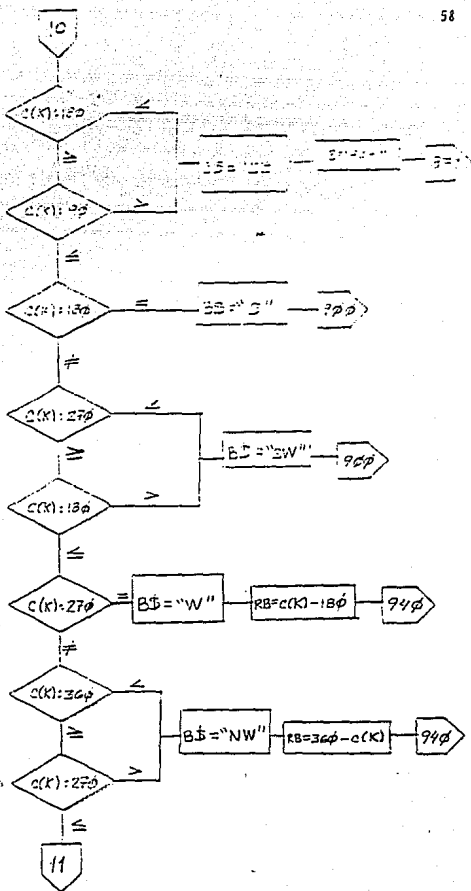


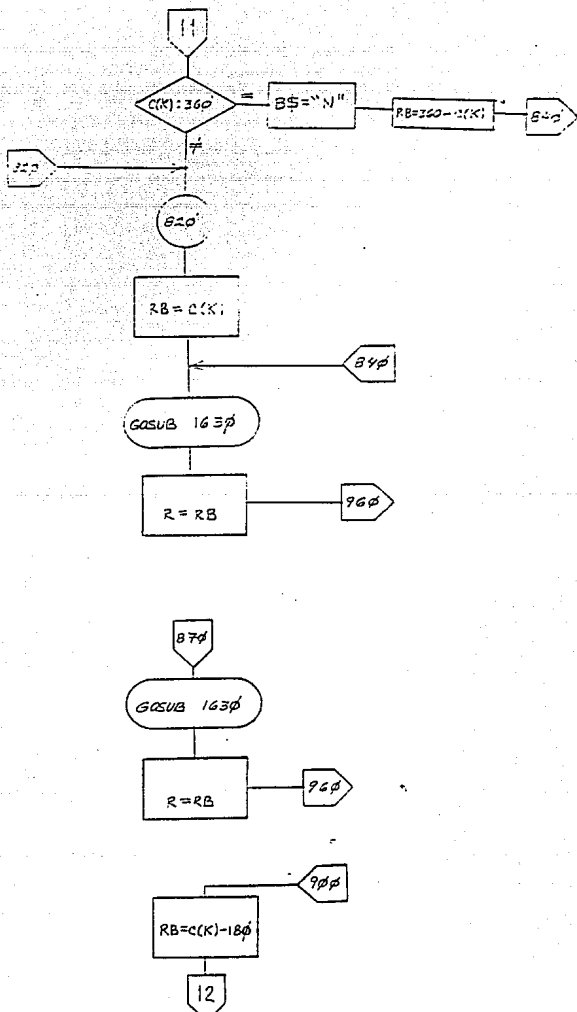
9

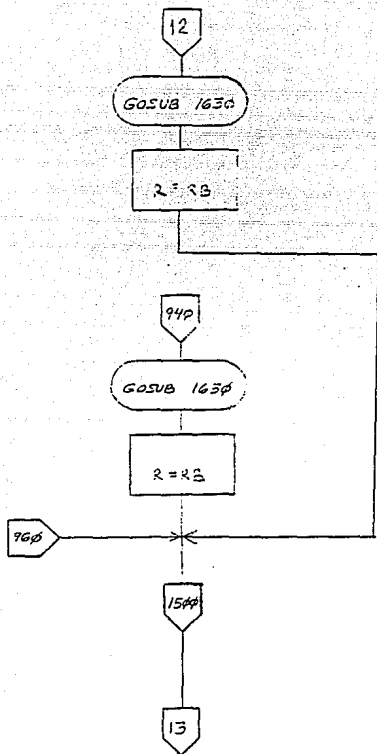
710

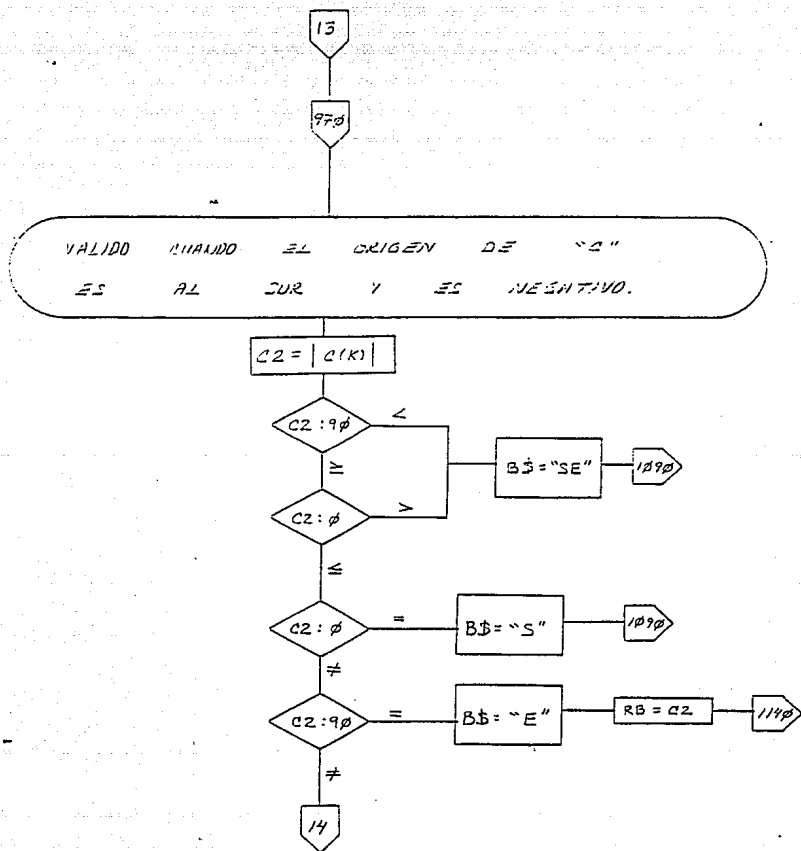
VALDO MINDO EL DRISEN SE "1"
 EL AL UCRTE I SE POSITIVO.

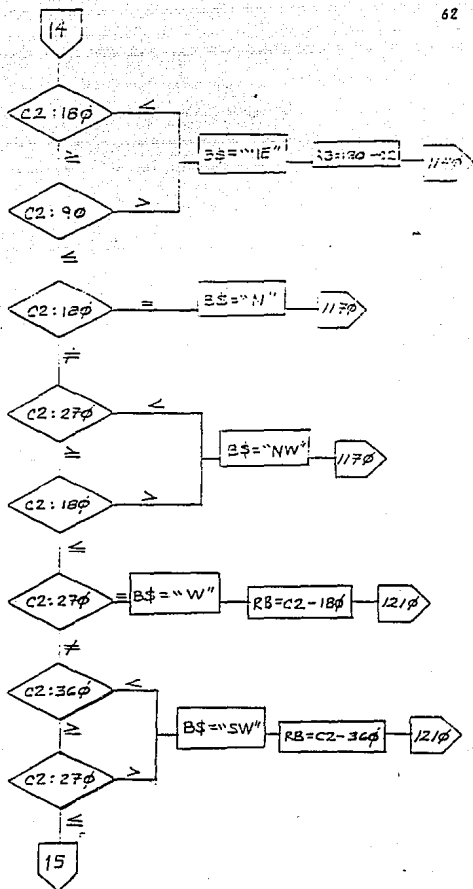


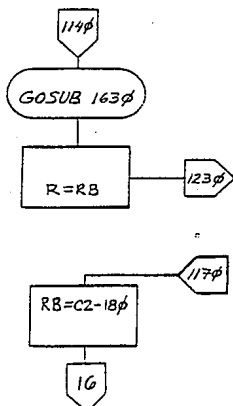
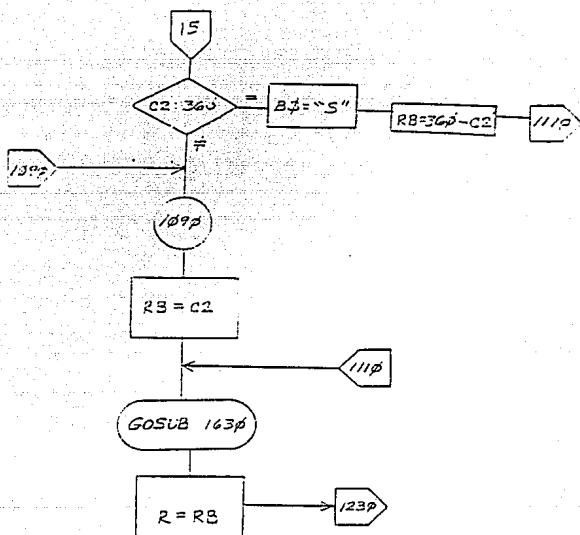


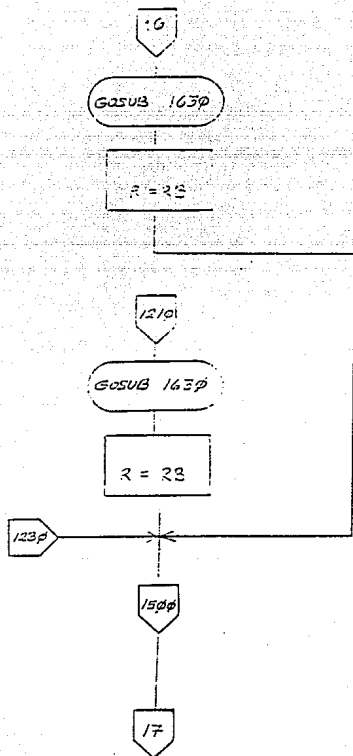












17

124φ

VALIDO CUANDO EL ORIGEN DE "C"
 ES AL NORTE O ES NEGATIVO.

C3 = |C(K)|

C3 = 9φ

L

N

Y

Bφ = "NW"

136φ

C3 = φ

N

C3 = φ

=

Bφ = "N"

136φ

C3 = 9φ

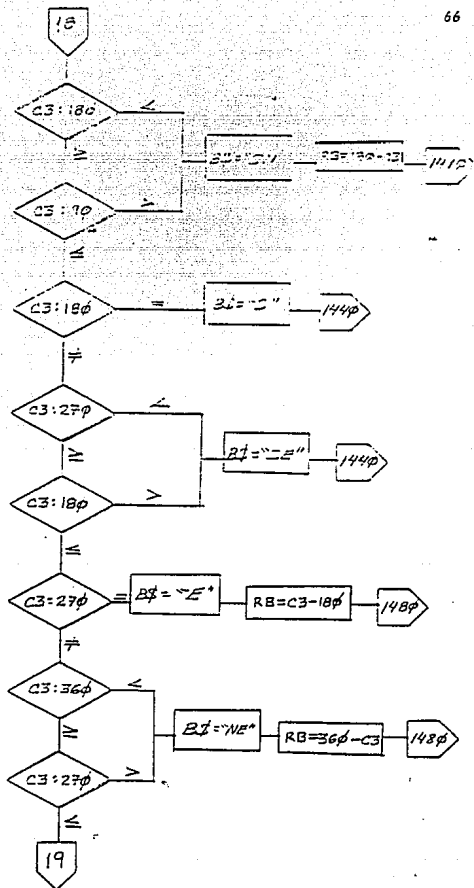
=

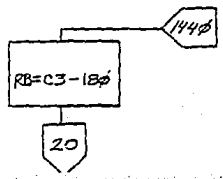
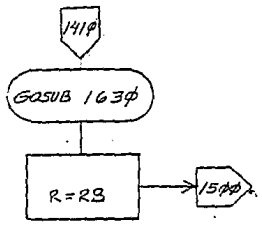
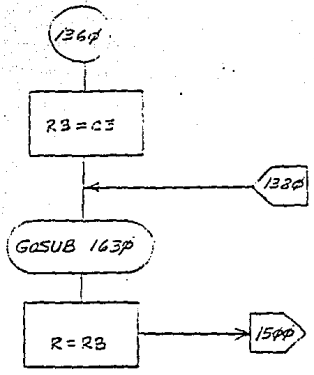
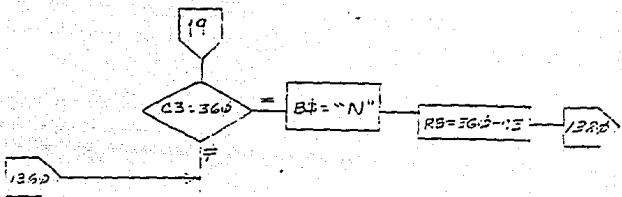
Bφ = "W"

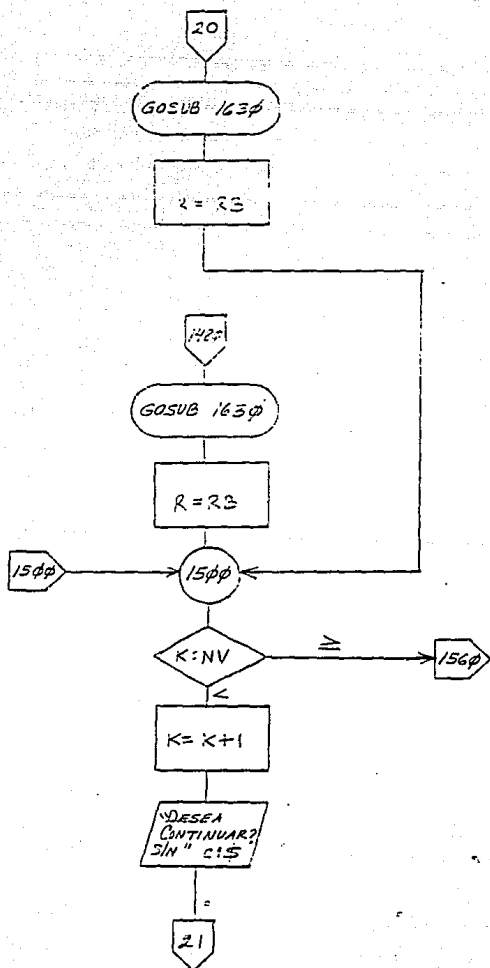
RB = C3

141φ

18







22

SUB 1630

SUBROUTINE SUB HELPER GRAD-MIN-SES

$R1 = INT(R5)$
 $R2 = R5 - R1$
 $R3 = R2(60)$
 $R4 = INT(R3)$
 $R5 = R3 - R4$
 $R6 = R5(60)$

$=$ $K : NV$
 \neq

$K, K+1, B\$,$
 $R1, R4, R6$

$K, K-(NV-1),$
 $B\$, R1, R4, R6$

RETURN

23

23

SUB 1760

SUBROUTINA QUE CALCULA GRAB - MIN - IEE

R7 = INT (R)
R8 = R - R7
R9 = R8 (60)
R10 = INT (R9)
R11 = R9 - R10
R12 = R11 (60)

NV, K, B\$,
R7, R10, R12

RETURN

IV.8. CODIFICACION.

```

10 REM PROGRAMA PARA EL CALCULO DE RUMBOS
20 REM METODO: TRANSPORTE DE RUMBOS
30 REM CONDICION: ANGULOS MEDIDOS A LA DERECHA
40 DIM C(100) 'ADMITE 100 VERTICES
50 DIM C(100)
60 INPUT "TECLEAR EL NUMERO DE VERTICES=";NV
70 PRINT "EL NUMERO DE VERTICES=";NV
80 PRINT " "
90 PRINT "DE" TAB(0) "A" TAB(15) "CUAD." TAB(22) "RBO. MAGNET. CALC."
100 PRINT TAB(24) "S" TAB(28) "M" TAB(34) "S"
110 PRINT " "
120 K=1
130 INPUT "SE TOMO EL RUMBO AL FINAL? S/N ";C3
140 IF C3="S" THEN GOTO 210
150 INPUT "TECLEAR RUMBO INICIAL 1-2 EN GRADOS DECIMALES=";R
160 INPUT "TECLEAR EL CUADRANTE DEL RUMBO INICIAL 1-2=";B$
170 RB=R
180 GOSUB 1600
190 :
200 K=K+1:GOTO 260
210 :
220 INPUT "TECLEAR EL RUMBO FINAL EN GRADOS DECIMALES=";R
230 INPUT "TECLEAR EL CUADRANTE DEL RUMBO FINAL=";B$
240 :
250 GOSUB 1760
260 :
270 PRINT "TECLEAR EL ANGULO";K;"EN GRADOS DECIMALES="
280 INPUT O(K)
290 :
300 IF B$="NE" GOTO 340
310 IF B$="SW" GOTO 370
320 IF B$="NW" GOTO 390
330 IF B$="SE" GOTO 420
340 :
350 C(K)=O(K) + R
360 GOTO 450
370 C(K)=O(K) + R
380 GOTO 710
390 C(K)=O(K) - R
400 IF C(K) < 0 GOTO 970
410 GOTO 450
420 C(K)=O(K) - R
430 IF C(K) < 0 GOTO 1240
440 GOTO 710
450 :
460 REM "VALIDO CUANDO EL ORIGEN DE C ES AL SUR Y ES +"
470 IF (C(K) < 90) AND (C(K) > 0) THEN B$="SW":GOTO 560
480 IF C(K)=0 THEN B$="S":GOTO 560

```

```

490 IF C(K)=90 THEN B$="W":RB=C(K):GOTO 610
500 IF (C(K) < 180) AND (C(K) > 90) THEN B$="NW":RB=180-C(K):GOTO 510
510 IF C(K)=180 THEN B$="N":GOTO 540
520 IF (C(K) < 270) AND (C(K) > 180) THEN B$="NE":GOTO 540
530 IF C(K)=270 THEN B$="E":RB=C(K)-180:GOTO 580
540 IF (C(K) < 360) AND (C(K) > 270) THEN B$="SE":RB=360-C(K):GOTO 580
550 IF C(K)=360 THEN B$="S":RB=360-C(K):GOTO 580
560 :
570 RB=C(K)
580 GOSUB 1630
590 R=RB
600 GOTO 700
610 GOSUB 1630
620 R=RB
630 GOTO 700
640 RB=C(K)-180
650 GOSUB 1630
660 R=RB
670 GOTO 700
680 GOSUB 1630
690 R=RB
700 GOTO 1300
710 :
720 REM "VALIDO CUANDO EL ORIGEN DE C ES AL NORTE Y ES +"
730 IF (C(K) < 90) AND (C(K) > 0) THEN B$="NE":GOTO 820
740 IF C(K)=0 THEN B$="N":GOTO 820
750 IF C(K)=90 THEN B$="E":RB=C(K):GOTO 870
760 IF (C(K) < 180) AND (C(K) > 90) THEN B$="SE":RB=180-C(K):GOTO 870
770 IF C(K)=180 THEN B$="S":GOTO 900
780 IF (C(K) < 270) AND (C(K) > 180) THEN B$="SW":GOTO 900
790 IF C(K)=270 THEN B$="W":RB=C(K)-180:GOTO 940
800 IF (C(K) < 360) AND (C(K) > 270) THEN B$="NW":RB=360-C(K):GOTO 940
810 IF C(K)=360 THEN B$="N":RB=360-C(K):GOTO 840
820 :
830 RB=C(K)
840 GOSUB 1630
850 R=RB
860 GOTO 960
870 GOSUB 1630
880 R=RB
890 GOTO 960
900 RB=C(K)-180
910 GOSUB 1630
920 R=RB
930 GOTO 960
940 GOSUB 1630
950 R=RB
960 GOTO 1500
970 :
980 REM "VALIDO CUANDO EL ORIGEN DE C ES AL SUR Y ES -"
990 C2=ABS(C(K))
1000 IF (C2 < 90) AND (C2 > 0) THEN B$="SE":GOTO 1090
1010 IF C2=0 THEN B$="S":GOTO 1090

```

```

1020 IF C2=90 THEN B$="E":RB=C2:GOTO 1140
1030 IF (C2 < 180) AND (C2 > 90) THEN B$="NE":RB=180-C2:GOTO 1140
1040 IF C2=180 THEN B$="N":GOTO 1170
1050 IF (C2 < 270) AND (C2 > 180) THEN B$="NW":GOTO 1170
1060 IF C2=270 THEN B$="W":RB=C2-180:GOTO 1210
1070 IF (C2 < 360) AND (C2 > 270) THEN B$="SW":RB=360-C2:GOTO 1210
1080 IF C2=360 THEN B$="S":RB=360-C2:GOTO 1110
1090 :
1100 RB=C2
1110 GOSUB 1630
1120 R=RB
1130 GOTO 1230
1140 GOSUB 1630
1150 R=RB
1160 GOTO 1230
1170 RB=C2-180
1180 GOSUB 1630
1190 R=RB
1200 GOTO 1230
1210 GOSUB 1630
1220 R=RB
1230 GOTO 1500
1240 :
1250 REM "VALIDO CUANDO EL ORIGEN DE C ES AL NORTE Y ES -"
1260 C3=ABS(C1/C2)
1270 IF (C3 < 90) AND (C3 > 0) THEN B$="NW":GOTO 1360
1280 IF C3=0 THEN B$="N":GOTO 1360
1290 IF C3=90 THEN B$="W":RB=C3:GOTO 1410
1300 IF (C3 < 180) AND (C3 > 90) THEN B$="SW":RB=180-C3:GOTO 1410
1310 IF C3=180 THEN B$="S":GOTO 1440
1320 IF (C3 < 270) AND (C3 > 180) THEN B$="SE":GOTO 1440
1330 IF C3=270 THEN B$="E":RB=C3-180:GOTO 1480
1340 IF (C3 < 360) AND (C3 > 270) THEN B$="NE":RB=360-C3:GOTO 1480
1350 IF C3=360 THEN B$="N":RB=360-C3:GOTO 1380
1360 :
1370 RB=C3
1380 GOSUB 1630
1390 R=RB
1400 GOTO 1500
1410 GOSUB 1630
1420 R=RB
1430 GOTO 1500
1440 RB=C3-180
1450 GOSUB 1630
1460 R=RB
1470 GOTO 1500
1480 GOSUB 1630
1490 R=RB
1500 :
1510 IF K>=NV THEN GOTO 1560
1520 K=K+1
1530 LINE INPUT "DESEA CONTINUAR? S/N:";C1$

```



```

1540 IF C13="N" THEN GOTO 1610
1550 GOTO 260
1560 :
1570 LINE INPUT "DESEA COMPROBAR S/N: ";C22;
1580 IF C22="N" THEN GOTO 1610
1590 K=1
1600 GOTO 260
1610 PRINT "FIN DEL CALCULO"
1620 END
1630 REM SUBROUTINA QUE CALCULA GRADOS-MIN-SEG
1640 R1=INT (R8)
1650 R2=R8-R1
1660 R3=R2*60
1670 R4=INT (R3)
1680 R5=R3-R4
1690 R6=R5*60
1700 IF K=NV THEN GOTO 1730
1710 PRINT K TAB(7) K+1 TAB(16) B$ TAB(22) R1 TAB(27) R4;
TAB(33) USING "##.##";R6
1720 GOTO 1750
1730 PRINT K TAB(7) K-(NV-1) TAB(16) B$ TAB(22) R1 TAB(27) R4;
TAB(33) USING "##.##";R6
1740 PRINT " "
1750 RETURN
1760 REM SUBROUTINA QUE CALCULA GRADOS-MIN-SEG
1770 R7=INT (R)
1780 R8=R-R7
1790 R9=R8*60
1800 R10=INT (R9)
1810 R11=R9-R10
1820 R12=R11*60
1830 PRINT NV TAB(7) K TAB(16) B$ TAB(22) R7 TAB(27) R10;
TAB(33) USING "##.##";R12
1840 RETURN

```

IV.9. PRUEBA DE ESCRITORIO

PROYECTO	TESIS	ARCHIVO	LRBO.7	FECHA	1993
PROGRAMA	Gwbasic	ELABORO	MPGA	HORA	76 DE 83

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
NV	6					
K	1					
C\$	"S"					
GOTO 210						
K	77.01667					
B\$	"NIV"					
GOSUB	1760					
R7	77					
R8	1.667023 ⁻²					
R9	1.000214					
R10	1					
R11	2.136231 ⁻⁴					
R12	0.01					
O(K)	92.33334	86.76667	134.5667	164.1833	159.15	81.2
GOTO	390	370	420	340	370	370
C(K)	15.31667	102.0833	56.65007	220.8334	201.7333	102.9333
GOTO	450	710	710	450	450	450

