

76
2 ej



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

"SISTEMA EXPERTO PARA EL ANALISIS
DE EXCAVACIONES"

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO CIVIL

P R E S E N T A:
ALMA ROSA PINEDA CONTRERAS



MEXICO, D.F.

1996

TESIS CON
FALLA DE ORIGEN

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 esta 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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERIA
DIRECCION
60-1-139/94

Señorita
ALMA ROSA PINEDA CONTRERAS
Presente.

En atención a su solicitud me es grato hacer de su conocimiento el tema que propuso el profesor **M.I. ROBERTO MAGAÑA DEL TORO**, que aprobó esta Dirección, para que lo desarrolle usted como tesis de su examen profesional de **INGENIERO CIVIL**.

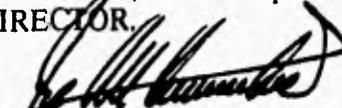
"SISTEMA EXPERTO PARA EL ANALISIS DE EXCAVACIONES"

- I. INTRODUCCION**
- II. METODO DE ANALISIS**
- III. SISTEMA EXPERTO DESARROLLADO**
- IV. APLICACION A UN CASO PRACTICO**
- V. CONCLUSIONES**

Ruego a usted cumplir con la disposición de la Dirección General de la Administración Escolar en el sentido de que se imprima en lugar visible de cada ejemplar de la tesis el título de ésta.

Asimismo le recuerdo que la Ley de Profesiones estipula que deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito para sustentar Examen Profesional.

Atentamente
"POR MI RAZA HABLARA EL ESPIRITU"
Cd. Universitario, a 26 de septiembre de 1994.
EL DIRECTOR.


ING. JOSÉ MANUEL COVARRUBIAS SOLIS

JMCS/RCR*nl

A mi madre, por todo lo que me ha dado

A mis hermanos

A mi Pueblo; a los olvidados sin oportunidad de educación

Agradecimientos

Mi agradecimiento al M en I Roberto Magaña Del Toro y al Dr. Miguel P. Romo por la oportunidad.

A las personas del Instituto de Ingeniería, amigos y compañeros, por su compañía y ayuda en la realización de este trabajo.

ÍNDICE

1.	INTRODUCCIÓN	1
1.1	Objetivo	3
2.	MÉTODO DE ANÁLISIS	5
2.1	Análisis de la presión lateral de tierras sobre elementos de soporte	6
2.1.1	Teoría de Rankine en suelos cohesivos	6
2.1.2	Teoría de coulumb	8
2.2	Movimientos de fondo	9
2.2.1	Levantamiento de fondo	9
2.2.2	Falla de fondo por subpresión	10
2.3	Métodos experimentales	11
2.4	Fundamentos para el estudio de los movimientos	13
2.4.1	Generalidades del método del elemento finito	13
2.4.2	Características a considerar en el análisis de excavaciones	18
2.5	Metodología de análisis.	18
3.	SISTEMA EXPERTO DESARROLLADO	23
3.1	Antecedentes	23
3.1.1	Subcampos de la IA	24
3.2	Sistemas Expertos	25
3.2.1	Tipología de Sistemas Expertos	25
3.2.2	Características de los Sistemas Expertos	26
3.3	Arquitectura de un Sistema Experto	27
3.3.1	Base de conocimientos	27
3.3.2	Motor de inferencia	28
3.3.3	Base de hechos	28
3.3.4	Módulos de comunicación	28
3.4	Representación del conocimiento	29
3.4.1	Reglas de producción	30
3.4.2	Redes semánticas	31
3.4.3	Marcos	31

3.5	Estrategias de control	32
3.5.1	Mecanismos de búsqueda	32
3.5.2	Estrategias de selección de reglas	33
	a) Metaconocimiento	33
	b) Metarreglas	34
3.5.3	Lógica	34
3.5.4	Evaluación del conocimiento	35
3.6	Lenguajes y sistemas de programación	35
3.7	Metodología de construcción de Sistemas Expertos	38
3.8	Desarrollo del Sistema Experto "SEANEX"	42
4.	APLICACIÓN A UN CASO PRÁCTICO	50
4.1	Descripción del problema	51
4.2	Características de la malla utilizada	56
4.3	Construcción de la malla	57
4.4	Construcción de archivo de datos	57
4.5	Graficación de resultados	58
5.	CONCLUSIONES	68
5.1	Recomendaciones	69
	REFERENCIAS	70
	ANEXO I	72
	ANEXO II	79
	ANEXO III	117
	ANEXO IV	130
	ANEXO V	133

1. INTRODUCCIÓN

Desde sus orígenes el hombre ha estado constantemente empeñado por la búsqueda de la resolución de problemas que se le presentan en su vida cotidiana (científica y social); que van desde la explicación de comportamientos de fenómenos, hasta aquellos que son creados por curiosidad o distracción (paradojas, acertijos, adivinanzas, juegos, etc.), esto ha ocupado el tiempo y requerido esfuerzos adicionales del hombre.

En la década de los cincuenta aparece en psicólogos y pedagogos un interés especial por encontrar métodos generales de resolución de problemas, que junto con los avances en el área de la computación, permitieran desarrollar procesos casi automáticos para la resolución

de problemas; desde este momento la Inteligencia Artificial hace su aparición. Se empiezan a escribir programas de tipo simbólico para la resolución automática de problemas, el conocimiento ya no es tratado solamente bajo el punto de vista numérico.

Los Sistemas Expertos (SE) son una rama de la Inteligencia Artificial, representan la transición del procesamiento de datos al procesamiento del conocimiento y sustituyen los algoritmos por mecanismos de inferencia, teniendo como característica, gracias a su gran capacidad de comunicar, el explicar porqué se procede de una u otra forma y el justificar los resultados.

Un SE es un conjunto de programas y herramientas capaces de simular el comportamiento de un Experto Humano en el momento de resolver un problema en un campo delimitado del conocimiento.

En realidad un SE sólo se comporta como un Experto Humano de manera aproximada, ya que hasta el momento se desconoce gran parte de los procesos mentales y motores de búsqueda que permiten al cerebro buscar y seleccionar las relaciones convenientes que hacen posible solucionar un problema.

En la mayor parte de los casos, hace falta aún emplear lo que hace el propio hombre y que la inteligencia artificial no puede todavía reemplazar: la inteligencia natural.

Los SE tienen aplicación práctica en: la interpretación de datos, el diagnóstico, la corrección, la monitorización, el control, la predicción, la planificación, el diseño y la enseñanza.

Los SE adquieren el conocimiento especializado a través de un experto, el cual es almacenado en una base de conocimiento; en ésta no sólo se almacenan datos individuales, sino que contiene también objetos complejos, sus cualidades, relaciones entre objetos y reglas para el procesamiento del conocimiento. La formalización del conocimiento en un SE se lleva a cabo de forma independiente del procesamiento, para ello se dispone de diferentes formas de representación del conocimiento, redes semánticas, reglas de producción, objetos estructurados.

El tratamiento del conocimiento se realiza mediante un motor de inferencia quien es el encargado de buscar y seleccionar de la base de conocimiento las reglas que se deben aplicar, cómo y cuándo, mediante estrategias de solución de problemas tales como: encadenamiento hacia adelante, encadenamiento hacia atrás o la combinación de estos dos mecanismos, para llegar finalmente a la mejor solución posible según el estado de conocimiento del dominio considerado (o quizás ninguna).

Todos los datos que describen el problema, las reglas utilizadas, los nuevos datos deducidos, las soluciones y propuestas de solución son almacenados en la base de conocimiento.

Es importante señalar que casi desde cualquier lenguaje de programación es posible construir un SE, las ventajas de unos sobre otros radican fundamentalmente, en la

complejidad que representa para el programador construir la lógica de funcionamiento del SE.

De manera generalizada se puede decir que la toma de decisiones en Geotecnia suele realizarse a través de una combinación de: teorías científicas y técnicas analíticas, métodos experimentales, juicio y sentido común; Lo que lleva a considerar que la información es escasa, que hay una gran diversidad de fenómenos y variables que intervienen en el comportamiento del problema y que las teorías existentes están fundamentadas en criterios y condiciones de aplicación específicos.

Como en los últimos años se han utilizado técnicas numéricas como el método del elemento finito que permite establecer procedimientos de análisis confiables, como es el caso del diseño de excavaciones profundas apuntaladas con caras verticales como las que se realizan en la ciudad de México para construcción de cimentaciones profundas y para las líneas del sistema de transporte colectivo (metro); el capturar el conocimiento de una forma computarizada logra la disponibilidad de éste y permite agilizar la toma de decisiones.

El estudio de los movimientos inducidos por las excavaciones a cielo abierto son de gran interés para el ingeniero geotecnista, ya que el reducir los movimientos del terreno a niveles que las estructuras colindantes lo asimilen sin presentar daño alguno permitirá el uso de procedimientos más eficientes.

Estos análisis implican una serie de pasos adicionales para plantear al problema deseado, lo que para el analista representa tiempo y por lo tanto retraso en la obtención de resultados finales.

1.1 OBJETIVO

En el presente trabajo se desarrolló un Sistema Experto el cual conjunta mediante módulos, la información de entrada y salida para evaluar el estado de desplazamientos causados por las excavaciones, apoyándose para estos análisis en el programa de computadora de elementos finitos TEST92 que simula procesos constructivos en suelos elasto-plásticos.

Para poder alcanzar el objetivo deben cumplirse los siguientes aspectos:

- 1) Automatizar parte de los análisis en el estudio de los movimientos causados por una excavación, para servir de apoyo a los Expertos Humanos.
- 2) Ser parte del proceso de entrenamiento del personal necesario en este tipo de análisis.
- 3) Construir un medio de difusión del conocimiento.

Los módulos que estructuran el sistema son independientes entre sí y a la vez son controlados por un módulo principal que permita ir incorporando nuevos módulos en futuras investigaciones.

2. MÉTODO DE ANÁLISIS

La evaluación de los movimientos del terreno aledaño a una excavación a cielo abierto son de gran interés para predecir la estabilidad de la misma. La estabilidad significa prevención de la falla, y esto se logra con el uso de métodos constructivos que reduzcan los movimientos del terreno a niveles que las estructuras colindantes los asimilen sin presentar daño alguno.

El diseño de las medidas que proporcionen estabilidad incluye la determinación de las presiones de tierra y de la falla potencial de fondo, y la selección de un sistema de drenaje que evite la subpresión en el fondo de la excavación y el flujo de agua hacia la misma a lo largo del muro. El control de los movimientos que se inducen al terreno generalmente se basa en el tipo de construcciones que existan a lo largo de la excavación y las facilidades que se tengan para realizar los trabajos.

Los aspectos que interesan conocer de las excavaciones profundas son la estabilidad lateral de la excavación, los movimientos de fondo y los asentamientos de colindancia, en donde lo único que puede evaluarse con certeza usando métodos tradicionales es el movimiento de fondo de la excavación. Así, los movimientos del fondo de una excavación se estiman dentro de un rango de tolerancia.

La utilización del Método del Elemento Finito para estimar esfuerzos y deformaciones en obras térreas está validado bajo casos historia bien documentados que permiten avanzar en el conocimiento sobre el comportamiento de este tipo de obras así como establecer procedimientos constructivos alternos innovadores. El método del elemento finito permite reproducir con buena aproximación los resultados reales, como es el caso de movimientos horizontales y verticales de una excavación. Por consiguiente constituye una herramienta útil que puede usarse como ayuda en el diseño de excavaciones a cielo abierto en la arcilla de la Ciudad de México.

2.1 ANÁLISIS DE LA PRESIÓN LATERAL DE TIERRAS SOBRE ELEMENTOS DE SOPORTE

Para mantener los desplazamientos laterales en el suelo dentro de los límites de tolerancia basta con restablecer, en el suelo, el estado de esfuerzos correspondiente al empuje de tierras en reposo; en base a esto se puede calcular la precarga en cada puntal.

Existen dos teorías fundamentales que permiten evaluar la presión de tierras sobre elementos de soporte: Rankine y Coulomb. Estas teorías suponen que 1) El muro puede desplazarse por giro o desplazamiento por cualquiera de sus aristas, en una distancia suficiente como para que se alcance a desarrollar un estado de equilibrio plástico. 2) La presión de poro es despreciable. 3) Las constantes del suelo que aparecen en las fórmulas del empuje tienen valores definidos y pueden determinarse con exactitud.

Conviene añadir desde ahora que en estas teorías no se considera cualquier efecto del agua que se pudiera acumular en el relleno, de manera que tales efectos, si existen habrán de considerarse de manera independiente, superponiéndolos a los empujes del suelo "seco".

2.1.1 Teoría de Rankine en suelos cohesivos.

La teoría de Rankine supone que no existe fricción entre el contacto muro-suelo, que la superficie del suelo es horizontal y que el paramento del muro es vertical; además establece que la falla es alcanzada cuando el círculo de Mohr toca la envolvente de falla.

Si se considera el estado de esfuerzo es reposo el material puede llegar a la falla por dos caminos. El primero permite que el elemento se deforme lateralmente por disminución de la presión horizontal, hasta el valor mínimo compatible con el equilibrio. El segundo consiste en aumentar la presión horizontal hasta que, después de sobrepasar el valor γz , alcance un

esfuerzo tal que el nuevo círculo de esfuerzos resulte también tangente a la envolvente horizontal de falla; de ésta manera, caben dos estados plásticos prácticos. El que se tiene cuando el esfuerzo horizontal alcanza el valor mínimo $K_a \gamma z$ (estado activo) y el que ocurre cuando dicha presión llega al valor máximo $K_p \gamma z$ (estado pasivo).

De acuerdo con Rankine se dirá que un suelo está en estado plástico cuando se encuentra en estado de falla generalizado.

Las fórmulas para las presiones activas se pueden relacionar con el empuje de suelos sobre muros, en tanto que las pasivas se relacionan con los casos en que los muros presionan al relleno tras ellos.

Para efectos de cálculo de la estabilidad del muro considerado como un elemento rígido, se puede considerar que el volumen de presiones los sustituyen fuerzas concentradas, cuya magnitud queda dada por E_A y E_P ;

$$E_a = \frac{1}{2} \gamma H^2 - 2cH \quad (2.1)$$

$$E_p = \frac{1}{2} \gamma H^2 + 2cH \quad (2.2)$$

dada la distribución lineal que para ambas presiones se tiene en la teoría de Rankine, se infiere que el punto de aplicación de tales fuerzas está a un tercio de la altura del muro, medido a partir de la base.

Las fórmulas 2.1 y 2.2 únicamente serían aplicables si fuera horizontal la superficie del relleno tras el muro y si los estados plásticos correspondientes se desarrollan por completo en el relleno.

La máxima altura a que puede llegarse en un corte vertical de material "cohesivo" sin soporte y sin derrumbe es:

$$H_c = \frac{4c}{\gamma} \quad (2.3)$$

La ecuación 2.3 da valores un poco altos de la altura estable real y en caso de que se use deberá ser afectada por un factor de seguridad de 2 como mínimo.

El conocimiento de las condiciones de frontera es esencial para estimar las presiones sobre estructuras de retención; tales condiciones deben representar el procedimiento de

construcción y la interacción entre el suelo y la estructura, en vez de ser simples condiciones analíticas, tales como "muro rígido", "muro liso" o "muro rugoso".

2.1.2. Teoría de Coulomb.

Ésta, considera que el empuje sobre un muro, se debe a una cuña de suelo, limitada por el paramento del muro, la superficie del relleno y una superficie de falla desarrollada dentro del relleno, a la que se supone plana, fig 2.1.

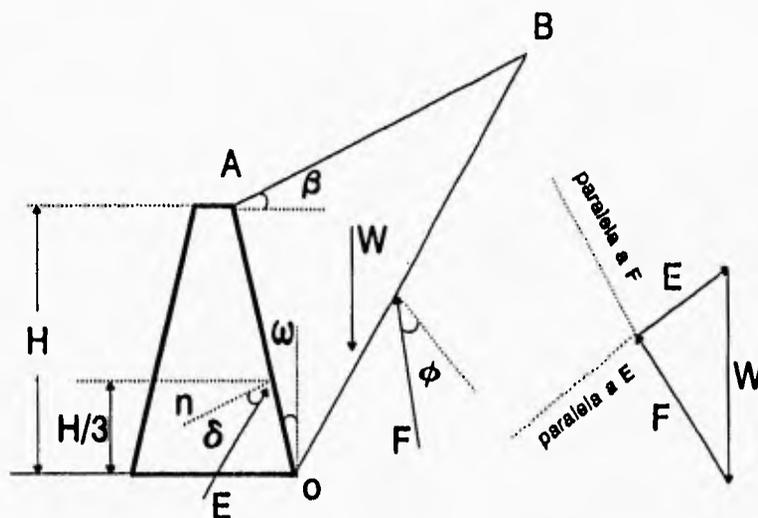


Fig 2.1 Mecanismo de empuje de suelos "friccionantes" según Coulomb

La cuña OAB, tiende a deslizar bajo el efecto de su peso y por esa tendencia se producen esfuerzos de fricción tanto en el respaldo del muro, como a lo largo del plano OB. Si suponemos que las resistencias friccionantes se desarrollan por completo, las fuerzas EA y FB, resultan inclinadas, respecto a las normales correspondientes a los ángulos δ y ϕ .

Siguiendo indicaciones de Terzaghi, el valor de δ puede tomarse como:

$$\frac{\phi}{2} \leq \delta \leq \frac{2}{3} \phi \quad (2.4)$$

considerando el equilibrio de la cuña, se ve que el polígono dinámico constituido por W, F y E debe cerrarse. Como W es conocida en dirección y magnitud, y se conocen previamente las direcciones de E y F; dicho polígono puede construirse para una cuña dada. Así puede conocerse la magnitud del empuje sobre el muro. Es claro que no hay ninguna razón para que la cuña escogida sea la que produce el empuje máximo. Se ve, así, que el método de trabajo, desemboca en un procedimiento de tanteos, dibujando diferentes cuñas, calculando el empuje correspondiente a cada una y llegando así, a una aproximación razonable para el valor máximo, producido por la cuña crítica.

Además de estos métodos para calcular empujes, existen otros más, como el método gráfico de Culman, método del círculo de fricción interna, método de la espiral logarítmica, y el método semiempírico de Terzaghi.

Otros métodos y teorías clásicas de empuje de tierra que pueden ser de interés en algunos casos particulares. Estas teorías requieren que toda la masa de suelo esté en equilibrio límite y que la resistencia del mismo se exprese en términos de la teoría de falla de Mohr-Coulomb.

2.2 MOVIMIENTOS DE FONDO

2.2.1 Levantamiento de fondo

En excavaciones a cielo abierto realizadas en arcilla blanda debe considerarse la posibilidad de que se produzca la falla de fondo por levantamiento, debido a que el peso de los bloques de arcilla que constituyen los lados de la excavación tiende a desplazar el fondo hacia arriba. Esto se relaciona con la precarga que se aplica por medio de los puntales, si esta resulta ser muy pequeña los desplazamientos laterales son grandes y en consecuencia generarse zonas de plastificación amplias, las cuales, producen irremediablemente un mecanismo de falla generalizada.

Existen varias teorías que permiten estimar la falla potencial de fondo para excavaciones en arcilla, entre éstas están:

Terzaghi (Juárez B., E. y Rico R., A. (1974)) analizó el factor de seguridad contra levantamiento del fondo de excavaciones apuntaladas, basándose en la capacidad de carga última por unidad de longitud.

$$FS = \frac{Qu}{Q} = \frac{1}{H} \left(\frac{5.7c}{\gamma - \frac{c}{0.7B}} \right) \quad (2.5)$$

La ecuación anterior es válida cuando el estrato de arcilla es homogéneo, al menos hasta una profundidad de $0.7B$ por debajo del nivel de desplante. Sin embargo, si un estrato duro de roca o de material similar es localizado a una profundidad $D < 0.7B$, el factor de seguridad puede modificarse de la siguiente forma:

$$FS = \frac{1}{H} \left(\frac{5.7c}{\gamma - \frac{c}{D}} \right) \quad (2.6)$$

Bjerrum y Eide propusieron un factor de seguridad para estratos homogéneos por debajo del fondo de la excavación.

$$FS = \frac{cN_c}{\gamma H} \quad (2.7)$$

donde N_c depende de la relación H/B y L/B .

Cuando existen estratos de arcilla resistentes a poca profundidad la superficie de falla debajo de la excavación podrá ser controlada por la resistencia no-drenada c_1 y c_2 . Para esta condición el factor de seguridad está dado por

$$FS = \frac{c_1(N_c F_d) F_s}{\gamma H} \quad (2.8)$$

donde:

N_c : Factor de capacidad de carga en una excavación infinitamente larga.

F_d : Factor de profundidad, el cual es una función de H/B (Juárez B., E. y Rico R., A. (1974)).

F_s : Factor de forma

c_1 : Cohesión no-drenada de arcilla blanda.

c_2 : Cohesión del estrato resistente a poca profundidad.

El factor de forma, F_s , es dado como:

$$F_s = 1 + 0.2 \frac{B}{L} \quad (2.9)$$

2.2.2 Falla de fondo por subpresión

Este tipo de falla se puede presentar debido a la existencia de estratos delgados de permeabilidad mucho mayor que de las arcillas. En estos estratos, las presiones artesianas, respecto al fondo del corte, producidas por el alivio de las presiones ocasionadas por la

excavación, puede provocar el levantamiento del material que está debajo de él, o que el flujo hacia arriba disminuya el poder portante del suelo.

El factor de seguridad contra la falla de fondo por efecto de la subpresión se expresa mediante la relación:

$$(FS) = \frac{P + S}{U} \quad (2.10)$$

donde:

P: Peso saturado del prisma de suelo bajo el fondo.

S: Fuerza cortante resistente en las caras verticales del prisma del fondo.

U: Fuerza total de subpresión en la base del prisma del fondo.

2.3 MÉTODOS EXPERIMENTALES

Con el objeto de verificar el comportamiento del suelo en la zona de excavación y adyacente a la misma, y tener los elementos suficientes que permitan juzgar los criterios de análisis adoptados, fundamentando en su caso la toma de decisiones con respecto al cambio o ratificación del procedimiento constructivo empleado y brinde la oportunidad de evaluar la confiabilidad de métodos analíticos para calcular esfuerzos y deformaciones en este tipo de obras construidas en la arcilla blanda de la Ciudad de México, es recomendable la instrumentación de la zona que proporcione un control significativo de los movimientos inducidos por la excavación.

La instrumentación permite la determinación de: a) la evaluación con el tiempo de las deformaciones verticales y horizontales, b) la variación con el tiempo de la presión de poro en los estratos más significativos para estimar la evaluación de los esfuerzos efectivos de la masa de suelo, conforme al progreso de la construcción.

La confiabilidad de los instrumentos para medir desplazamientos depende de la familiarización de todas las posibles causas de mal funcionamiento de los aparatos.

La calibración del instrumento es importante, por lo que todos los fabricantes deben proporcionar listas detalladas en donde se cubran todas las etapas de instalación, calibración en ceros y las toma de lecturas. La toma de lectura es el punto de falla de la confiabilidad de los instrumentos porque se puede dar datos falsos, especialmente cuando se utilizan transductores eléctricos en la recopilación de datos de asentamientos, es por esto que se recomienda el uso de sistemas mecánicos para tener cierto punto de seguridad de los registros.

En la tabla 2.1 se muestra las características generales de algunos dispositivos de los existentes para medir estos movimientos.

Grupo	Dispositivo	Función
Dispositivos para medir presión de poro	-Piezómetro de tubo vertical	miden los niveles de agua en suelos permeables y semi-permeables.
	-Piezómetro abierto en perforación previa	mide presión de poro
	-Piezómetro neumático	mide presiones de poro positivas en suelos totalmente saturados
Dispositivos para medir presiones laterales	-Celda de presión eléctrica para tierra	mide esfuerzos horizontales
	-Celda hidráulica de carga	define la magnitud de la carga aplicada a los troqueles.
	-Celda de presión tipo Carlson	mide esfuerzos en los contactos de las estructuras con el suelo.
Instrumentos para medir movimientos de masas de tierra	-Referencias superficiales	se detectan desplazamientos horizontales y verticales
	-Puntos de referencia subterráneos	se permite conocer los movimientos en el fondo de la excavación.
	-Bancos de nivel	se obtienen lecturas directas de los asentamientos.
	-Bancos de nivel profundo	miden los desplazamientos verticales.
Detectores de superficies de deslizamientos	-Deflectómetros	miden desplazamientos relativos al eje de la perforación.
	-Inclinómetro	permite conocer la distribución con la profundidad de los desplazamientos horizontales.
	-Banco de nivel flotante	determina los movimientos verticales causados por las excavaciones y hundimientos generales en el fondo de la excavación.