PROYECTO	TESIS	ARCHIVO	LRBO.7	FECHA	1993
PROGRAMA	Gwbasic	ELABORO	MPGA	HORA	77 DE 83

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
C	S y +	N y +	N y +	S y +	S y +	S y +
B ϕ	"SW"	"SE"	"NE"	"NE"	"NE"	"NI"
GOTO	56 ϕ	87 ϕ	82 ϕ	64 ϕ	61 ϕ	61 ϕ
KB	15.31667	102.0833	56.65004	220.8333	201.7853	102.7853
50SUB 16 ϕ						
R1	15	77	50	40	21	77
R2	0.316733	0.7166565	0.6500397	0.8333735	0.7833705	0.0166626
R3	19.0004	54.99939	39.00238	50.00051	47.00073	0.9997557
R4	19	54	39	50	47	0
R5	3.76728 ⁻⁴	0.99939	2.380371 ⁻³	6.103512 ⁻³	4.272701 ⁻⁴	0.9997557
R6	0.01	59.99	0.01	0.05	0.03	59.99
R	15.31667	102.0833	56.65004	220.8333	201.7853	102.7853
GOTO	7 ϕ ϕ	96 ϕ	96 ϕ	7 ϕ ϕ	7 ϕ ϕ	7 ϕ ϕ
GOTO	15 ϕ ϕ	15 ϕ ϕ	15 ϕ ϕ	15 ϕ ϕ	15 ϕ ϕ	15 ϕ ϕ
K \geq NV	1 < 6	2 < 6	3 < 6	4 < 6	5 < 6	6 \geq 6
	:	:	:	:	:	6070 15 ϕ ϕ
K	2	3	4	5	5	
C1 ϕ	"S"	"S"	"S"	"S"	"S"	
GOTO 26 ϕ						

IV.10. IMPRESION DEL PROGRAMA.

```

10 REM PROGRAMA PARA EL CALCULO DE RUMBOS
20 REM METODO: TRANSPORTE DE RUMBOS
30 REM CONDICION: ANGULOS MEDIDOS A LA DERECHA
40 DIM C(100) 'ADMITE 100 VERTICES
50 DIM C(100)
60 INPUT "TECLEAR EL NUMERO DE VERTICES=":NV
70 LPRINT "EL NUMERO DE VERTICES=":NV
80 LPRINT " "
90 LPRINT "DE" TAB(6) "A" TAB(15) "CUAD." TAB(22) "RBO. MAGNET. CALC."
100 LPRINT TAB(24) "S" TAB(28) "M" TAB(34) "S"
110 LPRINT " "
120 K=1
130 INPUT "SE TOMO EL RUMBO AL FINAL? S/N ":CS
140 IF CS="S" THEN GOTO 210
150 INPUT "TECLEAR RUMBO INICIAL 1-2. EN GRADOS DECIMALES=":R
160 INPUT "TECLEAR EL CUADRANTE DEL RUMBO INICIAL 1-2=":B$
170 RB=R
180 GOSUB 1630
190 :
200 K=K+1:GOTO 260
210 :
220 INPUT "TECLEAR EL RUMBO FINAL EN GRADOS DECIMALES=":R
230 INPUT "TECLEAR EL CUADRANTE DEL RUMBO FINAL=":B$
240 :
250 GOSUB 1760
260 :
270 PRINT "TECLEAR EL ANGULO";K;" EN GRADOS DECIMALES="
280 INPUT O(K)
290 :
300 IF B$="NE" GOTO 340
310 IF B$="SW" GOTO 370
320 IF B$="NW" GOTO 390
330 IF B$="SE" GOTO 420
340 :
350 C(K)=O(K) + R
360 GOTO 450
370 C(K)=O(K) + R
380 GOTO 710
390 C(K)=O(K) - R
400 IF C(K) < 0 GOTO 970
410 GOTO 450
420 C(K)=O(K) - R
430 IF C(K) < 0 GOTO 1240
440 GOTO 710
450 :
460 REM "VALIDO CUANDO EL ORIGEN DE C ES AL SUR Y ES +"
470 IF (C(K) < 90) AND (C(K) > 0) THEN B$="SW":GOTO 560
480 IF C(K)=0 THEN B$="S":GOTO 560
490 IF C(K)=90 THEN B$="W":RB=C(K):GOTO 610

```

```

500 IF (C(K) < 180) AND (C(K) > 90) THEN B$="NW":RB=180-C(K):GOTO 610
510 IF C(K)=180 THEN B$="N":GOTO 640
520 IF (C(K) < 270) AND (C(K) > 180) THEN B$="NE":GOTO 640
530 IF C(K)=270 THEN B$="E":RB=C(K)-180:GOTO 680
540 IF (C(K) < 360) AND (C(K) > 270) THEN B$="SE":RB=360-C(K):GOTO 680
550 IF C(K)=360 THEN B$="S":RB=360-C(K):GOTO 680
560 :
570 RB=C(K)
580 GOSUB 1630
590 R=RB
600 GOTO 700
610 GOSUB 1630
620 R=RB
630 GOTO 700
640 RB=C(K)-180
650 GOSUB 1630
660 R=RB
670 GOTO 700
680 GOSUB 1630
690 R=RB
700 GOTO 1500
710 :
720 REM "VALIDO CUANDO EL ORIGEN DE C ES AL NORTE Y ES +"
730 IF (C(K) < 90) AND (C(K) > 0) THEN B$="NE":GOTO 820
740 IF C(K)=0 THEN B$="N":GOTO 820
750 IF C(K)=90 THEN B$="E":RB=C(K):GOTO 870
760 IF (C(K) < 180) AND (C(K) > 90) THEN B$="SE":RB=180-C(K):GOTO 870
770 IF C(K)=180 THEN B$="S":GOTO 900
780 IF (C(K) < 270) AND (C(K) > 180) THEN B$="SW":GOTO 900
790 IF C(K)=270 THEN B$="W":RB=C(K)-180:GOTO 940
800 IF (C(K) < 360) AND (C(K) > 270) THEN B$="NW":RB=360-C(K):GOTO 940
810 IF C(K)=360 THEN B$="N":RB=360-C(K):GOTO 840
820 :
830 RB=C(K)
840 GOSUB 1630
850 R=RB
860 GOTO 960
870 GOSUB 1630
880 R=RB
890 GOTO 960
900 RB=C(K)-180
910 GOSUB 1630
920 R=RB
930 GOTO 960
940 GOSUB 1630
950 R=RB
960 GOTO 1500
970 :
980 REM "VALIDO CUANDO EL ORIGEN DE C ES AL SUR Y ES -"
990 C2=ABS(C(K))
1000 IF (C2 < 90) AND (C2 > 0) THEN B$="SE":GOTO 1090
1010 IF C2=0 THEN B$="S":GOTO 1090

```

```

1020 IF C2=90 THEN B$="E":RB=C2:GOTO 1140
1030 IF (C2 < 180) AND (C2 > 90) THEN B$="NE":RB=180-C2:GOTO 1140
1040 IF C2=180 THEN B$="N":GOTO 1170
1050 IF (C2 < 270) AND (C2 > 180) THEN B$="NW":GOTO 1170
1060 IF C2=270 THEN B$="W":RB=C2-180:GOTO 1210
1070 IF (C2 < 360) AND (C2 > 270) THEN B$="SW":RB=360-C2:GOTO 1210
1080 IF C2=360 THEN B$="S":RB=360-C2:GOTO 1110
1090 :
1100 RB=C2
1110 GOSUB 1630
1120 R=RB
1130 GOTO 1230
1140 GOSUB 1630
1150 R=RB
1160 GOTO 1230
1170 RB=C2-180
1180 GOSUB 1630
1190 R=RB
1200 GOTO 1230
1210 GOSUB 1630
1220 R=RB
1230 GOTO 1500
1240 :
1250 REM "VALIDO CUANDO EL ORIGEN DE C ES AL NORTE Y ES -"
1260 C3=ABS(C(K))
1270 IF (C3 < 90) AND (C3 > 0) THEN B$="NW":GOTO 1360
1280 IF C3=0 THEN B$="N":GOTO 1360
1290 IF C3=90 THEN B$="W":RB=C3:GOTO 1410
1300 IF (C3 < 180) AND (C3 > 90) THEN B$="SW":RB=180-C3:GOTO 1410
1310 IF C3=180 THEN B$="S":GOTO 1440
1320 IF (C3 < 270) AND (C3 > 180) THEN B$="SE":GOTO 1440
1330 IF C3=270 THEN B$="E":RB=C3-180:GOTO 1480
1340 IF (C3 < 360) AND (C3 > 270) THEN B$="NE":RB=360-C3:GOTO 1480
1350 IF C3=360 THEN B$="N":RB=360-C3:GOTO 1380
1360 :
1370 RB=C3
1380 GOSUB 1630
1390 R=RB
1400 GOTO 1500
1410 GOSUB 1630
1420 R=RB
1430 GOTO 1500
1440 RB=C3-180
1450 GOSUB 1630
1460 R=RB
1470 GOTO 1500
1480 GOSUB 1630
1490 R=RB
1500 :
1510 IF K>=NV THEN GOTO 1560
1520 K=K+1
1530 LINE INPUT "DESEA CONTINUAR? S/N: ";C1$
1540 IF C1$="N" THEN GOTO 1610

```

```

1550 GOTO 260
1560 :
1570 LINE INPUT "DESEA COMPROBAR? S/N: ";C25
1580 IF C25="N" THEN GOTO 1610
1590 K=1
1600 GOTO 260
1610 LPRINT "FIN DEL CALCULO"
1620 END
1630 REM SUBROUTINA QUE CALCULA GRADOS-MIN-SEG
1640 R1=INT (R8)
1650 R2=R8-R1
1660 R2=R2*60
1670 R4=INT (R2)
1680 R3=R2-R4
1690 R6=R3*60
1700 IF K=NV THEN GOTO 1730
1710 LPRINT K TAB(7) K+1 TAB(16) B$ TAB(22) R1 TAB(27) R4;
      TAB(33) USING "##.##";R6
1720 GOTO 1750
1730 LPRINT K TAB(7) K-(NV-1) TAB(16) B$ TAB(22) R1 TAB(27) R4;
      TAB(33) USING "##.##";R6
1740 LPRINT " "
1750 RETURN
1760 REM SUBROUTINA QUE CALCULA GRADOS-MIN-SEG
1770 R7=INT (R8)
1780 R8=R-R7
1790 R9=R8*60
1800 R10=INT (R9)
1810 R11=R9-R10
1820 R12=R11*60
1830 LPRINT NV TAB(7) K TAB(16) B$ TAB(22) R7 TAB(27) R10;
      TAB(33) USING "##.##";R12
1840 RETURN

```


IV.11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE VERTICES= 6

DE	A	CUAD.	RBO.	MAGNET.	CALC.
			G	M	S
6	1	NW	77	1	0.01
1	2	SW	18	19	0.02
2	3	SE	77	54	59.96
3	4	NE	56	39	0.14
4	5	NE	40	50	0.04
5	6	NE	21	47	0.03
6	1	NW	77	0	59.99

FIN DEL CALCULO

CAPITULO V
PROGRAMA PARA CALCULAR AZIMUTES

V.1 GENERALIDADES.

En el Capitulo IV , se mencionó que la dirección de una línea, puede definirse por su Rumbo , de igual modo , puede definirse por su AZIMUT y también, puede ser magnético o astronómico.

Un azimut, es el ángulo que forma una línea con la dirección Norte-Sur, medido de 0 a 360 grados, a partir del Norte, en el sentido de las manecillas del reloj.

Unicamente, en el primer cuadrante, coinciden el Rumbo y el Azimut, en valor numérico.

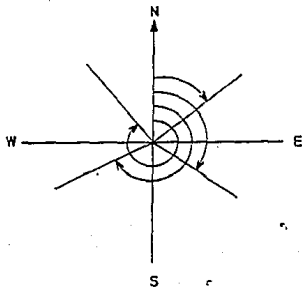


FIG. (16) AZIMUTES.

V.2. ALGORITMO.

El programa corre con instrucciones "DATA". Para cada problema, deberán modificarse las líneas 170 - 400, que corresponden a dichas instrucciones.

Una vez que se teclea el Número de Vértices, el programa usa los "DATA", con ellos trabaja y muestra los resultados.

Como condición, inicialmente deben conocerse los Rumbo. Después de indicar el número de vértices, se imprime una Tabla.

La instrucción "READ", toma los datos correspondientes al Angulo (grados-minutos-segundos), al Rumbo (en grados decimales) y al Cuadrante del Rumbo.

Compara los cuadrantes y de acuerdo a ellos, calcula el Azimut en grados decimales y luego lo convierte en grados, minutos y segundos.

Otra instrucción "READ", toma los valores del Rumbo (grados-minutos-segundos).

Imprime los datos y los resultados, en forma de tabla.

Finalmente, continúa calculando, por medio de iteraciones, hasta agotar las instrucciones DATA y de acuerdo al Número de Vértices indicado.

V.3. LIMITES DEL PROGRAMA.

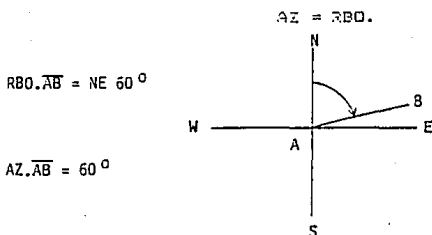
- a) Deben conocerse TODOS los Rumbos, como condición inicial. Además, deberán calcularse en decimales.
- b) También, deben conocerse todos los ángulos, aunque es opcional (ya que no influye en los cálculos) y, en las instrucciones "DATA", podrían indicarse como cero.
- c) El acimut en decimales, muestra cinco dígitos decimales: como grado de precisión.
- d) Las iteraciones admitidas, son hasta INFINITO. Queriendo decir que, el número de vértices es hasta " N ". Para esto, sólo deben aumentarse y cambiarse las instrucciones "DATA".
- e) En las instrucciones "DATA", el orden de los datos, debe ser el siguiente:
 - 1.- Grados del Angulo.
 - 2.- Minutos del Angulo.
 - 3.- Segundos del Angulo.
 - 4.- Rumbo en grados decimales.
 - 5.- Cuadrante del Rumbo (entre comillas " ").
 - 6.- Grados del Rumbo.
 - 7.- Minutos del Rumbo.
 - 8.- Segundos del Rumbo.

V.4. FORMULAS MATEMATICAS

V.4.1. FORMULAS GENERALES Y BASICAS.

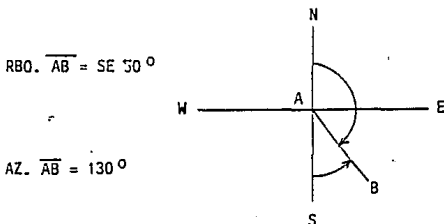
Si se sabe de antemano el Rumbo de un lado de una poligonal, puede definirse su Acimut correspondiente, de la siguiente manera:

a) El Acimut (AZ) sale directo, únicamente cuando el Rumbo (RBO.) está en el primer cuadrante, es decir, cuando el Rumbo sea: " NE ".

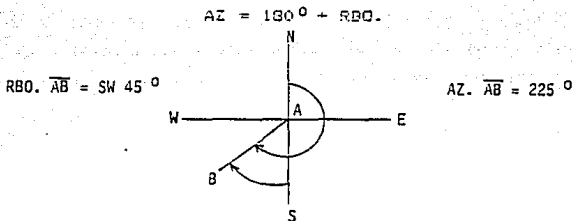


b) Restando a 180 grados, el Rumbo. Es decir, cuando éste sea: " SE ".

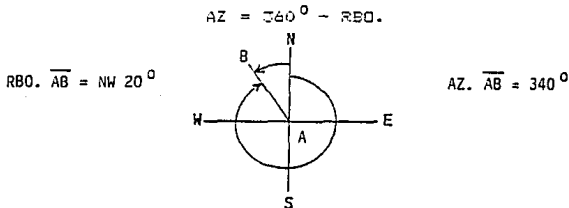
$$AZ = 180^\circ - RBO.$$



c) Sumando a 180 grados, el Rumbo. Es decir, cuando el Rumbo sea: " SW ".



d) Restando a 360 grados, el Rumbo. Es decir, cuando el Rumbo sea: " NW ".



V.4.2. FORMULAS USADAS EN EL PROGRAMA.

a) Si el Rumbo está en el cuadrante "NE" : El Azimut (AZ), es igual al Rumbo (R) :

$$AZ = R$$

b) Si el Rumbo está en el cuadrante "SE" :

$$AZ = 180^{\circ} - R$$

c) Si el Rumbo está en el cuadrante "SW" :

$$AZ = 180^{\circ} + R$$

d) Si el Rumbo está en el cuadrante "NW" :

$$AZ = 360^{\circ} - R$$

V.S. EJEMPLO.

Datos = Rumbos conocidos :

DE	A	R.M.C.	CUADRANTE
1	2	15 ° 19' 00''	SW
2	3	77 ° 55' 00''	SE
3	4	56 ° 39' 00''	NE
4	5	40 ° 50' 00''	NE
5	6	21 ° 47' 00''	NE
6	1	77 ° 01' 00''	NW

Aplicando las fórmulas indicadas en el inciso anterior (V.4.2.), tenemos lo siguiente:

$$\begin{aligned} \text{Cuadrante de 1-2} = \text{"SW"} : \text{AZ} &= 180^\circ + R \\ &= 180^\circ + 15^\circ 19' 00'' \\ &= 195^\circ 19' 00'' \end{aligned}$$

$$\begin{aligned} \text{Cuadrante de 2-3} = \text{"SE"} : \text{AZ} &= 180^\circ - R \\ &= 180^\circ - 77^\circ 55' 00'' \\ &= 102^\circ 05' 00'' \end{aligned}$$

$$\begin{aligned} \text{Cuadrante de 3-4} = \text{"NE"} : \text{AZ} &= R \\ &= 56^\circ 39' 00'' \end{aligned}$$

$$\begin{aligned} \text{Cuadrante de 4-5} = \text{"NE"} : \text{AZ} &= R \\ &= 40^\circ 50' 00'' \end{aligned}$$

$$\begin{aligned} \text{Cuadrante de 5-6} = \text{"NE"} : \text{AZ} &= R \\ &= 21^\circ 47' 00'' \end{aligned}$$

$$\begin{aligned} \text{Cuadrante de 6-1} = \text{"NW"} : \text{AZ} &= 360^\circ - R \\ &= 360^\circ - 77^\circ 01' 00'' \\ &= 282^\circ 59' 00'' \end{aligned}$$

V.3. VARIABLES UTILIZADAS.

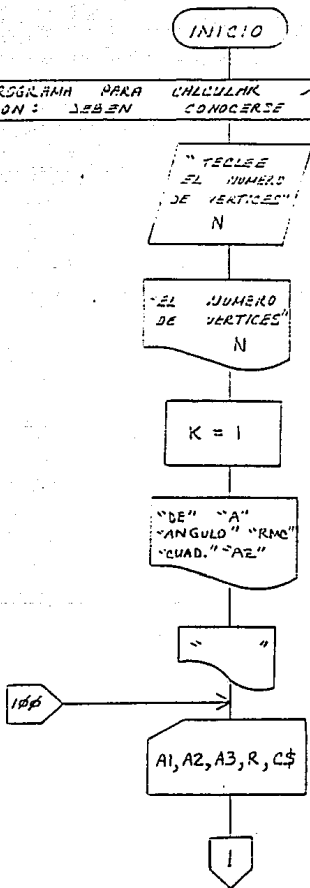
VARIABLE:	DEFINICION:
N	Número de vértices.
K	Contador.
A1	Grados del ángulo.
A2	Minutos del ángulo.
A3	Segundos del ángulo.
R	Rumbo en grados decimales.
C3	Cuadrante del rumbo.
AZ	Azimut.

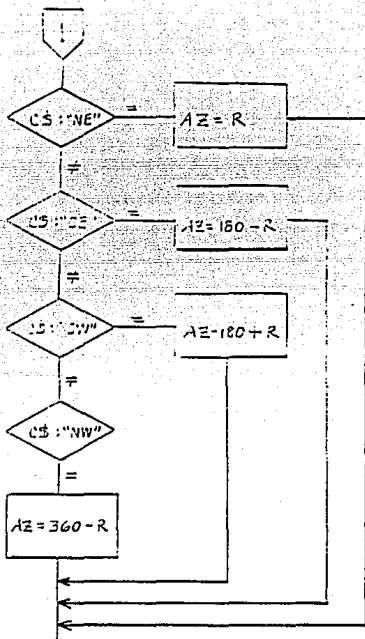
 CONVERSION A GRADOS-MINUTOS-SEGUNDOS :

A4,A7,A10	Números enteros.
A5,A8	Diferencias.
A6,A9	Multiplicados por 60.

R1	Grados del Rumbo.
R2	Minutos del Rumbo.
R3	Segundos del Rumbo.

PROGRAMA PARA CALCULAR AZIMUTES.
CONDICION: DEBEN CONOCERSE LOS RUMBOS.





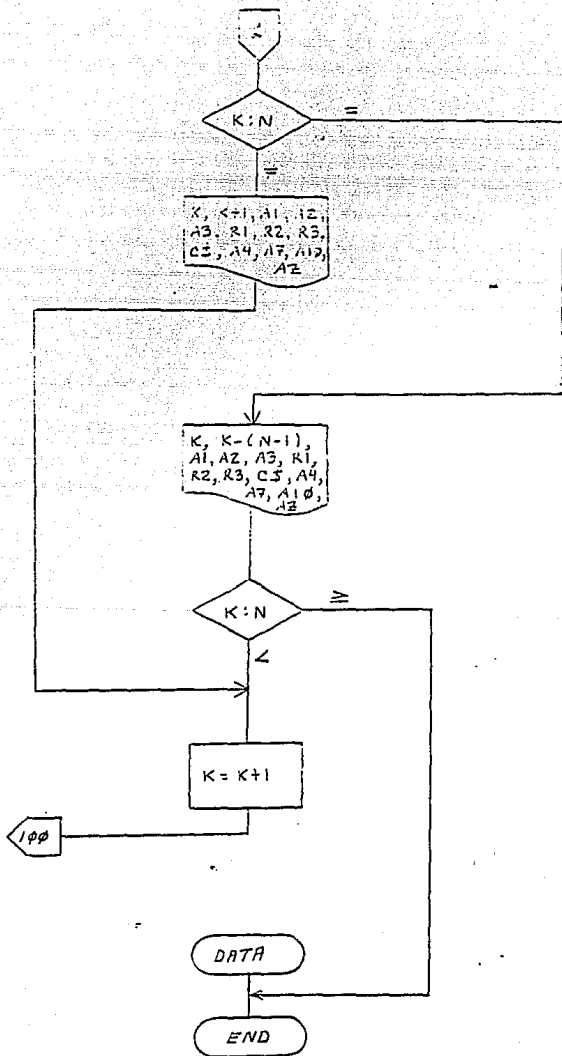
CONVERSION A GRADOS, MINUTOS Y SEGUNDOS

```

A4 = INT (AZ)
A5 = AZ - A4
A6 = A5 * 60
A7 = INT (A6)
A8 = A6 - A7
A9 = A8 * 60
A10 = INT (A9)
  
```

R1, R2, R3

2



```

10 REM PROGRAMA PARA CALCULAR AZIMUTES
20 REM CONDICION: DEBEN CONOCERSE LOS RUMBOS
30 INPUT 'TECLEE EL NUMERO DE VERTICES=":N
40 PRINT 'EL NUMERO DE VERTICES=":N
50 Z=1
60 PRINT "SE" TAB(7) "A" TAB(12) "ANGULO" TAB(27) "O.N.C." TAB(37) "CUAD."
   TAB(48) "AZIMUT" TAB(63) "AZIMUT"
70 PRINT TAB(12)"g" TAB(16)"a" TAB(20)"s" TAB(26)"g" TAB(30)"a" TAB(33)"s":
   TAB(47)"g" TAB(51)"a" TAB(55)"s" TAB(62)"decimales"
80 PRINT ' '
90 PRINT ' '
100 :
110 READ A1,A2,A3,R,C1
120 IF C1="NE" GOTO 160
130 IF C1="SE" GOTO 180
140 IF C1="SW" GOTO 200
150 IF C1="NW" GOTO 220
160 AZ = R
170 GOTO 240
180 AZ=180 - R
190 GOTO 240
200 AZ=180 + R
210 GOTO 240
220 AZ=720 - R
230 :
240 REM CONVERSION A GRADOS-MINUTOS-SEGUNDOS
250 A4 = INT (AZ)
260 A5 = AZ - A4
270 A6 = A5 * 60
280 A7 = INT (A6)
290 A8 = A6 - A7
300 A9 = A8 * 60
310 A10 = INT (A9)
320 READ R1,R2,R3
330 IF K = N THEN GOTO 360 ELSE GOTO 340
340 PRINT K TAB(6) K+1 TAB(10)A1 TAB(15) AZ TAB(19) A3 TAB(24) R1 TAB(28)R2:
   TAB(32) R3 TAB(40) C1 TAB(45)A4 TAB(50) A7 TAB(54) A10 TAB(61):
   USING "###.#####";AZ
350 GOTO 330
360 PRINT K TAB(6) K-(N-1) TAB(10)A1 TAB(15) AZ TAB(19) A3 TAB(24) R1 TAB(28)R2:
   TAB(32) R3 TAB(40) C1 TAB(45)A4 TAB(50) A7 TAB(54) A10 TAB(61):
   USING "###.#####";AZ
370 IF K < N THEN GOTO 380 ELSE GOTO 500
380 K=K+1:GOTO 100
390 DATA 92,20,00,15.3167,"SW",15,19,00
400 DATA 66,46,00,77.91666667,"SE",77,55,00
410 DATA 134,34,00,56.65,"NE",56,37,00
420 DATA 154,11,00,40.8334,"NE",40,50,00
430 DATA 160,57,00,21.78334,"NE",21,47,00
440 DATA 81,12,00,77.01666667,"NW",77,01,00
500 END

```

V.9. PRUEBA DE ESCRITORIO

PROYECTO	TESIS	ARCHIVO	LAZB.BAS	FECHA	1993
PROGRAMA	GW BASIC	ELABORO	MPGM	HOJA	95 DE 98

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
N	6					
K	1	2	3	4	5	6
A1	92	86	134	164	160	81
A2	20	46	34	10	50	11
A3	0	0	0	0	0	0
R	15.3167	77.91666667	56.65	40.8337	21.78337	77.01666667
C\$	"SW"	"SE"	"NE"	"NE"	"NE"	"NW"
	GOTO 200	GOTO 180	GOTO 160	GOTO 140	GOTO 120	GOTO 220
AZ	195.3167	102.0833	56.65	40.8337	21.78337	282.9833
	GOTO 240	GOTO 240	GOTO 240	GOTO 240	GOTO 240	GOTO 240
A4	195	102	56	40	21	282
A5	0.3166762	0.08333588	0.6500015	0.8334008	0.7833705	0.9833377
A6	19.00177	5.000153	39.00009	50.00405	47.00093	57.00025
A7	19	5	39	50	47	59
A8	0.00177002	0.0001525879	9.155277 ⁻⁵	0.009043579	0.0007212761	0.0002911406
A9	0.1062012	0.009155277	0.005993164	0.2426175	0.02503777	0.01967377
A10	0	0	0	0	0	0


```

10 REM PROGRAMA PARA CALCULAR AZIMUTES
20 REM CONDICION: DEBEN CONOCERSE LOS RUMDOS
30 INPUT "TECLEE EL NUMERO DE VERTICES=";N
40 LPRINT "EL NUMERO DE VERTICES=";N
50 K=1
60 LPRINT "SE" TAB(7) "A" TAB(10) "ANGULO" TAB(27) "RUMBO" TAB(77) "COORD."
  TAB(48) "AZIMUT" TAB(55) "AZIMUT"
70 LPRINT TAB(12)"a" TAB(16)"a" TAB(20)"a" TAB(24)"a" TAB(30)"a" TAB(33)"a":
  TAB(47)"a" TAB(51)"a" TAB(55)"a" TAB(62)"decimales"
80 LPRINT " "
90 LPRINT " "
100 :
110 READ A1,A2,A3,R,C
120 IF C="NE" GOTO 130
130 IF C="SE" GOTO 130
140 IF C="SW" GOTO 200
150 IF C="NW" GOTO 220
160 AZ = R
170 GOTO 240
180 AZ=180 - R
190 GOTO 240
200 AZ=180 + R
210 GOTO 240
220 AZ=360 - R
230 :
240 REM CONVERSION A GRADOS-MINUTOS-SEGUNDOS
250 A4 = INT (AZ)
260 A5 = AZ - A4
270 A6 = A5 * 60
280 A7 = INT (A6)
290 A8 = A6 - A7
300 A9 = A8 * 60
310 A10 = INT (A9)
320 READ R1,R2,R3
330 IF K = N THEN GOTO 360 ELSE GOTO 340
340 LPRINT K TAB(6) K+1 TAB(10)A1 TAB(15) A2 TAB(19) A3 TAB(24) R1 TAB(28)R2;
  TAB(32) R3 TAB(40) C; TAB(45)A4 TAB(50) A7 TAB(54) A10 TAB(61);
  USING "###.#####";AZ
350 GOTO 380
360 LPRINT K TAB(6) K-(N-1) TAB(10)A1 TAB(15)A2 TAB(19) A3 TAB(24) R1 TAB(28)R2;
  TAB(32) R3 TAB(40) C; TAB(45)A4 TAB(50) A7 TAB(54) A10 TAB(61);
  USING "###.#####";AZ
370 IF K < N THEN GOTO 380 ELSE GOTO 500
380 K=K+1;GOTO 100
390 DATA 92.20,00,15.3167,"SW",15,19,00
400 DATA 86.46,00,77.71666667,"SE",77,54,00
410 DATA 174.74,00,56.65,"NE",56,37,00
420 DATA 164.10,00,40.9554,"NE",40,50,00
430 DATA 160.56,00,21.78374,"NE",21,47,00
440 DATA 31.11,00,77.01666667,"NW",77,01,00
500 END

```

7.11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE VERTICES= 6												
DE	A	ANGULO			R.M.C.			CUAD.	AZIMUT			AZIMUT decimales
		g	m	s	g	m	s		g	m	s	
1	2	92	20	0	15	19	0	SW	195	19	0	195.31670
2	3	86	48	0	77	54	0	SE	102	5	0	102.08330
3	4	134	34	0	58	39	0	NE	56	39	0	56.65000
4	5	164	10	0	40	50	0	NE	40	50	0	40.83340
5	6	160	56	0	21	47	0	NE	21	47	0	21.78334
6	1	81	11	0	77	1	0	NW	282	59	0	282.98330

CAPITULO VI

PROGRAMA PARA COMPENSAR LA POLIGONAL

VI.1. GENERALIDADES.

En el Capítulo III, se vio como un polígono debe cerrar angularmente; ahora, veremos como debe cerrar linealmente. Para que esto suceda, se tendrá en cuenta que la suma algebraica de las proyecciones de cada uno de los lados del polígono, sea igual a CERO.

Si el error está dentro de la tolerancia, el trabajo se compensará. Esto puede hacerse por medio de los dos caminos más empleados: la Regla de la Brújula y la Regla del Tránsito.

De la última regla mencionada, tratará el presente Capítulo.

VI.2. ALGORITMO.

Para comenzar, el programa pide el número total de vértices de la poligonal; el cual puede contener hasta un máximo de 100 vértices, de lo contrario, marcará un error.

Después, el programa, se divide en las siguientes partes:

- a) Comprobación del cierre lineal y Cálculo de la Precisión,
- b) Compensación,
- c) Cálculo de las Proyecciones Corregidas,
- d) Cálculo de las Coordenadas,
- e) Subrutinas para salida de resultados.

a) Por medio de la instrucción cíclica, va leyendo la distancia y la va sumando.

Luego pide el Azimut (en grados decimales). Con esos datos, calcula las proyecciones (N-S y E-W), los errores (en Y / en X), las sumatorias: N, S, E y W.

Así, lee el ángulo, el rumbo y el azimut de cada lado e imprime esos datos junto con las proyecciones originales.

Después, calcula el error total (ET) y la precisión del trabajo efectuado (PT).

Pregunta si desea compensar, para continuar o, manda a imprimir las coordenadas con las proyecciones originales.

b) Calcula las constantes para "Y" y para "X". Luego pregunta los signos para las Correcciones en "Y" y en "X". Esto es de acuerdo a los valores de las sumatorias de las proyecciones: N, S, E y W. Después, va sumando dichas correcciones y manda a imprimir las sumatorias totales de las correcciones.

En caso de que la sumatoria de la corrección "Y" ó "X" y el error en "Y" ó "X", difiera de CERO, se tendrá la opción de sumar o restar a la primer corrección, otra cifra, indicado el total con su signo respectivo. Así, se tendrá la seguridad de que las coordenadas saldrán exactas.

c) A cada proyección la calcula con su signo final respectivo, sumándole o restándole la corrección antes calculada.

d) Indica teclear las coordenadas iniciales "Y" y "X".

Luego, calcula las coordenadas, tomando en cuenta las proyecciones corregidas de cada lado. Imprime los resultados correspondientes, por medio de subrutinas.

Pregunta si desea volver a compensar, en caso de que las coordenadas sean erróneas o bien, que no "cierren", es decir, que lleguen al final, con alguna diferencia, respecto a las coordenadas iniciales.

Dependiendo del error, se suma o se resta (la diferencia), para compensar. O bien, se teclean otras coordenadas.

e) Imprimen los resultados en forma de tabla, de la siguiente manera: distancia, ángulo horizontal, rumbo, cuadrante, proyecciones corregidas, coordenadas y vértices de cada lado.

VI.C. LIMITES DEL PROGRAMA.

- El número máximo de vértices es 100.
- Todos los azimutes, deben conocerse y además, deberán calcularse en grados decimales.
- Todos los valores numéricos, ocupan 8 espacios, contando el punto decimal. Ejemplo:

9 9 9 9 . 9 9 9	(números con punto decimal)
1 2 3 4 5 6 7 8	(espacios)

Esto, debe tenerse en cuenta, para el momento de cambiar las instrucciones : "DATA" , respecto a los valores numéricos de las distancias, como de las coordenadas.

De este modo, si el valor numérico de cualquier distancia o coordenada , rebasa los 8 espacios , la computadora los truncará.

d) El orden de los datos, para modificar las instrucciones "DATA" , será:

- 1) Distancia,
- 2) Grados del ángulo horizontal,
- 3) Minutos del ángulo horizontal,
- 4) Segundos del ángulo horizontal,
- 5) Grados del rumbo,
- 6) Minutos del rumbo,
- 7) Segundos del rumbo,
- 8) Cuadrante del rumbo (entre comillas " "),
- 9) Grados del Azimut,
- 10) Minutos del azimut y,
- 11) Segundos del azimut.

VI.4. FORMULAS MATEMATICAS.

La Regla del Tránsito, está basada en que todos los errores cometidos en un levantamiento, son accidentales y en que la

precisión de las medidas angulares son mayores que las medidas de longitud.

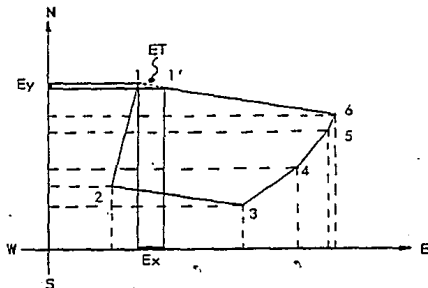
De este modo, los errores de cierre de las proyecciones, se corrigen proporcionalmente a las proyecciones de cada lado.

VI.4.1. FORMULAS GENERALES Y BASICAS.

El procedimiento a seguir, para aplicar la Regla del Tránsito, es la siguiente:

- a) Cálculo de las proyecciones para cada lado:
- Proy. sobre el Eje Y (N-S) = $\text{Dist.} \times (\cos \text{Az.})$
- Proy. sobre el Eje X (E-W) = $\text{Dist.} \times (\sin \text{Az.})$

- b) Condición del cierre lineal. Sea la figura (17):



$$\sum \text{Proy. N.} - \sum \text{Proy. S.} = \text{CERO}$$

$$\sum \text{Proy. E.} - \sum \text{Proy. W.} = \text{CERO}$$

Donde:

Σ = Sumatoria algebraica de cada Proyección.

c) Si sus diferencias no son CERO, entonces:

$$E_y = \left| \sum N - \sum S \right|$$

$$E_x = \left| \sum E - \sum W \right|$$

Donde:

E_y, E_x = Error en "Y" ó en "X".

$| \quad |$ = Valor absoluto.

\sum = Sumatoria algebraica de cada Proyección.

d) Error Total Lineal :

$$ET = \sqrt{E_x^2 + E_y^2}$$

e) Precisión del trabajo (PT) :

$$PT = \frac{1}{\sum L / ET}$$

Donde:

$\sum L$ = Suma total de la distancia.

ET = Error total.

La Precisión anterior, se compara con la tolerancia, que pueden ser, cualquiera de las siguientes, de acuerdo a las especificaciones y al trabajo realizado :

$$\frac{1}{500}, \frac{1}{1,000}, \frac{1}{5,000}, \frac{1}{10,000}, \text{ etc.}$$

f) Compensación (por la Regla del Tránsito) :

$$K_y = \frac{E_y}{\sum N + \sum S} ; \quad K_x = \frac{E_x}{\sum E + \sum W}$$

$$C_y = K_y (P_y) ; \quad C_x = K_x (P_x)$$

Donde:

K_y , K_x = Constante del error por unidad de proyección.
Ya sea "Y" ó "X".

E_y , E_x = Error en "Y" ó "X".

$\sum N + \sum S$ = Suma aritmética de todas las proyecciones,
en el Eje Y.

$\sum E + \sum W$ = Suma aritmética de todas las proyecciones,
en el Eje X.

C_y , C_x = Corrección a la proyección de un lado : "Y" ó
"X".

P_y , P_x = Proyección del lado tratado.

g) Comprobación.

$$\sum C_y = E_y$$

$$\sum C_x = E_x$$

Donde:

$\sum C_y$, $\sum C_x$: Sumatorias aritméticas de las correcciones
calculadas.

E_y , E_x : Error en "Y" ó en "X".

El signo de las correcciones calculadas , será así :

(POSITIVO) + A las Proyecciones cuya SUMA es MENOR.
 (NEGATIVO) - A las Proyecciones cuya SUMA es MAYOR.

Después, se procede a calcular las proyecciones corregidas. Cuya suma, debe cumplir con la Condición del Cierre Lineal.

h) Cálculo de las Coordenadas.

Finalmente, utilizando las proyecciones corregidas, pueden calcularse las Coordenadas respectivas de cada vértice.

Recordar que la suma o la resta de los valores de las Proyecciones Corregidas, va a ser de acuerdo a la Proyección que se trate, así será :

Proyección:	Signo:
N y E	+
S y W	-

Con la Coordenada inicial, se partirá para calcular las demás coordenadas, tomando el valor de la proyección siguiente y así sucesivamente, hasta llegar nuevamente a la coordenada inicial.

VI.4.2. FORMULAS USADAS EN EL PROGRAMA.

a) Cálculo de las Proyecciones para cada lado :

$$PY = \text{COS} (AZ * P) * B$$

$$PX = \text{COS} (AZ * P) * B$$

Donde:

PY : Proyección del lado tratado sobre el Eje Y (N-S).

PX : Proyección del lado tratado sobre el Eje X (E-W).

AZ : Acimut del lado tratado.

P : Constante = $\pi / 180$; para convertir radianes a grados.

B : Distancia del lado tratado.

b) Condición del Cierre Lineal.

$$SN - SS = 0$$

$$SE - SW = 0$$

Donde:

SN,SS,SE,SW : Sumatorias algebraicas de las Proyecciones :
N, S, E y W.

c) Cálculo del error.

Si sus diferencias son diferentes de CERO, entonces :

$$EY = SN - SS$$

$$EX = SE - SW$$

d) Error Total Lineal (ET) .

$$ET = \sqrt{EY^2 + EX^2}$$

e) Precisión del Trabajo (PT) .

$$PT = \frac{SUMB}{ET}$$

Donde:

SUMB : Sumatoria de la distancia.

ET : Error Total.

f) Compensación (por la Regla del Tránsito) .

$$KY = \left| \frac{EY}{SUMY} \right| \quad ; \quad KX = \left| \frac{EX}{SUMX} \right|$$

Si es POSITIVO:

$$CY = \left| KY (PY) \right| \quad ; \quad CX = \left| KX (PX) \right|$$

Si es NEGATIVO:

$$CY = -KY \left| PY \right| \quad ; \quad CX = -KX \left| PX \right|$$

Donde:

Ky , Kx : Constante de la corrección. Ya sea "Y" ó "X".

Ey , Ex : Error en "Y" ó "X".

SUMY : Suma aritmética de todas las proyecciones,
(N - S).

SUMX : Suma aritmética de todas las proyecciones.
(E - W).

Cy , Cx : Corrección a la proyección de un lado : "Y" ó
"X".

Py , Px : Proyección del lado tratado. Ya sea "X" ó
"Y".

| | : Valor absoluto.

g) Comprobación.

$$SCY = | EY | \quad ; \quad SCX = | EX |$$

Donde:

SCY : Sumatoria de la Corrección en "Y".

SCX : Sumatoria de la Corrección en "X".

EY : Error en "Y".

EX : Error en "X".

| | : Valor absoluto.

h) Cálculo de las Proyecciones Corregidas.

Si la Proyección Original es POSITIVA, entonces:

$$PNS = | PY | + CY \quad ; \quad PEW = | PX | + CX$$

Si la Proyección Original es NEGATIVA, entonces:

$$PNS = - (| PY | + CY) \quad ; \quad PEW = - (| PX | + CX)$$

Donde:

PNS, PEW : Proyecciones Corregidas N-S y E-W.

PY, PX : Proyecciones Originales "Y" ó "X".

CY, CX : Correcciones en "Y" ó en "X".

i) Cálculo de las Coordenadas.

Como datos originales, se indicarán las siguientes coordenadas : Y_1 , X_1 . Y con ellas, se calcularán las demás coordenadas.

$$Y_2 = Y_1 + PNS(1)$$

$$X_2 = X_1 + PEW(1)$$

Donde:

Y_2, X_2 : Coordenadas siguientes.

Y_1, X_1 : Coordenadas iniciales.

$PNS(1), PEW(1)$: Proyecciones iniciales en el Eje "Y" ó "X"

VI.5. EJEMPLO.

Datos: Dist. 1-2 = 200 m.

Az. 1-2 = $195^\circ 19'$

a) Cálculo de las Proyecciones.

$$PY_{1-2} = \text{COS } AZ_{1-2} \times \text{dist}_{1-2}$$

$$= (-0.96448062) 200$$

$$= -192.8961$$

$$PX_{1-2} = \text{SEN } AZ_{1-2} \times \text{Dist}_{1-2}$$

$$= (-0.26415361) 200$$

$$= -52.8307$$

Así, se van calculando las demás Proyecciones, hasta quedar como la siguiente tabla :

L A D O DE A	DISTANCIA (m.)	AZIMUT		PROYECCIONES ORIGINALES		
		g	m	N-S	E-W	
1	2	200.00	195	19	-192.8961	-52.8307
2	3	231.33	102	05	-48.4253	+226.2047
3	4	117.69	56	39	+64.7003	+98.3098
4	5	85.95	40	30	+65.0310	+56.1993
5	6	55.43	21	47	+32.9001	-13.1480
6	1	350.00	282	59	+78.6337	-341.0524
SUMAS =		1020.40			482.5866	737.7449

b) Condición del Cierre Lineal .

$$\begin{aligned} \text{SN} - \text{SS} &= \text{CERO} \\ &= 241.2652 - 241.3214 \\ &= -0.05625153 \end{aligned}$$

$$\begin{aligned} \text{SE} - \text{SW} &= \text{CERO} \\ &= 393.8618 - 393.8831 \\ &= -0.02133179 \end{aligned}$$

Como se ve, ambos resultados son diferentes de CERO.

c) Cálculo del error .

$$\begin{aligned} \text{EY} &= \text{SN} - \text{SS} \\ &= -0.05625153 \end{aligned}$$

$$\begin{aligned} \text{EX} &= \text{SE} - \text{SW} \\ &= -0.02133179 \end{aligned}$$

d) Error Total Lineal .

$$\begin{aligned}
 ET &= \sqrt{EY^2 + EX^2} \\
 &= \sqrt{(-0.05625153)^2 + (-0.02133179)^2} \\
 &= 0.06016045
 \end{aligned}$$

e) Precisión del Trabajo .

$$PT = \frac{SUMB}{ET}$$

$$PT = \frac{1020.40}{0.06016045}$$

$$PT = 1 : 16,961$$

f) Compensación .

Cálculo de las constantes :

$$KY = \left| EY / SUMY \right|$$

$$= \left| \frac{-0.05625153}{482.5866} \right|$$

$$= 1.16562561 \text{ E-04}$$

$$KX = \left| EX / SUMX \right|$$

$$= \left| \frac{-0.02133179}{787.7449} \right|$$

$$= 2.707956599 \text{ E-05}$$

Correcciones 1-2 en "Y" y en "X", con signo negativo :

$$CY_{1-2} = - KY \left| (PY_{1-2}) \right|$$

$$= -1.16562561 \text{ E-04} \left| (-192.8961) \right|$$

$$= -0.02248447$$

$$\begin{aligned}
 CX_{1-2} &= - KX \left| (PX_{1-2}) \right| \\
 &= -2.707954599 E-05 \left| (-52.8307) \right| \\
 &= -0.001430633
 \end{aligned}$$

Calculando de la misma forma, se obtienen los siguientes resultados :

L A D O		C O R R E C C I O N E S	
DE	A	Y	X
1	2	-2.248447E-02	-1.430633E-03
2	3	-5.644575E-03	+6.125524E-03
3	4	+7.541635E-03	+2.662185E-03
4	5	+7.580185E-03	+1.521854E-03
5	6	+3.934917E-03	+3.560419E-04
6	1	+9.165747E-03	-9.235549E-03

g) Comprobación.

$$SCY = |EY|$$

$$5.625153 E-02 = \left| -5.625153 E-02 \right|$$

$$SCX = |EX|$$

$$2.133179 E-02 = \left| -2.133179 E-02 \right|$$

Por lo tanto, cumplen con la igualdad, es decir, con la Condición del Cierre Lineal.

h) Cálculo de las Proyecciones Corregidas.

Las Proyecciones Originales 1-2, en "Y" y en "X", son negativas, entonces :

$$\begin{aligned}
 PNS_{1-2} &= - (|PY_{1-2}| + CY_{1-2}) \\
 &= - (|-192.8761| + (-0.022)) \\
 &= - 192.8736
 \end{aligned}$$

$$\begin{aligned}
 FEW_{1-2} &= - (| PX_{1-2} | + CX_{1-2}) \\
 &= - (| -52.8307 | + (-0.001)) \\
 &= - 52.82939
 \end{aligned}$$

Así, sucesivamente se van calculando, hasta quedar como sigue :

L A D O		PROYECCIONES CORREGIDAS	
DE	A	N-S	E-W
1	2	-192.8736	-52.82939
2	3	-48.41964	+226.2108
3	4	+64.70786	+98.31242
4	5	+65.03863	+56.20087
5	6	+32.70391	+13.14835
6	1	+78.64288	-341.0431

i) Cálculo de las Coordenadas.