Tabla 2.1 Dispositivos de medición de movimientos

2.4 FUNDAMENTOS PARA EL ESTUDIO DE LOS MOVIMIENTOS

Uno de los aspectos que afectan notablemente el comportamiento de obras civiles construidas en suelos blandos es el proceso constructivo. El efecto de los procedimientos que se sigan y la calidad de la mano de obra se manifiesta no sólo durante la construcción sino que repercute durante la operación de la estructura. Esto se debe a que el comportamiento de los suelos depende de manera importante de la historia de esfuerzos a la que se le somete. En algunos casos basta con modificar la trayectoria de esfuerzos para limitar la generación de presiones de poro, disminuir las deformaciones, e incluso, evitar la falla. Por estos motivos, los análisis de esfuerzos y deformaciones en excavaciones deben considerar de manera fidedigna el procedimiento constructivo.

2.4.1 Generalidades del Método del Elemento Finito

El Método del Elemento Finito es un procedimiento general de discretización de los problemas continuos planteados por expresiones definidas matemáticamente.

El estudio del comportamiento de cualquier fenómeno físico en muchos casos puede ser representado con un modelo adecuado utilizando un número finito de componentes bien definidos. A tales problemas, se les denomina discretos. En otros, la subdivisión prosigue con un número infinito de elementos implicados que conducen a ecuaciones diferenciales o expresiones equivalentes. A tales sistemas se les llama continuos.

El Método del Elemento Finito es un método de aproximación que permite estimar esfuerzos y deformaciones en un medio continuo (dominio de las variables de las ecuaciones diferenciales), mediante la resolución de ecuaciones diferenciales de problemas de valores en la frontera o de valores en la frontera e iniciales.

El medio se discretiza en varias regiones denominadas elementos finitos, de formas convenientes (líneas, triángulos, cuadriláteros, tetraedros, etc), interconectados en sus fronteras por nodos. Las variables de la ecuación diferencial se aproximan mediante una combinación lineal de funciones de interpolación especificados en los puntos nodales. La matriz de rigidez del elemento se obtiene a partir de la interpolación. Posteriormente se ensamblan las matrices de rigideces de los elementos para obtener la matriz de rigidez global. Por otra parte, el vector de cargas se aplica en los nodos y se resuelven las ecuaciones de equilibrio para los desplazamientos nodales. Por último, se calculan los esfuerzos y deformaciones de cada elemento a partir de dichos desplazamientos.

Considerando un elemento triangular de tres nodos (fig 2.2), se tiene que las componentes del vector de desplazamientos, para un caso bidimensional, queda expresado por:

$$\begin{bmatrix} u \\ v \end{bmatrix} = [N][d_e] \quad (2.11)$$

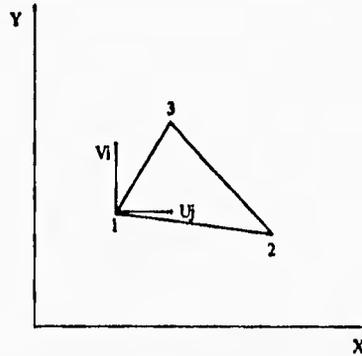


Fig 2.2 Desplazamientos en los nodos de un elemento triangular

donde la matriz N es la función de interpolación que permite aproximar la solución exacta de los desplazamientos dentro del elemento y d_e es el vector de desplazamientos nodales.

Encontrando los desplazamientos u y v , para dos grados de libertad, a partir de la consideración de que estos son funciones lineales de las coordenadas, se tiene entonces:

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y & \approx & u = N_1 u_1 + N_2 u_2 + N_3 u_3 \\ v &= \alpha_4 + \alpha_5 x + \alpha_6 y & \approx & v = N_1 v_1 + N_2 v_2 + N_3 v_3 \end{aligned}$$

sustituyendo las coordenadas de los tres nodos

$$\begin{aligned} u &= (x = x_1) = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 \\ u &= (x = x_2) = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \\ u &= (x = x_3) = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 \end{aligned}$$

expresando en forma matricial para obtener el valor de cada constante

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

resolviendo el sistema de ecuaciones:

$$\begin{aligned}
 u = & \frac{1}{2a} [x_1 y_1 - x_2 y_2 + (y_1 - y_2)x + (x_1 - x_2)y] u_1 + \\
 & \frac{1}{2a} [x_1 y_1 - x_2 y_2 + (y_1 - y_2)x + (x_1 - x_2)y] u_2 + \\
 & \frac{1}{2a} [x_1 y_1 - x_2 y_2 + (y_1 - y_2)x + (x_1 - x_2)y] u_3,
 \end{aligned} \tag{2.12}$$

de donde

$$u = N_1 u_1 + N_2 u_2 + N_3 u_3$$

de la misma forma se obtiene para v:

$$v = N_1 v_1 + N_2 v_2 + N_3 v_3$$

de esta manera la ecuación 2.9 queda expresada como:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}$$

De acuerdo con la Teoría de la Elasticidad, las deformaciones unitarias están dadas (para deformación plana) por

$$\epsilon_x = \frac{\partial u}{\partial x}, \quad \epsilon_y = \frac{\partial v}{\partial y}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{2.13}$$

De los desplazamientos, se obtiene, derivando u con respecto a "x", que:

$$\epsilon_x = \frac{1}{2a} [(y_1 - y_2)u_1 + (y_1 - y_2)u_2 + (y_1 - y_2)u_3]$$

derivando v con respecto a "y"

$$\varepsilon = \frac{1}{2a} [(x_2 - x_1)v_1 + (x_1 - x_2)v_2 + (x_2 - x_1)v_3]$$

derivando u respecto a "y" y v respecto a "x"

$$\gamma_{xy} = \frac{1}{2a} [(x_2 - x_1)u_1 + (x_1 - x_2)u_2 + (x_2 - x_1)u_3] + [(y_2 - y_1)v_1 + (y_1 - y_2)v_2 + (y_2 - y_1)v_3]$$

expresando en forma matricial

$$\varepsilon = Bd \quad (2.14)$$

$$\begin{bmatrix} \varepsilon \\ \gamma_{xy} \end{bmatrix} = \frac{1}{2a} \begin{bmatrix} B_1 & 0 & B_2 & 0 & B_3 & 0 \\ 0 & B_4 & 0 & B_5 & 0 & B_6 \\ B_4 & B_1 & B_5 & B_2 & B_6 & B_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}$$

donde la matriz B depende de las coordenadas de los nodos de los elementos, es decir:

$$B_1 = (y_2 - y_3)$$

$$B_2 = (y_3 - y_1)$$

$$B_3 = (y_1 - y_2)$$

$$B_4 = (x_2 - x_1)$$

$$B_5 = (x_1 - x_3)$$

$$B_6 = (x_3 - x_2)$$

Los esfuerzos dentro del elemento están en función de los desplazamientos de los nodos, por la Ley de Hooke se tiene:

$$\sigma = D\varepsilon$$

es decir

$$\sigma = DBd, \quad (2.15)$$

Donde la matriz D mide el cambio geométrico del elemento y es obtenida a partir de las ecuaciones constitutivas de un material sólido, elástico, lineal e isotrópico

Para la determinación de la matriz de rigidez de un elemento, se puede recurrir a la valuación del trabajo originado por las fuerzas externas y del trabajo interno producido por la deformación acumulada.

El trabajo provocado por las fuerzas externas es:

$$W = \frac{1}{2} (P_1 u_1 + Q_1 v_1 + P_2 u_2 + Q_2 v_2 + P_3 u_3 + Q_3 v_3)$$

Expresado matricialmente

$$W = \frac{1}{2} d^T P,$$

donde:

$$P = \begin{bmatrix} P_1 \\ Q_1 \\ P_2 \\ Q_2 \\ P_3 \\ Q_3 \end{bmatrix}$$

El trabajo interno acumulado dentro del elemento por unidad de volumen:

$$\dot{W}_u = \frac{1}{2} (\epsilon_x \sigma_x + \epsilon_y \sigma_y + \gamma_{xy} \tau_{xy})$$

Expresado matricialmente

$$\dot{W}_u = \frac{1}{2} \epsilon^T \sigma$$

igualando el trabajo interno con el externo

$$d^T P_i = d^T \int_{\omega} B^T \sigma dV$$

$$P_i = \int_{\omega} B^T DB dV$$

si se considera

- 1) Un espesor unitario del elemento
- 2) Que el tamaño del elemento decrece

$$P_i = B^T DB dA$$

donde

A: área del elemento

Llamando a K_e matriz de rigidez del elemento, y estando definida como:

$$K_i = B^T DBA$$

entonces se tiene:

$$P_i = K_i d$$

Así pues, para determinar la matriz de rigidez global, se superponen las rigideces individuales de cada elemento por adición directa, formando de esta manera, una matriz K .

Si se denomina U al vector de desplazamientos de todo el medio y P al vector de carga, entonces se podrá obtener para todo el entorno la siguiente expresión:

$$P = KU$$

La expresión anterior origina un sistema de ecuaciones donde las incógnitas son los desplazamientos en los nodos de los elementos. Su solución permite conocer dichos desplazamientos, y así determinar las deformaciones unitarias y los esfuerzos en cada elemento utilizando las ecuaciones 2.14 y 2.15 respectivamente.

2.4.2 Características a considerar en el análisis de excavaciones

Para llevar a cabo el análisis de esfuerzos y deformaciones en obras térreas se emplea un programa de computadora de elementos finitos.

Una de las características prominentes de las estructuras de tierra es que los procedimientos constructivos son muy importantes para su comportamiento durante y después de la construcción, e inclusive hasta pueden provocar condiciones de falla; es por ello

que un análisis de esfuerzos y deformaciones en las obras geotécnicas se enfoca no solamente en las estructuras hechas sino también, muchas veces con mayor importancia, en las estructuras en construcción.

Debe señalarse que un análisis que incluya los aspectos constructivos es complicado debido a que: 1) La frontera del dominio de análisis cambia continuamente. En el caso de excavaciones el suelo del fondo va siendo removido continuamente a medida que se avanza la obra. 2) Durante el proceso de construcción y excavación, otros elementos estructurales (ademes, puntales, muros de retención, etc) suelen instalarse para garantizar la estabilidad de la excavación, de tal manera que una gran variedad y un gran número de elementos que no necesariamente son de tierra intervienen en el análisis. 3) El ambiente de construcción o excavación dista de ser aislado. Una excavación abierta puede efectuarse en un sitio donde a su alrededor se encuentran edificios. Así que el análisis de una excavación puede involucrar una o varias estructuras vecinas. 4) El comportamiento de los suelos durante la excavación no es fácil de modelarlo ya que es sometido a diversas trayectorias de esfuerzos de extensión, en contraste al estado de compresión que generalmente inducen las cargas transmitidas por el edificio construido. Zonas de tensión y por consiguiente de agrietamiento también pueden encontrarse.

Tomando en consideración todos estos aspectos, no es posible en todos los casos dar una solución analítica del problema en cuestión. Además los métodos de análisis y de diseño simplificados en muchas ocasiones no pueden representar las condiciones reales que se encuentran en el campo y requieren un análisis más refinado.

2.5 METODOLOGÍA DE ANÁLISIS.

Para estimar las deformaciones inducidas al suelo aledaño por una excavación apuntalada con paredes verticales se utiliza un programa de computadora de elementos finitos llamado TEST92 con el cual se pueden simular los aspectos más relevantes de los procedimientos constructivos en suelos elasto-plásticos.

Programa empleado

El programa de computadora TEST92 tiene las siguientes características:

- 1) Existe la opción de modelar el proceso constructivo por etapas o la forma de aplicación de cargas secuenciales
- 2) El problema es bidimensional y estático.
- 3) Consta de diferentes tipos de elementos: sólidos, viga y barra.

- 4) Cuenta con un algoritmo incremental e iterativo para analizar problemas no lineales. Los materiales pueden ser elásticos-lineales y/o elasto-plásticos perfectos.
- 5) El programa maneja el concepto de almacenamiento dinámico en el sentido de que la máxima memoria central está disponible en la etapa de solución del sistema de ecuaciones.

Es importante señalar que el TEST92 toma en cuenta elementos estructurales que no son parte de la excavación, simulando los muros como vigas y los puntales como cargas concentradas.

El comportamiento del suelo se asimila a un modelo elasto-plástico perfecto, combinado con la Ley de Falla de Von Mises y suponiendo una regla de flujo asociada. Este tipo de modelo da resultados satisfactorios cuando se usa para reproducir el comportamiento no-drenado de arcillas saturadas.

Generación de mallas

La discretización de una malla de elementos finitos dentro de la región o dominio que se pretende estudiar, está condicionada comúnmente por dos factores: el grado de exactitud de la solución a obtener y el ahorro computacional de la misma.

Aunque no existen reglas fijas que permitan decidir el tamaño y la forma de los elementos a emplear en un determinado problema, se atiende primeramente al ahorro computacional utilizando elementos complejos y mallas sencillas.

Los elementos finitos se pueden clasificar de acuerdo a su forma geométrica, la cual puede ser muy variada y por lo regular ésta depende de la geometría del modelo original. El programa de computadora empleado TEST92 plantea su formulación con elementos cuadriláteros.

La malla es trazada en papel milimétrico a una escala conveniente donde se especifica la geometría de cada elemento y son vaciadas las coordenadas de los nodos pertenecientes a cada elemento; se especifican los elementos auxiliares (muros, cargas).

Simulación de la excavación

Para la simulación de la excavación se toman en cuenta los siguientes pasos. 1) Se define el dominio de análisis (características geométricas de la excavación. 2) Se discretiza el medio en formas geométricas convenientes, como se explicó en el punto precedente. 3) Se establecen las condiciones de frontera, que restringen el desplazamiento o rotación del suelo. 4) Se describen las propiedades de los materiales de los que está constituido el medio, tales como: peso específico, cohesión, módulo de elasticidad. 5) se especifican las condiciones

iniciales de esfuerzo: ordenada en la dirección de la gravedad, valor del esfuerzo total vertical, valor del coeficiente de empuje en reposo, valor de la presión en el agua, valor de la presión preconsolidada, así como también las características de carga. 7) Por último establecer el tipo de elemento finito.

Archivo de datos

Una vez descritas las características de análisis del problema se construye el archivo de datos con la ayuda de un editor del sistema operativo MS-DOS, este archivo consta de una serie de líneas, en el anexo I se presenta un ejemplo.

Resultados obtenidos

De la ejecución del TEST92 se obtienen: desplazamientos que permiten conocer la estabilidad de las paredes verticales y la influencia de los asentamientos de colindancia en el dominio de análisis; así como los movimientos de fondo de una excavación. A partir del estado de esfuerzos se conoce la zona de plastificación del medio considerado. Factor de seguridad, el factor de seguridad se define como la relación entre la resistencia al corte no-drenada (S_u) y el esfuerzo cortante máximo inducido en el suelo por el proceso de excavación. Estos resultados permiten establecer con precisión la superficie de falla, representada por la zona de factor de seguridad igual a uno. Otro resultado interesante obtenido con el programa son los elementos mecánicos (momentos flexionantes) a los que se ve sometido el muro.

Desde un shell que se ejecuta desde la estación de trabajo se seleccionan los desplazamientos de los puntos nodales de acuerdo al campo de estudio.

Es de suma importancia mencionar que de acuerdo a los estudios realizados en el Instituto de Ingeniería (Rodríguez, R. (1995)) los resultados que proporciona el Método del Elemento Finito reproducen con aproximación suficiente los desplazamientos medidos en campo, justificando que con el procedimiento numérico es posible reproducir con buen grado de confianza los movimientos en el suelo aledaño a una excavación.

Graficación de resultados

Los resultados que se obtienen de los movimientos inducidos por una excavación son graficados de la siguiente manera:

- 1) Profundidad vs desplazamiento: muestra los desplazamientos laterales del muro por etapas de excavación.

- 2) Desplazamiento vertical vs distancia desde la excavación: muestra los asentamientos causados en el terreno junto a la excavación.
- 3) Desplazamiento vertical vs distancia que comprende el ancho de la excavación: grafica los movimientos de fondo.

De las gráficas obtenidas se puede observar el comportamiento de la excavación para así realizar de manera racional el diseño de los elementos estructurales tales como muros, puntales, evitando de esta forma una sobreestimación en el diseño y logrando con ello el abatimiento de los costos de construcción.

3. SISTEMA EXPERTO DESARROLLADO

3.1 ANTECEDENTES

Desde la antigüedad el hombre ha deseado liberarse de las tareas más pesadas y laboriosas que realiza; esto ha llevado a los investigadores a un esfuerzo continuo por encontrar métodos generales de resolución de problemas que eliminen eficientemente la costosa y lenta programación de cada paso de procesamiento. Este fue el punto de partida de los investigadores en los dominios de la Inteligencia Artificial (IA).

La IA es la ciencia que estudia el funcionamiento de la inteligencia humana teniendo como objetivo principal la simulación de la misma, para lo cual se vale del diseño de máquinas

inteligentes que logren desarrollar las actividades intelectuales (manipulación, razonamiento, percepción, aprendizaje, creación, etc) realizadas por el cerebro humano.

3.1.2 Subcampos de la IA

a) Sistema de lenguaje natural

Es una tarea de la IA que estriba en disponer a la máquina de capacidades ligadas a la comprensión inteligente del lenguaje y facilitar la comunicación entre el hombre y la máquina; es un problema complejo pues intervienen distintos procesos como son: la comprensión del lenguaje, la síntesis y el análisis de la voz, el resumen y la traducción.

b) Sistemas reconocedores de imágenes.

En estos sistemas debe reconocerse el significado de imágenes con ayuda de procesos exactamente definidos. Aún no existen sistemas reconocedores de imágenes de aplicación general. Los procesos desarrollados hasta el momento sólo permiten aplicaciones especiales avanzadas (por ejemplo, robótica, automatismo industrial, análisis de fotografías aéreas, etc).

En la actualidad están apareciendo los primeros lenguajes especializados y se están diseñado los primeros ordenadores de propósito específico.

La programación automática, estudia las formas automáticas de generar programas que resuelvan los problemas planteados cumpliéndose una serie de especificaciones.

Los juegos, resuelve problemas cuya solución no necesita de explicación y justificación, sino que basta con que se cumplan una serie de leyes o reglas del juego.

c) Robótica.

Área en la que se desarrollan procesos mecánicos y repetitivos, y tareas manuales de las cuales es capaz el hombre.

d) Sistemas expertos

Área en la que se simulan los procesos intelectuales llevados a cabo por un Experto Humano durante la búsqueda, la elección y comunicación de una solución a un determinado problema en un campo finito del conocimiento.

Los Sistemas Expertos sólo se comportan de manera aproximada a los Expertos Humanos ya que hasta el momento se desconocen gran parte de los procesos mentales y motrices del cerebro humano.

3.2 SISTEMAS EXPERTOS

La solución de una gran variedad de problemas en el área de ciencia y tecnología es a través del uso de procedimientos basados en la utilización de la computadora, generalmente procedimientos numéricos.

Sin embargo, existe otro tipo de problemas cuya naturaleza de solución no corresponde con los requerimientos de las técnicas de computación convencionales, es decir, la solución no puede ser expresada en términos de un algoritmo numérico.

La aplicación de los Sistemas Expertos es adecuada allí donde los Expertos Humanos dispongan de conocimientos complejos en un área estrechamente delimitada, donde no existan algoritmos elaborados (o donde los existentes no puedan solucionar algunos problemas) y no existan teorías completas.

Otro campo de aplicación es allí donde hay teorías, pero resulta prácticamente imposible analizar todos los casos teóricamente imaginables mediante algoritmos y en un espacio de tiempo razonable.

Por Sistema Experto (SE) se entiende un conjunto de programas y herramientas que imitan el comportamiento del Experto Humano (EH) en el momento de solucionar un problema. Puede almacenar conocimientos de expertos para un campo determinado, y muy limitado, y solucionar un problema mediante deducción lógica de conclusiones.

3.2.1 Tipología de SE.

Dependiendo del problema que se presente, la aplicación de la mayoría de los Sistemas Expertos pueden clasificarse dentro de las siguientes categorías:

a) De interpretación: infiere significados de datos y situaciones a partir de su contexto. Esta categoría incluye la comprensión de mensajes, análisis de imágenes, interpretación de señales, etc. En Geotecnia pueden emplearse estos sistemas en la interpretación de pruebas de laboratorio de todo tipo.

b) De predicción: esta clase de sistemas infiere las probables consecuencia de una situación dada. En Geotecnia pueden emplearse para predecir las propiedades de un suelo con ciertas características.

c) De diagnóstico: identifican las causas que provocan el mal funcionamiento de un sistema a partir de la interpretación de datos previos sobre el funcionamiento, la estructura del proceso y/o sistema a diagnosticar. El campo de aplicación geotécnico es en el diagnóstico de estructuras de cimentaciones ya construidas.

d) De diseño: desarrollan configuraciones que resuelven un problema, a través de procesos iterativos, satisfaciendo sus restricciones. El diseño de cimentaciones, de excavaciones, entre otros, son un campo de aplicación.

e) De planeación: permiten diseñar acciones interrelacionadas para lograr algún objetivo, utilizando ciertos recursos y sujetándose a ciertas restricciones.

f) De reparación: desarrollan y ejecutan planes para corregir fallas de toda índole.

g) De control: esta categoría tiene como tarea el cumplimiento de un objetivo dado, a través de interpretar la situación actual, comparar con la norma, detectar desviaciones, formular el plan de corrección y monitorear su ejecución.

h) De instrucción o tutoriales: utilizados para crear medios lógicos de comunicación didácticos, que permitan interactuar al usuario con el sistema.

3.2.2 Características de los Sistemas Expertos

Se observa que hay un contraste entre las técnicas de programación tradicionales y los SE. Los primeros utilizan cálculo numérico con algoritmos comprensibles, y los segundos, la inferencia o la manipulación de datos es a través de reglas que son, aplicadas sobre la información pudiendo dar conclusiones que no estaban programados en los SE explícitamente, es decir la resolución del problema se basa en el razonamiento simbólico de los datos.

Los SE representan la transición del procesamiento de datos al procesamiento de conocimientos y sustituye los algoritmos por mecanismos de inferencia, teniendo como característica, gracias a su gran capacidad de comunicar, el explicar por qué procede de una u otra forma y el justificar los resultados obtenidos.

Con base en este razonamiento, podemos ver de forma detallada las características generales que posee el SE para realizar inferencias a partir del conocimiento del EH.

1) Adquisición del conocimiento

Los SE actuales en general no son capaces de obtener el conocimiento por sí mismos mediante la práctica, como lo hace un ser humano, si no hay que formalizar el conocimiento del EH para que éste sea aplicable a la solución de los problemas y pueda ser comunicado al usuario del sistema.

2) La heurística

La heurística es un conjunto de reglas empíricas concernientes a una determinada área que el Experto Humano aprende o descubre y que permiten encontrar la mejor solución (teniendo en cuenta el estado del conocimiento sobre el dominio considerado) y no la solución óptima.

3) Dominio del conocimiento

Los SE son especialistas en un ámbito del conocimiento y no en una tarea, como ocurre con los programas tradicionales; sin embargo al igual que los Expertos Humanos los SE sólo dominan un área finita del conocimiento. Mientras que esta limitación en los expertos esta dada por el tiempo necesario para el aprendizaje, en los SE la limitación es de carácter físico manifestándose en el tamaño de la memoria y velocidad de procesamiento que están relacionadas con el tipo de máquina.

4) Resolución de problemas

A un experto le basta con un conjunto de reglas finitas para poder afrontar cualquier problema dentro de su campo. En un SE la estrategia general de resolución se encuentra en relacionar las unidades elementales del conocimiento unas con otras que permitan conocer cual de ellas ha actuado, cuando y por qué (el control del sistema se denomina **motor de inferencia**).

3.3 ARQUITECTURA DE UN SISTEMA EXPERTO

Una característica decisiva de los SE es la separación entre conocimientos (reglas, hechos) por un lado y su procesamiento por el otro; lo que da origen a que la estructura de un SE esté organizada principalmente por los siguientes elementos.

3.3.1 Base de Conocimientos

La base de conocimientos de un SE contiene el conocimiento de los hechos y de las experiencias de los expertos en un dominio determinado. El conocimiento tiene que estar representado de una forma sencilla, con el fin de que se pueda manipular con facilidad, ocupe poco espacio en memoria; a la vez que sea independiente para que su modificación o la exclusión de una unidad de conocimiento no afecte al resto de la base de conocimientos ni al resto del SE. El conocimiento puede ser representado de diferentes formas como se verá más adelante.

3.3.2 Motor de Inferencia

Es el núcleo del SE, ya que pone en acción los elementos de la base de conocimientos que por medio de un conjunto de reglas lógicas infiere nuevos hechos y relaciones que amplían el conocimiento y resuelven el problema planteado.