Sean las coordenadas iniciales : $Y_1 = 400$
 $X_1 = 400$

$$\begin{aligned}
 Y_2 &= Y_1 + PNS_{1-2} \\
 &= 400 + (-192.8736) \\
 &= 207.1264
 \end{aligned}$$

$$\begin{aligned}
 X_2 &= X_1 + FEW_{1-2} \\
 &= 400 + (-52.82939) \\
 &= 347.1706
 \end{aligned}$$

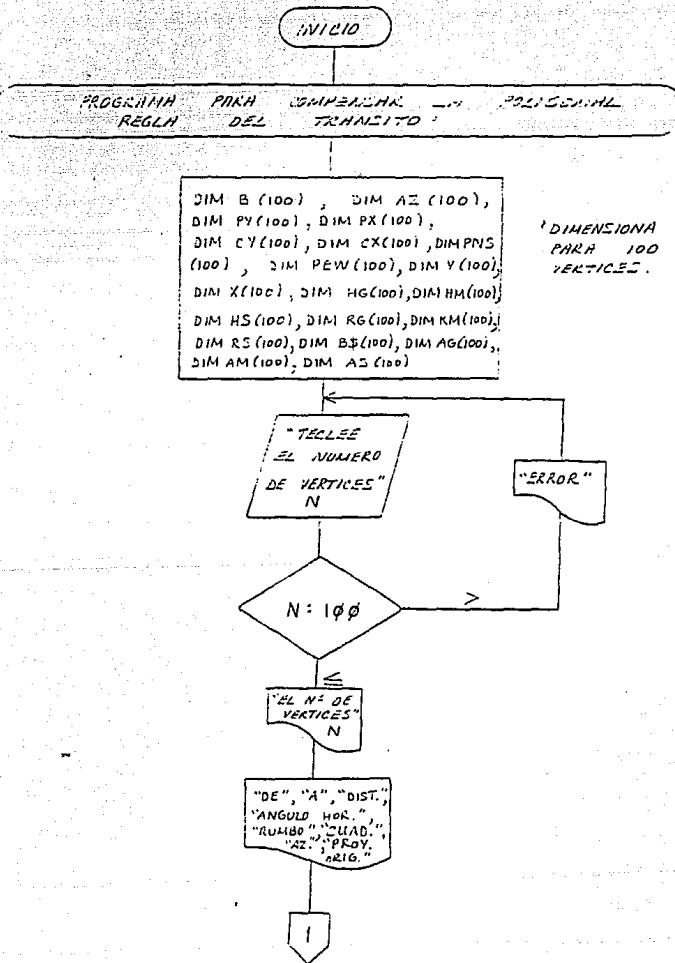
De este modo, se calculan las demás coordenadas, quedando finalmente :

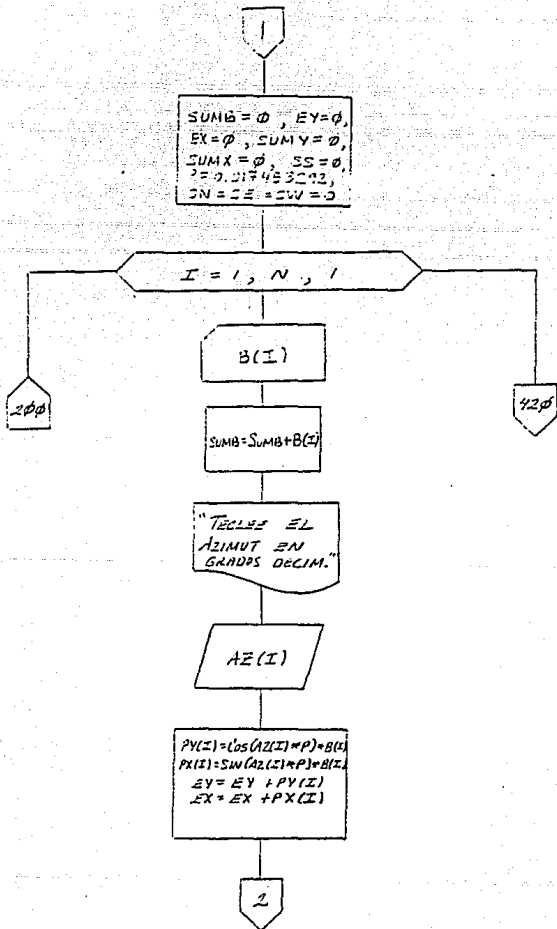
VERTICE	C O O R D E N A D A S	
	X	Y
1	400	400
2	207.1264	347.1706
3	158.7067	573.3815
4	223.4146	671.6939
5	288.4532	727.8947
6	321.3571	741.0431
1	400	400

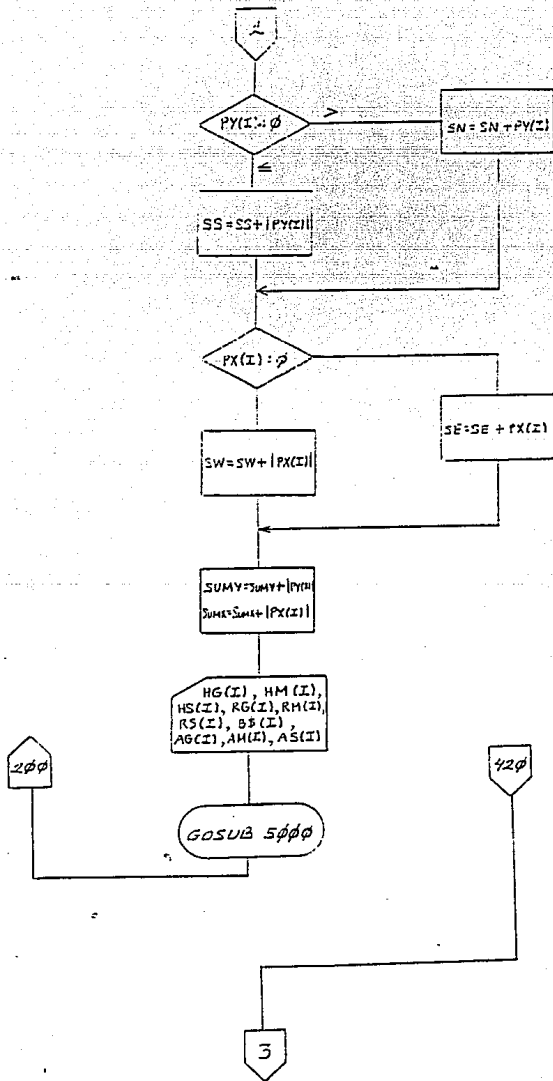
VI.6. VARIABLES UTILIZADAS.

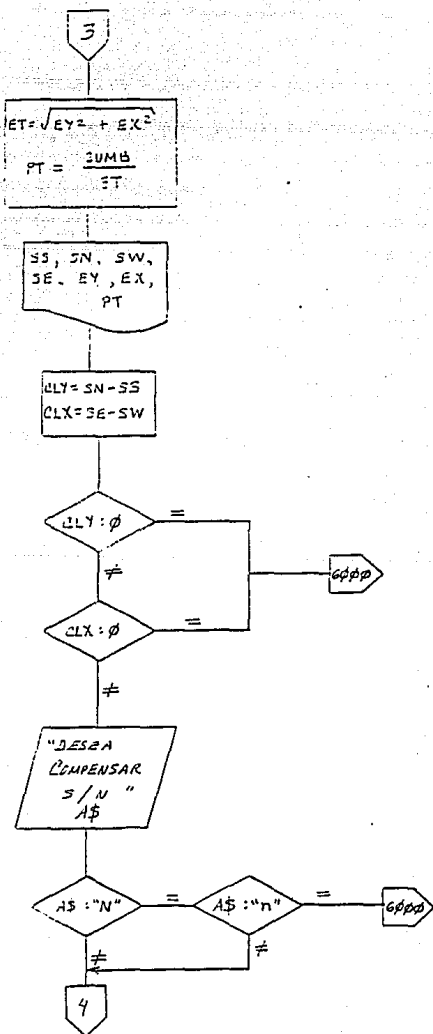
VARIABLE:	DEFINICION:
N	Número de vértices.
P	$\pi / 180^\circ$. Constante para convertir radianes a grados.
B (I)	Distancia.
SUMB	Sumatoria aritmética de la distancia.
AZ (I)	Azimut en grados decimales.
PY (I)	Proyección N - S.
PX (I)	Proyección E - W.
EY	Error en Y.
EX	Error en X.
SS	Sumatoria aritmética de la Proyección SUR.
SN	Sumatoria aritmética de la Proyección NORTE.
SW	Sumatoria aritmética de la Proyección OESTE (West) .
SE	Sumatoria aritmética de la Proyección ESTE.
SUMY	Sumatoria algebraica N - S .
SUMX	Sumatoria algebraica E - W .
HG (I)	Grados del ángulo.
HM (I)	Minutos del ángulo.
HS (I)	Segundos del ángulo.
RG (I)	Grados del rumbo.
RM (I)	Minutos del rumbo.
RS (I)	Segundos del rumbo.
B# (I)	Cuadrante del rumbo.

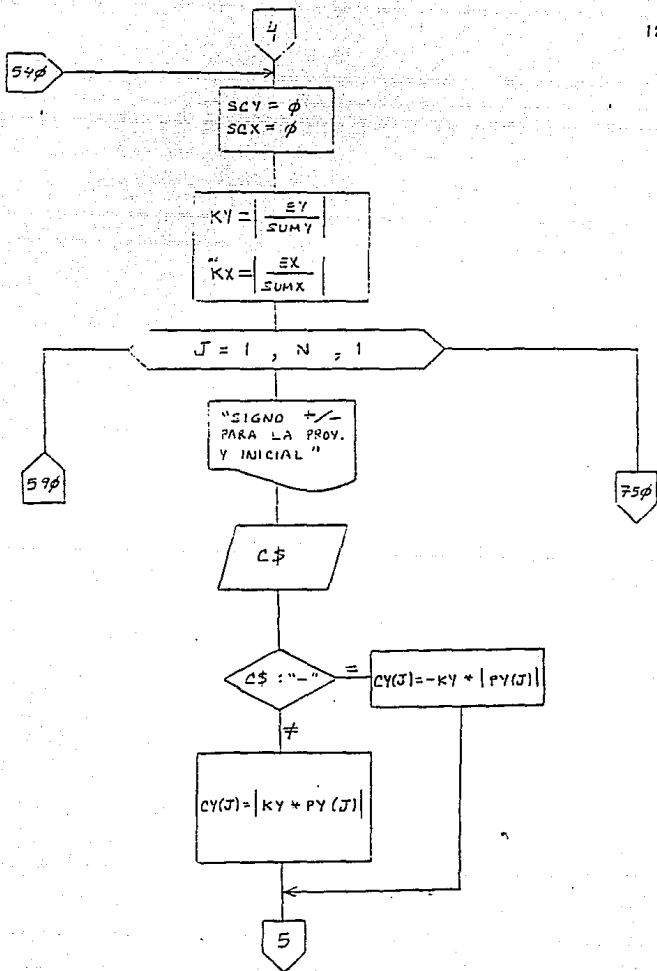
VARIABLE:	DEFINICION:
AG (I)	Grados del acimut.
AM (I)	Minutos del acimut.
AS (I)	Segundos del acimut.
ET	Error lineal total.
PT	Precisión del trabajo.
CLY	Diferencia de proyecciones, para la Condición del cierre lineal: $GN - SS = 0$
CLX	Diferencia de proyecciones, para la Condición del cierre lineal: $SE - SW = 0$
A\$,C\$,D\$,A1\$	Variables de Cadena.
I, J	Contadores numéricos.
KY	Constante de la Corrección en " Y " .
KX	Constante de la Corrección en " X " .
CY (I)	Corrección en Y .
CX (I)	Corrección en X .
SCY	Sumatoria aritmética de la corrección en " Y " .
SCX	Sumatoria aritmética de la corrección en " X " .
C1, C2	Correcciones adicionales para las proyecciones iniciales : "Y" ó "X" .
PNS (I)	Proyección corregida N - S .
PEW (I)	Proyección corregida E - W .
Y (I)	Coordenada en Y.
X (I)	Coordenada en X.

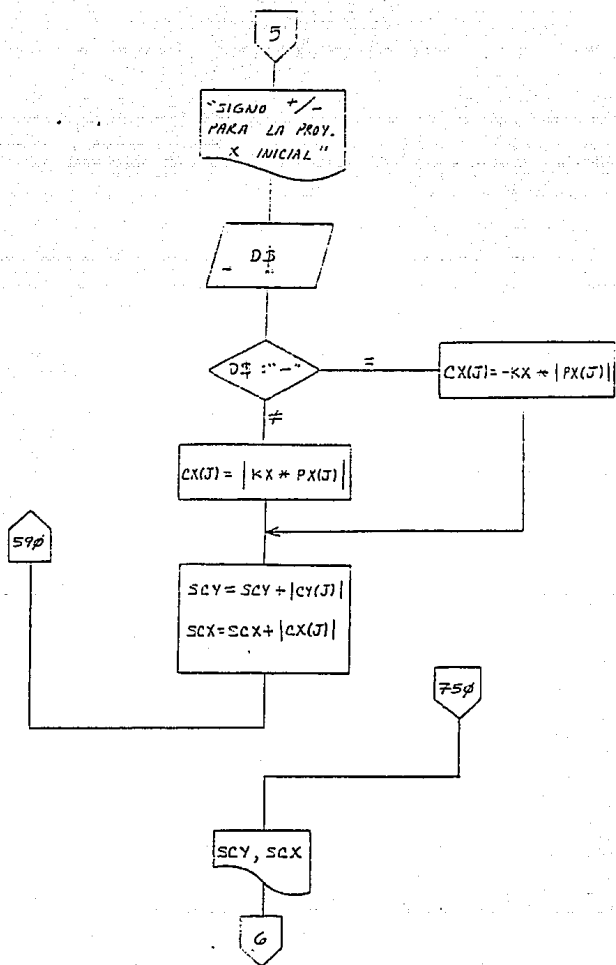


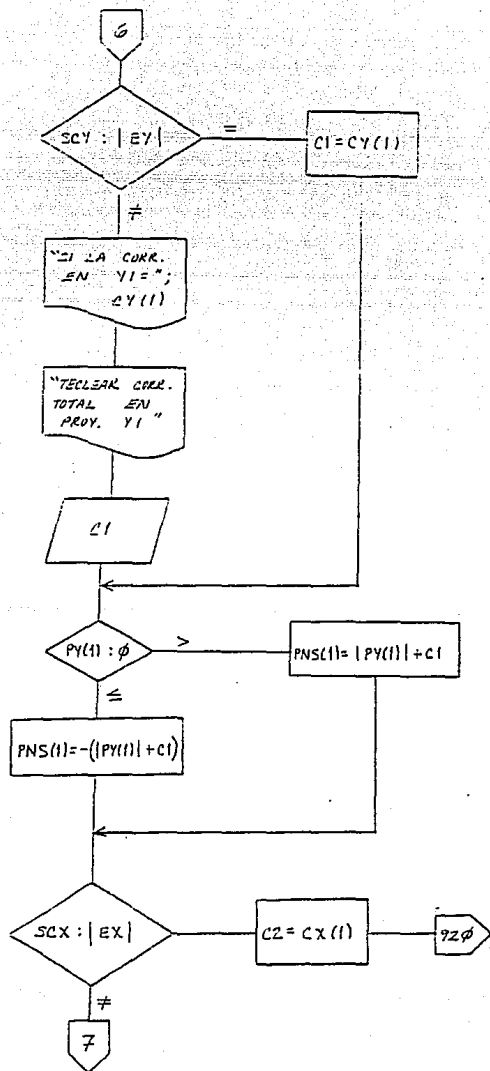


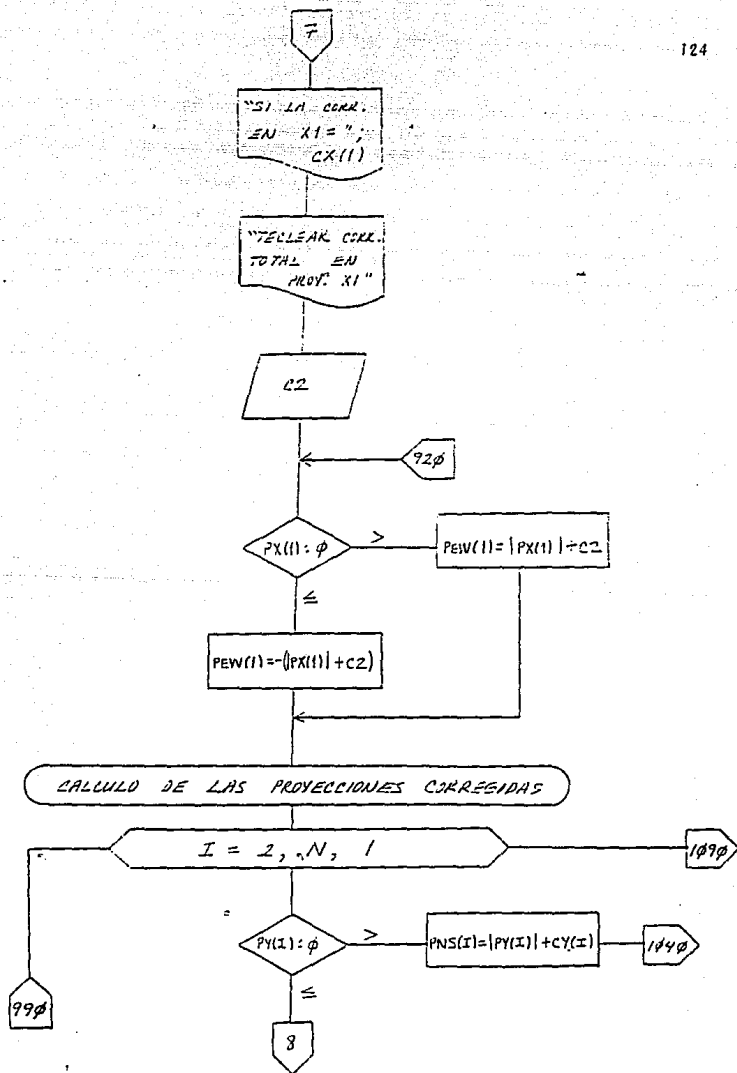


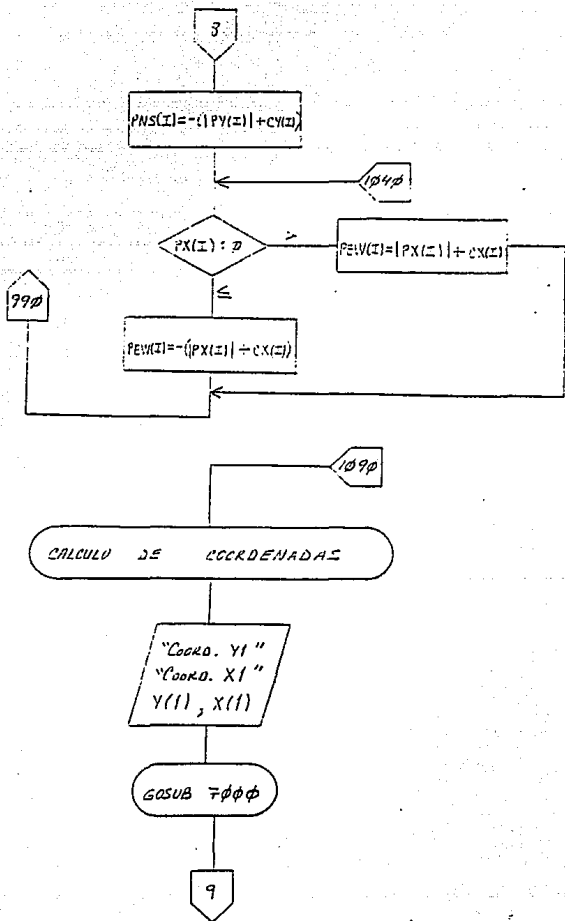


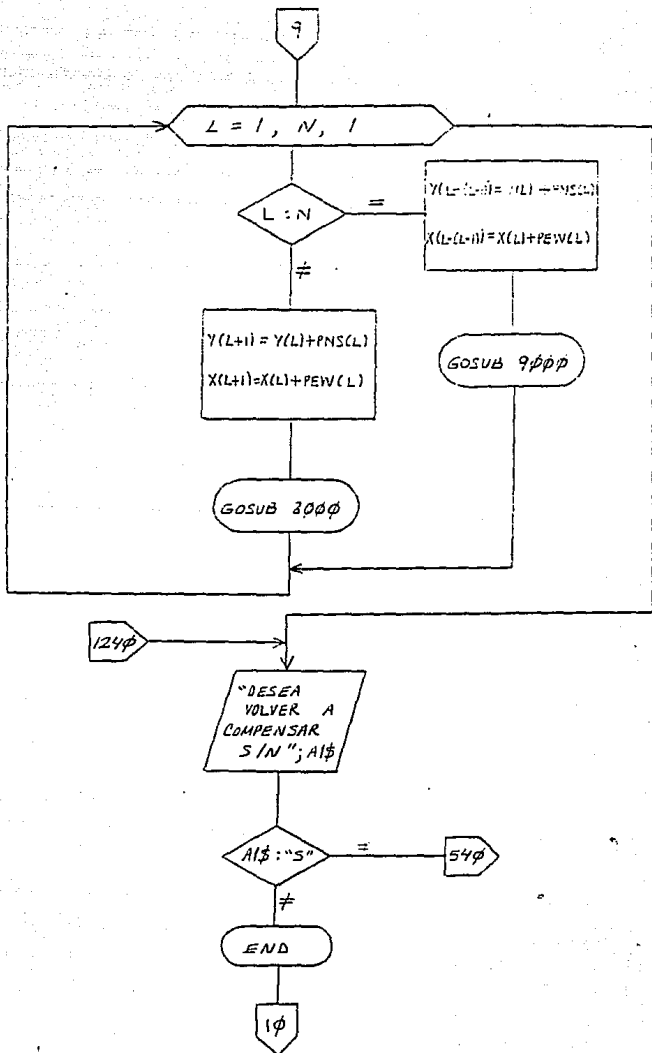


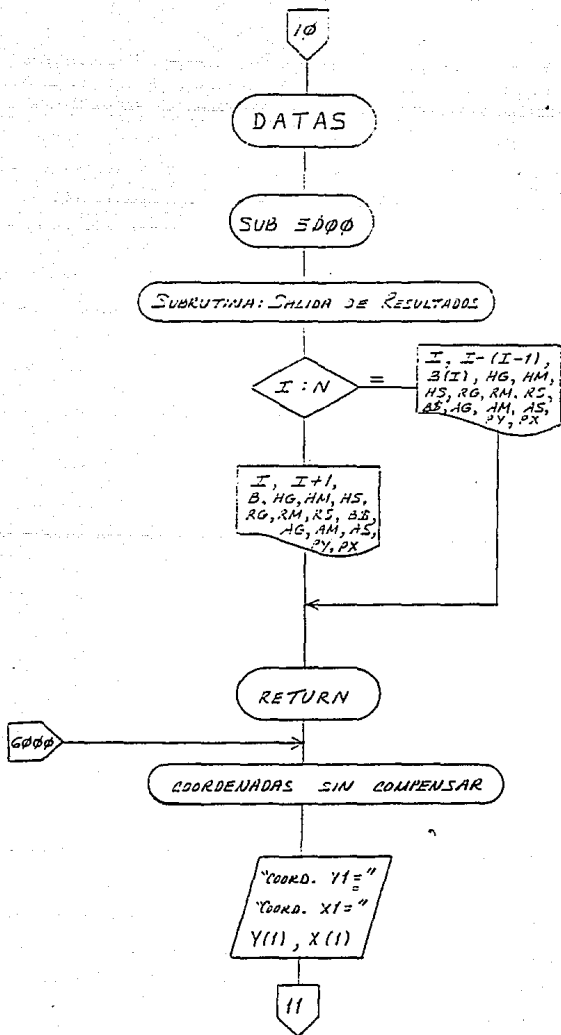


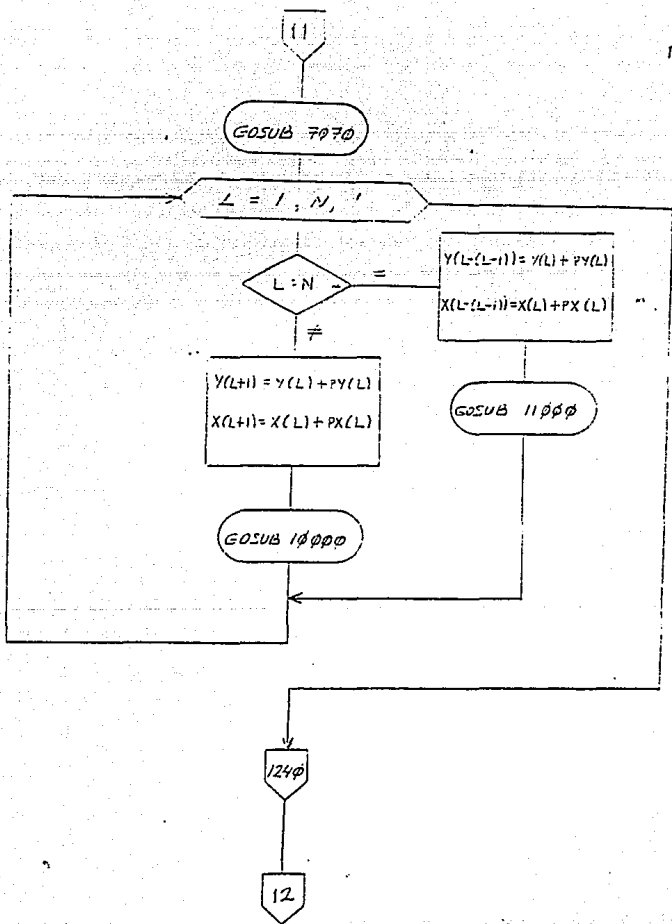


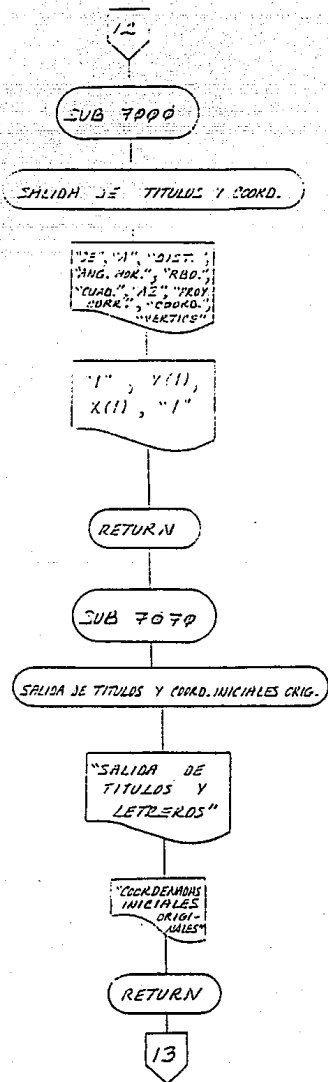


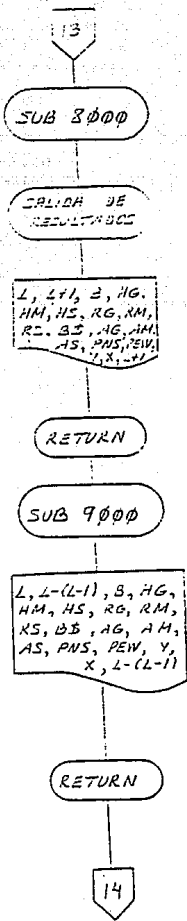












14

SUB 10000

L, L+1, A, "ANGULO"
"HORIZONTAL"; "RBO",
"AZIMUT", PY(L),
PX(L), Y(L+1),
X(L+1), L+1

RETURN

SUB 11000

L, L-(L-1), B,
"ANG. HOR.", "RBO",
"AZ.", PY(L),
PX(L), Y(1),
X(1), L-(L-1)

RETURN

VI.3. CODIFICACION.

```

10 REM PROGRAMA PARA COMPENSAR LA POLIGONAL
20 REM ** REGLA DEL TRANSITO **
30 DIM B(100): DIM AZ(100): DIM PY(100): DIM PX(100)
40 DIM CY(100): DIM CX(100): DIM PMS(100): DIM PEM(100)
50 DIM Y(100): DIM X(100)
60 DIM HG(100): DIM HM(100): DIM HS(100)
70 DIM R6(100): DIM RM(100): DIM RS(100)
80 DIM B4(100): DIM A6(100): DIM AM(100): DIM AS(100)
90 INPUT "TECLEE EL NUMERO DE VERTICES:"; N
100 IF N > 100 THEN PRINT "ERROR : Excede el numero lizats": GOTO 70
110 PRINT "EL NUMERO DE VERTICES=";N
120 PRINT " "
130 PRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." TAB(37) ;
    "RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES ORIGINALES"
140 PRINT TAB(23)"y" TAB(27)"x" TAB(31)"s" TAB(35)"g" TAB(39)"a" TAB(43)"s" ;
    TAB(54)"g" TAB(58)"s" TAB(62)"s" TAB(70)"N-S" TAB(83)"E-W"
150 SUMB=0: EY=0: EX=0: SUNY=0: SUMX=0: P=.0174532928
160 SS=0: SM=0: SE=0
170 PRINT " "
180 :
190 'Comprobacion del Cierre Lineal
200 FOR I = 1 TO N
210 READ B(I) 'Lee la distancia
220 SUMB=SUMB + B(I) 'Sumatoria de la distancia
230 PRINT "Teclen en grados decimales,el AZ.";I;:"
240 INPUT AZ(I)
250 PY(I)= COS(AZ(I) * P) * B(I) 'Proyeccion N-S
260 PX(I)= SIN(AZ(I) * P) * B(I) 'Proyeccion E-W
270 EY=EY + PY(I) 'Error en Y
280 EX=EX + PX(I) 'Error en X
290 IF PY(I) > 0 THEN GOTO 320
300 SS=SS + ABS(PY(I))
310 SOTO 330
320 SM=SM + PY(I)
330 IF PX(I) > 0 THEN GOTO 360
340 SM=SM + ABS(PX(I))
350 SOTO 370
360 SE=SE + PX(I)
370 SUNY= SUNY + ABS(PY(I)) 'Sumatoria N-S
380 SUMX= SUMX + ABS(PX(I)) 'Sumatoria E-W
390 READ HG(I),HM(I),HS(I),R6(I),RM(I),RS(I),B4(I),A6(I),AM(I),AS(I)
400 SOSUB 5000
410 NEXT I
420 ET=SGR((EY^2) + (EX^2)) 'Error Total
430 PT = (SUMB/ET) 'Precision del Trabajo
440 PRINT " "
450 PRINT TAB(67) "SS=";SS
460 PRINT TAB(67) "SM=";SM
470 PRINT TAB(80) "SE=";SE
480 PRINT TAB(80) "S=";S

```

```

490 PRINT TAB(67) "EY=";EY
500 PRINT TAB(80) "EX=";EI
510 PRINT "LA PRECISION ES="; "1 : ";PT
515 CLY=SN-SS : CLX=SE-SM
516 IF CLY=0 AND CLX=0 THEN GOTO 6000
520 INPUT "Desee Condensar S/N";AS
530 IF AS = "M" OR AS = "n" THEN GOTO 6000
540 :
550 SCY=0; SCX=0
560 'Condensacion
570 KY=ABS(EY/SUMY) 'Constante Y
580 KX=ABS(EX/SUMX) 'Constante X
590 FOR J = 1 TO N
600 PRINT "SIGNO +/- PARA LA PROYECCION Y";J;"="
610 INPUT C$
620 IF C$ = "-" GOTO 650
630 CY(J) = ABS(KY * PY(J)) 'Correccion en Y
640 GOTO 660
650 CY(J) = -KY * ABS(PY(J))
660 PRINT "SIGNO +/- PARA LA PROYECCION X";J;"="
670 INPUT D$
680 IF D$ = "-" GOTO 710
690 CX(J) = ABS(KX * PX(J)) 'Correccion en X
700 GOTO 720
710 CX(J) = -KX * ABS(PX(J))
720 SCY = SCY + ABS(CY(J)) 'Suma de la correccion en Y
730 SCX = SCX + ABS(CX(J)) 'Suma de la correccion en X
740 NEXT J
750 PRINT "Sumatoria CY=";SCY
760 PRINT "Sumatoria CX=";SCX
770 IF SCY = ABS(EY) THEN C1=CY(1) :GOTO 820
780 PRINT "Si la Correccion en Y1=";CY(1)
790 PRINT "Teclar Correccion Total en Proyeccion Y!"
800 PRINT "Ejemplo: +0.015 , -0.023 ,etc."
810 INPUT C1
820 :
830 IF PY(1) > 0 THEN GOTO 860
840 PMS(1) = - (ABS(PY(1)) + C1)
850 GOTO 870
860 PMS(1) = ABS(PY(1)) + C1
870 IF SCX = ABS(EX) THEN C2=CX(1):GOTO 920
880 PRINT "Si la Correccion en X1=";CX(1)
890 PRINT "Teclar Correccion Total en Proyeccion X!"
900 PRINT "Ejemplo: +0.015 , -0.023 ,etc."
910 INPUT C2
920 :
930 IF PX(1) > 0 THEN GOTO 960
940 PEW(1) = - (ABS(PX(1)) + C2)
950 GOTO 970
960 PEW(1) = ABS(PX(1)) + C2
970 :

```

```

790 'Calculo de las Proyecciones Corregidas
790 FOR I = 2 TO N
1000 IF PY(I) > 0 THEN GOTO 1030
1010 PMS(I) = - (ABS(PY(I)) + CY(I))
1020 GOTO 1040
1030 PMS(I) = ABS(PY(I)) + CY(I) 'Proyeccion Corregida N-S
1040 IF PX(I) > 0 THEN GOTO 1070
1050 PEM(I) = - (ABS(PX(I)) - CX(I))
1060 GOTO 1080
1070 PEM(I) = ABS(PX(I)) + CX(I) 'Proyeccion Corregida E-W
1080 NEXT I
1090 :
1100 'Calculo de Coordenadas
1110 INPUT 'Coordenada Y inicial=';Y(I)
1120 INPUT 'Coordenada X inicial=';X(I)
1130 GOSUB 7000
1140 FOR L = 1 TO N
1150 IF L = N THEN GOTO 1200
1160 Y(L+1) = Y(L) + PMS(L)
1170 X(L+1) = X(L) + PEM(L)
1180 GOSUB 8000
1190 GOTO 1230
1200 Y(L-L-1) = Y(L) + PMS(L)
1210 X(L-L-1) = X(L) + PEM(L)
1220 GOSUB 9000
1230 NEXT L
1240 INPUT 'Desea Volver a Compensar S/N';A1$
1250 IF A1$ = 'S' OR A1$ = 's' THEN GOTO 540
1260 END
1270 DATA 200,92,20,0,15,19,0,'SW',195,19,0
1280 DATA 231.33,86,46,0,77,55,0,'SE',102,5,0
1290 DATA 117.69,134,34,0,56,39,0,'NE',56,39,0
1300 DATA 85.95,164,11,0,40,50,0,'NE',40,50,0
1310 DATA 35.43,160,57,0,21,47,0,'NE',21,47,0
1320 DATA 350,81,12,0,77,01,0,'NW',232,59,0
5000 REM Subrutina : Salida de Resultados :
5010 IF I = N THEN GOTO 5040
5020 PRINT I TAB(6) I+1 TAB(11)B(I) TAB(21)HG(I) TAB(26)HM(I) TAB(30)HS(I);
TAB(34)RG(I) TAB(38)RM(I) TAB(42)RS(I) TAB(48) 90(I) TAB(52)AG(I) ;
TAB(57)AM(I) TAB(61)AS(I) TAB(66) USING '*###.### ' ;PY(I),PX(I)
5030 GOTO 5050
5040 PRINT I TAB(6) I-(I-1) TAB(11)B(I) TAB(21)HG(I) TAB(26)HM(I) TAB(30)HS(I);
TAB(34)RG(I) TAB(38)RM(I) TAB(42)RS(I) TAB(48) 90(I) TAB(52)AG(I) ;
TAB(57)AM(I) TAB(61)AS(I) TAB(66) USING '*###.### ' ;PY(I),PX(I)
5050 RETURN
6000 'Coordenadas sin Compensar
6010 INPUT 'Coordenada Y inicial=';Y(I)
6020 INPUT 'Coordenada X inicial=';X(I)
6030 GOSUB 7070
6040 FOR L = 1 TO N

```

```

5050 IF L = N THEN GOTO 6100
5060 Y(L+1) = Y(L) + PY(L)
5070 X(L+1) = X(L) + PX(L)
5080 GOSUB 10000
5090 GOTO 6130
5100 Y(L-L-1) = Y(L) - PY(L)
5110 X(L-L-1) = X(L) + PX(L)
5120 GOSUB 11000
6130 NEXT L
5140 GOTO 1240
7000 'SUBROUTINA : Salida de Titulos y Coordenadas Iniciales :
7010 PRINT ' ': PRINT ' ': PRINT ' '
7020 PRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." ;
TAB(37)"RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES CORR." ;
TAB(92)"COORDENADAS" TAB(108)"VERTICE"
7030 PRINT TAB(23)"g" TAB(27)"s" TAB(31)"s" TAB(35)"g" TAB(39)"s" TAB(43)"s" ;
TAB(54)"g" TAB(58)"s" TAB(62)"s" TAB(70)"M-S" TAB(80)"E-W" TAB(92)"Y" ;
TAB(102)"X"
7040 PRINT ' '
7050 PRINT TAB(7)"*" TAB(68) Y(1) TAB(98) X(1) TAB(111)"*"
7060 RETURN
7070 'SUBROUTINA : Salida de Titulos y Coordenadas Iniciales sin corr. :
7080 PRINT ' ': PRINT ' ': PRINT ' '
7090 PRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." ;
TAB(37)"RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES ORIG." ;
TAB(92)"COORDENADAS" TAB(108)"VERTICE"
7100 PRINT TAB(23)"g" TAB(27)"s" TAB(31)"s" TAB(35)"g" TAB(39)"s" TAB(43)"s" ;
TAB(54)"g" TAB(58)"s" TAB(62)"s" TAB(70)"M-S" TAB(80)"E-W" TAB(92)"Y" ;
TAB(102)"X"
7110 PRINT ' '
7120 PRINT TAB(7)"*" TAB(68) Y(1) TAB(98) X(1) TAB(111)"*"
7130 RETURN
9000 'SUBROUTINA : Salida de Resultados :
9010 PRINT L TAB(6) L+1 TAB(11) B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AN(L) ;
TAB(61)AS(L) TAB(66) PHS(L) TAB(76)PEW(L) TAB(88) Y(L+1) TAB(98) X(L+1) ;
TAB(110) L+1
9020 RETURN
9000 'SUBROUTINA : Salida de Resultados :
9010 PRINT L TAB(6) L-(L-1) TAB(11)B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AN(L) ;
TAB(61)AS(L) TAB(66) PHS(L) TAB(76)PEW(L) TAB(88) Y(1) TAB(98) X(1) TAB(110) ;
L-(L-1)
9020 RETURN
10000 'SUBROUTINA : Salida de Resultados :
10010 PRINT L TAB(6) L+1 TAB(11) B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AN(L) ;
TAB(61)AS(L) TAB(66) PY(L) TAB(76) PX(L) TAB(88) Y(L+1) TAB(98) X(L+1) ;
TAB(110) L+1
10020 RETURN
11000 'SUBROUTINA : Salida de Resultados :
11010 PRINT L TAB(6) L-(L-1) TAB(11)B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AN(L) ;
TAB(61)AS(L) TAB(66) PY(L) TAB(76) PX(L) TAB(88) Y(1) TAB(98) X(1) TAB(110) ;
L-(L-1)
11020 RETURN

```

11.9. PRUEBA DE ESCRITORIO

PROYECTO	TESIS	ARCHIVO	L.C.I. 10. Bas	FECHA	1993
PROGRAMA	Gwbasic	ELABORO	MPGIA	HOJA	136 DE 144

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
N	6					
P	$\pi/180^\circ$					
I	1	2	3	4	5	6
B(I)	200	231.33	117.69	85.95	35.23	350
SUMB	200	431.33	549.02	634.77	670.70	1020.70
AZ(I)	195.3166667	102.0833333	56.65	40.83333333	21.78333333	282.9833333
PY(I)	-192.8961	-48.4253	164.7003	165.0310	132.4001	178.6337
PX(I)	-52.8307	1226.2047	178.3078	156.1793	113.1450	-341.0524
EY	-192.8961	-241.3214	-176.6211	-111.5901	-78.68198	5.625153E-02
EX	-52.8307	1173.374	1271.6837	1327.8831	1341.031	-2.133179E-02
PY(I) > ϕ	NO	NO	GOTO 320	GOTO 320	GOTO 320	GOTO 320
SS	192.8961	241.3214	—	—	—	241.3214
SN	—	—	64.70032	129.7314	162.6514	241.2652
PX(I) > ϕ	NO	GOTO 360	GOTO 360	GOTO 360	GOTO 360	NO
SW	52.8307	—	—	—	—	393.8831
SE	—	226.2047	324.5147	380.7138	393.8618	393.8618
SUMY	192.8961	241.3214	306.0217	371.0528	403.9529	482.5866
SUMX	52.8307	277.0354	377.3452	433.5445	446.6725	187.7447
ET	6.016045E-02					
PT	16,961.31					

PROYECTO	TESIS	ARCHIVO	LCL10.Bos	FECHA	1993
PROGRAMA	Gwbasic	ELABORO	MPGM	HORA	137 DE 144

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
CLY = ϕ	\neq					
CLX = ϕ	:					
A ϕ	"S"					
KY	1.165626E-04					
KX	2.707956E-05					
J	1	2	3	4	5	6
C ϕ	"-"	"-"	"+"	"+"	"+"	"+"
	GOTO 65 ϕ	GOTO 65 ϕ	:	:	:	:
LY(J)	-2.248447E-02	-5.644575E-03	7.541635E-03	7.580185E-03	3.834717E-03	7.165147E-03
D ϕ	"-"	"+"	"+"	"+"	"+"	"-"
	GOTO 71 ϕ	:	:	:	:	GOTO 71 ϕ
CX(J)	-1.430633E-03	6.125524E-03	2.662185E-03	1.521854E-03	3.560714E-04	-7.235541E-03
SCY	5.625153E-02				:	
SCX	2.133179E-02					
SCY = EY	S1					
CI	-2.248447E-02					
	GOTO 820					
PY(I) > ϕ	NO					
PNS(I)	-192.8736					
	GOTO 870					

VI.10. IMPRESION DEL PROGRAMA.

```

10 REM PROGRAMA PARA COMPENSAR LA POLIGONAL
20 REM ** RECLA DEL TRANSITO **
30 WIDTH "LPT1:",120
40 OPEN "LPT1:"FOR OUTPUT AS #1
50 DIM B(100): DIM AZ(100): DIM PY(100): DIM PX(100)
60 DIM CY(100): DIM CX(100): DIM PMS(100): DIM PEM(100)
70 DIM Y(100): DIM X(100)
80 DIM HG(100): DIM HM(100): DIM HS(100)
90 DIM RG(100): DIM RM(100): DIM RS(100)
100 DIM Bx(100): DIM Ag(100): DIM Am(100): DIM As(100)
110 INPUT "TECLEE EL NUMERO DE VERTICES:"; N
120 IF N > 100 THEN PRINT "ERROR : Excede el nuero limite": GOTO 110
130 LPRINT "EL NUMERO DE VERTICES=";N
140 LPRINT " "
150 LPRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." TAB(37) ;
      "RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES ORIGINALES"
160 LPRINT TAB(23)"g" TAB(27)"a" TAB(31)"s" TAB(35)"j" TAB(39)"a" TAB(43)"s" ;
      TAB(54)"g" TAB(58)"a" TAB(62)"s" TAB(70)"N-S" TAB(83)"E-W"
170 SUMB=0: EY=0: EX=0: SUMY=0: SUMX=0: P=-.017453292#
180 SS=0: SN=0: SM=0: SE=0
190 LPRINT " "
200 :
210 'Coaprobacion del Cierre Lineal
220 FOR I = 1 TO N
230   READ B(I)           'Lee la distancia
240   SUMB=SUMB + B(I)    'Suatoria de la distancia
250   PRINT "Teclee en grados decimales.el AZ.;"I";"
260   INPUT AZ(I)
270   PY(I)= COS(AZ(I) * P) * B(I)   'Proyeccion N-S
280   PX(I)= SIN(AZ(I) * P) * B(I)   'Proyeccion E-W
290   EY=EY + PY(I)                'Error en Y
300   EX=EX + PX(I)                'Error en X
310   IF PY(I) > 0 THEN GOTO 340
320   SS = SS + ABS(PY(I))
330   GOTO 350
340   SN = SN + PY(I)
350   IF PX(I) > 0 THEN GOTO 380
360   SM = SM + ABS(PX(I))
370   GOTO 390
380   SE = SE + PX(I)
390   SUMY = SUMY + ABS(PY(I))        'Suatoria N-S
400   SUMX = SUMX + ABS(PX(I))        'Suatoria E-W
410   READ HG(I),HM(I),HS(I),RG(I),RM(I),RS(I),Bx(I),Ag(I),Am(I),As(I)
420   GOSUB 5000
430 NEXT I
440 ET=SGR((EY^2) + (EX^2))           'Error Total
450 PT = (SUMB/ET)                   'Precision del Trabajo
460 LPRINT " "
470 LPRINT TAB(67) "SS=";SS
480 LPRINT TAB(67) "SN=";SN
490 LPRINT TAB(60) "SM=";SM
500 LPRINT TAB(60) "SE=";SE

```

```

510 LPRINT TAB(67) "CY=";EY
515 CLY=SN-SS : CLI=SE-SW
516 IF CLY=0 AND CLI=0 THEN GOTO 6000
520 LPRINT TAB(80) "EX=";EX
530 LPRINT "LA PRECISION ES=";PI : ' ;PT
540 INPUT "Desear Compensar S/N";A$
550 IF A$ = "N" OR A$ = "n" THEN GOTO 6000
560 :
570 SCY=0; SCX=0
580 'Compensacion
590 KY=ABS(EY/SUMY) 'Constante Y
600 KX=ABS(EX/SUMX) 'Constante X
610 FOR J = 1 TO N
620 PRINT "SIGNO +/- PARA LA PROYECCION Y";J;"="
630 INPUT C$
640 IF C$ = "-" GOTO 670
650 CY(J) = ABS(KY * PY(J)) 'Correccion en Y
660 GOTO 680
670 CY(J) = -KY * ABS(PY(J))
680 PRINT "SIGNO +/- PARA LA PROYECCION X";J;"="
690 INPUT D$
700 IF D$ = "-" GOTO 730
710 CX(J) = ABS(KX * PX(J)) 'Correccion en X
720 GOTO 740
730 CX(J) = -KX * ABS(PX(J))
740 SCY = SCY + ABS(CY(J)) 'Suav de la correccion en Y
750 SCX = SCX + ABS(CX(J)) 'Suav de la correccion en X
760 NEXT J
770 LPRINT "Sumatoria CY=";SCY
780 LPRINT "Sumatoria CX=";SCX
790 IF SCY = ABS(EY) THEN C1=CY(1) :GOTO 840
800 PRINT "Si la Correccion en Y=";CY(1)
810 PRINT "Teclear Correccion Total en Proyeccion Y1"
820 PRINT "Ejemplo: +0.015 , -0.023 ,etc."
830 INPUT C1
840 :
850 IF PY(1) > 0 THEN GOTO 880
860 PMS(1) = - (ABS(PY(1)) + C1)
870 GOTO 890
880 PMS(1) = ABS(PY(1)) + C1
890 IF SCX = ABS(EX) THEN C2=CX(1):GOTO 940
900 PRINT "Si la Correccion en X=";CX(1)
910 PRINT "Teclear Correccion Total en Proyeccion X1"
920 PRINT "Ejemplo: +0.015 , -0.023 ,etc."
930 INPUT C2
940 :
950 IF PX(1) > 0 THEN GOTO 980
960 PEM(1) = - (ABS(PX(1)) + C2)
970 GOTO 990
980 PEM(1) = ABS(PX(1)) + C2
990 :
1000 'Calculo de las Proyecciones Corregidas

```

```

1010 FOR I = 2 TO N
1020 IF PY(I) > 0 THEN GOTO 1050
1030 PMS(I) = - (ABS(PY(I)) + CY(I))
1040 GOTO 1060
1050 PMS(I) = ABS(PY(I)) + CY(I) 'Proyeccion Corregida N-S
1060 IF PX(I) > 0 THEN GOTO 1090
1070 PEM(I) = - (ABS(PX(I)) + CX(I))
1080 GOTO 1100
1090 PEM(I) = ABS(PX(I)) + CX(I) 'Proyeccion Corregida E-W
1100 NEXT I
1110 :
1120 'Calculo de Coordenadas
1130 INPUT "Coordenada Y inicial=";Y(1)
1140 INPUT "Coordenada X inicial=";X(1)
1150 GOSUB 7000
1160 FOR L = 1 TO N
1170 IF L = N THEN GOTO 1220
1180 Y(L+1) = Y(L) + PMS(L)
1190 X(L+1) = X(L) + PEM(L)
1200 GOSUB 8000
1210 GOTO 1250
1220 Y(L-L-1) = Y(L) + PMS(L)
1230 X(L-L-1) = X(L) + PEM(L)
1240 GOSUB 9000
1250 NEXT L
1260 INPUT "Desea Volver a Compensar S/N";A1$
1270 IF A1$ = "S" OR A1$ = "s" THEN GOTO 580
1280 END
1290 DATA 290,92,20,0,15,19,0,"SW",195,19,0
1300 DATA 231,33,86,46,0,77,55,0,"SE",102,5,0
1310 DATA 117,69,134,34,0,56,39,0,"NE",56,37,0
1320 DATA 85,95,164,11,0,40,50,0,"NE",40,50,0
1330 DATA 35,43,160,57,0,21,47,0,"NE",21,47,0
1340 DATA 350,81,12,0,77,01,0,"NW",282,59,0
5000 REM Subrutina + Salida de Resultados +
5010 IF I = N THEN GOTO 5040
5020 LPRINT I TAB(6) I+1 TAB(11)B(I) TAB(21)HG(I) TAB(26)HM(I) TAB(30)HS(I);
TAB(34)RG(I) TAB(38)RM(I) TAB(42)RS(I) TAB(48) B(I) TAB(52)AG(I) ;
TAB(57)AM(I) TAB(61)AS(I) TAB(66) US:HG "#####.#### " ;PY(I),PX(I)
5030 GOTO 5050
5040 LPRINT I TAB(6) I-(I-1) TAB(11)B(I) TAB(21)HG(I) TAB(26)HM(I) TAB(30)HS(I);
TAB(34)RG(I) TAB(38)RM(I) TAB(42)RS(I) TAB(48) B(I) TAB(52)AG(I) ;
TAB(57)AM(I) TAB(61)AS(I) TAB(66) USING "#####.#### " ;PY(I),PX(I)
5050 RETURN
5000 'Coordenadas sin Compensar
5010 INPUT "Coordenada Y inicial=";Y(1)
5020 INPUT "Coordenada X inicial=";X(1)
5030 GOSUB 7070
5040 FOR L = 1 TO N
5050 IF L = N THEN GOTO 6100
5060 Y(L+1) = Y(L) + PY(L)
5070 X(L+1) = X(L) + PX(L)
5080 GOSUB 10000