El motor de inferencia frente a un problema a resolver detecta los conocimientos que tendrán lugar, el orden en que lo harán, y cómo se aplicarán, es decir, determina cómo y cuándo se procesarán las reglas, y dado el caso, también la elección de qué reglas deberán procesarse. Determina a la vez que estrategias de búsqueda se implementarán para estructurar el mejor plan de resolución.

3.3.3 Base de Hechos

La base de hechos es el conjunto de información introducida por el usuario o ya incorporada al sistema, que sirve como datos de entrada y/o condición para solucionar un problema en un dominio específico. Esta base (memoria temporal) no se conserva (salvo por necesidades del usuario) y depende exclusivamente de la situación estudiada.

3.3.4 Módulos de Comunicación

Una forma de lograr la comunicación entre el SE y el usuario es a través de módulos que permitan la entrada de datos y salida de soluciones, así como ayuda en línea que contenga información relevante del funcionamiento del sistema. Los módulos deben ser rápidos, potentes, sencillos y adecuados con el fin de que admitan estructuras flexibles que sean cercanas al lenguaje natural, logrando trabajar en el nivel adecuado de cada uno de los usuarios que empleen el sistema.

a) Módulo del Usuario

Es el que gobierna el diálogo entre el usuario y el sistema de forma sencilla y aproximándose lo más posible a un lenguaje natural.

Este módulo comunica al motor de inferencia las consultas del usuario, y a este último, los resultados de la consulta.

Dentro de las tareas más importantes que desempeña este módulo están: entrada de datos, salida de explicaciones, salida de justificaciones, salida de soluciones.

b) Módulo del Experto

Este módulo mantiene una comunicación entre el experto y el programador, con el objeto de que este último pueda recibir el conocimiento, verificar la verosimilitud de éste, organizar la coherencia de la base de conocimientos y transformar los datos en conocimientos incorporados al sistema, es decir, éste módulo permite el mantenimiento, validación y configuración del sistema.

El ingeniero de conocimiento deberá estar algo familiarizado con el componente de adquisición de conocimientos para poder realizar modificaciones sencillas por sí solo.

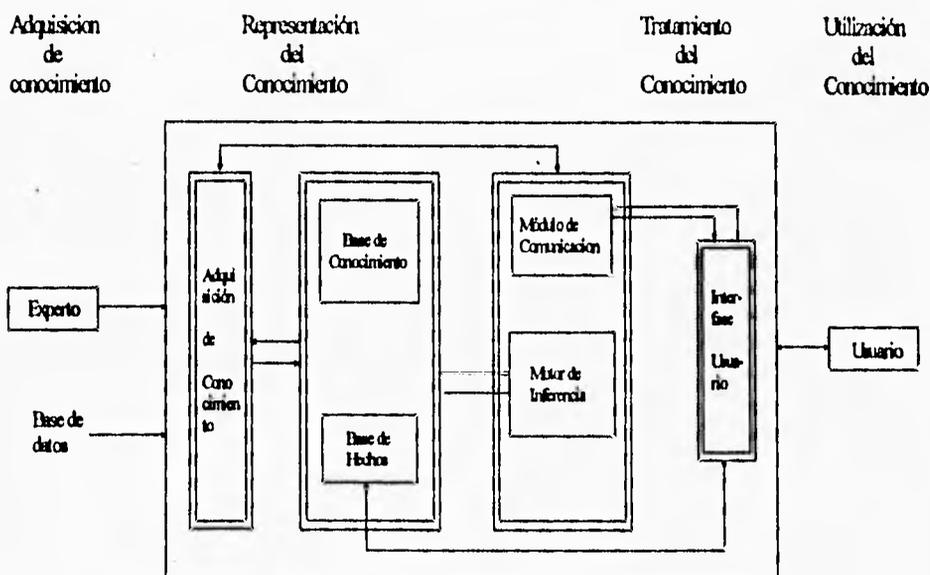


Fig 3.1 Arquitectura de un SE

Una representación gráfica de los componentes de un SE es la que se presenta en la figura 3.1, donde se puede observar la estrecha vinculación que existe entre la Base de Hechos, la Base de Conocimiento y el Motor de Inferencia.

3.4 REPRESENTACIÓN DEL CONOCIMIENTO

Para el procesamiento y la manipulación del conocimiento un SE es necesario que éste se encuentre formalizado y estructurado.

La formalización del conocimiento debe estar sustentada bajo criterios fijos que lo hagan sencillo y objetivo. Por ello se han desarrollado procedimientos de representación del conocimiento que puedan ofrecer un apoyo eficiente a la estructuración y al procesamiento del saber. Estos conocimientos crean fragmentos de conocimiento independientes unos de otros que, por consiguiente, son fácilmente modificables. Los conocimientos se combinan después, mediante un mecanismo general de razonamiento y deducción.

3.4.1 Reglas de producción

La forma más comprensible de representación del conocimiento se basa en reglas de producción.

Una regla consta de un conjunto de acciones o efectos que son ciertos cuando se cumplen un conjunto de condiciones o causas. Una regla traduce una relación a una acción condicional.

Las reglas de producción están redactadas de tal manera que cada regla ignora la existencia de las demás y por lo tanto no se activan por su nombre, sino por sus condiciones de aplicabilidad. Ello permite añadir o suprimir reglas sin preocuparse de los efectos de estas modificaciones.

Una regla de producción tiene generalmente la siguiente forma:

si < condiciones > ENTONCES < conclusiones o acciones >

La parte izquierda expresa las condiciones de aplicabilidad de la regla. La conclusión se suele referir a la creación de un nuevo hecho válido, mientras que la acción suele referirse a la transformación de un hecho.

Un buen mecanismo de inferencia destacará por sus eficientes estrategias (de solución de problemas) para la elección de una regla a partir de una serie de posibles reglas. Este conocimiento acerca de las reglas de producción se denomina metarreglas. Cuando la base de conocimientos está construida en forma modular las metarreglas facilitan el trabajo, pues pueden inhibir parte de los módulos lo que hace más rápida la búsqueda de soluciones.

Entre las ventajas que presentan las reglas de producción están su sencillez y su independencia que permite modificaciones sin que se vea afectado el resto de la Base de Conocimientos.

El principal problema de las reglas de producción reside en el crecimiento muy rápido del número de reglas lo que relentiza el proceso de inferencia y la gran facilidad con que pueden producirse contradicciones.

3.4.2 Redes semánticas

Las redes semánticas son un método de representación del conocimiento sobre las relaciones de los objetos. Los nudos de una red semántica corresponden a conceptos (objetos o sucesos) y las ramas describen relaciones entre estos conceptos, estas relaciones pueden ser de herramientas o de descripción.

Al igual que en las reglas de producción se logra la inferencia de nuevos hechos, esta inferencia se realiza por seguimiento de los enlaces o por la acción de las herencias.

Las ventajas de las redes semánticas son su potencia al momento de definir relaciones; sus desventajas son su poca flexibilidad, lo que dificulta las modificaciones y la lectura es compleja cuando la base de conocimientos es grande.

No es recomendable la construcción de SE únicamente mediante redes semánticas; sino utilizando conjuntamente reglas de producción.



Fig 3.2 Red semántica

3.4.3 Marcos

Las formas de representación de conocimiento que se acaban de mencionar se destacan por el aspecto declarativo del conocimiento así como por su naturaleza inferencial. Cuando los conocimientos no son inferenciales (por ejemplo variantes sobre un prototipo), la mejor forma de representar el conocimiento es mediante objetos estructurados. Esta representación se puede considerar como una extensión de las redes semánticas.

Los marcos, que corresponden a los objetos estructurados, son estructuras de datos que sirven para representar una situación estereotipada. Un marco es la división de situaciones, en sus componentes; las características de los componentes son introducidos en ranuras (Slots) las cuales guardan ciertas relaciones de herencia entre ellos y pueden contener valores y procedimientos para inferir estos valores al igual que en la red semántica.

Los marcos son modulares por naturaleza y admiten la representación del conocimiento en forma declarativa o procedimental.

Un marco puede llenarse por defecto, es decir que para aquellos valores que no se definen de forma explícita se toma el valor de la ranura modelo o se infiere de otra.

Conferencia

Fecha		
Lugar		
Tema		
Participante		

ranura Valor de la ranura

Fig 3.3 Representación del conocimiento en marcos

3.5 ESTRATEGIAS DE CONTROL

Como se ha mencionado, el motor de inferencia es la parte central del SE; es el que alimentado por la base de conocimientos, construye dinámicamente el razonamiento, diciendo qué reglas se activan y en qué orden; para ello se consideran ciertas estrategias de búsqueda y de elección de conocimiento.

3.5.1 Mecanismo de búsqueda

Entre las estrategias de búsqueda de soluciones existen dos posibilidades.

a) La búsqueda no ordenada de soluciones.

En esta estrategia existen dos tipos de búsqueda aleatoria y heurística; la primera es de forma exhaustiva y la segunda, la base de conocimientos se divide en módulos, buscando solamente en alguna de estas partes según lo que indique el metaconocimiento.

b) La búsqueda ordenada de soluciones.

La búsqueda ordenada se basa en el encadenamiento del conocimiento, esto es, cuando la evaluación lógica de una regla se convierte en el antecedente de la siguiente regla a considerar.

Las estructuras de control que manipulan el razonamiento son:

Encadenamiento hacia adelante (afirmando el antecedente se afirma el consecuente): El motor de inferencia parte de los hechos para llegar a los resultados, seleccionando el conjunto de reglas aplicables según las condiciones del problema; este razonamiento es práctico en casos de tipo numérico.

Encadenamiento hacia atrás (negando el consecuente se niega el antecedente): El motor de inferencia parte del objetivo y trata de volver a los hechos y conocimientos con el fin de comprobar el objetivo. En este tipo de encadenamiento las soluciones deben ser conocidas de antemano.

Encadenamiento mixto: es el más empleado en la construcción de los SE. Consiste en buscar un conjunto de soluciones mediante un encadenamiento hacia adelante, tras lo cual mediante un encadenamiento hacia atrás se verifican estas soluciones.

El uso de encadenamiento mixto puede provocar búsquedas infinitas por lo que hay que dotar a estos sistemas de detectores de este tipo de falla.

3.5.2 Estrategias de selección de reglas

a) Metaconocimiento

El metaconocimiento es la parte del SE que le indica al motor de inferencia que conocimiento debe ser elegido y aplicado en cada momento asegurando que éste se ha aplicado correctamente.

El metaconocimiento se puede presentar en cuatro formas:

Fijo y Ciego: el metaconocimiento está sumergido en el propio motor de inferencia.

Seccionable antes de la ejecución del programa: el usuario elige entre una serie de alternativas las más cercanas a las características del problema a resolver.

Seccionable durante la ejecución del programa: el sistema preguntará al usuario características secundarias del problema a medida que va obteniendo resultados parciales y de esta manera redireccionar la búsqueda si es necesario.

Externo: el SE permanece en modo interactivo durante todo el proceso con el usuario.

b) Metareglas

Las metareglas (que expresan un metaconocimiento) permiten guiar la elección de las reglas durante la resolución de un problema, traduciendo la estrategia del experto. Permiten seguir una estrategia dinámica que evoluciona en función del estado del sistema.

Existen dos clases de metareglas:

Metareglas ciegas: contienen conocimiento sobre la estructura de las reglas.

Metareglas no ciegas o inteligentes: contienen conocimientos sobre el contenido de las reglas.

3.5.3 Lógica

La lógica es una forma de representar la realidad que permite determinar la veracidad o falsedad de los resultados con base en una serie de hechos o axiomas de partida, es decir, permite realizar desarrollos concretos.

El motor de inferencia puede utilizar diferentes clases de lógica para toma de decisiones, entre ellas existen:

Según el tipo de variables

Lógica booleana: admite sólo dos valores (verdadero y falso).

Lógica trivalente: es aquella en la que se admiten tres valores (verdadero, falso o neutro.)

Lógica multivalente: las variables pueden tomar muchos valores discretos.

Lógica difusa o borrosa: es una generalización de la lógica multivalente y permite la toma de valores continuos entre dos valores límite.

Según el tipo de conocimientos

Lógica monótona: es aquella en la que los nuevos conocimientos no pueden entrar en contradicción con los ya existentes.

Lógica no monótona: es utilizada cuando se emplean conocimientos parciales o incompletos y admite la incorporación de nuevos conocimientos a la base de hechos, aunque entre en contradicción con los ya existentes.

Según la temporalidad

Lógica atemporal: no se considera la influencia del tiempo en los hechos.

Lógica temporal: Los hechos van cambiando en el transcurso del tiempo.

Según el tipo de valores

Lógica de orden 0: admite proposiciones, representando condiciones Objeto-Valor.

Lógica de orden 0+: las condiciones son representadas como Objeto-Atributo-Valor.

Lógica de orden 1 o lógica de predicados: incorpora en la representación del conocimiento un nuevo valor que es la incertidumbre.

3.5.4 Evaluación del conocimiento

El motor de inferencia puede evaluar el conocimiento de diferentes maneras. Una vez que el motor de inferencia ha determinado que parte de la base de conocimientos se va a utilizar para resolver el problema en cuestión, prosigue a evaluar este conocimiento.

La evaluación puede ser:

Determinista. Las condiciones son determinantes en el resultado total.

Probabilística. Es necesario conocer previamente las probabilidades de cada una de las hipótesis respecto a cada uno de los hechos.

Aproximada. Cuando los resultados dependen del grado de veracidad de los hechos y conocimientos utilizados.

Por analogía. En la que las propiedades de un hecho las adquiere otro por similitud.

Hereditarias. Cuando un hecho adquiere las propiedades del hecho que lo procede.

3.6 LENGUAJES Y SISTEMAS DE PROGRAMACIÓN

Los Sistemas Expertos pueden ser desarrollados en cualquier lenguaje de programación, desde un lenguaje ensamblador hasta un sistema desarrollado.

Existen varios niveles de lenguajes especializados (que se describirán a continuación) y el uso de cualquiera de estos no significa de ninguna forma que el programa sea un SE, es decir no es la herramienta la que otorga las características de un SE sino la estructura del programa. El programa desarrollado será finalmente procesado en forma de código de máquina que el microprocesador ejecutará secuencialmente; por lo tanto desde cualquier lenguaje de programación se pueden construir SE con mayor o menor facilidad.

Dentro de los diferentes niveles de lenguajes existen los siguientes.

Lenguajes Imperativos

Los lenguajes imperativos, son aquellos en el que el control del programa pasa siempre a la siguiente línea del programa salvo que se le ordene lo contrario. El flujo de la ejecución de las instrucciones es definida por el programador mediante etiquetas.

Dentro de este grupo de lenguajes están:

- El BASIC, por su sencillez
- El PASCAL por su estructuración
- El C por su portabilidad y alto rendimiento.

Lenguajes Funcionales

Estos lenguajes son aquellos donde el flujo del programa viene marcado por las necesidades que aparecen al evaluar una función.

La realización de un motor de inferencia resulta más sencilla a partir de un control funcional que a partir de un control de tipo imperativo.

Los lenguajes más conocidos son el LISP y el LOGO.

Lenguajes Orientados al Objeto

La programación orientada a objetos se caracteriza porque no existe distinción entre los procedimientos y los datos.

Los programas están formados por los objetos que son a la vez los procedimientos (comportamientos, que pueden ser algoritmos) y los datos (facetas); los objetos envían los mensajes a otros objetos. Cada objeto interpreta el mensaje que le llega y estos heredan las propiedades del prototipo que lo han originado.

Los objetos tienen una serie de atributos y cada uno de estos posee facetas, como pueden ser los valores permitidos.

El control en un lenguaje orientado a los objetos es el siguiente: entrega los datos, se propone una serie de hipótesis, se ordenan las hipótesis, se verifican, y si no queda ninguna más por verificar se propone una solución con base en las hipótesis válidas. Durante la evaluación del programa, se pueden crear y modificar los objetos, siendo esto, punto importante para el aprendizaje de los programas.

El ser esta una nueva forma de programación crea un inconveniente en la filosofía de los programadores, creando en algunos de ellos dificultad para programar.

Lenguajes Declarativos

Se describe el objeto que queremos demostrar, especificando en el programa los conocimientos sobre el que debe demostrarlo y las reglas que puede utilizar.

Este lenguaje representa el conocimiento en forma de reglas de producción y cuenta con un motor de inferencia que es el propio interprete o compilador. Los lenguajes más conocidos en la actualidad son el PROLOG y el OPS.

Estos lenguajes son muy compactos, al no necesitar ninguna estructura de control.

Dentro de los inconvenientes podemos citar.

- Las operaciones de entrada/salida son algo complejas.
- Poco fiable al no aparecer el comportamiento del programa, pues el motor de inferencia está inmerso dentro del programa.

Sistemas desarrollados

Los sistemas desarrollados; se forman por un conjunto de modos de representación del conocimiento y motor de inferencia incluido programable o parametrizable, lo que facilita la construcción de SE.

Si se parte de este nivel en el desarrollo de un SE, no es necesario realizar las etapas de tratamiento simbólico, construcción del motor de inferencia, detallar la justificación y explicación de los procesos necesarios para llegar a una solución, sólo es necesario realizar la representación del conocimiento con base en reglas y determinar las relaciones entre los módulos.

La mayoría de los entornos se fabrican de tal modo que los usuarios se familiaricen con ellos rápidamente. Sus principales ventajas son su gran rapidez de desarrollo, así como la comodidad y sencillez que ofrece su manejo, cuenta con una serie de dispositivos que se usan en los momentos que se requiera (demostraciones, editores, compiladores).

Las desventajas más sobresalientes son su gran rigidez y su poca portabilidad.

Los sistemas vacíos son programas a los que únicamente se les introduce el conocimiento para hacerlos operativos, por lo que es posible desarrollar un SE sin conocimientos profundos de programación.

La desventaja principal es su nula flexibilidad, lo que obliga a comprar productos específicos para cada requerimiento.

La clasificación de los entornos desarrollados y sistemas vacíos es difícil de establecerse. Una manera podría ser el grado de dificultad que presenta para los usuarios que pretenden construir un SE.

La mayoría de estas herramientas están programadas en LISP, lenguajes imperativos y un reducido grupo de ellos está desarrollado en PROLOG.

3.7 METODOLOGÍA DE CONSTRUCCIÓN DE SISTEMAS EXPERTOS

a) Estudio de la demanda

En esta etapa se determina el porqué de la construcción de un sistema experto, se estudia y clasifican las características del problema a resolver. Entre los puntos de partida más importantes en la toma de decisiones se encuentran:

- Construir un SE con el fin de obtener experiencia de un grupo de expertos que permitan formar a especialistas en un campo determinado del conocimiento.
- La necesidad de automatizar la resolución de problemas en un área del conocimiento que por medio de las técnicas tradicionales no es posible representar.
- Cuando el conocimiento evoluciona rápidamente que con las técnicas tradicionales de programación no es posible incorporar modificaciones de tal manera que no modifiquen profundamente el comportamiento.
- Cubrir la demanda de usuarios no expertos en un área del saber.

b) Análisis del problema

En esta fase se determina si es posible resolver el problema y plantear los objetivos que se quieren alcanzar con el desarrollo del SE para así determinar el campo en el que se desea la construcción del mismo, las tareas y funciones que desempeñará el sistema, según las características del problema, así como el tipo de conocimiento a utilizar.

Evaluación de la posible aplicación.

- El problema debe involucrar procesamiento simbólico significativo, complejidad, juicios e incertidumbre.
- El problema debe tener solución, y ésta, reportar beneficios a un grupo de usuarios.
- El conocimiento empleado no está contenido en libros o manuales. No existe un método definido y único que solucione el problema.
- No existen ecuaciones que definan su comportamiento, ni modelos estadísticos que se adapten.
- Debe ser un problema concreto, con el fin de que el conocimiento requerido para su resolución sea limitado, es decir que puedan ser definidas las fronteras en la base de conocimiento.

Algunas de estas condiciones pueden no tomarse en cuenta cuando el proyecto tiene una aplicación concreta, como lo es la difusión de conocimientos recientes (tutoriales), o por el grado de desarrollo en el que se encuentra la investigación a la que se desea aplicar el SE.

c) Elección del conocimiento

El sistema debe ser construido de una fuente de experiencia; por ello es necesario primero identificar los expertos que estén dispuestos a aportar sus conocimientos; una vez que se haya extraído el conocimiento del experto y de las fuentes escritas, el ingeniero de conocimiento se familiarizará con él para pasarlo por escrito y posteriormente representarlo en el soporte seleccionado.

La adquisición del conocimiento es una de las fases más complejas, requiere de intercambios constantes de conocimientos entre el experto y el ingeniero de conocimiento lo que implica frecuentes marchas atrás. El ingeniero de conocimiento debe contar con una buena metodología y sobre todo, gran habilidad y experiencia para obtener del experto información concreta.

De la literatura se pueden precisar siete pasos a cubrir en esta etapa.

- 1) Familiarización del ingeniero de conocimiento con el problema
- 2) Delimitar el problema.
- 3) Determinar entre los expertos disponibles los más idóneos para la aportación del conocimiento.

- 4) Una vez identificados los expertos, motivarlos en el desarrollo del SE desde el comienzo de su elaboración hasta el final de la misma.
- 5) Plantear de forma general la resolución del problema con las aportaciones del grupo de expertos.
- 6) Resolver dudas sobre el contenido de la base de conocimientos. Obtener conocimiento al detalle.
- 7) Completar y corregir la base de conocimientos sobre el soporte del SE; este paso se realiza normalmente sobre el prototipo.

d) Selección del soporte

Una vez que se tienen los conceptos claves y las relaciones entre ellos, se trata de formalizar el conocimiento con la ayuda de algún tipo de herramienta. El tipo de herramienta muchas veces es impuesto por el análisis del problema.

Entonces es aquí donde el ingeniero de conocimiento determina las características del motor de inferencia, del lenguaje y tipo de herramienta a utilizar (reglas de producción, marcos) para la representación del conocimiento que le parezcan corresponder al problema planteado.

e) Construcción del prototipo del SE

El prototipo del SE permite probar la validez del conocimiento y, por consiguiente, de la herramienta elegida. El prototipo facilita encontrar en menor tiempo los errores existentes en el momento de llegar a una solución del problema en cuestión.

f) Validación del prototipo

En este punto se detectan los errores que pudieran existir en el SE. Para ello se enfrenta el SE contra el experto humano. Este último estudia el comportamiento del sistema frente a una gama de problemas comparando la resolución que da con la que él realizaría.

En efecto, la fase de validación puede provocar revisiones en los diferentes componentes del prototipo, principalmente en la base de conocimientos, si los resultados entre el SE y los expertos no son los esperados.

g) Construcción final del SE

Una vez que sea revisado y encontrado los inconvenientes del prototipo, se hace un refinamiento de éste, modificando la base de conocimientos y al mismo tiempo la estructura de control (si esto es posible).

El resultado de estas modificaciones debe producir un mejoramiento en el funcionamiento del SE, sin que esto signifique que las modificaciones sean drásticas.

En la figura 3.4 se presenta un diagrama de bloques de la metodología de construcción propuesta.

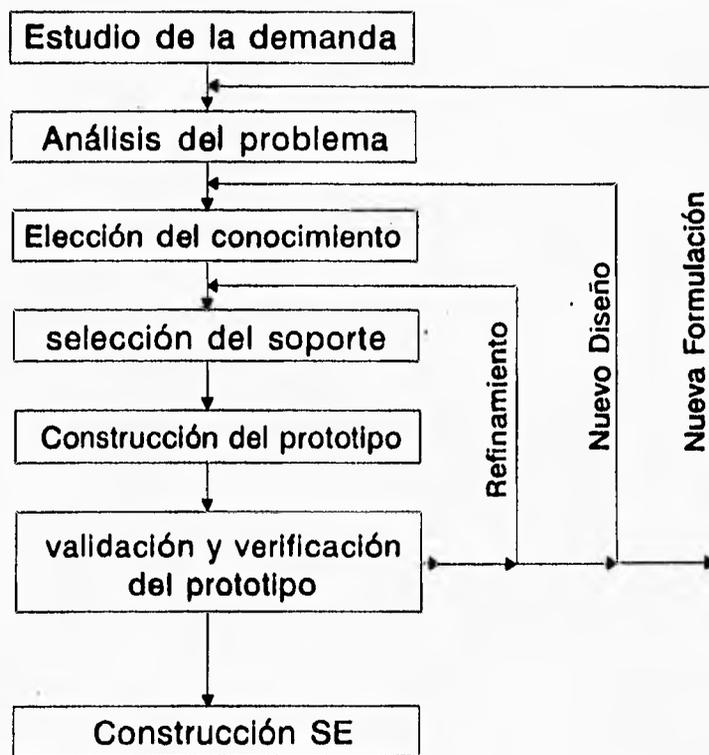


Fig 3.4 Ciclo de desarrollo de un SE

3.8 DESARROLLO DEL SISTEMA EXPERTO "SEANEX"

a) Estudio de la demanda.