```

```

5090      GOTO 5130
5100      Y(L-(L-1)) = Y(L) + PY(L)
5110      X(L-(L-1)) = X(L) + PX(L)
5120      SOSUB 11000
5130      NEXT L
5140      GOTO 1260
7000      'SUBROUTINA # Salida de Titulos y Coordenadas Iniciales #
7010      LPRINT " " ; LPRINT " " ; LPRINT " "
7020      LPRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." ;
          TAB(37)"RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES CORR." ;
          TAB(92)"COORDENADAS" TAB(108)"VERTICE"
7030      LPRINT TAB(23)"g" TAB(27)"s" TAB(31)"s" TAB(35)"g" TAB(39)"a" TAB(43)"s" ;
          TAB(54)"g" TAB(58)"s" TAB(62)"s" TAB(70)"N-S" TAB(80)"E-W" TAB(92)"Y" ;
          TAB(102)"Y"
7040      LPRINT " "
7050      LPRINT TAB(7)"*" TAB(88) Y(1) TAB(98) X(1) TAB(111)"*"
7060      RETURN
7070      'SUBROUTINA # Salida de Titulos y Coordenadas Iniciales sin corr. #
7080      LPRINT " " ; LPRINT " " ; LPRINT " "
7090      LPRINT "DE" TAB(7)"A" TAB(11)"DISTANCIA" TAB(22)"ANGULO HOR." ;
          TAB(37)"RUMBO" TAB(47)"CUAD." TAB(56)"AZIMUT" TAB(66)"PROYECCIONES ORIG." ;
          TAB(92)"COORDENADAS" TAB(108)"VERTICE"
7100      LPRINT TAB(23)"g" TAB(27)"s" TAB(31)"s" TAB(35)"g" TAB(39)"a" TAB(43)"s" ;
          TAB(54)"g" TAB(58)"a" TAB(62)"s" TAB(70)"N-S" TAB(80)"E-W" TAB(92)"Y" ;
          TAB(102)"Y"
7110      LPRINT " "
7120      LPRINT TAB(7)"*" TAB(88) Y(1) TAB(98) X(1) TAB(111)"*"
7130      RETURN
9000      'SUBROUTINA # Salida de Resultados #
9010      LPRINT L TAB(6) L+1 TAB(11) B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
          TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AM(L) ;
          TAB(61)AS(L) TAB(66) PNS(L) TAB(76)PEW(L) TAB(88) Y(L+1) TAB(98) X(L+1) ;
          TAB(110) L+1
9020      RETURN
9000      'SUBROUTINA # Salida de Resultados #
9010      LPRINT L TAB(6) L-(L-1) TAB(11)B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
          TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AM(L) ;
          TAB(61)AS(L) TAB(66) PNS(L) TAB(76)PEW(L) TAB(88) Y(1) TAB(98) X(1) TAB(110) ;
          L-(L-1)
9020      RETURN
10000      'SUBROUTINA # Salida de Resultados #
10010      LPRINT L TAB(6) L+1 TAB(11) B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
          TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AM(L) ;
          TAB(61)AS(L) TAB(66) PY(L) TAB(76) PX(L) TAB(88) Y(L+1) TAB(98) X(L+1) ;
          TAB(110) L+1
10020      RETURN
11000      'SUBROUTINA # Salida de Resultados #
11010      LPRINT L TAB(6) L-(L-1) TAB(11)B(L) TAB(21)HG(L) TAB(26)HM(L) TAB(30)HS(L) ;
          TAB(34)RG(L) TAB(38)RM(L) TAB(42)RS(L) TAB(48) B*(L) TAB(52)AG(L) TAB(57)AM(L) ;
          TAB(61)AS(L) TAB(66) PY(L) TAB(76) PX(L) TAB(88) Y(1) TAB(98) X(1) TAB(110) ;
          L-(L-1)
11020      RETURN

```

VI.11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE VERTICES= 5

DE	A	DISTANCIA	ANGULO HOR.			RUMBO			CUAD.	AZIMUT			PROYECCIONES ORIGINALES	
			g	m	s	g	m	s		g	m	s	N-S	E-W
1	2	200	92	20	0	15	19	0	SW	195	19	0	-192.8961	-52.8307
2	3	231.33	36	46	0	77	55	0	SE	102	5	0	-48.4253	+226.2047
3	4	117.69	134	34	0	56	39	0	NE	56	39	0	+54.7003	+98.2099
4	5	95.95	164	11	0	40	50	0	NE	40	50	0	+45.0310	+56.1993
5	6	35.43	160	57	0	21	47	0	NE	21	47	0	+32.9001	+13.1480
6	1	350	91	12	0	77	1	0	NW	282	59	0	+78.6337	-341.0524

SS= 241.3214

SN= 241.2652

SW= 393.6931

SE= 393.8618

EY=-5.625153E-02

EX=-2.133179E-02

LA PRECISION ES=1 : 16961.31

Sumatoria CY= 5.625153E-02

Sumatoria CX= 2.133179E-02

DE	A	DISTANCIA	ANGULO HOR.			RUMBO			CUAD.	AZIMUT			PROYECCIONES CORR.		COORDENADAS		VERTICE
			g	m	s	g	m	s		g	m	s	N-S	E-W	Y	X	
	1														400	400	1
1	2	200	92	20	0	15	19	0	SW	195	19	0	-192.8736	-52.82939	207.1264	347.1706	2
2	3	231.33	36	46	0	77	55	0	SE	102	5	0	-48.41964	226.2108	158.7067	573.3815	3
3	4	117.69	134	34	0	56	39	0	NE	56	39	0	64.70786	98.31242	223.4146	671.6739	4
4	5	95.95	164	11	0	40	50	0	NE	40	50	0	65.03863	56.20087	288.4532	727.8947	5
5	6	35.43	160	57	0	21	47	0	NE	21	47	0	32.90391	13.14835	321.3571	741.0431	6
6	1	350	91	12	0	77	1	0	NW	282	59	0	78.64288	-341.0431	400	400	1

CAPITULO VII

PROGRAMA PARA CALCULAR RUMBOS Y DISTANCIAS

VII.1. GENERALIDADES.

Conociendo las coordenadas de dos vértices, de una poligonal topográfica, puede determinarse su distancia y su rumbo.

VII.2. ALGORITMO.

Primero, se proporciona el número de vértices totales, después, el programa pide las coordenadas iniciales "X" e "Y", con las que calcula el rumbo y la distancia.

Luego, pide las coordenadas siguientes y nuevamente calcula su rumbo y su distancia. Y así sucesivamente, hasta que otra vez pide las coordenadas iniciales y hace el cálculo final.

VII.3. LIMITES DEL PROGRAMA.

- a) El número de vértices puede ser hasta infinito.
- b) Las coordenadas, admiten 7 dígitos, más el punto decimal.
- c) La distancia final, puede tener 5 decimales.
- d) La poligonal topográfica, de preferencia, debe ser cerrada.

VII.4. FORMULAS MATEMATICAS.

VII.4.1. FORMULAS GENERALES Y BASICAS.

La siguiente fórmula, es para hallar el rumbo:

$$RBO.1-2 = \frac{X2 - X1}{Y2 - Y1} \arctan$$

La siguiente, es para hallar la distancia:

$$DIST.1-2 = \sqrt{(X2-X1)^2 + (Y2-Y1)^2}$$

Donde:

X1, X2 = Coordenadas en el EJE E-W.

Y2, Y1 = Coordenadas en el EJE N-S.

VII.4.2. FORMULAS USADAS EN EL PROGRAMA.

Cálculo del rumbo:

$$AR = \frac{F}{G} (\text{ARCTAN})$$

$$\frac{F}{G} = \left| \frac{D}{E} \right| \quad ; \quad \begin{array}{l} D = Z - X \\ E = W - Y \end{array}$$

Donde:

Coordenada X1 = X ; Coordenada Y1 = Y

Coordenada X2 = Z ; Coordenada Y2 = W

AR = Rumbo en radianes.

Cálculo de la distancia:

$$B = \sqrt{F^2 + G^2}$$

Donde:

B = Distancia.

F = Coordenadas X.

G = Coordenadas Y.

VII.5. EJEMPLO.

DATOS:

X1=400

Y1=400

X2=347.171

Y2=207.126

$$RBO.1-2 = (-52.829 / -192.874) * (ARCTAN)$$

$$RBO.1-2 = (0.273904206) * (ARCTAN)$$

$$RBO.1-2 = S 15^{\circ} 19' 04'' W$$

$$DIST.1-2 = \sqrt{(-52.829)^2 + (-192.874)^2}$$

$$DIST.1-2 = 199.9782 \text{ m.}$$

DATOS:

X2=347.171

Y2=207.126

X3=573.382

Y3=158.707

$$RBO.2-3 = S 77^{\circ} 55' 06'' E$$

$$DIST.2-3 = 231.3349 \text{ m.}$$

DATOS: X3=573.362 Y3=158.707
 X4=671.694 Y4=223.415

RBO.3-4= N 56° 38' 50" E DIST.3-4= 117.6741 m.

DATOS: X4=671.694 Y4=223.415
 X5=727.895 Y5=288.453

RBO.4-5= N 40° 49' 52" E DIST.4-5= 65.95339 m.

DATOS: X5=727.895 Y5=288.453
 X6=741.043 Y6=321.357

RBO.5-6= N 21° 46' 51" E DIST.5-6= 35.45364 m.

DATOS: X6=741.043 Y6=321.357
 X1=400 Y1=400

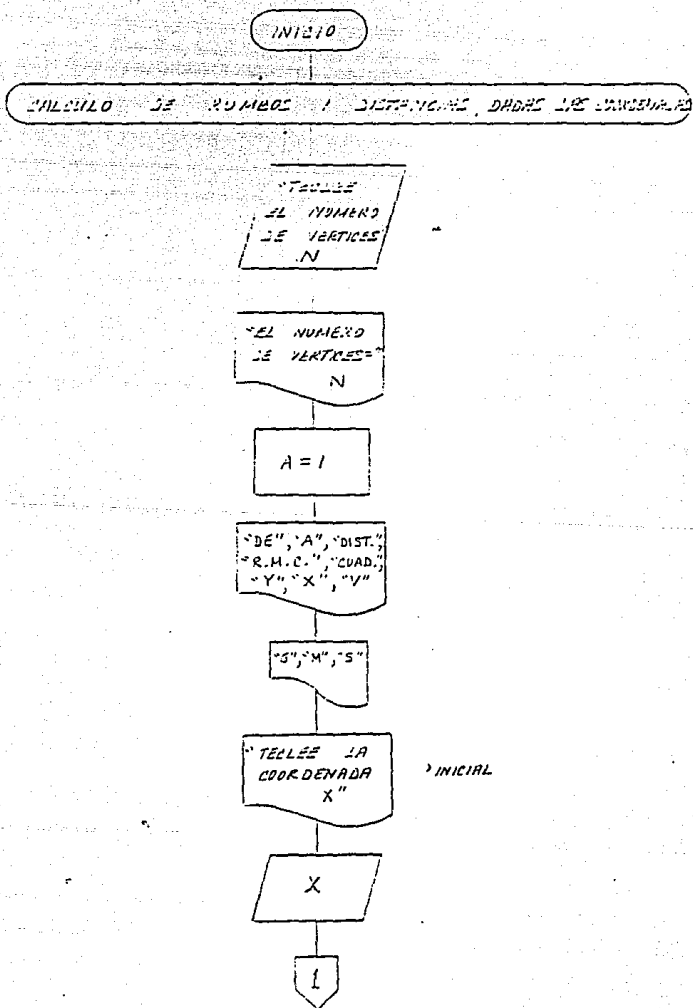
RBO.6-1= N 77° 00' 53" W DIST.6-1= 349.993 m.

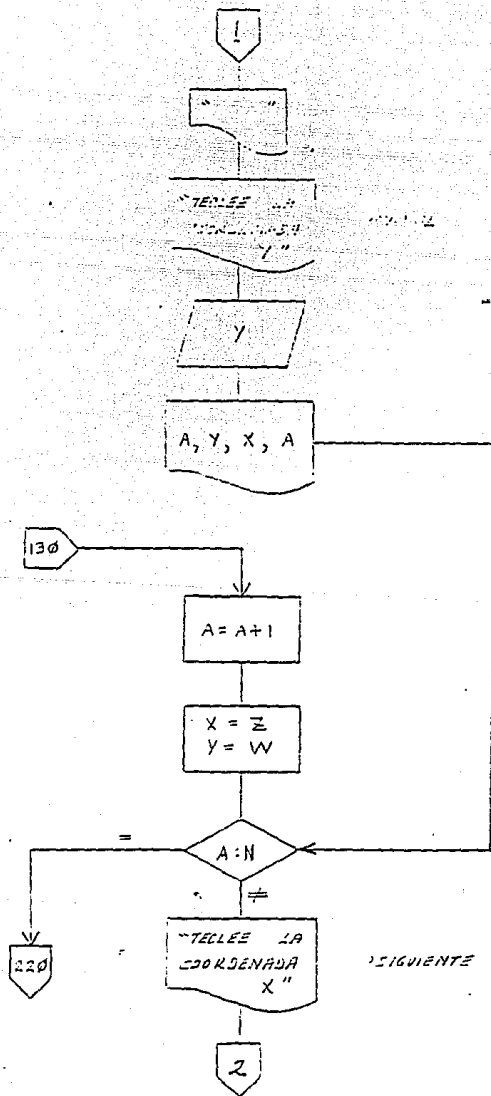
VII.6. VARIABLES UTILIZADAS.

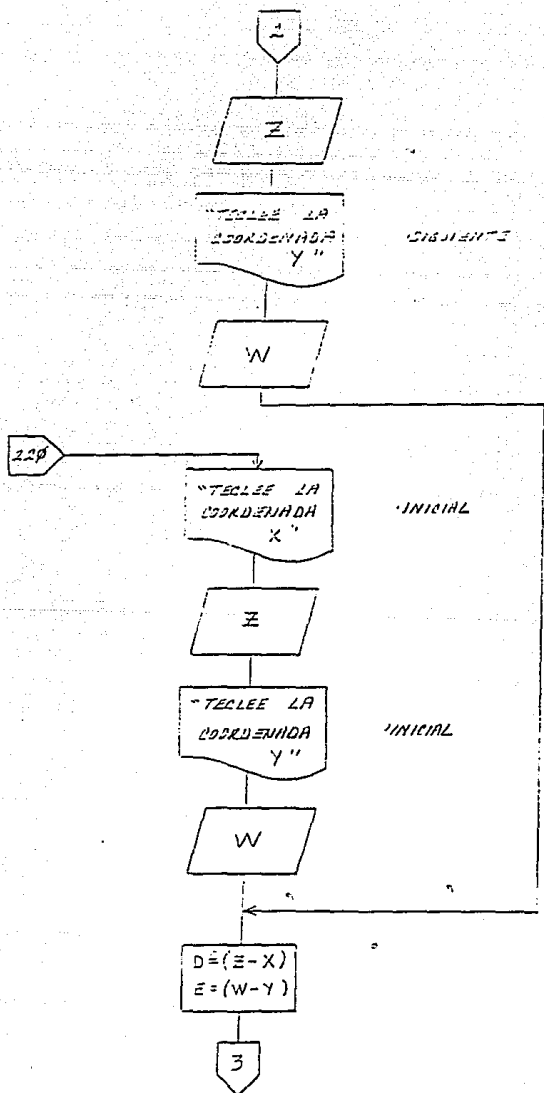
VARIABLE:	DEFINICION:
N	Número de vértices.
A	Contador.
X,Z	Coordenadas E-W.
Y,W	Coordenadas N-S.
D	Diferencia de X.
E	Diferencia de Y.
F	Valor absoluto de D.
G	Valor absoluto de E.
AR	Rumbo en radianes.
AG	Rumbo en grados decimales.
B1,B2,A5	Rumbo en grados-minutos-segundos.
B	Distancia.
B1,B2,A1-A5	Variables para convertir a gra-min-seg.

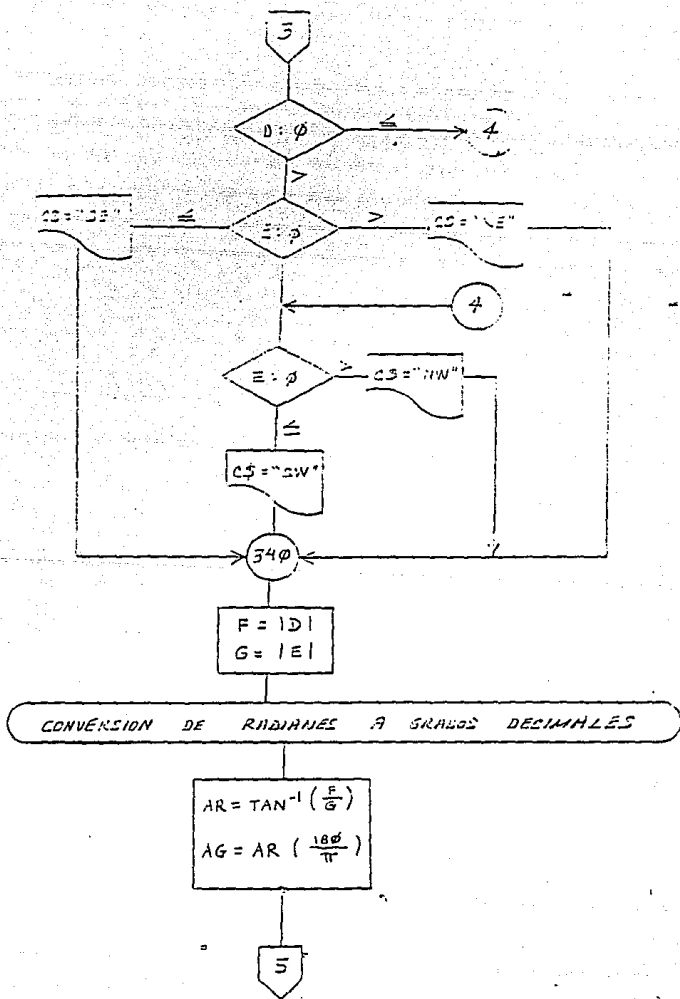
DONDE:

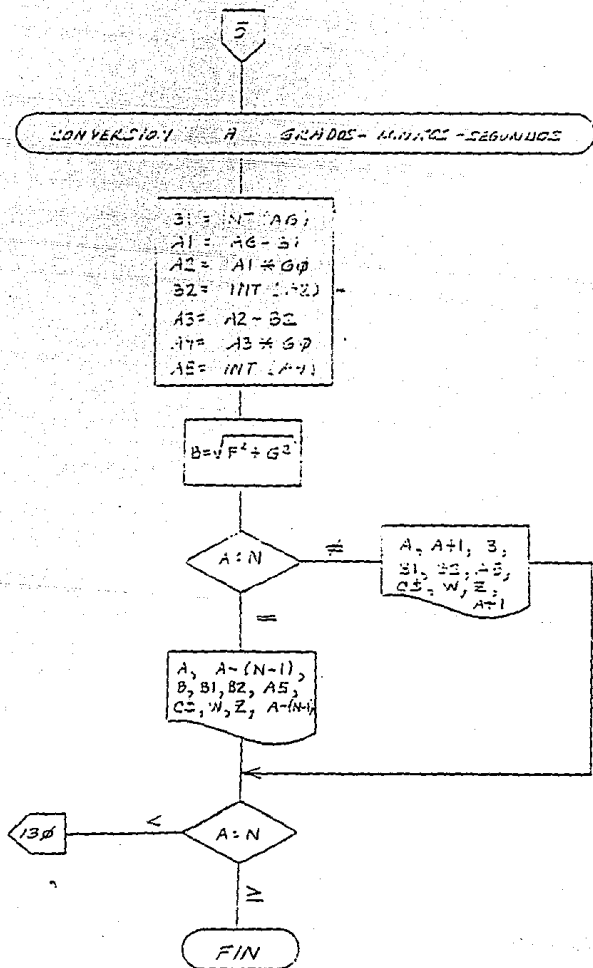
B1,B1,A5	Enteros.
A1,A3	Diferencias.
A2,A4	Multiplicados por 60.












```

10 REM programa para calcular rumbos y distancias dadas las coordenadas
20 INPUT "TECLEE EL NUMERO DE VERTICES:";N
30 PRINT "EL NUMERO DE VERTICES=";N
40 A=1
50 PRINT "DE" TAB(7)"A" TAB(12)"DISTANCIA" TAB(25)"RUMBO N.O." TAB(38)"CUAD.:"
  TAB(50)"Y" TAB(53)"X" TAB(75)"Y"
60 PRINT TAB(25)"5" TAB(29)"M" TAB(32)"S"
70 PRINT "TECLEE LA COORDENADA I:";A
80 INPUT I
90 PRINT "
100 PRINT "TECLEE LA COORDENADA J:";A
110 INPUT J
120 PRINT TAB(6) A TAB(46) Y TAB(59) X TAB(74) A;GOTO 160
170 A=A+1
140 I=1
150 Y=4
160 IF A=N THEN GOTO 220 ELSE GOTO 170
170 PRINT "TECLEE LA COORDENADA I:";A+1
180 INPUT I
190 PRINT "TECLEE LA COORDENADA Y:";A+1
200 INPUT Y
210 GOTO 230
220 PRINT "TECLEE LA COORDENADA I:";A-(N-1)
230 INPUT I
240 PRINT "TECLEE LA COORDENADA Y:";A-(N-1)
250 INPUT Y
260 D=(Z-X)
270 E=(W-Y)
280 IF D>0 THEN GOTO 290 ELSE GOTO 320
290 IF E>0 THEN C$="NE";GOTO 340 ELSE GOTO 300
300 C$="SE"
310 GOTO 340
320 IF E<0 THEN C$="NW";GOTO 340 ELSE GOTO 330
330 C$="SW"
340 F=ABS (D)
350 G=ABS (E)
360 REM CONVERSION DE RADIANTES A GRADOS DECIMALES
370 AR=ATN(F/G) 'EN RADIANTES
380 AG=AR*(57.295779514) 'EN GRADOS DECIMALES
390 REM CONVERSION DE ANGULO DECIMAL A ANGULO HEXADECIMAL GRAB-MIN-SEG
400 B1=INT(AG)
410 A1=A6-B1
420 A2=A1*60
430 B2=INT(A2)
440 A3=A2-B2
450 A4=A3*60
460 A5=INT(A4)
470 B=SQR((F^2)+(G^2))
480 IF A=N THEN GOTO 510 ELSE GOTO 490
490 PRINT A TAB(6) A+1 TAB(12)B TAB(24) B1 TAB(28)B2 TAB(32)A5 TAB(39) C$;
  TAB(46) W TAB(59) Z TAB(74) A+1
500 GOTO 520
510 PRINT A TAB(6) A-(N-1) TAB(12)B TAB(24) B1 TAB(28)B2 TAB(32)A5 TAB(39);
  C$ TAB(46) W TAB(59) Z TAB(74) A-(N-1)
520 IF A=N THEN GOTO 130 ELSE GOTO 530
530 END

```

PROYECTO	TESIS	ARCHIVO	PRD20.Bos	FECHA	1993
PROGRAMA	GW- Basic	ELABORO	MPGM	INDIA	156 DE 159

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
N	6					
A	1	2	3	4	5	6
X	400	347.171	573.382	671.694	727.895	741.043
Y	400	207.126	158.707	223.415	238.753	321.357
	GOTO 160	:	:	:	:	:
A=N	1 ≠ 6	2 ≠ 6	3 ≠ 6	4 ≠ 6	5 ≠ 6	6 = 6
	:	:	:	:	:	GOTO 220
Z	347.171	573.382	671.694	727.895	741.043	400
W	207.126	158.707	223.415	238.753	321.357	400
	GOTO 260	GOTO 260	GOTO 260	GOTO 260	GOTO 260	:
D	-52.829	226.211	98.312	56.201	13.178	-341.043
E	-192.874	-48.419	64.708	65.038	32.704	78.643
D > 0	NO	SI	SI	SI	SI	NO
E > 0	NO	NO	SI	SI	SI	SI
C\$	"SW"	"SE"	"NE"	"NE"	"NE"	"NIV"
F	52.829	226.211	98.312	56.201	13.178	341.043
G	192.874	48.419	64.708	65.038	32.704	78.643
AR	0.2673472	1.359935	0.9836848	0.7126381	0.5801507	1.344163
AG	15.31787	77.91851	56.64747	40.83116	21.73101	77.01734

PROYECTO	TESIS		ARCHIVO	PRD20_Bos	FOLIO	1993
PROGRAMA	GW-Basic	EI ABORO	MPGM	H01A	157	DE 159

VARIABLE	ITERACION Nº 1	ITERACION Nº 2	ITERACION Nº 3	ITERACION Nº 4	ITERACION Nº 5	ITERACION Nº 6
B1	15	77	56	40	21	77
A1	0.3178673	0.9185104	0.6774686	0.8311537	0.7810116	0.01788717
A2	77.07204	55.11063	38.84812	49.80725	46.8057	0.8903500
B2	17	55	38	47	40	0
A3	0.07203675	0.1106262	0.848117	0.8612332	0.8305777	0.8903500
A4	4.322205	6.637573	50.88684	52.15373	51.6717	53.72102
A5	4	6	50	52	51	53
B	199.9782	231.3349	117.6761	85.75637	35.43567	377.743
A = N	1 ≠ 6	2 ≠ 6	3 ≠ 6	4 ≠ 6	5 ≠ 6	6 = 6
	:	:	:	:	:	GOTO 510
A < N	1 < 6	2 < 6	3 < 6	4 < 6	5 < 6	6 ≥ 6
	GOTO 130	GOTO 130	GOTO 130	GOTO 130	GOTO 130	GOTO 530
						END

```

10 REM programa para calcular cuadrados y distancias dadas las coordenadas
20 INPUT "TECLEE EL NUMERO DE VERTICES:";N
30 LPRINT "EL NUMERO DE VERTICES=";N
40 A=1
50 LPRINT "DE" TAB(7)"A" TAB(12)"DISTANCIA" TAB(25)"RUMBO A.C." TAB(35)"CUAD.:"
   TAB(50)"V" TAB(63)"X" TAB(75)"Y"
60 LPRINT TAB(25)"S" TAB(29)"M" TAB(33)"S"
70 PRINT "TECLEE LA COORDENADA X";A
80 INPUT X
90 LPRINT "
100 PRINT "TECLEE LA COORDENADA Y";A
110 INPUT Y
120 LPRINT TAB(6) A TAB(16) Y TAB(59) X TAB(74) A;GOTO 150
130 A=A+1
140 X=Z
150 Y=W
160 IF A=N THEN GOTO 220 ELSE GOTO 170
170 PRINT "TECLEE LA COORDENADA X";A+1
180 INPUT Z
190 PRINT "TECLEE LA COORDENADA Y";A+1
200 INPUT W
210 GOTO 150
220 PRINT "TECLEE LA COORDENADA X";A-(N-1)
230 INPUT Z
240 PRINT "TECLEE LA COORDENADA Y";A-(N-1)
250 INPUT W
260 G=(Z-X)
270 E=(W-Y)
280 IF D>0 THEN GOTO 290 ELSE GOTO 320
290 IF E>0 THEN C1="NE";GOTO 340 ELSE GOTO 300
300 C1="SE"
310 GOTO 340
320 IF E<0 THEN C1="NW";GOTO 340 ELSE GOTO 330
330 C1="SW"
340 F=ABS (D)
350 G=ABS (E)
360 REM CONVERSION DE RADIANTES A GRADOS DECIMALES
370 AR=ATN(F/G) 'EN RADIANTES
380 AG=AR*(57.295779518) 'EN GRADOS DECIMALES
390 REM CONVERSION DE ANGULO DECIMAL A ANGULO HEXADECIMAL GRAD-MIN-SEG
400 B1=INT(AG)
410 A1=AG-B1
420 A2=A1*60
430 B2=INT(A2)
440 A3=A2-B2
450 A4=A3*60
460 A5=INT(A4)
470 B=SQR((F^2)+(G^2))
480 IF A=N THEN GOTO 510 ELSE GOTO 490
490 LPRINT A TAB(6) A+1 TAB(12) B TAB(24) B1 TAB(28) B2 TAB(32) A5 TAB(39) C1;
   TAB(46) W TAB(59) Z TAB(74) A+1
500 GOTO 520
510 LPRINT A TAB(6) A-(N-1) TAB(12) B TAB(24) B1 TAB(28) B2 TAB(32) A5 TAB(39);
   C1 TAB(46) W TAB(59) Z TAB(74) A-(N-1)
520 IF A<N THEN GOTO 130 ELSE GOTO 530
530 END

```

VII. 11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE VERTICES= 5

DE	A	DISTANCIA	RUMBO N.C.		CUAD.			
			3	4	5			
	1					400	400	1
1	2	199.9782	15	19	4	SW	207.125	347.171
2	3	231.3349	77	55	3	SE	159.707	573.332
3	4	117.5961	55	38	50	NE	227.415	671.694
4	5	35.95639	40	49	52	NE	238.453	727.395
5	5	35.43364	21	46	51	NE	721.357	741.343
5	1	319.993	77	0	53	NW	400	400

CAPITULO VIII
PROGRAMA PARA CALCULAR EL AREA

VIII.1. GENERALIDADES.