El análisis de movimientos en excavaciones profundas con caras verticales requiere de la automatización de una serie de pasos aislados que hacen el proceso de análisis un tanto engorroso.

La construcción del SE no tratará de sustituir al experto o de mantener usuarios y expertos alejados entre si, sino más bien se trata de poner en las manos de los usuarios una herramienta efectiva que libere a los expertos de los trabajos rutinarios y proporcione a los usuarios no expertos la posibilidad de conocer en forma constante los conocimientos de los expertos en esta área.

El contar con un medio automático para la formación del personal, repercutiría indudablemente en liberar parte del tiempo que el EH destina para tal propósito.

b) Análisis del problema

El problema planteado es posible resolverlo con la construcción de un SE que por un lado funcione como un asistente del usuario para la toma de decisiones y por otro comunique nuevos conocimientos.

También se lograría automatizar gran parte de la metodología empleada, que a pesar de tener características de programación tradicional, el tipo de información que se maneja requiere de un control asistido por el usuario.

Objetivos del sistema

- Automatización de la mayor parte de la metodología utilizada en el análisis de excavaciones, reduciendo substancialmente el tiempo requerido para ello.
- Crear un medio de comunicación entre el experto y el usuario con el fin de que este último se socialice con los conocimientos existentes y al mismo tiempo ayude a deducir nuevos datos. El sistema debe funcionar como asistente del usuario en la toma de decisiones.

Alcance del problema

El SE a desarrollar se encuentra en el campo educativo y de diseño, éste deberá permitir reducir el tiempo de análisis, así como brindar apoyo a los usuarios de los métodos que se emplean actualmente para estudiar los movimientos en excavaciones profundas a cielo

abierto. El usuario tendrá beneficios en la utilización del SE porque forjará en él criterios para el diseño de excavaciones, cumpliendo así uno de los objetivos de la parte didáctica. Otro de los alcances del sistema es lograr que la resolución del problema se vaya procesando por partes para que brinde al usuario la oportunidad de validar los resultados obtenidos.

En la construcción de un SE el entorno de programación es muy importante debido a las constantes modificaciones que se deben realizar al mismo, por este motivo es necesario que pueda ponerse en marcha el SE sin la necesidad de contar con el ambiente de programación y utilizando solamente comandos del MS-DOS.

Que los procesos lógicos puedan ser reutilizados en la construcción de otros sistemas expertos.

Formulación particular de la solución

La estructuración del sistema mediante módulos permitirá dar solución al problema por pasos. Se propone que la solución al problema sea mediante módulos para que el crecimiento del SE sea posible sin la alteración de las demás partes que lo componen. Entonces se proponen:

- Un módulo que cuente con información general del método que sustenta la solución.
- Otro módulo que permita al usuario especificar al sistema las condiciones a las cuales está sujeto su problema.
- Un módulo que presente los resultados gráficamente de una forma amigable, que permita al diseñador dar soluciones concretas al problema.
- Por último, un módulo principal que ligue los demás módulos.

c) Elección del conocimiento

El desarrollo de esta fase comprendió la recopilación de información tanto de Expertos Humanos en el área involucrada al problema, como la consulta de fuentes escritas relacionadas con el análisis de excavaciones y de Sistemas Expertos.

En lo que se refiere a las fuentes escritas la información es bastante escueta y no proporcionan un método exacto que definen a ciencia cierta el análisis de excavaciones. En lo que se refiere al conocimiento del elemento finito la información es abundante y no hubo problema.

En cuanto al conocimiento extraído del experto no hubo problema alguno y, se consideró, fue el de mayor nivel, lo que significó una ayuda importante para la construcción del sistema.

d) Selección del soporte

Dentro de los requerimientos que se presentan para conocer el tipo de herramienta requerido por el motor de inferencia se tomaron en cuenta los siguientes aspectos.

La representación del conocimiento se propone mediante reglas de producción (causa-efecto), el manejo de las reglas requiere utilizar lógica de tipo booleana y atemporal, donde éstas sólo admiten dos valores (verdadero o falso) y el tiempo de acción no se considera, es decir, si un hecho toma un valor verdadero o falso lo es para siempre.

El mecanismo de búsqueda es preciso que se realice con una búsqueda en profundidad y encadenamiento hacia adelante, porque dependiendo de los hechos considerados se desprenderán soluciones.

El tipo de soporte a emplear queda definido fundamentalmente porque se requiere que el SE sea lo más autónomo del entorno de programación

El lenguaje considerado fue TURBO PASCAL.

TURBO PASCAL versión 7 (Borland Pascal) es un lenguaje de tipo estructurado que aunque el orden de ejecución determina el flujo de la solución, su estructura permite un diseño modular.

La programación estructurada utiliza un número limitado de estructuras de control que minimizan la complejidad de los programas, y por consiguiente, reduce los errores; hace los programas más fáciles de escribir, verificar, leer y mantener, dando características de un lenguaje de alto nivel.

El lenguaje cuenta con algunas características de programación denominada al objeto, que permitirá según crezcan las necesidades, convertir los programas fuentes a esta forma de programación.

Considerando que la programación en TURBO PASCAL permite:

- 1) Programación de tipo modular
- 2) Manejo de lógica booleana
- 3) Es relativamente sencillo crear interfases de comunicación con el usuario
- 4) Puede controlar pantallas y crear ventanas
- 5) Es capaz de utilizar gráficos y animar imágenes
- 6) El ambiente de programación y el de compilar pueden trabajar de manera conjunta, reduciendo drásticamente los tiempos necesarios para compilar y depurar programas.
- 7) Es posible crear archivos ejecutables e independientes del entorno de programación
- 8) Es posible construir interfases con programas codificados en otro lenguaje.

Se optó por utilizar TURBO PASCAL versión 7 ya que uno de los objetivos de la construcción del SE y de mayor peso, se refiere a lograr la mayor independencia del sistema del entorno de programación.

En el transcurso del desarrollo del SE se encontraron algunos inconvenientes con el lenguaje utilizado.

El motor de inferencia se construyó a sabiendas que quedaría inmerso en el cuerpo del sistema y muy difícilmente podría ser reutilizado en otro sistema.

La interfase que se requería realizar con un programa codificado en Fortran 77 no se pudo llevar a cabo desde la PC, puesto que este programa requiere de mayor memoria para ser ejecutado que la que ofrecen estas máquinas.

Se optó por trabajar desde el sistema operativo UNIX para lograr que la interfase estuviera contemplada desde el SE. Entonces se pensó en utilizar el lenguaje C, sin embargo no se contaba en esos momentos con librerías de gráficas cargadas en la estación de trabajo, lo que implicaría detener el desarrollo del sistema hasta que se dominara el lenguaje y se instalarán las librerías, por lo que se decidió continuar trabajando con TURBO PASCAL puesto que ya se dominaba gran parte de la estructura del lenguaje.

En lo que respecta a la base de conocimiento necesaria para construir el prototipo del SE, es posible manejarla bajo el lenguaje seleccionado debido a sus características de modularidad, pudiendo finalmente ensamblar los programas desarrollados bajo un programa principal.

El tipo de estructura del lenguaje permite que los módulos del sistema se vayan perfeccionando con el tiempo y/o se vaya ampliando el número de estos.

Otro soporte (lenguaje Prolog) fue considerado, pero no fue evaluado porque los antecedentes que se tenían de él demostraban desventajas en el momento de hacer interfase con programas codificados en Fortran 77.

Además como se pretende que con el tiempo alguno de los módulos pueda ser aumentado y/o corregido según las necesidades del usuario, el lenguaje utilizado debe ser tal que tenga características universales para los geotecnistas, ya que la mayoría de las veces se conoce más este tipo de lenguaje y es más fácil de aprender que los lenguajes funcionales y declarativos.

e) Construcción del prototipo

El prototipo SEANEX (Sistema Experto para Analizar Excavaciones) se estructura prácticamente en varias fases sucesivas de ejecución, cada una de ellas brinda información procesada (resultados) para la ejecución de la fase subsiguiente hasta la toma de decisión final.

El SE, a través de una comunicación con el usuario, obtiene de éste toda la información necesaria para la formulación del problema.

Las funciones operativas que realiza SEANEX para el procesamiento de información, se encuentran agrupadas en tres módulos.

Para que la comunicación entre el usuario y el sistema resultara lo más amena posible se utilizaron pantallas, ventanas y teclas especiales, así como explicación en línea.

1) Módulo Explicativo

El módulo explicativo tiene la intención de ilustrar al usuario cuales son los aspectos más relevantes que se deben tomar en cuenta en el análisis de desplazamientos de una excavación utilizando el Método del Elemento Finito, así como presentar las teorías existentes sobre el diseño de excavaciones y sus limitaciones.

Este módulo se llevó a cabo con la manipulación de archivos de textos en ASCII, estos son invocados desde SE, el cual activa una serie de instrucciones que logran ceder el control al usuario para que él mismo vaya consultando las diferentes partes del texto según sus necesidades de consulta. El usuario puede escapar de la consulta en el momento que lo desee con sólo activar la tecla ESC.

2) Módulo de Alimentación

El módulo de alimentación tiene como objetivo generar las mallas de elementos finitos y posteriormente generar un archivo con las características del problema en estudio. Los archivos generados por el sistema son leídos por el programa de computadora de elementos finitos (TEST92) para obtener estados de desplazamientos.

En este módulo existen cuatro opciones. Las tres primeras sirven para generar las mallas de elementos finitos y la última para detallar las características del problema en cuestión.

La generación de mallas se realiza por tres diferentes maneras:

1) (Rectangulares y proporcionales). Se generan elementos proporcionales con tan sólo introducir las coordenadas de frontera. La malla generada puede ser modificada

posteriormente por el usuario por medio de tres opciones: modificar, insertar y/o borrar líneas.

2) (Rectangulares no proporcionales). Se determina la posición en X y Y de los elementos que conforman la malla; para esto se debe contar con un bosquejo general de dicha malla, después el usuario puede refinarla con las opciones mencionadas arriba, según los requerimientos de éste.

Para indicar la posición de los elementos, existen dos opciones, por medio de la lectura de un archivo de datos o tecleando los datos directamente desde una ventana.

3) (Generador). La malla es generada por medio de un programa que es llamado por el SE y permite construir diferentes geometrías.

Por otra parte, para definir las características del problema a resolver se cuenta con una serie de tarjetas de datos que se presentan en ventanas, permitiendo que el usuario introduzca la información de una forma rápida y amena.

Las características introducidas son del tipo:

- Geométricas.
- De carga.
- De material.
- De frontera de análisis.

Es importante tomar en consideración que cada vez que el usuario haga una corrida, el archivo nuevo debe ser asignado con diferente nombre al anterior, de lo contrario la información precedente será borrada automáticamente.

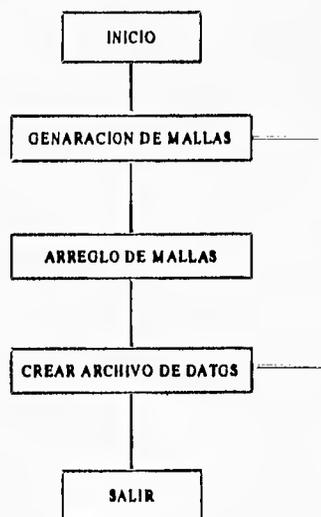


Fig 3.5 Diagrama de bloques del módulo de alimentación

3) Módulo de Graficación

El módulo de graficación permite preparar la información resultante de la ejecución del TEST92, esto es, el módulo selecciona y ordena los datos necesarios para la graficación de resultados. Una vez que la selección se realizó, el módulo grafica los resultados siguientes:

- a) Asentamiento de colindancia
- b) Desplazamientos laterales
- c) Malla deformada

Los desplazamientos se realizan considerando que el TEST92 permite simular una excavación por etapas; además los desplazamientos graficados son propios de una excavación profunda con paredes verticales.

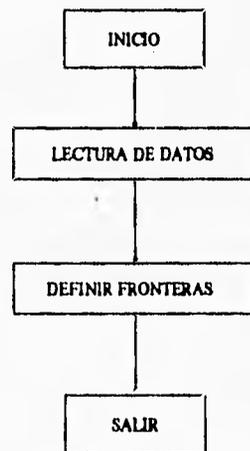


Fig 3.6 Diagrama de bloques del módulo de graficación

4) Módulo Principal

El módulo constituye la unión de los demás módulos. Cuenta con la capacidad de incorporar nuevos módulos en el momento que estos se desarrollen.

El módulo principal cuenta con una serie de archivos de textos y ejecutables, que son necesarios para que funcione el sistema, de lo contrario si algún archivo no está presente el sistema detectará el error y dará un aviso.

Cabe mencionar que cada uno de los módulos puede ser llamado desde el MS-DOS.

V. Los programas fuentes de cada uno de estos módulos se presentan en los anexos II al

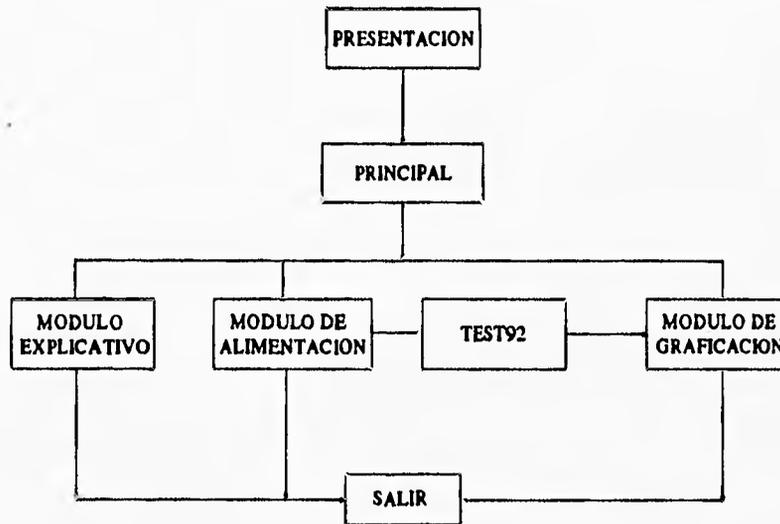


Fig 3.7 Diagrama de bloques del módulo principal

f) Validación del prototipo

Cuando el prototipo fue probado y revisado surgieron algunos inconvenientes en cuanto a los resultados finales que arrojaba, que posteriormente fueron corregidos sin alterar la parte esencial del sistema propuesto.

4. APLICACIÓN A UN CASO PRÁCTICO

Existe un gran número de obras de infraestructura (metro, drenes, colectores, interceptores, etc) que son alojadas en excavaciones a cielo abierto superficiales o profundas, en las cuales las condiciones de estabilidad dependen del procedimiento constructivo durante y después de la construcción e inclusive hasta pueden provocar condiciones de falla, por esto es importante evaluar el estado de esfuerzos y deformaciones al que está sujeta la excavación y las zonas aledañas a ella.

Un ejemplo típico donde el comportamiento de las excavaciones profundas a cielo abierto tiene suma importancia es en la construcción de cajones subterráneos para el sistema de transporte colectivo (metro); donde la predicción de las deformaciones causadas por excavación proporcionará una herramienta muy importante para la prevención de daños.

La localización de los sondeos realizados, así como las características estratigráficas y algunas de las propiedades de los suelos en el sitio del cruce se muestra en las figuras 4.3 a 4.4. El nivel freático se localiza a 2.50m del nivel del terreno natural y la primera capa dura se ubica a 36m. La arcilla tiene una cohesión promedio de $2.5t/m^2$ y un contenido natural de agua promedio de 300%. se puede apreciar lentes de arena intercalados en la formación arcillosa a diferentes profundidades y de varios espesores.

La excavación del suelo entre los muros Milán, previamente instalados, se lleva a cabo usualmente en talud. Sin embargo, en este caso los análisis de estabilidad de taludes y los cálculos de falla de fondo mostraron que para las condiciones existentes este procedimiento de excavación presentaba alto riesgo. Por tal motivo, se evaluó la alternativa de excavar a cielo abierto celdas formadas con muros transversales cuya separación entre sí varió en esta zona entre 6 y 8.5m.

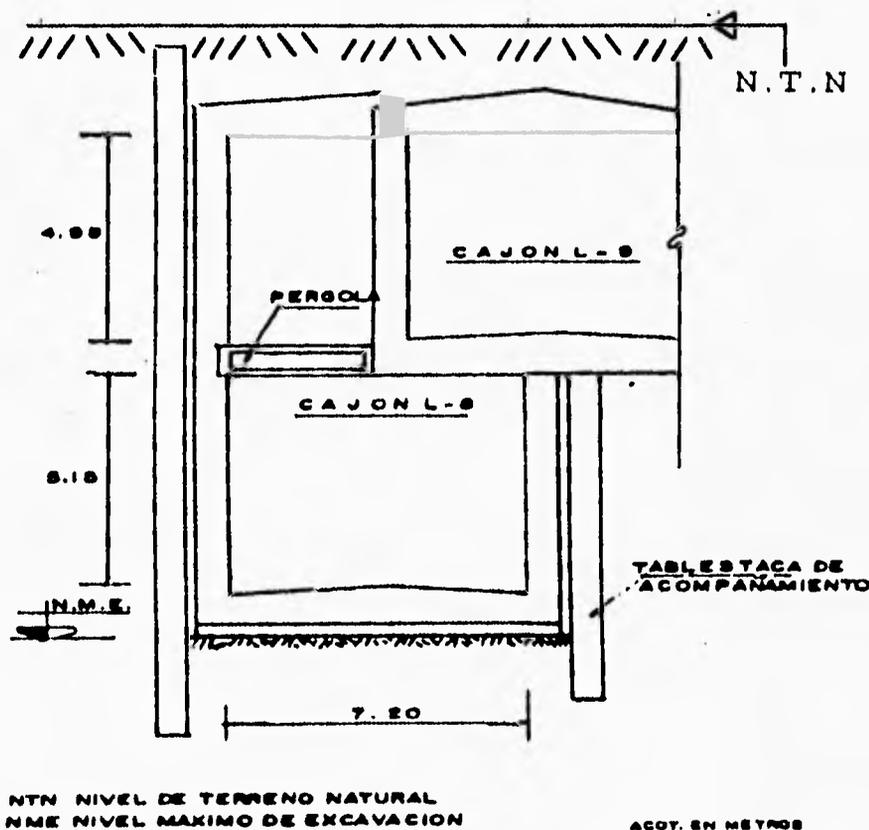


Fig 4.2 Cajón en cruce de L-8 con L-9

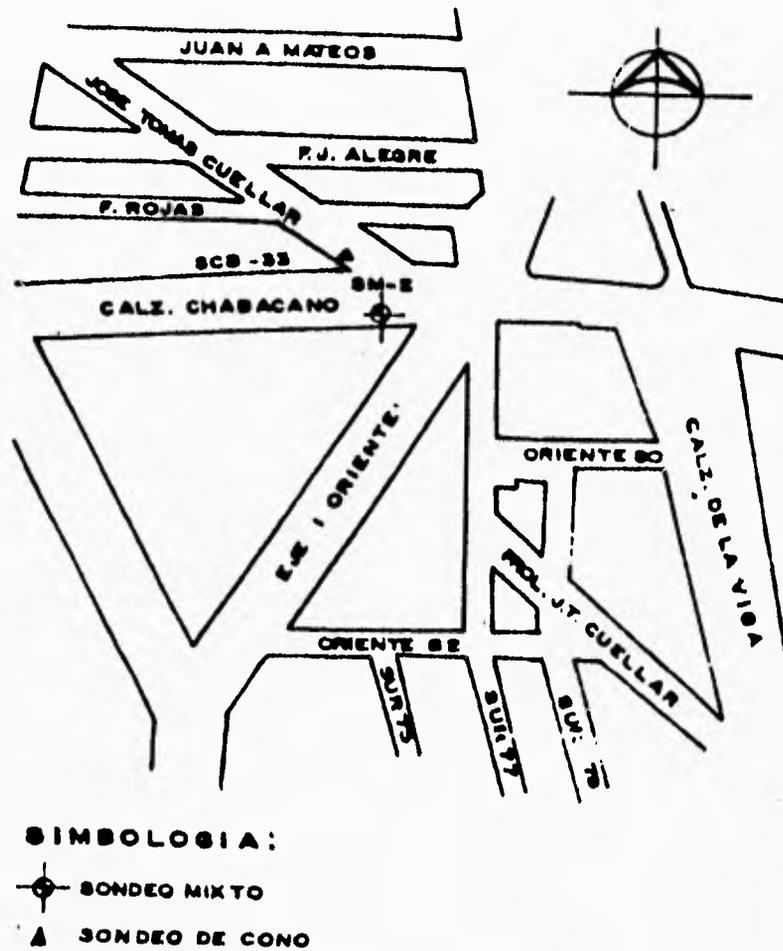


Fig 4.3 Localización de sondes

El procedimiento constructivo también contempló la instalación de troqueles a diferentes alturas de los muros Milán para ayudar a equilibrar los empujes del suelo y darle mayor estabilidad global a la excavación. En la fig 4.5 se presenta un esquema de la planta de las celdas, así como la localización de los inclinómetros que se instalaron para monitorear los desplazamientos horizontales del terreno inducidos por la excavación del suelo dentro de las celdas.

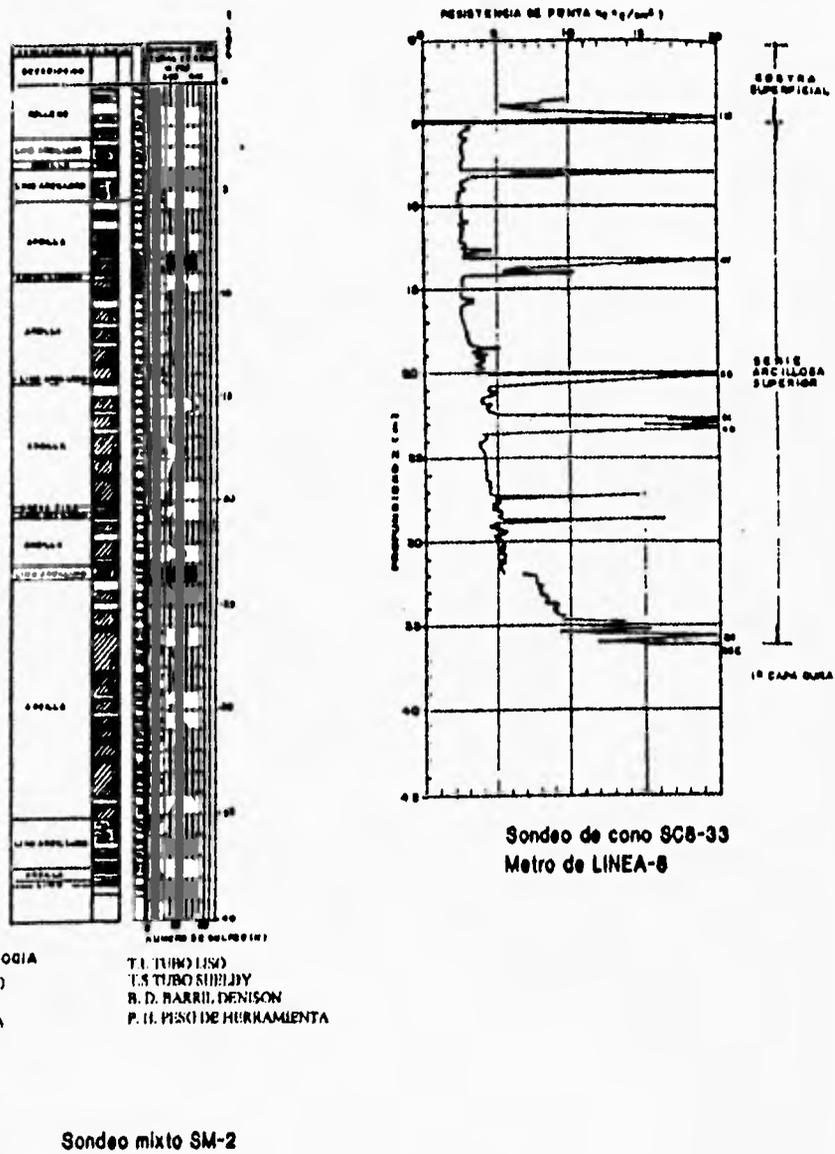


Fig 4.4 Sondeos

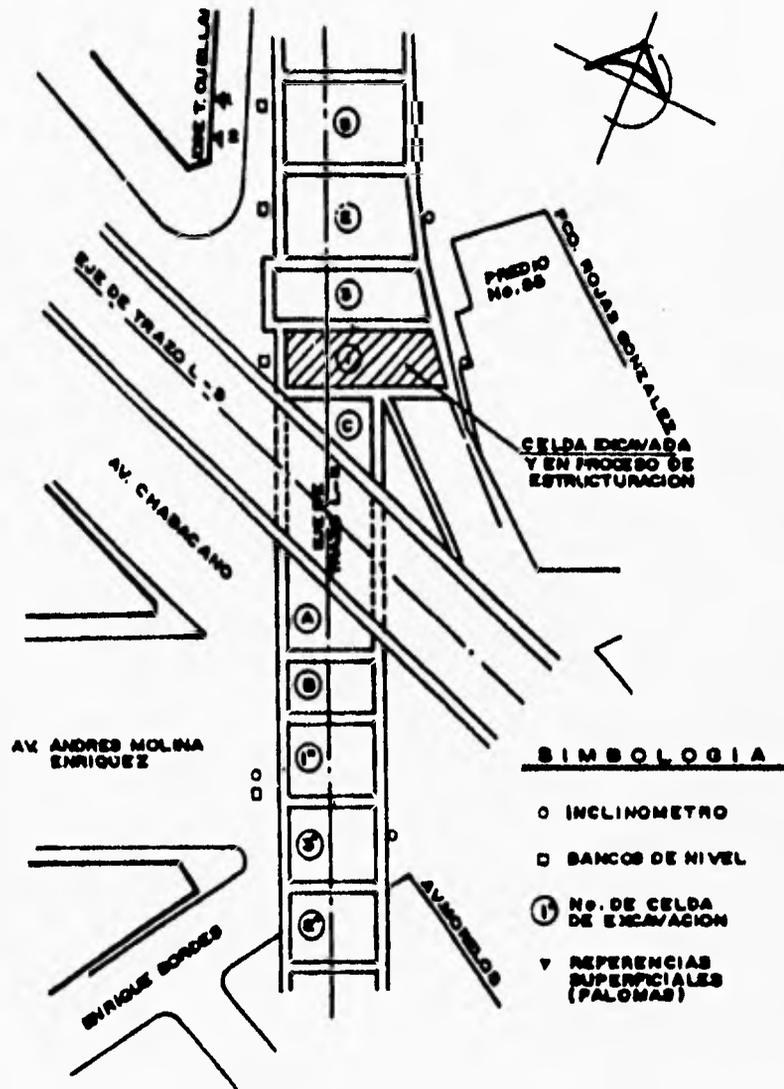


Fig 4.5 Localización de las celdas 1 y 2 y de los inclinómetros

4.2 CARACTERÍSTICAS DE LA MALLA UTILIZADA

La malla que se elaboró para la modelación numérica consta de 408 nodos y 121 elementos. El sistema entibamiento se toma como un muro apuntalado. El medio considerado (basado en el perfil de la zona, fig 4.4) es estratificado (6 estratos) y se proporcionan condiciones iniciales de esfuerzo, que corresponden a la condición isostática, debidas sólo al peso propio del suelo. La capa de arcilla es de 40m de profundidad. El nivel de aguas freáticas se ubica a 2.5m de profundidad. La carga aplicada por los puntales se simula con cargas concentradas en los nodos. Cinco niveles de puntales son usados. La excavación es de 8m de ancho por 15.8m de profundidad. Las características generales del sistema suelo-muro se indican en la tabla 4.1. Se emplea un sólido isoparamétrico con ocho nodos, material elásto-plástico perfecto para el suelo. Un elemento viga isoparamétrica de dos nodos, material elástico lineal para modelar el muro Milán. Se suponen condiciones de deformación plana.

Se consideraron cuatro etapas de construcción (excavación y aplicación de cargas en los puntales). En la primera etapa se considera a los elementos de la malla de elementos finitos como intactos. En la segunda etapa se construye el muro Milán, se excava a 2.5m de profundidad, se coloca el primer nivel de puntales con la precarga especificada y se aplica el efecto tridimensional calculado. En la tercera etapa se excava a 10.8m de profundidad y se coloca el segundo, tercero y cuarto nivel de puntales con la precarga especificada. En la cuarta y última etapa de construcción se alcanza el nivel máximo de excavación (15.8m de profundidad), se coloca el quinto y último de nivel de puntales con la precarga especificada y se construye la losa de fondo.

	Prof. (m)	E_s (t/m^2)	μ	γ (t/m^3)	c (t/m^2)			Prof. (m)	Q (t)
S	0.0-4.8	2000	0.350	1.30	10.0	P	1	2.5	30.0
U	4.8-10.8	600	0.495	1.20	3.0	U	2	4.8	30.0
E	10.8-15.0	340	0.495	1.20	1.7	N	3	7.8	30.0
L	15.0-25.0	1400	0.495	1.20	3.5	T	4	10.8	60.0
O	25.0-30.0	3000	0.350	1.20	10.0	A	5	13	60.0
	30.0-40.0	4500	0.350	1.20	15.0	L			

Tabla 4.1 Características generales del sistema muro-suelo

4.3 CONSTRUCCIÓN DE LA MALLA

Para la modelación numérica se corre el módulo de alimentación desde el cual se genera la malla de elementos finitos; para éste caso primero se genera con elementos proporcionales fig 4.7, posteriormente utilizando la opción de arreglo del módulo la malla es afinada para conseguir las características deseadas fig 4.8.

Las coordenadas e incidencias que sean arrojadas por la generación de la malla serán definitivas y contemplarán parte del archivo.

4.4 CONSTRUCCIÓN DE ARCHIVO DE DATOS

Desde el mismo módulo y con el uso de una ventana de trabajo, fig 4.6, las características del problema son introducidas. En las figuras 4.9 y 4.10 se muestra que con la ayuda gráfica las características de los grupos de material y las etapas de excavación son indicadas en la misma malla para conformar el archivo de datos final.

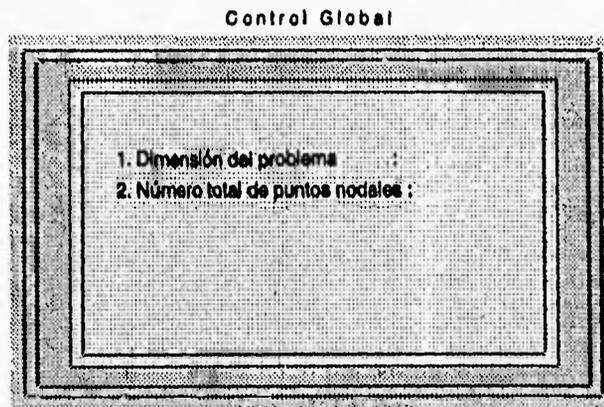


Fig 4.6 ventana de trabajo

4.5 GRAFICACIÓN DE RESULTADOS

Para definir la magnitud crítica de la estabilidad de la excavación TEST92 es ejecutado con el archivo de datos generado por el módulo de alimentación, en donde se realizan análisis de deformación tanto en el sentido vertical como horizontal, considerando la interrelación entre las propiedades del suelo, la rigidez del ademe, la precarga en los puntales y el espaciamiento entre estos elementos, definiendo los valores críticos a cuidar para que no sean rebasados durante el proceso constructivo.

Con los resultados obtenidos con TEST92 el módulo de graficación es activado para analizar los siguientes resultados.

- 1) Asentamientos de colindancia (fig 4.11)
- 2) Desplazamientos laterales (fig 4.12)
- 3) Malla de elementos finitos deformada (fig 4.13)

Finalmente la graficación de resultados permitirá al diseñador jugar con los parámetros involucrados de tal manera que la predicción de deformaciones proporcione la seguridad de la obra deseada.

En la Mecánica de Suelos existen otros problemas que también pueden ser analizados con el Método del Elemento Finito, tales como laderas, terraplenes, etc. El graficador de mallas que se encuentra integrado dentro del sistema permite graficar los mallas con el objeto de buscar errores en las coordenadas e incidencias de la misma. Las figuras 4.14, y 4.15 son dos ejemplos que muestran la aplicación útil del generador de mallas.

111	100	89	78	67	56	45	34	23	12	1
112	101	90	79	68	57	46	35	24	13	2
113	102	91	80	69	58	47	36	25	14	3
114	103	92	81	70	59	48	37	26	15	4
115	104	93	82	71	60	49	38	27	16	5
116	105	94	83	72	61	50	39	28	17	6
117	106	95	84	73	62	51	40	29	18	7
118	107	96	85	74	63	52	41	30	19	8
119	108	97	86	75	64	53	42	31	20	9
120	109	98	87	76	65	54	43	32	21	10
121	110	99	88	77	66	55	44	33	22	11