Mediante productos cruzados y de acuerdo a una tabulación ordenada de las coordenadas de los vértices, se puede obtener el área de una poligonal topográfica.

VIII.2. ALGORITMO.

Se proporciona el número de vértices y éste lo compara con el intervalo : entre 4 y 100 ,si está dentro de esos números, continúa el programa, de lo contrario, muestra un mensaje de error.

Luego, pide las coordenadas " X " e " Y ", calcula los productos cruzados e imprime los resultados.

Y así, sucesivamente pide las coordenadas hasta terminar.

Finalmente, calcula e imprime las sumatorias " X " e " Y " y el Area Total.

VIII.3. LIMITES DEL PROGRAMA.

a) Debe usarse para poligonales mayores a 3 y menores a 100 vértices.

b) Se aceptan coordenadas iniciales, hasta 1.000 con 3 decimales. Es decir, con 3 dígitos enteros y 3 decimales. O bien, con 3 dígitos enteros, sin decimales.

Ejemplos: 1000.123 , 10 000 000

Es decir, el campo asignado será de 3 espacios (contando el punto decimal); si rebasa esos espacios, truncará los valores numéricos de las coordenadas.

c) El grado de precisión de las constantes numéricas del programa (referentes a la DISTANCIA y a las SUMATORIAS de los productos en "X" e "Y"), tendrán el doble de precisión, de acuerdo a los datos proporcionados.

Por ejemplo: Si el dato es de 3 decimales: 347.173 , tendrá 6 dígitos de precisión.

VIII.4. FORMULAS MATEMATICAS.

VIII.4.1. FORMULAS GENERALES Y BASICAS.

La fórmula general, se obtiene formando trapecios, con cada lado. Cuyas bases, pueden ser las " X " o las " Y " de los vértices y sus alturas las diferencias respectivas.

Sean los vértices, con sus respectivas coordenadas:

A (X1 , Y1)

B (X2 , Y2)

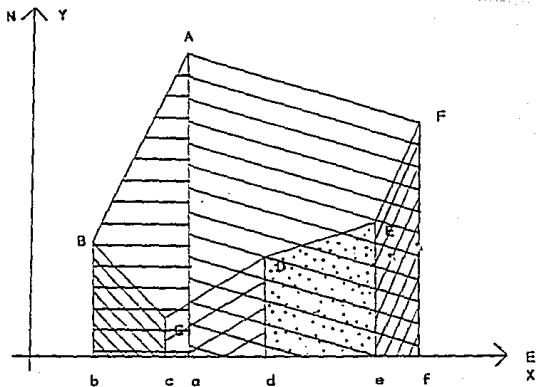
C (X3 , Y3)

D (X4 , Y4)

E (X5 , Y5)

F (X6 , Y6)

Sea la siguiente figura:



SUPERFICIE DEL POLIGONO:

$$\text{SUP} = \text{TRAPECIO } BbAa + \text{TRAPECIO } AaFf - \text{TRAPECIO } BbCc - \\ - \text{TRAPECIO } CcDd - \text{TRAPECIO } DdEe - \text{TRAPECIO } EeFf$$

$$\text{SUP} = \frac{Y2+Y1}{2} (X2-X1) + \frac{Y1+Y6}{2} (X1-X6) - \frac{Y3+Y2}{2} (X2-X3) - \\ - \frac{Y3+Y4}{2} (X3-X4) - \frac{Y4+Y5}{2} (X4-X5) - \frac{Y5+Y6}{2} (X5-X6)$$

$$\text{CSUP} = (Y2+Y1)(X2-X1) + (Y1+Y6)(X1-X6) - (Y2+Y3)(X2-X3) - \\ - (Y3+Y4)(X3-X4) - (Y4+Y5)(X4-X5) - (Y5+Y6)(X5-X6)$$

$$\text{CSUP} = Y2X2 - Y2X1 + Y1X2 - Y1X1 + Y1X1 - Y1X6 + Y6X1 - Y6X6 - \\ - Y3X2 + Y3X3 - Y3X3 + Y3X3 - Y3X3 + Y3X4 - Y4X3 + Y4X4 - \\ - Y4X4 + Y4X5 - Y5X4 + Y5X5 - Y5X5 + Y5X6 - Y6X5 + Y6X6$$

ORDENANDO Y SACANDO LAS (X) COMO FACTOR:

$$\text{CSUP} = X1(Y6-Y2) + X2(Y1-Y3) - X3(Y2-Y4) + X4(Y3-Y5) + \\ + X5(Y4-Y6) + X6(Y5-Y1)$$

La siguiente tabulación, es igual a la fórmula anterior:

PUNTOS	X	Y	PRODUCTOS ↘	PRODUCTOS ↗
A	X1	Y1		X1Y1
B	X2	Y2	X1Y2	X2Y2
C	X3	Y3	X2Y3	X3Y3
D	X4	Y4	X3Y4	X4Y4
E	X5	Y5	X4Y5	X5Y5
F	X6	Y6	X5Y6	X6Y6

Σ PROD. ↘

Σ PROD. ↗

$$(\Sigma \text{ PROD. } \swarrow) - (\Sigma \text{ PROD. } \nearrow)$$

SUP. POLIGONO =

VIII.4.2. FORMULAS USADAS EN EL PROGRAMA.

Primero se introducen los datos, en este caso, las coordenadas; luego se multiplican por separado (productos cruzados), como a continuación se indica:

ITERACION FINAL (EXCLUSIVAMENTE) :

$$P(I) = X(I) * Y(I-(M-1))$$

$$Q(I) = Y(I) * X(I-(M-1))$$

DEMÁS ITERACIONES :

$$P(I) = X(I) * Y(I-1)$$

$$Q(I) = Y(I) * X(I+1)$$

DONDE:

$P(I)$ - Productos en X

$Q(I)$ - Productos en Y

$X(I)$, $X(I+1)$, $X(I-(M-1))$ - Coordenadas en X

$Y(I)$, $Y(I+1)$, $Y(I-(M-1))$ - Coordenadas en Y

$SUMP = SUMP + P(I)$ - Sumatoria de los productos en X

$SUMQ = SUMQ + Q(I)$ - Sumatoria de los productos en Y

CALCULO DE LA SUPERFICIE TOTAL:

$$ST = \frac{SUMP - SUMQ}{2}$$

VIII.C. EJEMPLO.

Como datos, se proporcionan las coordenadas "X" e "Y" y se aplica la fórmula de manera tabular, como se indicó en el inciso anterior (VIII.4.1.).

EST.	X	Y	PROD.X	PROD.Y
1	400	400	82850.41	138868.4
2	347.171	207.126	55098.47	118762.3
3	573.382	158.707	128102.1	106602.5
4	671.694	223.415	193752.1	162622.7
5	727.895	288.453	233914.2	213756.1
6	741.043	321.357	296417.2	128542.8
SUMATORIA EN X=			990134.52734375	
SUMATORIA EN Y=				869154.796875

LA SUPERFICIE TOTAL ES:

$$\text{SUPERFICIE} = \frac{\text{SUMATORIA X} - \text{SUMATORIA Y}}{2}$$

$$\text{SUPERFICIE} = \frac{990134.52734375 - 869154.796875}{2}$$

$$\text{SUPERFICIE} = 60,489,865234375 \text{ m}^2$$

VIII.3. VARIABLES UTILIZADAS.

VARIABLES	DEFINICION
M	Número de vértices
I	Contador
SUMP	Sumatoria de los productos en X
SUMQ	Sumatoria de los productos en Y
X(I-(M-1)) , X(I)	Coordenadas X iniciales
Y(I-(M-1)) , Y(I)	Coordenadas Y iniciales
X(I+1)	Coordenadas X siguientes
Y(I+1)	Coordenadas Y siguientes
P(I)	Productos en X
Q(I)	Productos en Y
ST	Superficie total

CALCULO DEL AREA DE UN POLIGONO DERRALDO, EN FUNCION DE SUS VERTICES.

INICIO

DEFINICION
DE CTE. CAL
DISE. RECTANG.
S

SIM X(100)
SIM Y(100)
SIM P(100)
SIM Q(100)

¿TERMINA?
Nº VERTICES

M

4: M:100

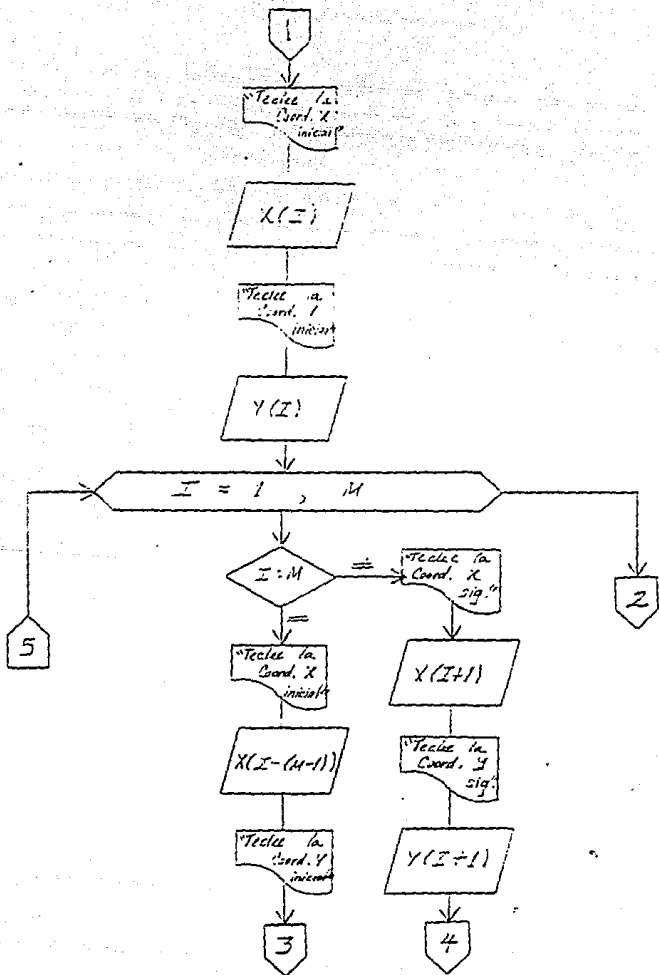
"ERROR:
INTERVALO
ENTRE 4
Y 100"

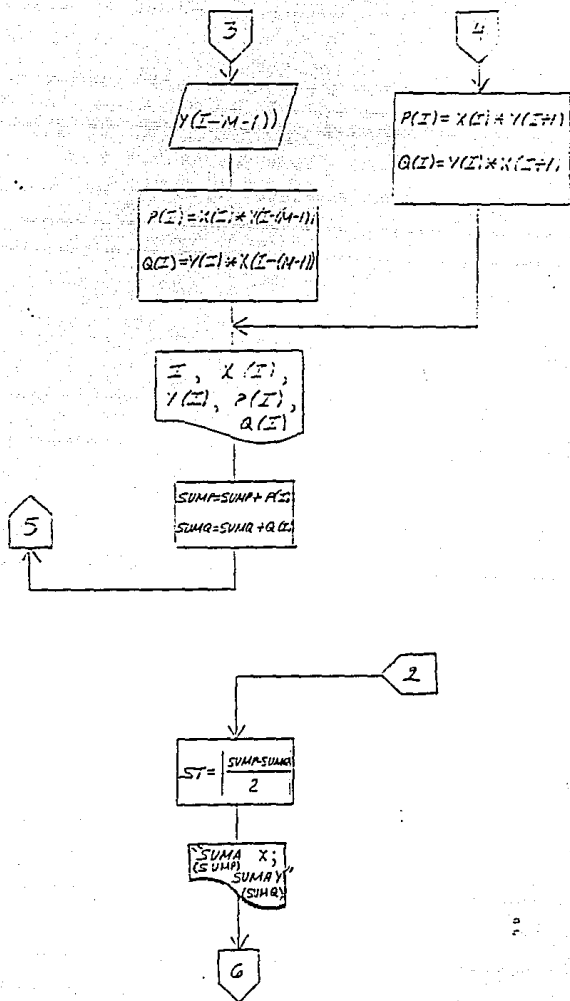
~ Nº VERTICES
M

"EST" ; "X"
"Y"
"PROD.X" ; "PROD.Y"

SUMP = 0
SUMQ = 0

1








```

10 REM PROGRAMA QUE CALCULA EL AREA DE UN POLIGONO CERRADO
20 REM EN FUNCION DE SUS COORDENADAS
30 DEFDBL S 'CONSTANTE CON DOBLE PRECISION
40 DIM X(100) 'MEMORIA PARA 100 VERTICES
50 DIM Y(100)
60 DIM P(100)
70 DIM Q(100)
80 PRINT "TECLEE EL NUMERO DE VERTICES"
90 INPUT M
100 IF M<4 OR M>100 THEN PRINT "ERROR:INTERVALO ENTRE 4 Y 100":GOTO 90
110 PRINT "EL NUMERO DE VERTICES ES:":M
120 PRINT " "
130 PRINT "EST. "TAB(9)"X"TAB(22)"Y"TAB(31)"PROD.X"TAB(51)"PROD.Y"
140 PRINT " "
150 SUMP=0:SUMQ=0:I=1
160 PRINT "TECLEE LA COORDENADA X":I
170 INPUT X(I)
180 PRINT "TECLEE LA COORDENADA Y":I
190 INPUT Y(I)
200 FOR I=1 TO M
210 IF I=M THEN GOTO 290 ELSE GOTO 220
220 PRINT "TECLEE LA COORDENADA X":I+1
230 INPUT X(I+1)
240 PRINT "TECLEE LA COORDENADA Y":I+1
250 INPUT Y(I+1)
260 P(I)=X(I)*Y(I+1)
270 Q(I)=Y(I)*X(I+1)
280 GOTO 350
290 PRINT "TECLEE LA COORDENADA X":I-(M-1)
300 INPUT X(I-(M-1))
310 PRINT "TECLEE LA COORDENADA Y":I-(M-1)
320 INPUT Y(I-(M-1))
330 P(I)=X(I)*Y(I-(M-1))
340 Q(I)=Y(I)*X(I-(M-1))
350 PRINT I TAB(5) X(I) TAB(18) Y(I) TAB(30) P(I) TAB(50) Q(I)
360 SUMP=SUMP+P(I) 'SUMATORIA DE LOS PRODUCTOS EN X
370 SUMQ=SUMQ+Q(I) 'SUMATORIA DE LOS PRODUCTOS EN Y
380 NEXT I
390 ST=ABS((SUMP-SUMQ)/2) 'SUPERFICIE TOTAL
400 PRINT " "
410 PRINT " "
420 PRINT "SUMATORIA EN X=" TAB(30);SUMP
430 PRINT "SUMATORIA EN Y=" TAB(50);SUMQ
440 PRINT " "
450 PRINT " "
460 PRINT "LA SUPERFICIE TOTAL ES:"
470 PRINT ST;"METROS CUADRADOS"
480 END

```


VIII.10. IMPRESION DEL PROGRAMA.

```

10 REM PROGRAMA QUE CALCULA EL AREA DE UN POLIGONO CERRADO
20 REM EN FUNCION DE SUS COORDENADAS
30 DEFDD S 'CONSTANTE CON DOBLE PRECISION
40 DIM X(100) 'MEMORIA PARA 100 VERTICES
50 DIM Y(100)
60 DIM P(100)
70 DIM Q(100)
80 PRINT "TECLEE EL NUMERO DE VERTICES"
90 INPUT M
100 IF M<4 THEN LPRINT "ERROR:INTERVALO ENTRE 4 Y 100":GOTO 30
110 LPRINT "EL NUMERO DE VERTICES ES: ";M
120 LPRINT " "
130 LPRINT "EST. "TAB(2)"X"TAB(22)"Y"TAB(31)"PROD.X"TAB(51)"PROD.Y"
140 LPRINT " "
150 SUMP=0:SUMQ=0:I=1
160 PRINT "TECLEE LA COORDENADA X";I
170 INPUT X(I)
180 PRINT "TECLEE LA COORDENADA Y";I
190 INPUT Y(I)
200 FOR I=1 TO M
210 IF I=M THEN GOTO 290 ELSE GOTO 220
220 PRINT "TECLEE LA COORDENADA X";I+1
230 INPUT X(I+1)
240 PRINT "TECLEE LA COORDENADA Y";I+1
250 INPUT Y(I+1)
260 P(I)=X(I)*Y(I+1)
270 Q(I)=Y(I)*X(I+1)
280 GOTO 350
290 PRINT "TECLEE LA COORDENADA X";I-(M-1)
300 INPUT X(I-(M-1))
310 PRINT "TECLEE LA COORDENADA Y";I-(M-1)
320 INPUT Y(I-(M-1))
330 P(I)=X(I)*Y(I-(M-1))
340 Q(I)=Y(I)*X(I-(M-1))
350 LPRINT I TAB(5) X(I) TAB(18) Y(I) TAB(30) P(I) TAB(50) Q(I)
360 SUMP=SUMP+P(I) 'SUMATORIA DE LOS PRODUCTOS EN X
370 SUMQ=SUMQ+Q(I) 'SUMATORIA DE LOS PRODUCTOS EN Y
380 NEXT I
390 ST=ABS((SUMP-SUMQ)/2) 'SUPERFICIE TOTAL
400 LPRINT " "
410 LPRINT " "
420 LPRINT "SUMATORIA EN X=" TAB(30);SUMP
430 LPRINT "SUMATORIA EN Y=" TAB(50);SUMQ
440 LPRINT " "
450 LPRINT " "
460 LPRINT "LA SUPERFICIE TOTAL ES:"
470 LPRINT ST:"METROS CUADRADOS"
480 END

```

VIII.11. IMPRESION DE DATOS Y RESULTADOS.

EL NUMERO DE VERTICES ES: 6

EST.	X	Y	PROD.X	PROD.Y
1	400	400	32850.41	133868.4
2	747.171	297.126	55098.47	118762.3
3	573.382	158.707	128102.1	106602.5
4	671.674	223.413	193752.1	162622.7
5	727.395	298.453	332914.2	213756.1
6	741.343	321.357	296417.2	128542.3

SUMATORIA EN X= 990134.52734375

SUMATORIA EN Y=

369154.796875

LA SUPERFICIE TOTAL ES:

60489.865234375 METROS CUADRADOS

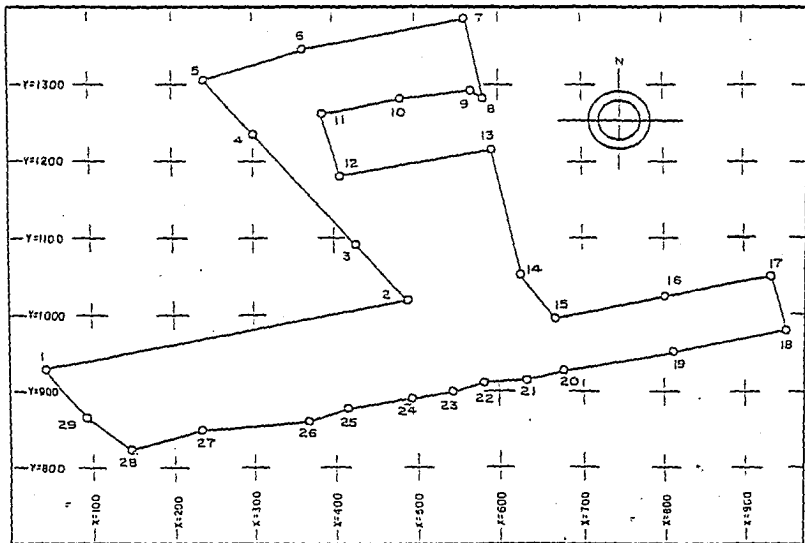
CAPITULO IX

RESULTADOS

De acuerdo al objetivo inicial, se muestra un ejemplo de una poligonal topográfica cerrada de 29 vértices.

Figura (18)

Levantamiento hecho en la Colonia "El Molino". Delegación Iztapalapa. México, D.F. 1993. Escala = 1 : 5000



A continuación, se muestran sólo los resultados de la poligonal mencionada.