Fig 4.7 Malla de elementos finitos inicial

Aplicación a un caso práctico

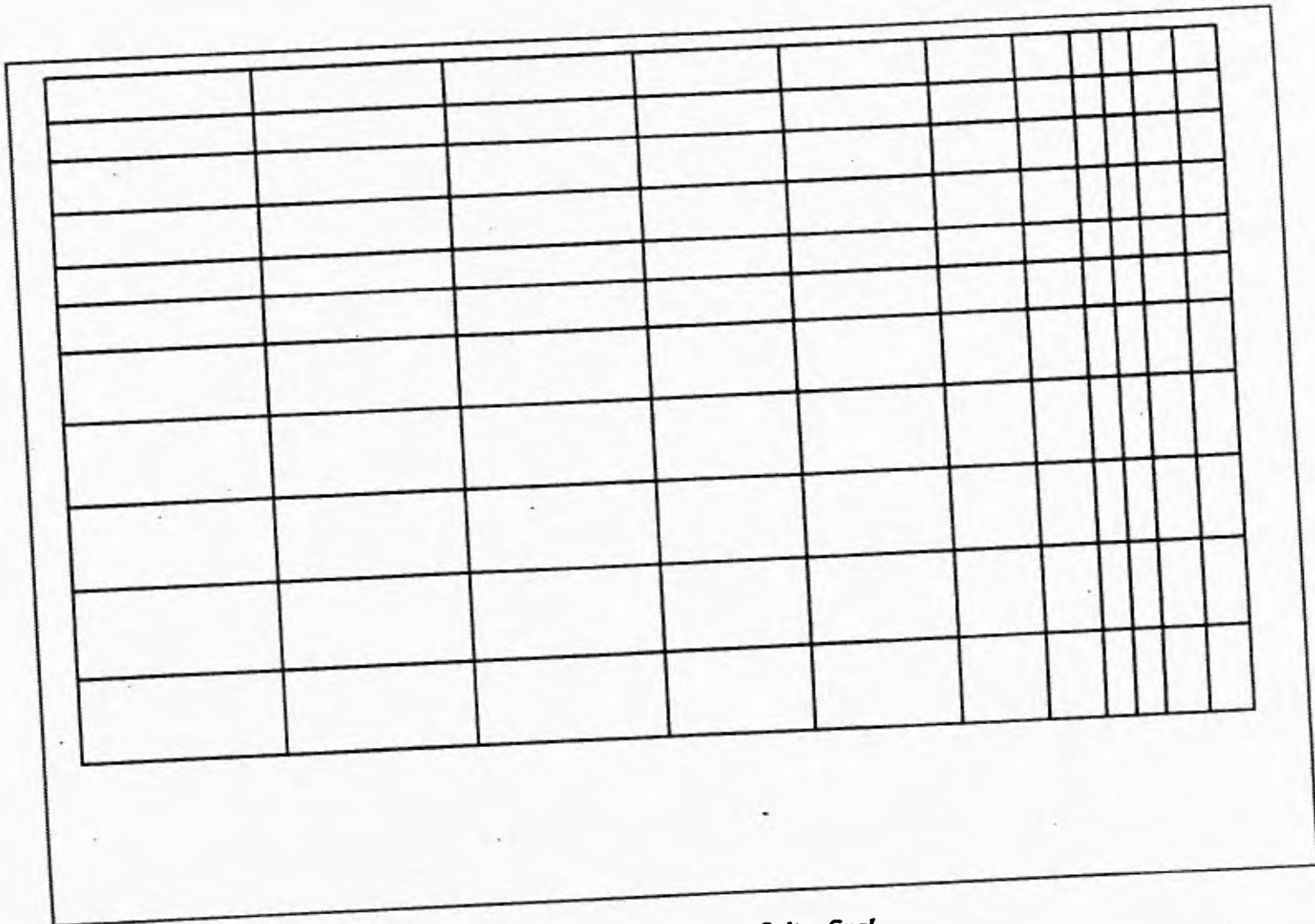


Fig 4.8 Malla de elementos finitos final

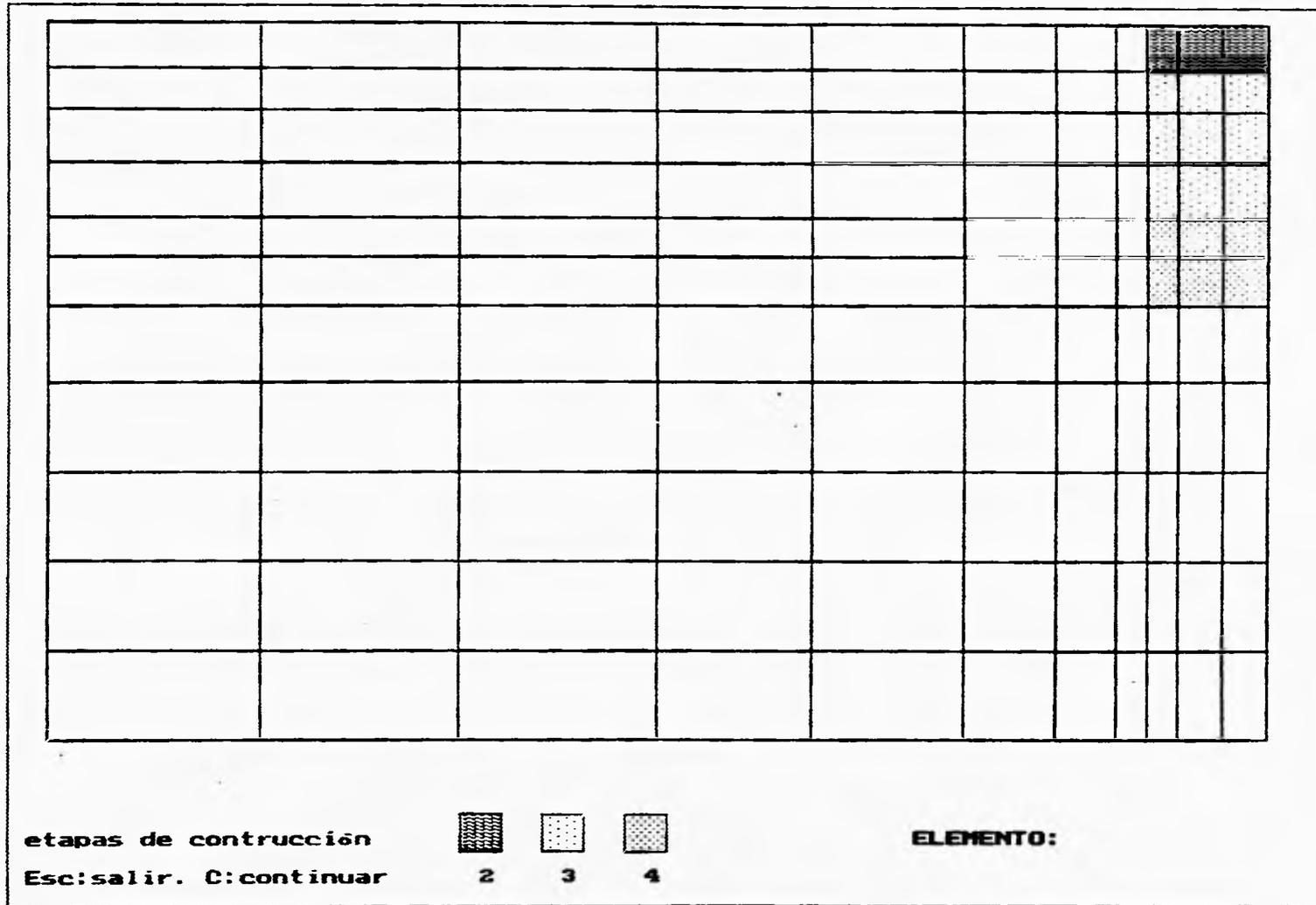


Fig 4.9 Etapas de excavación

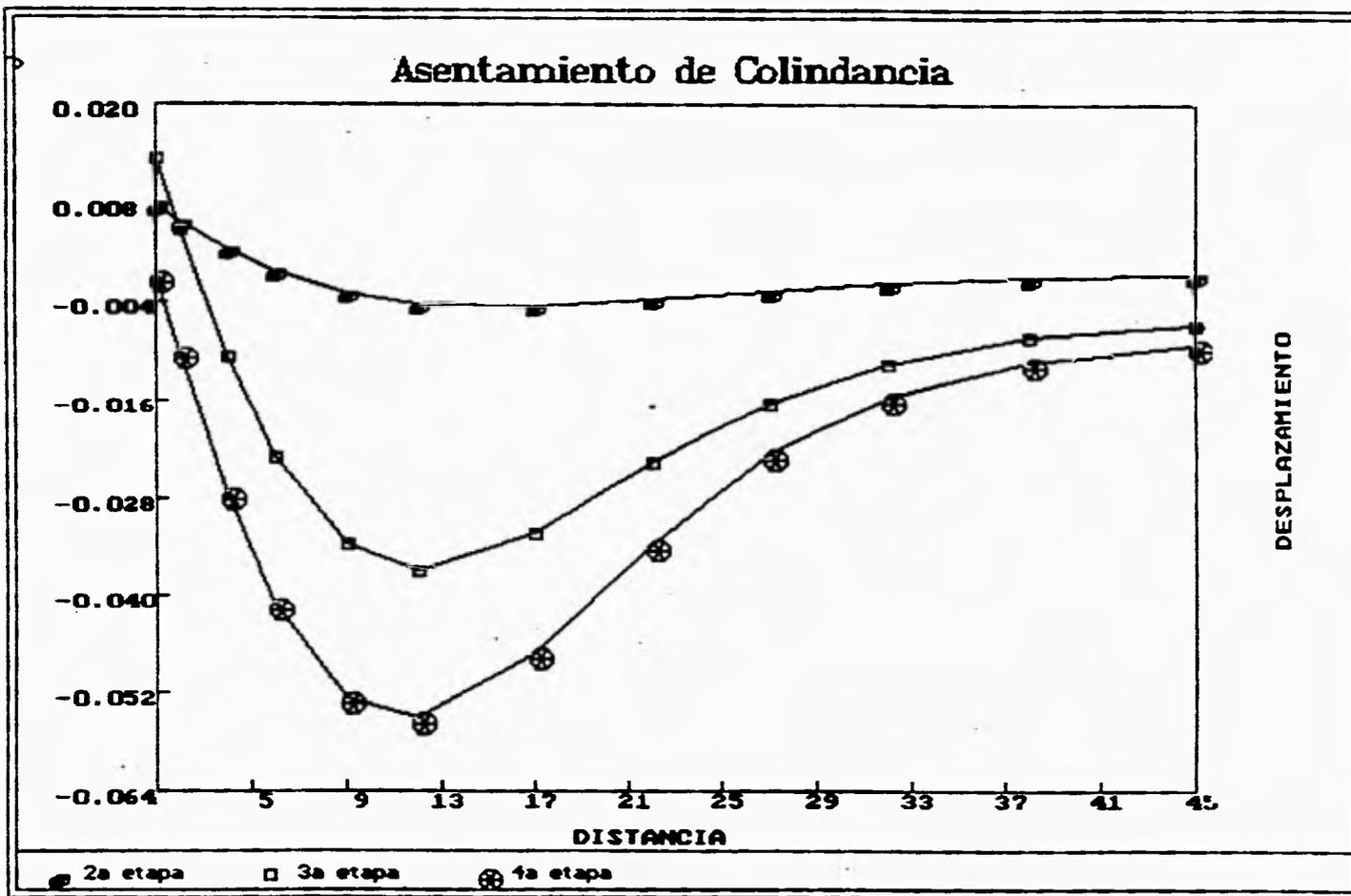


Fig 4.11 Movimientos resultantes

Aplicación a un caso práctico

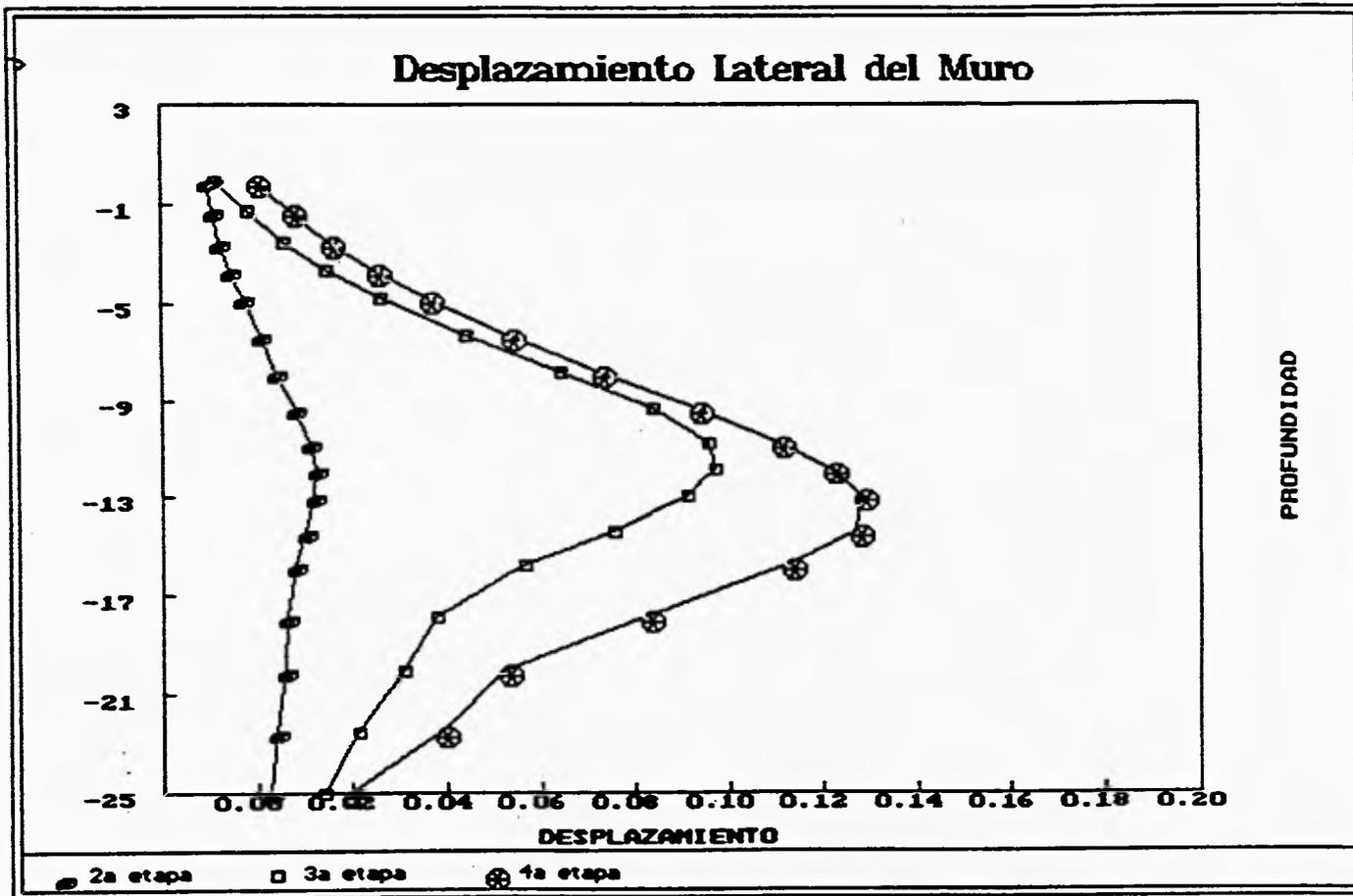


Fig 4.12 Movimientos analizados

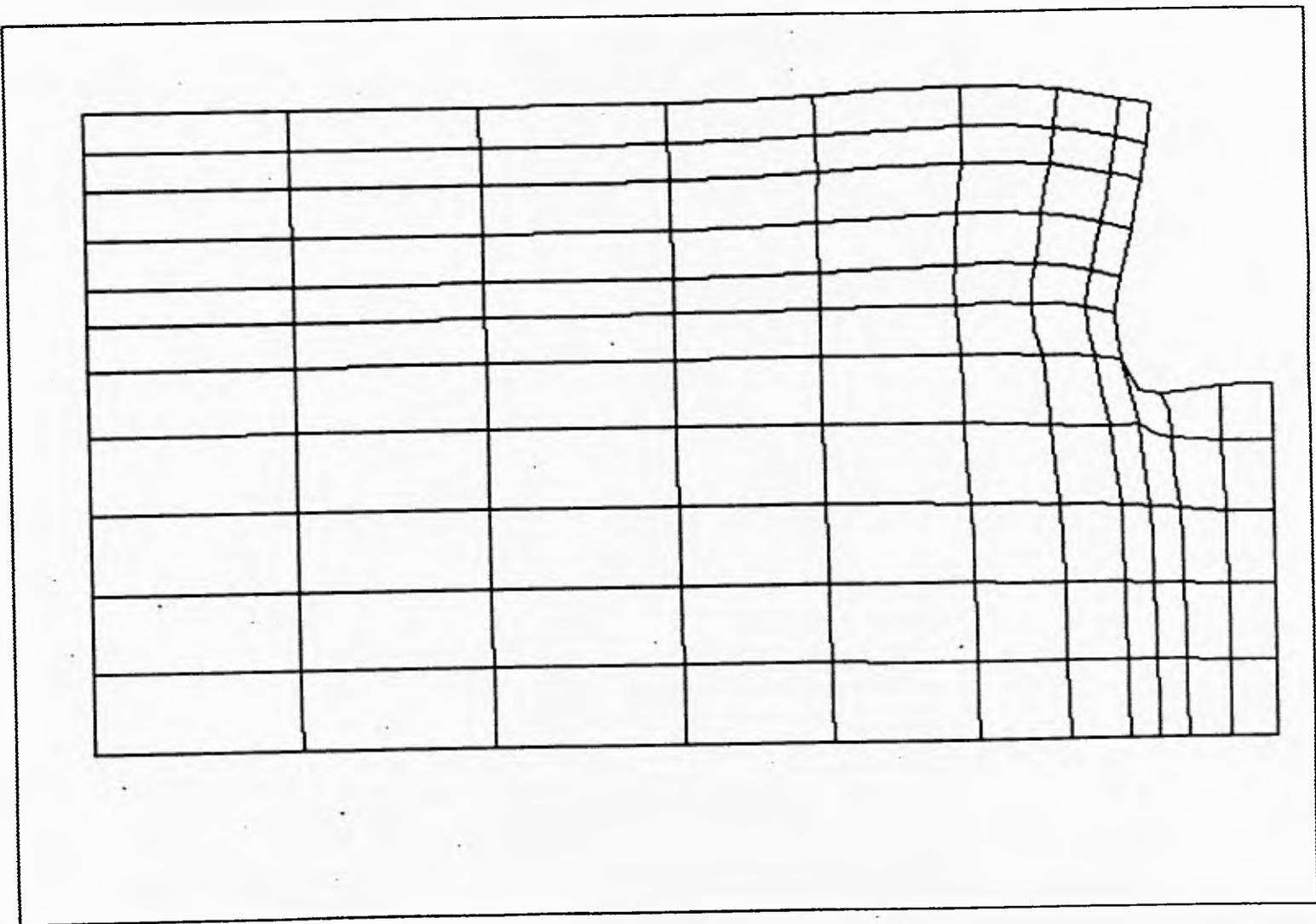


Fig 4.13 Malla deformada

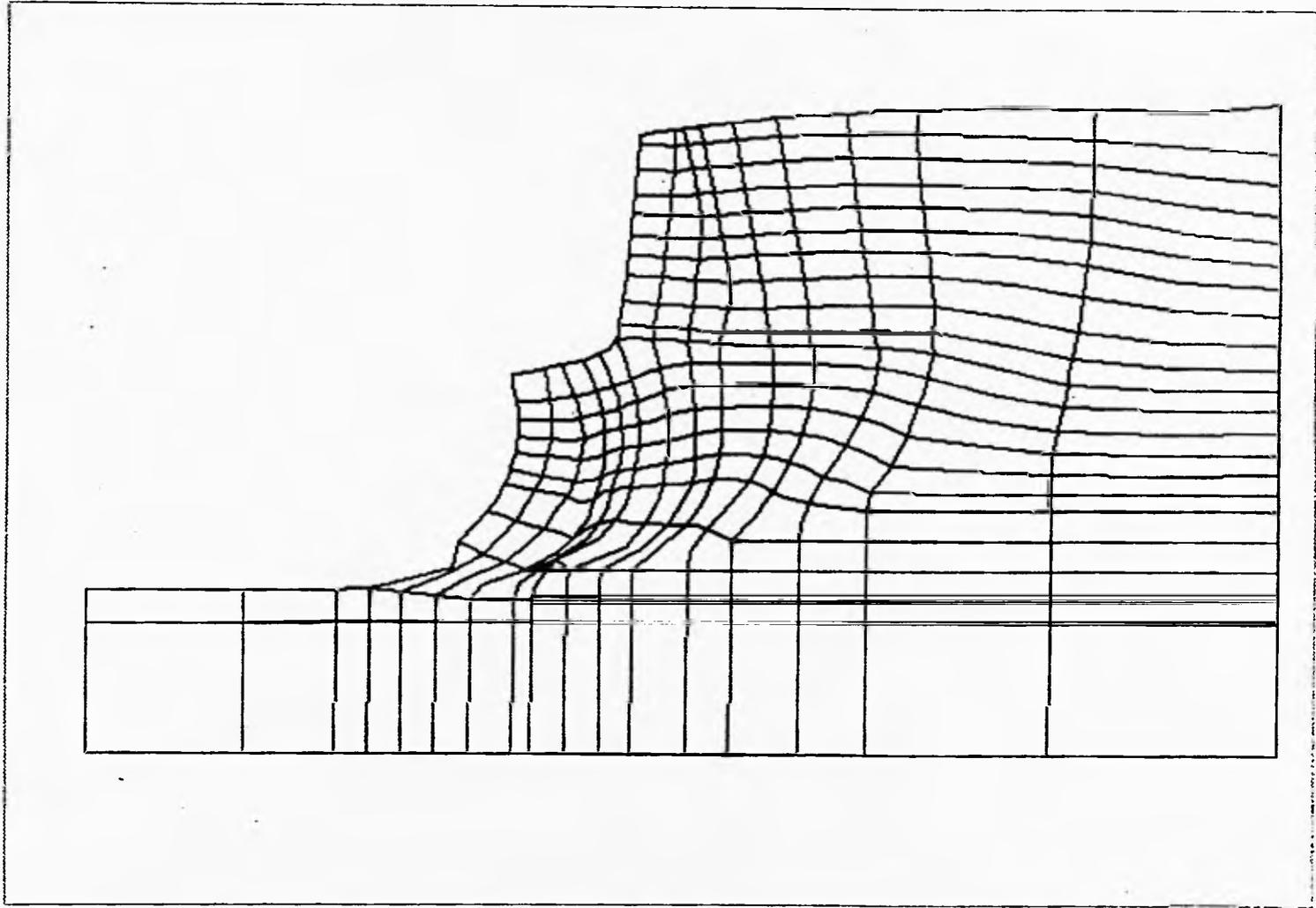


Fig 4.14 Malla de elementos finitos para analizar la estabilidad de un talud con "crib wall"

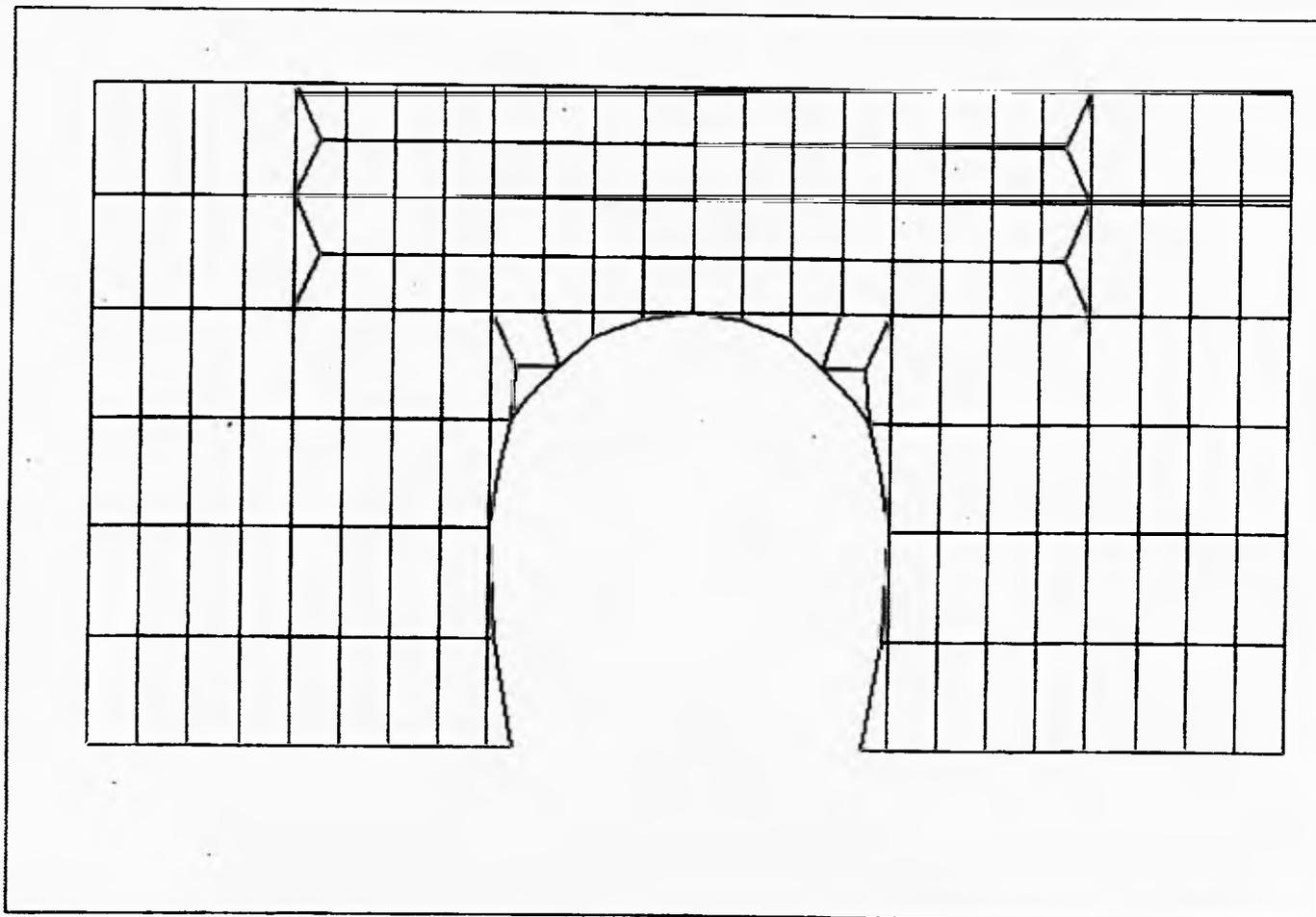


Fig 4.15 Malla de elementos finitos para analizar los esfuerzos a los que está sometida una alcantarilla

5. CONCLUSIONES

- 1) El SE desarrollado automatiza gran parte de los procesos utilizados en el estudio de los movimientos causados por una excavación, reduciendo significativamente los tiempos necesarios para llegar a conclusiones referente a este tipo de problemas y dando mayor flexibilidad a los datos de entrada y salida.
- 2) La estructuración del SE mediante módulos permite el crecimiento de éste, según las investigaciones se vayan desarrollando. La incorporación de estos módulos se puede llevar a cabo de una manera muy sencilla.

- 3) Es posible utilizar esta técnica de resolución de problemas para enfrentar una gran diversidad de problemas en las diferentes ramas de la Ingeniería Civil de manera flexible.
- 4) Los programas fuentes de los módulos en los que se dividió la solución es posible transformarlos posteriormente a lenguaje orientado al objeto para ser cargados en la estación de trabajo, y así se tenga mayor alcance del SE en cuanto a su crecimiento y uso.
- 5) Es posible recuperar parte de la lógica empleada para ser utilizada en la construcción de otros SE.
- 6) La construcción del SE demuestra que es posible crear un medio idóneo para la difusión de nuevos conocimientos.
- 7) El SE representa un elemento de partida para desarrollar otras investigaciones que permitan que el conocimiento pueda ser difundido y utilizado sin que éste quede solamente en las manos del investigador. Además lograr, entre la misma comunidad geotecnista y otras áreas del conocimiento, un trabajo interdisciplinario que de paso al desarrollo de tecnología de punta y así mismo crear un medio de retroalimentación entre los que construyen y los que utilizan el producto final.

5.1 RECOMENDACIONES

Uno de los siguientes pasos del desarrollo del SE es la implementación de éste en la estación de trabajo que permita incorporar el programa TEST92 dentro del cuerpo del SE y ampliar sus capacidades. Gracias a que la mayor parte de los pasos necesarios para analizar excavaciones apuntaladas se lograron automatizar, surge la idea de ampliar las investigaciones en el campo, como son:

- 1) Generación de mallas tridimensionales, con la que se mejorará el modelaje de la excavación y colocado de puntales.
- 2) El modelaje puede incluir casos elásticos y no lineales de comportamiento de suelos, así como su comparación con mediciones con inclinómetros, en excavaciones del metro.
- 3) Incluir información relativa al caso de túneles y lumbreras, así como la presentación gráfica de las mallas empleadas en los estudios axisimétricos.

REFERENCIAS

- 1) Chatain, J. N. y Dussauchoy, A.; "Sistemas Expertos. Métodos y Herramientas"; Ed. Paraninfo S.A. Madrid, España; 1988.
- 2) Cruz, J. R.; " Sistema Experto para el Manejo de Programas de Análisis Dinámico de Presas"; Tesis de licenciatura, Universidad Nacional Autónoma de México, México; 1993.
- 3) Goldstein, L. J. "Turbo Pascal. Introducción a la Programación Orientada a Objetos (versión 6.0)"; Ed. Prentice-Hall Hispanoamericana S.A.

- 4) Joyanes, Luis; "Programación en Turbo Pascal (versiones 4.0,5.0 y 5.5)"; Ed McGraw-Hill, España; 1990.
- 5) Juárez B., E., Rico R., A.; "Fundamentos de Mecánica de Suelos"; Ed. Limusa, Vol I 3a ed., México, D. F.; 1974.
- 6) Kemper, N.; " Sistema Experto Aplicado al Uso Eficiente de Energía en Iluminación Industrial"; tesis de licenciatura; Universidad Nacional Autónoma de México, México; 1991.
- 7) Lara,R.; "Utilización de Sistemas Expertos en la Geotecnia"; XV Reunión nacional de Mecánica de Suelos; San Luis Potosí, S.L.P, México; Vol 11, pp 277-281; 1990.
- 8) Levine, Robert I., Drang, D., Delson, B.; "Expert Systems: Turbo Pascal Edition"; Ed McGraw-Hill Book Company; 1988.
- 9) Li Xiangyue y Romo, M. P.; "TEST92: Un Programa de Computadora para Simular Procesos Constructivos en Suelos Elasto-Plásticos"; informe interno, Instituto de Ingeniería, UNAM; 1992.
- 10) Mompim, P.; J. " Inteligencia Artificial. Conceptos, Técnicas y Aplicaciones"; Ed. Marcombo S.A, Barcelona, España; 1988
- 11) Nebendahl, Dieter; "Sistemas Expertos. Introducción a la Técnica y Aplicación"; Ed. Marcombo S.A, Barcelona, España; 1988.
- 12) Rico Rodríguez, A. y Del Castillo, H.; "La Ingeniería de Suelos en las Vías Terrestres"; Ed. Limusa, S.A., México.
- 13) Rodríguez, R.; "Análisis de los Movimientos en el Terreno por Excavaciones Profundas en la Ciudad de México"; tesis de licenciatura, Universidad Nacional Autónoma de México, México, 1995.
- 14) Romo, M. P., Ruelas, S., Magaña, R.; "Comportamiento de la Excavación para el Cruce de las Líneas 8 y 9 del Metro"; XVI Reunión Nacional de Mecánica de Suelos, Casos Historia en; Zacatecas, Zac; México; Vol I pp 39-46; 1992.
- 15) Sánchez y Beltrán, J.P.; "Sistemas Expertos (Una metodología de Programación)"; Ed. Microbit, México; 1990.

ANEXO I

Archivo de datos generado por el módulo de alimentación

(* Título del problema *)

EXCAVACION, METRO, ELASTO-PLASTICO PERFECTO

(* Tarjeta de control global *)

2 408 3 2 4 9 14 0 8

(* Tarjeta de control de arreglo común *)

50000

(* Error de iteración *)

1.E-02

(* Coordenadas de puntos nodales, grados de libertad *)

80 40.00 1 0 1	78.5 15.00 0 0 1	75.5 29.20 0 0 1
80 38.75 1 0 1	78.5 10.00 0 0 1	75.5 27.00 0 0 1
80 37.50 1 0 1	78.5 5.00 0 0 1	75.5 24.20 0 0 1
80 36.35 1 0 1	78.5 0.00 1 1 1	75.5 20.00 0 0 1
80 35.20 1 0 1	77 40.00 0 0 1	75.5 15.00 0 0 1
80 33.70 1 0 1	77 38.75 0 0 1	75.5 10.00 0 0 1
80 32.20 1 0 1	77 37.50 0 0 1	75.5 5.00 0 0 1
80 30.70 1 0 1	77 36.35 0 0 1	75.5 0.00 1 1 1
80 29.20 1 0 1	77 35.20 0 0 1	74 40.00 0 0 1
80 28.10 1 0 1	77 33.70 0 0 1	74 38.75 0 0 1
80 27.00 1 0 1	77 32.20 0 0 1	74 37.50 0 0 1
80 25.60 1 0 1	77 30.70 0 0 1	74 36.35 0 0 1
80 24.20 1 0 1	77 29.20 0 0 1	74 35.20 0 0 1
80 22.10 1 0 1	77 28.10 0 0 1	74 33.70 0 0 1
80 20.00 1 0 1	77 27.00 0 0 1	74 32.20 0 0 1
80 17.50 1 0 1	77 25.60 0 0 1	74 30.70 0 0 1
80 15.00 1 0 1	77 24.20 0 0 1	74 29.20 0 0 1
80 12.50 1 0 1	77 22.10 0 0 1	74 28.10 0 0 1
80 10.00 1 0 1	77 20.00 0 0 1	74 27.00 0 0 1
80 7.50 1 0 1	77 17.50 0 0 1	74 25.60 0 0 1
80 5.00 1 0 1	77 15.00 0 0 1	74 24.20 0 0 1
80 2.50 1 0 1	77 12.50 0 0 1	74 22.10 0 0 1
80 0.00 1 1 1	77 10.00 0 0 1	74 20.00 0 0 1
78.5 40.00 0 0 1	77 7.50 0 0 1	74 17.50 0 0 1
78.5 37.50 0 0 1	77 5.00 0 0 1	74 15.00 0 0 1
78.5 35.20 0 0 1	77 2.50 0 0 1	74 12.50 0 0 1
78.5 32.20 0 0 1	77 0.00 1 1 1	74 10.00 0 0 1
78.5 29.20 0 0 1	75.5 40.00 0 0 1	74 7.50 0 0 1
78.5 27.00 0 0 1	75.5 37.50 0 0 1	74 5.00 0 0 1
78.5 24.20 0 0 1	75.5 35.20 0 0 1	74 2.50 0 0 1
78.5 20.00 0 0 1	75.5 32.20 0 0 1	74 0.00 1 1 1

Anexo I: Archivo de datos

73.0 40.00 0 0 1	70 30.70 0 0 1	63.0 32.20 0 0 1
73.0 37.50 0 0 1	70 29.20 0 0 1	63.0 29.20 0 0 1
73.0 35.20 0 0 1	70 28.10 0 0 1	63.0 27.00 0 0 1
73.0 32.20 0 0 1	70 27.00 0 0 1	63.0 24.20 0 0 1
73.0 29.20 0 0 1	70 25.60 0 0 1	63.0 20.00 0 0 1
73.0 27.00 0 0 1	70 24.20 0 0 1	63.0 15.00 0 0 1
73.0 24.20 0 0 1	70 22.10 0 0 1	63.0 10.00 0 0 1
73.0 20.00 0 0 1	70 20.00 0 0 1	63.0 5.00 0 0 1
73.0 15.00 0 0 1	70 17.50 0 0 1	63.0 0.00 1 1 1
73.0 10.00 0 0 1	70 15.00 0 0 1	60 40.00 0 0 1
73.0 5.00 0 0 1	70 12.50 0 0 1	60 38.75 0 0 1
73.0 0.00 1 1 1	70 10.00 0 0 1	60 37.50 0 0 1
72 40.00 0 0 0	70 7.50 0 0 1	60 36.35 0 0 1
72 38.75 0 0 0	70 5.00 0 0 1	60 35.20 0 0 1
72 37.50 0 0 0	70 2.50 0 0 1	60 33.70 0 0 1
72 36.35 0 0 0	70 0.00 1 1 1	60 32.20 0 0 1
72 35.20 0 0 0	68.0 40.00 0 0 1	60 30.70 0 0 1
72 33.70 0 0 0	68.0 37.50 0 0 1	60 29.20 0 0 1
72 32.20 0 0 0	68.0 35.20 0 0 1	60 28.10 0 0 1
72 30.70 0 0 0	68.0 32.20 0 0 1	60 27.00 0 0 1
72 29.20 0 0 0	68.0 29.20 0 0 1	60 25.60 0 0 1
72 28.10 0 0 0	68.0 27.00 0 0 1	60 24.20 0 0 1
72 27.00 0 0 0	68.0 24.20 0 0 1	60 22.10 0 0 1
72 25.60 0 0 0	68.0 20.00 0 0 1	60 20.00 0 0 1
72 24.20 0 0 0	68.0 15.00 0 0 1	60 17.50 0 0 1
72 22.10 0 0 0	68.0 10.00 0 0 1	60 15.00 0 0 1
72 20.00 0 0 0	68.0 5.00 0 0 1	60 12.50 0 0 1
72 17.50 0 0 1	68.0 0.00 1 1 1	60 10.00 0 0 1
72 15.00 0 0 1	66 40.00 0 0 1	60 7.50 0 0 1
72 12.50 0 0 1	66 38.75 0 0 1	60 5.00 0 0 1
72 10.00 0 0 1	66 37.50 0 0 1	60 2.50 0 0 1
72 7.50 0 0 1	66 36.35 0 0 1	60 0.00 1 1 1
72 5.00 0 0 1	66 35.20 0 0 1	55.0 40.00 0 0 1
72 2.50 0 0 1	66 33.70 0 0 1	55.0 37.50 0 0 1
72 0.00 1 1 1	66 32.20 0 0 1	55.0 35.20 0 0 1
71.0 40.00 0 0 1	66 30.70 0 0 1	55.0 32.20 0 0 1
71.0 37.50 0 0 1	66 29.20 0 0 1	55.0 29.20 0 0 1
71.0 35.20 0 0 1	66 28.10 0 0 1	55.0 27.00 0 0 1
71.0 32.20 0 0 1	66 27.00 0 0 1	55.0 24.20 0 0 1
71.0 29.20 0 0 1	66 25.60 0 0 1	55.0 20.00 0 0 1
71.0 27.00 0 0 1	66 24.20 0 0 1	55.0 15.00 0 0 1
71.0 24.20 0 0 1	66 22.10 0 0 1	55.0 10.00 0 0 1
71.0 20.00 0 0 1	66 20.00 0 0 1	55.0 5.00 0 0 1
71.0 15.00 0 0 1	66 17.50 0 0 1	55.0 0.00 1 1 1
71.0 10.00 0 0 1	66 15.00 0 0 1	50 40.00 0 0 1
71.0 5.00 0 0 1	66 12.50 0 0 1	50 38.75 0 0 1
71.0 0.00 1 1 1	66 10.00 0 0 1	50 37.50 0 0 1
70 40.00 0 0 1	66 7.50 0 0 1	50 36.35 0 0 1
70 38.75 0 0 1	66 5.00 0 0 1	50 35.20 0 0 1
70 37.50 0 0 1	66 2.50 0 0 1	50 33.70 0 0 1
70 36.35 0 0 1	66 0.00 1 1 1	50 32.20 0 0 1
70 35.20 0 0 1	63.0 40.00 0 0 1	50 30.70 0 0 1
70 33.70 0 0 1	63.0 37.50 0 0 1	50 29.20 0 0 1
70 32.20 0 0 1	63.0 35.20 0 0 1	50 28.10 0 0 1

50 27.00 0 0 1	40 0.00 1 1 1	21.0 0.00 1 1 1
50 25.60 0 0 1	34.0 40.00 0 0 1	14 40.00 0 0 1
50 24.20 0 0 1	34.0 37.50 0 0 1	14 38.75 0 0 1
50 22.10 0 0 1	34.0 35.20 0 0 1	14 37.50 0 0 1
50 20.00 0 0 1	34.0 32.20 0 0 1	14 36.35 0 0 1
50 17.50 0 0 1	34.0 29.20 0 0 1	14 35.20 0 0 1
50 15.00 0 0 1	34.0 27.00 0 0 1	14 33.70 0 0 1
50 12.50 0 0 1	34.0 24.20 0 0 1	14 32.20 0 0 1
50 10.00 0 0 1	34.0 20.00 0 0 1	14 30.70 0 0 1
50 7.50 0 0 1	34.0 15.00 0 0 1	14 29.20 0 0 1
50 5.00 0 0 1	34.0 10.00 0 0 1	14 28.10 0 0 1
50 2.50 0 0 1	34.0 5.00 0 0 1	14 27.00 0 0 1
50 0.00 1 1 1	34.0 0.00 1 1 1	14 25.60 0 0 1
45.0 40.00 0 0 1	27 40.00 0 0 1	14 24.20 0 0 1
45.0 37.50 0 0 1	27 38.75 0 0 1	14 22.10 0 0 1
45.0 35.20 0 0 1	27 37.50 0 0 1	14 20.00 0 0 1
45.0 32.20 0 0 1	27 36.35 0 0 1	14 17.50 0 0 1
45.0 29.20 0 0 1	27 35.20 0 0 1	14 15.00 0 0 1
45.0 27.00 0 0 1	27 33.70 0 0 1	14 12.50 0 0 1
45.0 24.20 0 0 1	27 32.20 0 0 1	14 10.00 0 0 1
45.0 20.00 0 0 1	27 30.70 0 0 1	14 7.50 0 0 1
45.0 15.00 0 0 1	27 29.20 0 0 1	14 5.00 0 0 1
45.0 10.00 0 0 1	27 28.10 0 0 1	14 2.50 0 0 1
45.0 5.00 0 0 1	27 27.00 0 0 1	14 0.00 1 1 1
45.0 0.00 1 1 1	27 25.60 0 0 1	7.0 40.00 0 0 1
40 40.00 0 0 1	27 24.20 0 0 1	7.0 37.50 0 0 1
40 38.75 0 0 1	27 22.10 0 0 1	7.0 35.20 0 0 1
40 37.50 0 0 1	27 20.00 0 0 1	7.0 32.20 0 0 1
40 36.35 0 0 1	27 17.50 0 0 1	7.0 29.20 0 0 1
40 35.20 0 0 1	27 15.00 0 0 1	7.0 27.00 0 0 1
40 33.70 0 0 1	27 12.50 0 0 1	7.0 24.20 0 0 1
40 32.20 0 0 1	27 10.00 0 0 1	7.0 20.00 0 0 1
40 30.70 0 0 1	27 7.50 0 0 1	7.0 15.00 0 0 1
40 29.20 0 0 1	27 5.00 0 0 1	7.0 10.00 0 0 1
40 28.10 0 0 1	27 2.50 0 0 1	7.0 5.00 0 0 1
40 27.00 0 0 1	27 0.00 1 1 1	7.0 0.00 1 1 1
40 25.60 0 0 1	21.0 40.00 0 0 1	0 40.00 1 0 1
40 24.20 0 0 1	21.0 37.50 0 0 1	0 38.75 1 0 1
40 22.10 0 0 1	21.0 35.20 0 0 1	0 37.50 1 0 1
40 20.00 0 0 1	21.0 32.20 0 0 1	0 36.35 1 0 1
40 17.50 0 0 1	21.0 29.20 0 0 1	0 35.20 1 0 1
40 15.00 0 0 1	21.0 27.00 0 0 1	0 33.70 1 0 1
40 12.50 0 0 1	21.0 24.20 0 0 1	0 32.20 1 0 1
40 10.00 0 0 1	21.0 20.00 0 0 1	0 30.70 1 0 1
40 7.50 0 0 1	21.0 15.00 0 0 1	0 29.20 1 0 1
40 5.00 0 0 1	21.0 10.00 0 0 1	0 28.10 1 0 1
40 2.50 0 0 1	21.0 5.00 0 0 1	0 27.00 1 0 1
0 25.60 1 0 1	0 17.50 1 0 1	0 7.50 1 0 1
0 24.20 1 0 1	0 15.00 1 0 1	0 5.00 1 0 1
0 22.10 1 0 1	0 12.50 1 0 1	0 2.50 1 0 1
0 20.00 1 0 1	0 10.00 1 0 1	0 0.00 1 1 1

(* Cargas concentradas *)

13 2 4 -1.06
30 2 4 -1.06
48 2 4 -1.06
65 2 4 -1.06
83 2 4 -1.06
100 2 4 -1.06
108 1 2 -8.0
108 1 3 -8.0
110 1 2 -8.0
110 1 3 -8.0
112 1 2 -8.0
112 1 3 -8.0
114 1 4 -16.0
116 1 4 -16.0

(* Condiciones iniciales *)

40. 0. 0.5 0. 0
37.5 -3.25 0.5 0. 0
35.2 -6.24 0.5 2.3 0
29.2 -13.44 0.5 8.3 0
24.2 -19.44 0.5 13.3 0
15. -30.48 0.5 22.5 0
10. -36.48 0.5 27.5 0
0. -51.48 0.5 37.5 0

(* Control global de elementos *)

6 121 8 2 6 5 5 4
3 14 2 3 1 5 3 1

(* Propiedades de materiales. Sólido isoparamétrico con 8 nodos *)

2000.0 0.350 1.30 10.0 100000.
600.0 0.495 1.20 3.0 100000.
340.0 0.495 1.20 1.7 100000.
1400.00 0.495 1.20 3.5 100000.
3000.00 0.350 1.20 10.0 100000.
4500.00 0.350 1.50 15.0 100000.

(* Conectividades, Incidencias, etapa de construcción, material*)

36	37	38	25	3	2	1	24	1	-2	192	193	194	173	159	158	157	172	5	1
38	39	40	26	5	4	3	25	1	-3	194	195	196	174	161	160	159	173	6	1
40	41	42	27	7	6	5	26	2	-3	196	197	198	175	163	162	161	174	6	1
42	43	44	28	9	8	7	27	2	-3	211	212	213	200	178	177	176	199	1	1
44	45	46	29	11	10	9	28	3	-4	213	214	215	201	180	179	178	200	1	1
46	47	48	30	13	12	11	29	3	-4	215	216	217	202	182	181	180	201	2	1
48	49	50	31	15	14	13	30	4	1	217	218	219	203	184	183	182	202	2	1
50	51	52	32	17	16	15	31	4	1	219	220	221	204	186	185	184	203	3	1
52	53	54	33	19	18	17	32	5	1	221	222	223	205	188	187	186	204	3	1
54	55	56	34	21	20	19	33	6	1	223	224	225	206	190	189	188	205	4	1
56	57	58	35	23	22	21	34	6	1	225	226	227	207	192	191	190	206	4	1
71	72	73	60	38	37	36	59	1	-2	227	228	229	208	194	193	192	207	5	1
73	74	75	61	40	39	38	60	1	-3	229	230	231	209	196	195	194	208	6	1
75	76	77	62	42	41	40	61	2	-3	231	232	233	210	198	197	196	209	6	1
77	78	79	63	44	43	42	62	2	-3	246	247	248	235	213	212	211	234	1	1
79	80	81	64	46	45	44	63	3	-4	248	249	250	236	215	214	213	235	1	1
81	82	83	65	48	47	46	64	3	-4	250	251	252	237	217	216	215	236	2	1
83	84	85	66	50	49	48	65	4	1	252	253	254	238	219	218	217	237	2	1
85	86	87	67	52	51	50	66	4	1	254	255	256	239	221	220	219	238	3	1
87	88	89	68	54	53	52	67	5	1	256	257	258	240	223	222	221	239	3	1
89	90	91	69	56	55	54	68	6	1	258	259	260	241	225	224	223	240	4	1
91	92	93	70	58	57	56	69	6	1	260	261	262	242	227	226	225	241	4	1
106	107	108	95	73	72	71	94	1	-2	262	263	264	243	229	228	227	242	5	1
108	109	110	96	75	74	73	95	1	-3	264	265	266	244	231	230	229	243	6	1
110	111	112	97	77	76	75	96	2	-3	266	267	268	245	233	232	231	244	6	1
112	113	114	98	79	78	77	97	2	-3	281	282	283	270	248	247	246	269	1	1
114	115	116	99	81	80	79	98	3	-4	283	284	285	271	250	249	248	270	1	1
116	117	118	100	83	82	81	99	3	-4	285	286	287	272	252	251	250	271	2	1
118	119	120	101	85	84	83	100	4	1	287	288	289	273	254	253	252	272	2	1
120	121	122	102	87	86	85	101	4	1	289	290	291	274	256	255	254	273	3	1
122	123	124	103	89	88	87	102	5	1	291	292	293	275	258	257	256	274	3	1
124	125	126	104	91	90	89	103	6	1	293	294	295	276	260	259	258	275	4	1
126	127	128	105	93	92	91	104	6	1	295	296	297	277	262	261	260	276	4	1
141	142	143	130	108	107	106	129	1	1	297	298	299	278	264	263	262	277	5	1
143	144	145	131	110	109	108	130	1	1	299	300	301	279	266	265	264	278	6	1
145	146	147	132	112	111	110	131	2	1	301	302	303	280	268	267	266	279	6	1
147	148	149	133	114	113	112	132	2	1	316	317	318	305	283	282	281	304	1	1
149	150	151	134	116	115	114	133	3	1	318	319	320	306	285	284	283	305	1	1
151	152	153	135	118	117	116	134	3	1	320	321	322	307	287	286	285	306	2	1
153	154	155	136	120	119	118	135	4	1	322	323	324	308	289	288	287	307	2	1
155	156	157	137	122	121	120	136	4	1	324	325	326	309	291	290	289	308	3	1
157	158	159	138	124	123	122	137	5	1	326	327	328	310	293	292	291	309	3	1
159	160	161	139	126	125	124	138	6	1	328	329	330	311	295	294	293	310	4	1
161	162	163	140	128	127	126	139	6	1	330	331	332	312	297	296	295	311	4	1
176	177	178	165	143	142	141	164	1	1	332	333	334	313	299	298	297	312	5	1
178	179	180	166	145	144	143	165	1	1	334	335	336	314	301	300	299	313	6	1
180	181	182	167	147	146	145	166	2	1	336	337	338	315	303	302	301	314	6	1
182	183	184	168	149	148	147	167	2	1	351	352	353	340	318	317	316	339	1	1
184	185	186	169	151	150	149	168	3	1	353	354	355	341	320	319	318	340	1	1
186	187	188	170	153	152	151	169	3	1	355	356	357	342	322	321	320	341	2	1
188	189	190	171	155	154	153	170	4	1	357	358	359	343	324	323	322	342	2	1
190	191	192	172	157	156	155	171	4	1	359	360	361	344	326	325	324	343	3	1

361 362 363 345 328 327 326 344 3 1	392 393 394 378 359 358 357 377 2 1
363 364 365 346 330 329 328 345 4 1	394 395 396 379 361 360 359 378 3 1
365 366 367 347 332 331 330 346 4 1	396 397 398 380 363 362 361 379 3 1
367 368 369 348 334 333 332 347 5 1	398 399 400 381 365 364 363 380 4 1
369 370 371 349 336 335 334 348 6 1	400 401 402 382 367 366 365 381 4 1
371 372 373 350 338 337 336 349 6 1	402 403 404 383 369 368 367 382 5 1
386 387 388 375 353 352 351 374 1 1	404 405 406 384 371 370 369 383 6 1
388 389 390 376 355 354 353 375 1 1	406 407 408 385 373 372 371 384 6 1
390 391 392 377 357 356 355 376 2 1	

(* Propiedades de materiales. Viga isoparamétrica *)
120000.0 0.30 1.2 0.60 0.018

(* Conectividades viga *)

106 107 1 2
107 108 1 2
108 109 1 2
109 110 1 2
110 111 1 2
111 112 1 2
112 113 1 2
113 114 1 2
114 115 1 2
115 116 1 2
116 117 1 2
117 118 1 2
118 119 1 2
119 120 1 2

En los siguientes anexos, se listan los programas de cada uno de los módulos de SEANEX. Los nombres de los procedimientos son combinación de palabras en español e identifican la función de cada uno de los procedimientos y no se pueden confundir con los comandos de Turbo Pascal.

ANEXO II

MÓDULO DE ALIMENTACIÓN

Lista de procedimientos utilizados en este módulo por orden alfabético y su funcionamiento.

Activar:	Procedimiento para iluminar la opción a elegir.
Adelante:	Procedimiento para activar la palabra siguiente.
Archivo1:	Escribe el archivo generado por las tarjetas al archivo final.
Archivo2:	Escribe el archivo generado por los elementos auxiliares al archivo final.
Agregar:	Agrega las coordenadas finales al archivo generado por el usuario.
AjustarX:	Procedimiento para ajustar coordenadas modo texto a modo gráfico. También (AjustarY).
Antes:	Permite desplegar la página anterior a la que se encuentra en pantalla.
Arreglo:	Procedimiento para afinar malla.
Borrar:	Procedimiento que permite especificar al usuario la dirección a borrar líneas.
BorrarX:	Procedimiento para especificar y borrar coordenadas en dirección X.
BorrarY:	Procedimiento para borrar coordenadas en dirección Y.
BorrarCoorY:	Borrar las coordenadas en dirección Y según el usuario lo requiera.
Cargar:	Procedimiento para detectar letreros utilizados.
Cargas_Con:	Captura cargas concentradas.
Cargas_Pre:	Captura cargas de presión.
Condicioni:	Captura condiciones iniciales.
Construcción:	Procedimiento que permite indicar los elementos que se excavan de manera gráfica.
CoorArc:	Procedimiento para interpretar las coordenadas de la malla de un archivo de MS-DOS.
CoorMalla:	Genera las coordenadas con elementos proporcionales.
CoorMallal:	Genera las coordenadas con elementos no proporcionales.

CoorY:	Genera coordenadas en dirección Y.
CoorYc:	Corrige las coordenadas en dirección Y.
Cuatronodos:	Genera incidencias para mallas con elementos de cuatro nodos.
Desactivar:	Procedimiento que desactiva la palabra parpadeante.
Detectar:	Inicializa los gráficos y determina el tipo de monitor que se va a utilizar.
Detener:	Aviso que indica que la información ha sido introducida.
Despues:	Permite desplegar la página siguiente a la que se encuentra en pantalla.
Desplegar:	Procedimiento para desplegar en pantalla las opciones.
Elemento:	Despliega página por página la información cargada en memoria.
Elemenviga:	Permite al usuario introducir los datos de elementos auxiliares.
Elegir:	Procedimiento para elegir la opción de cómo generar la malla.
Entrada:	Permite elegir la opción de malla a generar.
Enunciado:	Procedimiento que despliega los enunciados para introducir datos.
Escribir:	Despliega los datos, introducidos por el usuario, en pantalla.
Generación:	Dibuja la malla de elementos finitos.
Global:	Procedimiento que captura las características generales del problema en estudio.
Grados:	Genera los grados de libertad en dirección X y Y para cada elemento de la malla.
Iluminar:	Ilumina la posición del enunciado.
Imprimir:	Imprime en pantalla y en archivo las coordenadas generadas por la malla construida.
Incidencias:	Procedimiento para generar incidencias.
Info:	Procedimiento para dar información al usuario. Igual los procedimientos (Info2, Info4).
Insertar:	El usuario especifica la dirección a insertar líneas.
InsertarX:	Procedimiento para insertar coordenadas en X.
InsertarCoory:	Inserta las coordenadas en dirección Y según el usuario lo requiera.
InsertarY:	Permite que el usuario indique las líneas a insertar en dirección Y.
Instrucción:	Procedimiento para generar en la ventana de trabajo las instrucciones ejecución.
Leer:	Procedimiento para leer las coordenadas de la malla de un archivo de datos.

Leer1:	Procedimiento para localizar y leer el archivo de datos. Igual procedimiento (Leer2).
Llamada:	Crea la ventana auxiliar.
Limpiar2:	Limpia la ventana de trabajo.
Limpiartar:	Limpia la ventana de información.
Malla:	Permite dibujar la malla así como los elementos que se excavan y construyen.
Margen:	Crea un recuadro en la pantalla de trabajo.
Materiales:	Procedimiento que agrupa los materiales según el tipo de elemento de análisis.
Memoria:	Captura el espacio necesario en memoria.
Menu:	Relaciona la palabra elegida con la tecla especial para llamar el procedimiento en cuestión.
Modificar:	Permite indicar las líneas a modificar.
ModificarX:	Procedimiento para modificar coordenadas.
ModificarCoorY:	Modifica coordenadas en dirección Y.
ModificarY:	Permite al usuario indicar las modificaciones en dirección Y.
Mostrar:	Procedimiento para generar márgenes en ventana de información.
Nombre1:	Sale a MS-DOS y busca el nombre del archivo de coordenadas.
Nombre2:	Sale a MS-DOS y busca el nombre del archivo de incidencias.
Ochonodos:	Genera incidencias para mallas con elementos de ocho nodos.
Ordenar:	Procedimiento para buscar el número menor y mayor.
Pausa:	Procedimiento que permite al usuario ver la malla de forma pausada.
Procesarm:	Ordena las tareas a ejecutar dentro del módulo.
Proseguir:	Procedimiento para detectar las teclas de uso correcto.
Quedar:	Permite al usuario decidir que página desea desplegar en pantalla.
Regrasa:	procedimiento para activar la palabra anterior.
Restaurar:	Restaura la pantalla de modo gráfico a modo texto.
Resultitulo_Pro:	Procedimiento para escribir los datos introducidos por el usuario en el archivo auxiliar. Igual para los procedimientos Resulglobal, Resultmemo, Resultcargasp, Resultconi, Resultgloe, Resultelemenviga.
Tarea:	Procedimiento para ejecutar la opción elegida con la palabra parpadeante.

Tipo_Materi: Se indica gráficamente el tipo de material de que está constituida la malla.

Título_Pro: Procedimiento para capturar título del problema.

Transferencia: Acomoda los archivos auxiliares a un archivo final.

Truncardib: Ajusta las coordenadas de trabajo a coordenadas gráficas.

```
program Mallas(DATOS,OUTPUT);
($M 50000,0,30000)
```

```
uses crt,dos,graph;
```

```
Const
  nodos: array[1..8] of integer = (4,4,2,2,8,8,2,3);
```

```
type
  CCAL = array [0..700] of real;
  COORGR = array[1..700] of integer;
  CUADRITO = array[1..400,1..10] of integer;
  LETRA = String;
  ENTERO = integer;
```

```
var
```

```
J,J1,L,L2,K2,K,M,W,V,ELEMENTOB,ELEMENTOM,
ELEMENTOI:integer;
cefin,LLA,LIB,TIPO,FACE,COMIENZO,bb,b,envio,gen
ero:integer;
U,X,X1,X2,Y,Y1,Y2,ESPACIO,ESPACIOI:real;
XV,xf,xw,xamb,xa,YV,yf,yw,yamb,ya:CCAL;
XTM,YTM,XS,YS,MS,cviga,centrox,centroy:COORGR
;
CELDA:CUADRITO;
AL,BL:ENTERO;
salidarch:letra;
datos,datos1,datosf,datosa:TEXT;
```

```
procedure Detectar;
```

```
var
  D,M,maxcolors,ErrorCode:Integer;
```

```
begin
  D:=Detect;
  Initgraph(D,M,'BPBGF');
  ErrorCode:=GraphResult;
  if ErrorCode <> Grok then
    begin
      writeln('Error en el sistema de graficos. ');
      grapherrorMsg(errorcode);
      readln;
      halt(1);
    end;
  maxcolors:=getmaxcolor+1;
end;
```

```
procedure Pauss;
```

```
var
  CH:char;
```

```
begin
  while keypressed do CH:=readkey,
  CH:=readkey;
end;
```

```
procedure
Malla(XTM,YTM:COORGR;cefin,FACE:Integer);
```

```
const
  seleccion: set of char = ['c','C','#27];
```

```
type
  pasar = integer;
```

```
var
  G,estilo:integer;
  Punto,punto1:String;
  TECLA:char;
```

```
NUMERO,NUMERO1,NUMERO2,FINALX,FINALY,
CENTX,CENTY,nodo:pasar;
salir:boolean;
```

```
procedure Margen;
```

```
begin
  setcolor(1);
  setcolor(2);
  rectangle(1,1,getmaxx-1,getmaxy-1);
end;
```

procedure Generacion;

```
var
  NUM,I,R,CNFIN:integer;
```

```
begin
  Detectar;
  cleardevice;
  setbkcolor(1);
  Margen;
  case TIPO of
    1,2:begin
      CNFIN:=5;
      nodo:=3;
      end;
    5,6:begin
      CNFIN:=9;
      nodo:=5;
      end;
  end;
  setcolor(10);
  for NUM:= 1 to CEFIN do
    begin
      NUMERO1:=CELDA[NUM,1];
      NUMERO2:=CELDA[NUM,nodo];
      CENTX:=round((XTM[NUMERO1] +
      XTM[NUMERO2]) div 2);
      CENTY:=ROUND((YTM[NUMERO1] +
      YTM[NUMERO2]) div 2);
      str(NUM,PUNTO);
      outtextxy(CENTX,CENTY,PUNTO);
    end;
```

```
for I:= 1 to cefin do
  begin
    for R:= 1 to CNFIN do
      begin
        NUMERO:=CELDA[I,R];
        if R=1 then
          begin
            moveto(XTM[NUMERO],YTM[NUMERO]);
            FINALX:=XTM[NUMERO];
            FINALY:=YTM[NUMERO];
          end
        else if (R > 1) and (R<CNFIN) then
          begin
            Lineto(XTM[NUMERO],YTM[NUMERO]);
          end
        else
          begin
            Lineto(FINALX,FINALY);
          end;
      end;
    end;
```

```
end;
end;
```

procedure Construcion;

```
var
  Excava,Etapa:array[1..50] of integer;
  Contador,I,LL,llenado:integer;
  m:string;
```

```
begin
  Generacion;
  setcolor(2);
  for LL:=1 to b-1 do
    begin
      rectangle(180+(40*ll),430,200+(40*ll),450);
      setfillstyle(8+ll,3);
      floodfill(190+(40*ll),440,2);
      outtextxy(10,460,Esc:salir. C:continuar);
      str(ll+1,m);
      outtextxy(190+(40*ll),460,m);
      outtextxy(10,440,'etapas de contrucción');
    end;
    salir:=false;
    Contador:=0;
    I:=1;
    repeat
      setcolor(10);
      outtextxy(440,440,ELEMENTO: ');
      readln(Excava[I]);
      str(Excava[I],Punto);
      outtextxy(520,440,Punto);
      readln(etapa[I]);
      str(etapa[I],Punto1);
      outtextxy(536,440,punto1);
      Margen;
      delay(500);
      LLenado:=7+Etapa[I];
      setcolor(10);
      NUMERO1:=CELDA[EXCAVA[I],1];
      NUMERO2:=CELDA[EXCAVA[I],nodo];
      CENTX:=ROUND((XTM[NUMERO1] +
      XTM[NUMERO2]) div 2);
      CENTY:=ROUND((YTM[NUMERO1] +
      YTM[NUMERO2]) div 2);
      setbkcolor(1);
      setfillstyle(llenado,3);
      floodfill(CENTX,CENTY,10);
      setcolor(1);
      rectangle(510,440,550,450);
      setfillstyle(0,1);
      floodfill(530,445,1);
    repeat
      Tecla:=readkey;
      if not (Tecla in seleccion) then
        write(chr(7));
      until (tecla in seleccion);
      if Tecla = #27 then
        salir:=true;
```

```

I:=I+1;
until(salir);
I:=1;
while Contador < CEFIN+1 do
begin
  Contador:=Contador + 1;
  if Contador <> Excava[I] then
  begin
    case tipo of
      1,2 : Celda[Contador,6]:=1;
      5,6 : Celda[Contador,10]:=1;
    end;
  end
  else
  begin
    case tipo of
      1,2 : Celda[Contador,6]:= -Etapas[I];
      5,6 : Celda[Contador,10]:= -Etapas[I];
    end;
    I:=I+1;
  end;
end;
Closegraph;
end;

procedure Tipo_Materi;

var
  TMaterial,CMATE,LLENADO,LL,Contador:integer;
  M:string;

begin
  Generacion;
  setcolor(2);
  for LL:= 1 to bb do
  begin
    rectangle(180+(40*LL),430,200+(40*LL),450);
    setfillstyle(4+ll,3);
    floodfill(190+(40*LL),440,2);
    outtextxy(10,440,'Materiales:');(. C:continuar');
    outtextxy(10,460,'Esc:salir C:continuar');
    str(LL,M);
    outtextxy(190+(40*ll),460,M);
  end;
  salir:=false;
  CMATE:=0;
  repeat
  CMATE:=1+CMATE;
  setcolor(10);
  outtextxy(520,440,'ELEMENTO: ');
  str(CMATE,Punto);
  outtextxy(600,440,Punto);
  readln(TMaterial);
  Margen;
  delay(500);
  LLENADO:= 4+TMaterial;
  case tipo of
    1,2 : Celda[CMATE,5]:= TMaterial;
    5,6 : Celda[CMATE,9]:= TMaterial;
  end;