III.1. PROGRAMA PARA COMPROBAR EL CIERRE ANGULAR :

EL NUMERO DE ANGULOS MEDIDOS ES= 29

LA TOLERANCIA ANGULAR ES= +/-
 0 GRADOS 5 MINUTOS 00 SEGUNDOS

EL ANGULO 1	= 65.48333	65	28	59.99
EL ANGULO 2	= 299.5367	299	34	0.16
EL ANGULO 3	= 180.5	180	30	0.00
EL ANGULO 4	= 180.9667	180	58	0.14
EL ANGULO 5	= 65.08333	65	4	59.98
EL ANGULO 6	= 174.5	174	18	0.01
EL ANGULO 7	= 89.31667	89	37	0.01
EL ANGULO 8	= 40.81667	40	49	0.01
EL ANGULO 9	= 225.78333	225	46	59.86
EL ANGULO 10	= 183.98333	183	58	59.90
EL ANGULO 11	= 272.05	272	2	59.96
EL ANGULO 12	= 266.4667	266	28	0.14
EL ANGULO 13	= 90.9	90	54	0.01
EL ANGULO 14	= 203.98333	203	58	59.90
EL ANGULO 15	= 247.3167	247	19	0.11
EL ANGULO 16	= 179.53333	179	31	59.86
EL ANGULO 17	= 89.23333	89	13	59.99
EL ANGULO 18	= 90.81668	90	49	0.05
EL ANGULO 19	= 178.5	178	36	0.02
EL ANGULO 20	= 184.48333	184	28	59.90
EL ANGULO 21	= 168.2	168	11	59.99
EL ANGULO 22	= 192.85	192	31	0.02
EL ANGULO 23	= 172.4667	172	28	0.14
EL ANGULO 24	= 182.18333	182	10	59.89
EL ANGULO 25	= 189.6667	189	40	0.13
EL ANGULO 26	= 163.8	163	48	0.01
EL ANGULO 27	= 192.5	192	30	0.00
EL ANGULO 28	= 126.58333	126	34	59.87
EL ANGULO 29	= 162.2667	162	16	0.10

LA SUMA ANGULAR ES=
 4860 GRADOS 0 MINUTOS 0 SEGUNDOS

 Angulos compensados
 LISTOS para el CALCULO de RUMBOS

IX.2. PROGRAMA PARA CALCULAR RUMBOS :

EL NUMERO DE VERTICES= 29

DE	A	CUAD.	RBO. MAGNET. CALC.		
			G	M	E
1	2	NE	78	46	0.01
2	3	NW	40	48	0.12
3	4	NW	41	18	0.12
4	5	SW	42	18	0.23
5	6	NE	72	28	59.81
6	7	NE	78	20	59.80
7	8	SE	11	16	0.31
8	9	NW	52	5	0.48
9	10	SW	82	7	59.64
10	11	SW	78	3	59.76
11	12	SE	12	34	0.25
12	13	NE	79	37	59.64
13	14	SE	11	16	0.31
14	15	SE	33	15	0.22
15	16	NE	77	25	59.65
16	17	NE	77	25	59.76
17	18	SE	11	20	0.15
18	19	SW	77	51	0.13
19	20	SW	79	15	0.11
20	21	SW	74	46	0.21
21	22	SW	96	34	0.16
22	23	SW	73	43	0.14
23	24	SW	91	15	0.00
24	25	SW	79	4	0.16
25	26	SW	59	24	0.03
26	27	SW	35	26	0.02
27	28	SW	73	8	0.02
28	29	NW	53	28	59.79
29	1	NW	35	44	59.89
1	2	NE	78	46	0.21

FIN DEL CALCULO

IX.5. PROGRAMA PARA CALCULAR AZIMUTES :

EL NUMERO DE VERTICES= 29

DE	A	ANGULO			R.H.C.			CUAD.	AZIMUT			AZIMUT decimales
		g	m	s	g	m	s		g	m	s	
1	2	35	29	0	78	46	0	NE	78	46	0	78.78667
2	3	299	54	0	40	48	0	NW	319	12	0	319.20000
3	4	180	30	0	41	18	0	NW	319	42	0	319.70000
4	5	180	38	0	42	16	0	NW	317	45	60	317.75330
5	6	95	3	0	72	39	0	NE	72	39	0	72.65000
6	7	174	18	0	78	21	0	NE	78	20	60	78.35300
7	8	89	37	0	11	16	0	SE	168	43	60	168.73330
8	9	40	49	0	52	3	0	NW	307	55	0	307.91670
9	10	225	47	0	32	3	0	SW	262	7	60	262.13330
10	11	183	59	0	78	9	0	SW	258	9	60	258.14990
11	12	272	7	0	17	54	0	SE	166	3	60	166.09990
12	13	266	28	0	79	38	0	NE	79	37	60	79.63323
13	14	90	54	0	11	16	0	SE	168	43	60	168.73330
14	15	203	59	0	35	13	0	SE	144	44	60	144.74990
15	16	247	19	0	77	26	0	NE	77	25	60	77.43325
16	17	179	32	0	77	54	0	NE	77	53	60	77.89990
17	18	39	14	0	11	20	0	SE	168	39	60	168.66660
18	19	70	49	0	77	31	0	SW	257	51	0	257.85000
19	20	178	36	0	79	15	0	SW	259	15	0	259.25000
20	21	184	29	0	74	46	0	SW	254	46	0	254.76670
21	22	168	12	0	36	34	0	SW	266	34	0	266.56670
22	23	192	51	0	75	43	0	SW	253	43	0	253.71670
23	24	172	28	0	81	15	0	SW	261	15	0	261.25000
24	25	182	11	0	79	4	0	SW	259	4	0	259.06670
25	26	189	40	0	69	24	0	SW	249	23	60	249.40000
26	27	163	48	0	35	36	0	SW	265	36	0	265.60000
27	28	192	30	0	73	6	0	SW	253	6	0	253.10000
28	29	126	35	0	53	29	0	NW	306	31	0	306.51670
29	1	162	16	0	35	43	0	NW	324	15	0	324.25000

IX. 4. PROGRAMA PARA COMPENSAR LA POLIGONAL :

EL NUMERO DE VERTICES= 29.

DE	A	DISTANCIA	ANGULO HOR.			RUMBO			CUAD.	MINUT			PROYECCIONES ORIGINALES	
			q	r	s	q	r	s		q	r	s	N-S	E-W
1	2	453.05	55	29	0	78	46	0	NE	73	46	0	+88.2544	+444.3705
2	3	77.227	299	34	0	40	48	0	NW	719	12	0	+73.8004	-63.5301
3	4	191.574	190	30	0	41	19	0	NW	313	42	0	-143.7227	-125.4371
4	5	72.665	190	58	0	42	16	0	NW	717	44	0	+68.5741	-62.3249
5	6	131.255	65	5	0	72	37	0	NE	72	37	0	+39.1413	+123.2330
6	7	201.248	174	19	0	73	21	0	NE	78	21	0	+40.5386	+197.1022
7	8	103.075	39	27	0	11	16	0	SE	168	44	0	-101.0886	+20.1384
8	9	16.189	40	49	0	52	5	0	NW	707	55	0	+9.9484	-12.7716
9	10	86.906	225	47	0	82	9	0	SW	262	9	0	-11.3947	-36.0891
10	11	95.797	193	59	0	78	9	0	SW	259	9	0	-19.6721	-33.7554
11	12	91.761	272	3	0	13	54	0	SE	136	5	0	-79.5667	+19.5414
12	13	189.392	266	23	0	79	73	0	NE	79	23	0	-34.0808	+185.5004
13	14	169.354	90	54	0	11	16	0	SE	168	44	0	-166.0903	-53.0877
14	15	68.531	203	59	0	35	15	0	SE	144	45	0	-95.9652	+39.2524
15	16	136.773	247	19	0	77	26	0	NE	77	26	0	+29.3022	+133.6916
16	17	139.642	179	22	0	77	54	0	NE	77	54	0	+29.2718	+136.3396
17	18	73.767	39	14	0	11	20	0	SE	138	40	0	-72.3286	+14.4965
18	19	143.392	90	49	0	77	51	0	SW	257	51	0	-30.1800	-140.1800
19	20	136.282	178	26	0	79	15	0	SW	259	15	0	-25.4199	-153.2903
20	21	45.539	184	29	0	74	46	0	SW	254	46	0	-11.9654	-43.9389
21	22	53.916	168	12	0	86	34	0	SW	266	34	0	-3.2288	-53.8192
22	23	39.275	192	51	0	73	43	0	SW	253	43	0	-11.0122	-37.6996
23	24	50.266	172	23	0	81	15	0	SW	261	15	0	-7.6619	-49.7798
24	25	77.599	182	11	0	79	4	0	SW	259	4	0	-14.7179	-76.1905
25	26	52.249	199	40	0	69	24	0	SW	249	24	0	-18.3834	-49.9082
26	27	130.956	165	48	0	85	36	0	SW	265	36	0	-10.0468	-130.5700
27	28	89.433	192	20	0	73	5	0	SW	253	5	0	-25.9984	-35.5707
28	29	72.443	126	35	0	53	29	0	NW	306	31	0	+43.1077	-58.2213
29	1	79.587	162	16	0	25	45	0	NW	324	15	0	+64.5907	-46.4987

SS= 665.0207

SN= 664.9351

SM= 1350.177

SE= 1350.204

EY= -.0856781

EI= 2.710343E-02

LA PRECISION ES=1 : 36716

Ecuatoria CY= 3.567809E-02

Ecuatoria CX= 2.710342E-02

DE	A	DISTANCIA	ANGULO HOR.			RUMBO	CUAD.	AZIMUT			PROYECCIONES CORR.		COORDENADAS		VERTICE		
			g	l	s			g	l	s	N-S	E-W	Y	X			
1	2	453.05	65	29	0	78	46	0	NE	78	46	0	38.26226	44.3659	930.74	45.31	1
2	7	37.227	279	34	0	40	48	0	NW	719	12	0	77.80511	-63.53075	1019.002	490.1759	2
3	4	191.574	180	30	0	41	18	0	NW	719	42	0	145.312	-126.4404	1272.607	426.6452	3
4	5	92.565	180	38	0	42	16	0	NW	717	44	0	88.57856	-62.3255	1276.639	300.0048	4
5	6	131.255	65	5	0	72	39	0	NE	72	39	0	79.1438	125.2318	1305.118	237.8793	5
6	7	201.248	174	18	0	73	21	0	NE	73	21	0	-0.64119	197.1002	1344.262	363.161	6
7	3	103.075	59	37	0	11	16	0	SE	168	44	0	-101.3821	20.13819	1384.303	560.2513	7
8	2	16.189	40	49	0	32	5	0	NW	207	55	0	9.949021	-12.77171	1293.621	580.3995	8
9	10	36.306	225	47	0	32	3	0	SW	262	3	0	-11.39397	-36.089	1293.77	567.6277	9
10	11	95.797	193	59	0	78	3	0	SW	258	9	0	-19.67083	-93.75633	1291.376	481.5337	10
11	12	81.761	272	3	0	13	54	0	SE	166	6	0	-79.36161	19.64122	1262.205	787.7824	11
12	13	189.372	266	23	0	79	38	0	NE	79	38	0	74.08304	186.2985	1182.344	407.4236	12
13	14	169.334	30	54	0	11	16	0	SE	168	44	0	-166.0796	33.08739	1216.727	593.7221	13
14	15	68.651	203	59	0	25	15	0	SE	144	45	0	-35.36159	27.55205	1050.847	526.3095	14
15	16	136.973	247	19	0	77	26	0	NE	77	26	0	29.80408	133.6902	994.8854	566.3616	15
16	17	139.642	179	32	0	77	54	0	NE	77	54	0	29.27368	136.5332	1024.689	600.0518	16
17	18	73.767	89	14	0	11	20	0	SE	168	40	0	-72.3239	14.49639	1053.963	936.59	17
18	19	145.272	30	49	0	77	51	0	SW	257	51	0	-30.17803	-140.1814	981.6392	951.0864	18
19	20	136.282	178	36	0	79	15	0	SW	259	15	0	-25.41825	-133.8917	951.4612	310.705	19
20	21	45.539	184	29	0	74	46	0	SW	254	46	0	-11.96461	-43.93938	926.0429	677.0134	20
21	22	53.916	168	12	0	36	34	0	SW	266	34	0	-3.228625	-53.81978	914.0783	633.074	21
22	23	39.275	192	51	0	73	43	0	SW	253	43	0	-11.01149	-37.69994	910.8497	579.2542	22
23	24	50.366	172	28	0	31	15	0	SW	261	15	0	-7.66137	-49.78032	899.9391	541.5543	23
24	25	77.599	182	11	0	79	4	0	SW	259	4	0	-14.71695	-76.19124	892.1768	491.774	24
25	26	52.249	189	40	0	69	24	0	SW	249	24	0	-18.3822	-48.90867	877.4598	515.5827	25
26	27	130.956	163	48	0	35	36	0	SW	265	36	0	-10.04619	-130.5714	859.0776	366.6741	26
27	28	89.433	192	30	0	73	6	0	SW	253	6	0	-25.39668	-95.57158	849.0314	236.1027	27
28	29	72.443	126	35	0	53	29	0	NW	306	31	0	43.11048	-58.22185	823.0347	150.5311	28
29	1	79.587	162	16	0	35	45	0	NW	324	15	0	84.5949	-46.49914	866.1451	92.30926	29
															930.7401	45.81013	1

IX.3. PROGRAMA PARA CALCULAR RUMBOS Y DISTANCIAS :

EL NUMERO DE VERTICES= 29

DE	A	DISTANCIA	RUMBO N. O.			CUAD.	Y	X	N
			G	M	S				
	1					730.74	45.31		1
1	2	453.0467	78	45	57	NE	1019.002	490.176	2
2	3	97.33112	40	47	54	NW	1092.307	426.545	3
3	4	191.5815	41	17	54	NW	1236.537	300.205	4
4	5	92.36942	42	15	54	NW	1305.113	237.379	5
5	6	171.2548	72	38	35	NE	1344.252	733.131	6
6	7	201.2463	73	20	37	NE	1384.703	560.231	7
7	8	103.3686	11	16	3	SE	1283.821	530.4	8
8	9	16.18972	52	4	57	NW	1293.77	567.628	9
9	10	86.70674	82	3	1	SW	1281.376	481.537	10
10	11	95.79737	78	9	2	SW	1262.208	387.785	11
11	12	91.75532	13	54	2	SE	1182.944	407.424	12
12	13	139.57	79	37	56	NE	1216.927	593.722	13
13	14	169.5439	11	13	2	SE	1050.847	626.31	14
14	15	38.52818	33	13	4	SE	794.385	666.362	15
15	16	136.7719	77	25	58	NE	1024.389	800.052	16
16	17	177.641	77	33	55	NE	1053.763	736.37	17
17	18	73.75245	11	20	1	SE	931.637	751.086	18
18	19	143.3925	77	51	3	SW	951.461	810.905	19
19	20	136.2833	79	15	3	SW	926.043	677.013	20
20	21	45.537	74	46	1	SW	714.078	633.074	21
21	22	53.91667	36	34	3	SW	910.85	579.254	22
22	23	37.27537	73	43	1	SW	979.838	541.554	23
23	24	50.36609	91	15	3	SW	892.177	491.774	24
24	25	77.37932	79	4	2	SW	877.46	415.583	25
25	26	52.24929	69	24	6	SW	857.078	366.674	26
26	27	130.757	35	33	37	SW	949.031	236.103	27
27	28	89.43355	73	6	6	SW	823.035	150.531	28
28	29	72.445	53	28	55	NW	366.145	92.309	29
29	1	79.59063	33	44	54	NW	930.74	45.81	1

IX.9. PROGRAMA PARA CALCULAR EL AREA :

EL NUMERO DE VERTICES ES: 29

EST.	X	Y	PROD.X	PROD.Y
1	45.31	330.74	46680.47	456225.4
2	490.174	1019.002	535569.3	474752.1
3	426.645	1092.607	527565.1	323006.1
4	300.205	1226.537	371603	294146.6
5	337.379	1305.112	319771.7	473756
6	365.161	1344.262	502742.3	753137.5
7	560.261	1324.903	719274.7	303777.7
8	590.4	1293.821	750904.1	723732.6
9	567.629	1293.77	727628.7	627000.3
10	481.539	1281.876	607801	497089.7
11	337.793	1262.205	458686.3	514252.6
12	407.424	1182.844	495805.3	702330.5
13	593.722	1216.927	623911	762782
14	626.81	1050.847	623603.7	700244.5
15	666.362	994.885	662813.3	795959.3
16	300.052	1024.689	343255.2	759713.4
17	336.59	1053.763	319393.2	1002409
18	351.086	981.639	304921.2	796016
19	310.905	951.461	290933	644151.5
20	677.013	926.043	618842.7	586253.6
21	633.074	914.078	576635.5	529483.4
22	579.254	910.85	521234.3	493274.5
23	541.554	999.839	483162.1	442517
24	491.774	892.177	431512	370773.6
25	415.583	877.46	357018.2	321741.3
26	366.674	859.078	311317.6	202830.9
27	236.103	849.031	194321	127805.5
28	150.531	823.035	120381.7	75973.53
29	92.309	666.145	35915.68	39678.11

SUMATORIA EN X=
SUMATORIA EN Y=

15143573.2046875

15460999.17578125

LA SUPERFICIE TOTAL ES:
158712.685546875 METROS CUADRADOS

CAPITULO X

DISCUSION DE RESULTADOS

En el Capitulo III (respecto al cierre angular), se observa a primera vista que los ángulos no cierran, pero lo que sucede es que la computadora utiliza muchos dígitos, que ocuparían mucho espacio para imprimirlos todos, de este modo trunca los números por la instrucción: "print using".

Observando los resultados del capítulo IV (cálculo de los rumbos), se aprecia que la computadora sale con un valor de: $77^{\circ}01'00''01$ y llega con otro valor: $77^{\circ}00'59''99$, habiendo un error de $00''02$; lo cual no afecta al redondear al minuto.

En el capítulo V, no hay variantes, ya que se utilizaron para su cálculo, los rumbos mencionados.

A continuación, se presenta un resultado diferente al mostrado anteriormente, en el capítulo VI (referente a la compensación de la poligonal), para comparar el grado de precisión que tiene uno con respecto al otro.

Se podrá comprobar que es mínima la diferencia y los resultados presentados antes son del todo confiables.

En el siguiente ejemplo, se utilizó una calculadora de bolsillo CASIO FX-3500 :

DE	A	DISTANCIA	ANGULO HOR.			RUMBO			AZIMUT		PROYECCIONES CORR.		COORDENADAS		VERTICE
			1	2	3	4	5	6	7	8	N-S	E-W	X	Y	
1	2	200.00	92	29	15	19	SW	195	19	-192.974	-32.830	400.000	400.000	1	
2	3	231.33	86	46	77	55	SE	102	95	-48.419	+26.211	207.123	347.170	2	
3	4	117.89	134	34	58	39	NE	58	39	+64.708	+98.313	223.418	671.574	3	
4	5	85.95	164	11	40	50	NE	40	50	+65.039	+56.200	238.454	727.594	4	
5	6	35.43	160	57	21	47	NE	21	47	-32.903	+13.149	221.557	741.343	5	
6	1	250.00	81	12	77	01	NW	282	39	-78.643	-341.343	400.000	400.000	1	

El ejemplo siguiente, es el programa mostrado antes, en el Capítulo VI :

DE	A	DISTANCIA	ANGULO HOR.			RUMBO			AZIMUT			PROYECCIONES CORR.		COORDENADAS		VERTICE	
			1	2	3	4	5	6	7	8	9	10	11	12	X		Y
1	2	200	92	20	0	15	19	0	SW	195	19	0	-192.8736	-32.82939	400	400	1
2	3	231.33	86	46	0	77	55	0	SE	102	5	0	-48.41964	226.2108	158.7067	573.5815	2
3	4	117.89	134	34	0	58	39	0	NE	58	39	0	64.70786	98.31242	223.4146	671.6739	3
4	5	85.95	164	11	0	40	50	0	NE	40	50	0	65.03863	56.20087	238.4532	727.8947	4
5	6	35.43	160	57	0	21	47	0	NE	21	47	0	32.90391	13.14835	221.5571	741.0431	5
6	1	250	81	12	0	77	1	0	NW	282	39	0	78.64288	-341.0431	400	400	1

En los valores de las proyecciones, de los ejemplos de la calculadora con el programa, se observa que es poca la diferencia, ya que ésta oscila entre 0.00091 y 0.00010 metros. Como puede apreciarse en los siguientes cuadros:

PROYECCIONES CORREGIDAS EN "Y"		DIFERENCIA ARITMETICA EN METROS
CALCULADORA	PROGRAMA	
-192.374	-192.3736	0.00040
-48.419	-48.41964	0.00064
+64.708	+64.70786	0.00014
+65.039	+65.03863	0.00037
+32.903	+32.90391	0.00091
+78.643	+78.64288	0.00012

PROYECCIONES CORREGIDAS EN "X"		DIFERENCIA ARITMETICA EN METROS
CALCULADORA	PROGRAMA	
-52.850	-52.82939	0.00061
+226.211	+226.2108	0.00020
+98.313	+98.31242	0.00058
+56.200	+56.20087	0.00087
+13.149	+13.14835	0.00065
-341.043	-341.0431	0.00010

En las coordenadas, la diferencia oscila entre 0.0008 y 0.0001 metros. Como se muestra a continuación:

COORDENADAS EN "Y"		DIFERENCIA ARITMETICA EN METROS
CALCULADORA	PROGRAMA	
207.126	207.1264	0.0004
158.707	158.7067	0.0003
223.415	223.4146	0.0004
288.454	288.4532	0.0008
321.357	321.3571	0.0001

C O O R D E N A D A S EN "X"		DIFERENCIA ARITMETICA
CALCULADORA	PROGRAMA	EN METROS
347.170	347.1706	0.0006
573.381	573.3815	0.0005
571.594	571.5937	0.0001
727.894	727.8947	0.0007
741.043	741.0431	0.0001

Del Capítulo VII, respecto al cálculo de las distancias y los rumbos, dadas las coordenadas de los vértices, siempre variará un poco la distancia y el rumbo originales con los que se obtengan por cálculo. Por ejemplo, las distancias inicial y final, son originalmente : 200.00 y 350.00 y por cálculo, son: 199.9782 y 349.793 respectivamente. En cuanto a los rumbos, no cambian si se redondean al minuto.

Asimismo, puede observarse lo mismo con el cálculo del área (como se indicó en el capítulo VIII):

PROGRAMA :	CALCULADORA:
60,489.365224375 m. ²	60,489.38517 m. ²

Habiendo una diferencia, entre ellas de 0.019935625 metros cuadrados.

Este valor resulta porque la computadora utiliza doble precisión y la calculadora tiene 10 dígitos de precisión. Además, siempre habrá variantes numéricas, pero como la diferencia es mínima, no afecta al resultado final, ya que ambos tienen el mismo propósito.

Refiriéndose al capítulo IX (Resultados), se muestra un ejemplo de un levantamiento realizado en la Delegación Iztapalapa, el cual contiene 29 vértices. Asimismo, se muestra su dibujo a escala (1:5000).

En el programa del cierre angular, se tiene que al redondear los segundos, se cumple con la condición de cierre.

En el programa de los rumbos, se proporcionó uno inicial, con un valor de: $N 78^{\circ} 46' 0.01'' E$ y finalmente, en la comprobación, se llega con un valor de: $N 78^{\circ} 46' 0.21'' E$, habiendo una diferencia de $0.20''$, la cual es mínima para las admitidas en topografía.

En el programa de los acimutes, no hay problema, ya que se calcularon a partir de los rumbos proporcionados.

En el programa para la compensación de la poligonal, se proporcionan las siguientes coordenadas iniciales:

$$Y1 = 930.74 \qquad X1 = 45.81$$

Después de calcular las demás coordenadas, se llega al final con las siguientes:

$$Y1 = 930.7401 \qquad X1 = 45.81013$$

Teniendo, entre ellas, una diferencia mínima que, puede despreciarse, ya que los valores de las coordenadas, generalmente sólo se proporcionan hasta los milímetros.

Luego, se presentan los resultados del problema inverso: el cálculo de los rumbos y las distancias, partiendo de las coordenadas. Aquí, siempre habrá variaciones pequeñas, como se aprecia en la sección IX.5.

En el programa para el cálculo del área, los resultados, pueden compararse con los obtenidos con una calculadora CASIO FX-1600, programable, con 10 dígitos de precisión.

Así, queda ilustrado de la siguiente manera:

PROGRAMA:	CALCULADORA:
158,712.865546875 m. ²	158,712.708 m. ²

Puede comentarse que hay una diferencia mínima de: 0.00245312 m.², la cual se debe a la precisión que tiene la computadora con respecto a la calculadora de bolsillo utilizada.

Y de esta manera, los programas descritos anteriormente, son precisos y confiables.

CAPITULO XI

CONCLUSIONES

1.- El lenguaje GW-Basic, lo tienen instalado la mayoría de las computadoras, así será sencillo disponer de los programas expuestos en este trabajo, para facilitar el cálculo de los elementos que intervienen en una poligonal topográfica cerrada.

2.- Para todo aquel que comience a familiarizarse con las computadoras y, sin profundizar mucho en otros lenguajes (que no sean BASIC), pueda utilizar los programas anteriores para comprender mejor la manera de solucionar el problema: calcular una poligonal topográfica cerrada.

Asimismo, puede transferir, posteriormente, los programas indicados en este trabajo, a otros lenguajes.

3.- Los programas descritos en este trabajo, son aplicables a la solución de cualquier poligonal topográfica cerrada. Únicamente, se cambiarán los valores de los "data":

4.- Se puede procesar por partes, un cálculo determinado, a saber: sólo el cierre angular; sólo los rumbos y los azimutes; sólo compensar la poligonal; sólo calcular el problema inverso: cálculo de rumbos y distancias ó, únicamente calcular el área.

5.- Todos los programas tienen una precisión confiable, ya que ésta oscila entre valores muy pequeños. Dependiendo del número de vértices, la precisión alcanzada, los aparatos utilizados, las condiciones del levantamiento, etc., pueden obtenerse diferencias mínimas. Como se muestra a continuación:

El ejemplo que sigue, tiene en total 6 vértices. Y es el tratado en los resultados de cada capítulo (desde el III - VIII):

CONCEPTO:	PRECISION (+/-):	UNIDAD:
Angulos	0.02	segundos de arco (")
Distancias	0.00091	metros (m.)
Area	0.019953625	metros cuadrados (m. ²)

El ejemplo siguiente, tiene en total 29 vértices. Y es el tratado en el capítulo IX (Resultados) :

CONCEPTO:	PRECISION (+/-):	UNIDAD:
Angulos	0.20	segundos de arco (")
Distancias	0.0036	metros (m.)
Area	0.02245312	metros cuadrados (m. ²)

Resultando valores mínimos, que caen dentro de la tolerancia y éstos pueden despreciarse, ya que no afectan los cálculos finales.

BIBLIOGRAFIA

- Albrecht, B. & Inman, D. 1991. SW-Basic a su alcance. España. McGraw-Hill. 379 p.
- DATAPRO Information Services Group. 1991. DATAPRO Manufacturing Automation Series: CAD / CAM / CAE Systems. USA. MacGraw-Hill. p.v.
- Duffy, T. 1990. Cuatro herramientas del Software-Plus. México. Grupo Editorial Iberoamérica. 800 p.
- International Business Machines (IBM). 1980. Historia de la Computación. (El siglo del procesador electrónico). México. IBM. 116 p.
- Joyanes Aguilar, L. 1987. Metodología de la Programación. (Diagramas de flujo, algoritmos y programación estructurada). México. McGraw-Hill. 248 p.
- Montes de Oca, M. 1982. Topografía. (4a. ed.) México. Representaciones y Servicios de Ingeniería, S.A. 344 p.