```

```

setcolor(10);
NUMERO1:=Celda[CMATE,1];
NUMERO2:=Celda[CMATE,nodo];
CENTX:=ROUND((XTM[NUMERO1] +
XTM[NUMERO2]) DIV 2);
CENTY:=ROUND((YTM[NUMERO1] +
YTM[NUMERO2]) DIV 2);
setfillstyle(LLENADO,3);
floodfill(CENTX,CENTY,10);
setcolor(1);
rectangle(600,440,610,450);
setfillstyle(0,1);
floodfill(606,445,1);
repeat
  Tecla:=READKEY;
  if not (tecla in seleccion) then
  write(chr(7));
  until (tecla in seleccion);
  if Tecla = #27 then
  Salir:=true;
  until(Salir);
Closegraph;
end;

begin
  if Face = 1 then
  begin
    Generacion;
    readln;
  end
  else
  begin
    Construcion;
    Tipo_Materi;
  end;
end;

{Procedimiento para generar archivo de datso}
{*****tarjetas*****}

procedure Mostrarar;

var
  P:integer;

begin
  window(13,20,62,20);
  textbackground(0);
  clrscr;
  textcolor(1);
  for p:= 1 to 49 do
  begin
    gotoxy(0+p,1);write('_');
  end;
  textcolor(11);
  gotoxy(45,1);write(')Esc(');
  window(13,21,62,24);
  textbackground(1);
  clrscr;

```

end;

procedure Instruccion;

```
begin
  window(7,3,67,3);
  textbackground(0);
  clrscr;
  textcolor(11);
  gotoxy(3,1);write('»Esc«');
  gotoxy(38,1);write('»Desplegar:AvPág RePág«');
end;
```

procedure LLamada;

```
begin
  window(15,8,48,18);
  textbackground(8);
  clrscr;
  window(14,7,47,17);
  textbackground(1);
  clrscr;
end;
```

procedure Limpiar2;

```
begin
  clrscr;
  window(1,1,80,25);
  textbackground(7);
  clrscr;
end;
```

procedure Mostrarlec;

```
begin
  Instruccion;
  window(8,5,68,19);
  textbackground(8);
  clrscr;
  window(7,4,67,18);
  textbackground(1);
  clrscr;
end;
```

procedure Limpiar1;

```
begin
  window(13,20,63,24);
  textbackground(7);
  clrscr;
  window(13,20,62,23);
  textbackground(7);
  clrscr;
end;
```

procedure Tarjetas(COMIENZO:integer);

type

```
F = array[1..10] of word;
F2 = array[1..10] of real;
FAUX = array[1..3,1..2] of integer;
PALABRA = array[1..10] of string[45];
LUGAR = array[1..10] of integer;
ca = string[25];
```

var

```
FICHERO,FICHEROE,FICHEROG:F;
FICHERO2:F2;
FICHEROA:FAUX;
L,RR,EV:INTEGER;
informacion,TITULO:ca;
regresa:boolean;
```

procedure Margen;

var

```
l:integer;
```

begin

```
  clrscr;
  textcolor(11);
  for l:= 1 to 75 do
  begin
    gotoxy(2+l,1); write(chr(205));
    gotoxy(3+l,25);write(chr(205));
    if l < 24 then
    begin
      gotoxy(2,1+i);write(chr(186));
      gotoxy(79,1+i); write(chr(186));
    end;
  end;
  gotoxy(79,25);write(chr(188));
  gotoxy(79,1);write(chr(187));
  gotoxy(2,1);write(chr(201));
  gotoxy(2,25);write(chr(200));
  gotoxy(3,1);write(chr(175));
  gotoxy(3,25);write(chr(175));
  gotoxy(77,1);write(chr(174));
  gotoxy(78,25);write(chr(174));
  window(3,1,78,1);
  textbackground(1);
  clrscr;
  textcolor(10);
  gotoxy(7,1);write('Esc: Siguiente');
  gotoxy(58,1);write('Retroceso:repetir');
end;
```

procedure Ventana(Informacion:ca;s,l:integer);

var

```

I:integer;
begin
  window(8,4,68,18);
  textbackground(11);
  clrscr;
  textcolor(11);
  for I:= 1 to 56 do
  begin
    gotoxy(3+I,1); write(chr(205));
    gotoxy(3+I,15);write(chr(205));
    if i < 14 then
    begin
      gotoxy(2,1+I);write(chr(186));
      gotoxy(60,1+I);write(chr(186));
    end;
  end;
  gotoxy(60,15);write(chr(188));
  gotoxy(60,1);write(chr(187));
  gotoxy(2,1);write(chr(201));
  gotoxy(2,15);write(chr(200));
  gotoxy(3,1);write(chr(175));
  gotoxy(3,15);write(chr(175));
  gotoxy(59,1);write(chr(174));
  gotoxy(59,15);write(chr(174));
  gotoxy(s,1);write(Informacion);
  window(11,5,65,17);
  textbackground(1);
  clrscr;
  textcolor(9);
  for I:= 1 to 51 do
  begin
    gotoxy(2+i,1);write(chr(196));
    gotoxy(2+i,13);write(chr(196));
    if i < 12 then
    begin
      gotoxy(2,1+i);write(chr(179));
      gotoxy(54,1+i);write(chr(179));
    end;
  end;
  gotoxy(2,1);write(chr(218));
  gotoxy(2,13);write(chr(192));
  gotoxy(54,1);write(chr(191));
  gotoxy(54,13);write(chr(217));
end;

```

Procedure Detener(a1,a2:integer);

```

begin
  textcolor(10);
  gotoxy(a1,a2);write(#220);
end;

```

Procedure Info;

```

begin
  Mostrar;
  textcolor(3);
  gotoxy(7,2); write(' NDM = 1 si unidimensional ');

```

```

  gotoxy(7,3); write(' NDM = 2 si es bidimensional ');
  gotoxy(7,4); write(' NDM = 3 si es tridimensional ');
end;

```

Procedure Info2;

```

begin
  mostrar;
  gotoxy(3,2);write(' Debe recordarse que la carga tiene signo');
  gotoxy(3,3);write(' y es congruente con la convención del signo');
  gotoxy(3,4);write(' del sistema global ');
end;

```

Procedure Info4;

```

begin
  mostrar;
  gotoxy(3,2);write('Es negativo si el esfuerzo es de compresión');
  gotoxy(3,3);write('Es positivo si la presión es de compresión');
end;

```

procedure Enunciado(A:integer;X,Y:LUGAR;PAL:PALABRA);

```

var
  T:integer;
begin
  for T:= 1 to A do
  begin
    gotoxy(X[T],Y[T]);write(PAL[T]);
  end;
end;

```

procedure Iluminar(I:integer;X,Y:LUGAR;PAL:PALABRA);

```

begin
  textcolor(11);
  gotoxy(X[I],Y[I]);write(PAL[I]);
end;

```

procedure Escribir(FICHEROE:F;I,B:integer);

```

var
  J:integer;
begin
  for J:= 1 to I-1 do
  begin
    gotoxy(50,J+B);write(FICHEROE[J]);
  end;

```

```

end;

procedure
Escribir2(FICHERO2:F2;D,I,B:INTEGER);

var
  J:integer;

begin
  for J:= D to I-1 do
    begin
      gotoxy(50,J+B);write(FICHERO2[J]:4:2);
    end;
  end;

procedure Proseguir(var adelante:boolean);

const
  seleccion: set of char = [#27,#8];

var
  tecla:char;

begin
  repeat
    tecla:=readkey;
    if not (tecla in seleccion) then
      write(chr(7));
    until(tecla in seleccion);
    if tecla = #27 then
      adelante:=true;
    end;

procedure Titulo_Pro(I:Integer);

var
  X,Y:lugar;
  pal:palabra;

begin
  clrscr;
  Margen;
  Ventana('Titulo',26,1);
  textcolor(3);
  X[1]:=15;Y[1]:=4;PAL[1]:=Nombre del archivo a
  generar ;
  X[2]:=15;Y[2]:=8;pal[2]:=Titulo del Problema ;
  Enunciado(2,X,Y,PAL);
  Iluminar(I,X,Y,pal);
end;

procedure Resultitulo_Pro;

var
  L:integer;

  tecla:char;
  adelante:boolean;

begin
  clrscr;
  Margen;
  L:=1;
  Titulo_Pro(i);
  adelante:=false;
  repeat
    for i:= 1 to 2 do
      begin
        case i of
          1:begin
            Titulo_Pro(i);
            gotoxy(15,6);readln(salidarch);
            end;
          2:begin
            assign(datos,'datos.a.dat');
            rewrite(datos);
            Titulo_Pro(i);
            gotoxy(15,6);write(salidarch);
            gotoxy(15,10);readln(TITULO);
            write(datos,TITULO);
            writeIn(datos);
            close(datos);
            end;
          end;
        end;
        Proseguir(adelante);
        until(adelante);
      end;

procedure Global(I:INTEGER);

var
  X,Y:LUGAR;
  PAL:PALABRA;
  T,A:integer;

begin
  Ventana('Control Global',24,1);
  textcolor(3);
  X[1]:=3;Y[1]:=3;PAL[1]:= Dimensión del problema
  ;
  X[2]:=3;Y[2]:=4;PAL[2]:= Número de puntos nodales
  ;
  X[3]:=3;Y[3]:=5;PAL[3]:= Número máximo de grados
  de libertad ;
  X[4]:=3;Y[4]:=6;PAL[4]:= Número total de grupos de
  elementos ;
  X[5]:=3;Y[5]:=7;PAL[5]:= Número total de etapas de
  análisis ;
  X[6]:=3;Y[6]:=8;PAL[6]:= Número máximo de
  iteraciones en cada etapa ;
  X[7]:=3;Y[7]:=9;PAL[7]:= Número total de cargas
  concentradas ;
  X[8]:=3;Y[8]:=10;PAL[8]:= Número total de cargas de
  presión ;

```

```

X[9]:=3;Y[9]:=11;PAL[9]:=' Número total de puntos
que definen ends
X[10]:=3;Y[10]:=12;PAL[10]:=' iniciales de esfuerzos
Enunciado(10,X,Y,PAL);
Iluminar(1,X,Y,PAL);
IF I=9 THEN
begin
gotoxy(3,11);write(PAL[9]);
gotoxy(3,12);write(PAL[10]);
end;
end;

```

procedure Resulglobal;

```

var
L:integer;
tecla:char;
adelante:boolean;

begin
clrscr;
Margen;
envio:=0;
I:=1;
Global(I);
adelante:=false;
repeat
for I:= 1 to 9 do
begin
case I of
1:begin
Info;
Global(I);
gotoxy(50,I+2);readln(FICHEROG[I]);
Limpiar;
end;
2..8:begin
Global(I);
Escribir(FICHEROG,I,2);
gotoxy(50,I+2);readln(FICHEROG[I]);
if i=5 then
B:=ficheroG[5];
if ficheroG[3]=3 then
envio:=1;
end;
9:begin
Global(I);
Escribir(FICHEROG,I,2);
gotoxy(50,12);readln(FICHEROG[I]);
end;
end;
end;
Detener(53,12);
Proseguir(adelante);
until(adelante);
Append(datos);
for I:= 1 to 9 do
begin

```

```

write(datos, ficheroG[i:5);
end;
writeln(datos);
close(datos);
end;

```

procedure Memoria(I:INTEGER);

```

var
X,Y:LUGAR;
PAL:PALABRA;

begin
Ventana(Memoria,26,1);
textcolor(3);
X[1]:=3;Y[1]:=5;PAL[1]:=' Número de espacio
necesario en memoria';
X[2]:=3;Y[2]:=6;PAL[2]:=' para el análisis';
X[3]:=3;Y[3]:=7;PAL[3]:=' Error previsto de
iteración';
Enunciado(3,X,Y,PAL);
IF I=1 THEN
Iluminar(1,X,Y,PAL);
Iluminar(I+1,X,Y,PAL);
end;

```

procedure Resultmemo;

```

var
I,A:integer;
adelante:boolean;
TECLA:CHAR;
FACTIBLE: array[1..2] of word;

begin
Margen;
I:=1;
Memoria(I);
adelante:=FALSE;
repeat
for I:= 1 to 2 do
begin
case I of
1:begin
Memoria(I);
gotoxy(45,I+5);readln(FACTIBLE[I]);
end;
2:begin
Memoria(I);
gotoxy(45,6);write(FACTIBLE[I-1]);
gotoxy(45,I+5);readln(FICHERO2[I]);
Detener(53,12);
end;
end;
end;
Proseguir(adelante);
until(adelante);
for A:= 1 to 2 do

```

```

begin
if A=1 then
begin
append(datos);
writeln(datos,FACTIBLE[A]);
end
else
write(datos,FICHERO2[A]:10);
end;
writeln(datos);
close(datos);
end;

```

procedure Cargas_Con(I:INTEGER);

```

var
X,Y:LUGAR;
PAL:PALABRA;

begin
Ventana('Cargas concentradas',20,1);
textcolor(3);
X[1]:=3;Y[1]:=6;pal[1]:=' Punto donde se aplica la
carga      :;
X[2]:=3;Y[2]:=7;pal[2]:=' Gdo de libertad de la
aplicación de la cga :;
X[3]:=3;Y[3]:=8;pal[3]:=' Número de etapa de la
aplicación de la cga :;
X[4]:=3;Y[4]:=9;pal[4]:=' Valor de la carga
:;
Enunciado(4,X,Y,PAL);
Iluminar(1,X,Y,PAL);
end;

```

procedure Resulcargasc;

```

var
I,r,a:integer;
adelante:boolean;
tecla:char;

begin
for r:= 1 to ficheroG[7] do
begin
if ficheroG[7] <> 0 then
begin
clrscr;
l:=1;
Cargas_Con(l);
adelante:=false;
repeat
for I:= 1 to 4 do
begin
case I of
1..3:begin
Cargas_Con(l);
Escribir(FICHERO,I,5);
gotoxy(50,I+5);readln(FICHERO[I]);

```

```

END;
4:begin
Info2;
Cargas_Con(l);
Escribir(FICHERO,I,5);
gotoxy(50,I+5);readln(FICHERO2[I]);
detener(53,12);
Limpiarar;
end;
end;
Proseguir(adelante);
until(adelante);
for a:= 1 to 4 do
begin
if a=1 then
append(datos);
if a < 4 then
write(datos,fichero[a]:5)
else
write(datos,fichero2[a]:10:2);
end;
writeln(datos);
close(datos);
end;
end;
end;

```

procedure Cargas_Pre(I:integer);

```

var
t:integer;
X,Y:LUGAR;
PAL:PALABRA;

begin
Ventana('Cargas de Presión',23,1);
textcolor(3);
X[1]:=3;Y[1]:=5;PAL[1]:=' Punto nodal donde se
aplica la carga      :;
X[2]:=3;Y[2]:=6;PAL[2]:=' Número de etapa de la
aplicación de la crga:;
X[3]:=3;Y[3]:=7;PAL[3]:=' Valor de la presión normal
:;
X[4]:=3;Y[4]:=8;PAL[4]:=' Valor de la presión
tangencial      :;
X[5]:=3;Y[5]:=9;PAL[5]:=' Valor de la presión
tangencial (si NDM > 2):;
Enunciado(5,X,Y,PAL);
Iluminar(1,X,Y,PAL);
end;

```

procedure Resulcargasp;

```

var
I,R,A:integer;
adelante:boolean;

```

```

begin
for R:= 1 to FICHEROG[8] do
begin
if FICHEROG[8] <> 0 then
begin
I:=1;
Cargas_Pre(I);
adelante:=FALSE;
repeat
for I:= 1 to 5 do
begin
case 1 OF
1,2:begin
Cargas_Pre(I);
Escribir(FICHERO,I,4);
gotoxy(50,I+4);readln(FICHERO[I]);
end;
3,4,5:begin
Cargas_Pre(I);
Escribir(FICHERO,3,4);
Escribir2(FICHERO2,3,I,4);
gotoxy(50,I+4);readln(FICHERO2[I]);
detener(53,12);
end;
end;
end;
Proseguir(ADELANTE);
until(ADELANTE);
for A:= 1 to 5 do
begin
if a=1 then
append(datos);
if A<3 then
write(datos,FICHERO[A]:5)
else
write(datos,FICHERO2[A]:10:2);
end;
writeln(datos);
close(datos);
end;
end;
end;

procedure Condicion1(I:Integer);

var
T:integer;
X,Y:LUGAR;
PAL:PALABRA;

begin
Ventana('Condiciones Iniciales',21,1);
textcolor(3);
X[1]:=3;Y[1]:=5;PAL[1]:= Ordenada en la dirección
de la gravedad ;
X[2]:=3;Y[2]:=6;PAL[2]:= Valor de esfuerzo total
vertical ;
X[3]:=3;Y[3]:=7;PAL[3]:= Valor del coeficiente de
empuje de reposo ;

```

```

X[4]:=3;Y[4]:=8;PAL[4]:= Valor de la presión del
agua ;
X[5]:=3;Y[5]:=9;PAL[5]:= Valor de la presión
preconsolidada ;
Enunciado(5,X,Y,PAL);
Iluminar(I,X,Y,PAL);
end;

```

procedure Resultoni;

```

var
I,R,A:integer;
adelante:boolean;

begin
for R:=1 to FICHEROG[9] do
begin
if FICHEROG[9] <> 0 then
begin
I:=1;
Condicion1(I);
adelante:=FALSE;
repeat
for I:= 1 to 5 do
begin
case 1 of
1,3 :begin
Condicion1(I);
Escribir2(FICHERO2,1,I,4);
gotoxy(50,I+4);readln(FICHERO2[I]);
Limpiarar;
end;
2,4,5:begin
Info4;
Condicion1(I);
Escribir2(FICHERO2,1,I,4);
gotoxy(50,I+4);readln(FICHERO2[I]);
if i=5 then
detener(53,12);
Limpiarar;
end;
end;
end;
Proseguir(adelante);
until(adelante);
for A:=1 to 5 do
begin
IF A=1 THEN
append(datos);
write(datos,FICHERO2[A]:10:2);
end;
writeln(datos);
close(datos);
end;
end;
end;
end;

```

```

procedure
Materiales(fichero:f;FICHEROA:FAUX);

type
    MATE = array[1..10] of real;

var
    I,R,M,A,LON,B:integer;
    PAL:PALABRA;
    MATERIAL:MATE;
    adelante:boolean;

begin
    Ventana('Propiedades',24,1);
    textcolor(3);
    PAL[1]:=Módulo de elasticidad ;
    PAL[2]:=Relación de Poisson ;
    PAL[3]:=Peso volumétrico ;
    PAL[4]:=Cohesión ;
    PAL[5]:=Resistencia a la tensión ;
    PAL[6]:=Factor de Warping ;
    PAL[7]:=Area transversal ;
    PAL[8]:=Momento de inercia ;
    adelante:=FALSE;
    for M:= 1 to FICHEROG[4] do
    begin
        for R:= 1 to FICHEROA[m,2] do
        begin
            repeat
                case FICHEROA[M,1] of
                    1,5 : begin
                        for I:= 1 to 4 do
                        begin
                            gotoxy(6,4+I);write(PAL[I]);
                            if I=4 then
                                begin
                                    gotoxy(6,4+I);write(PAL[I+1]);
                                end;
                            end;
                        for I:= 1 to 4 do
                        begin
                            if I<4 then
                                begin
                                    textcolor(11);
                                    gotoxy(6,4+I);write(PAL[I]);
                                end
                            else
                                begin
                                    textcolor(11);
                                    gotoxy(6,4+I);write(PAL[I+1]);
                                end;
                            end;
                        end;
                    end;
                end;
            gotoxy(38,4+I);readln(MATERIAL[I]);
            LON:=4;
            end;
            end;
            2,6:begin
                for I:= 1 to 5 do

```

```

begin
    gotoxy(6,4+I);write(PAL[I]);
end;
    for I:= 1 to 5 do
    begin
        textcolor(11);
        gotoxy(6,4+I);write(PAL[I]);
    end;
    gotoxy(38,4+I);readln(MATERIAL[I]);
    LON:=5;
    end;
    end;
    3:begin
        for I:= 1 to 5 do
        begin
            if I <= 2 then
                begin
                    gotoxy(6,4+I);write(PAL[I]);
                end
            else
                begin
                    gotoxy(6,4+I);write(PAL[3+I]);
                end;
            end;
        for I:= 1 to 5 do
        begin
            if I<3 then
                begin
                    textcolor(11);
                    gotoxy(6,4+I);write(PAL[I]);
                end
            else
                begin
                    textcolor(11);
                    gotoxy(6,4+I);write(PAL[3+I]);
                end;
            end;
        gotoxy(38,4+I);readln(MATERIAL[I]);
        LON:=5;
        end;
        end;
        7:begin
            for I:= 1 to 2 do
            begin
                if I=1 then
                    begin
                        gotoxy(6,4+I);write(PAL[I]);
                    end
                else
                    begin
                        gotoxy(6,4+I);write(PAL[7]);
                    end;
                end;
            for I:= 1 to 2 do
            begin
                if I=1 then
                    begin
                        textcolor(11);
                        gotoxy(6,4+I);write(PAL[I]);
                    end
                end
            end

```



```

end;
if eof(fichero) then
begin
close(fichero);
clrscr;
exit;
end;
end;

```

procedure Antes;

```

begin
if lia=1 then
begin
lia:=lia;
lib:=lib;
end
else
begin
lia:=lia-12;
lib:=lib-12;
end;
end;
end;

```

procedure Despues;

```

begin
lia:=lia+12;
lib:=lib+12;
end;

```

procedure Quedar(var regresa:boolean);

```

const
nulo=0;
arriba=#73;
abajo=#81;
casa=#119;
seleccion: set of char={#0,#27};
mover: set of char={arriba,abajo,casa};

```

```

var
X,Y:LUGAR;
PAL:PALABRA;
tecla:char;

```

```

begin
limpiar2;
lia:=1;
lib:=12;
Elemento;
regresa:=false;
repeat
repeat
tecla:=upcase(readkey);
until (tecla in seleccion);
case tecla of

```

```

#0:begin
tecla:=readkey;
if tecla in mover then
case tecla of
#81:begin
clrscr;
Despues;
Elemento;
end;
#73:begin
clrscr;
Antes;
Elemento;
end;
end
else
write(char(7));
end;
#27:regresa:=true;
end;
until(regresa);
Limpiar2;
end;

```

procedure Resultgloc(var FICHERO:A:FAUX);

```

const
CLAVE : array [1..8,1..5] of integer =
((4,2,3,3,2),(4,2,5,5,1),(2,3,5,3,1),(2,2,0,0,2),
(8,2,3,3,2),(8,2,5,5,4),(2,2,2,1,4),(3,2,0,0,3));

```

```

var
I,R,a:integer;
adelante:boolean;

```

procedure Identificador(var FICHERO:F);

```

var
G:integer;

```

```

begin
for G:= 1 to 5 do
begin
if G <= 2 then
begin
gotoxy(50,4+G);write(CLAVE[FICHERO{1},G]);
FICHERO[2+G]:=CLAVE[FICHERO{1},G];
end
else
Begin
gotoxy(50,5+G);write(CLAVE[FICHERO{1},G]);
FICHERO[3+G]:=CLAVE[FICHERO{1},G];
end;
end;
end;
end;

```

begin

```

for R:= 1 to fichero[4] do
begin
  I:=1;
  Control_Gloe(I);
  adelante:=FALSE;
  repeat
  gotoxy(49,3);write('Enter');
  readln;
  for I:= 1 to 8 do
  begin
    case I of
      1:begin
        Quedar(regresa);
        Margen;
        Control_GLOE(I);
        Escribir(FICHERO,I,2);
        gotoxy(50,I+2);readln(FICHERO[I]);
        FICHEROA[R,1]:=FICHERO[I];
        Limpiar;
        end;
      2,5:begin
        Control_Gloe(I);
        Identificador(FICHERO);
        Escribir(FICHERO,I,2);
        gotoxy(50,I+2);readln(FICHERO[I]);
        if (I=2) and (R>1) then
          EV:=FICHERO[2];
        if I=5 then
          begin
            FICHEROA[R,2]:=FICHERO[5];
            if r=1 then
              bb:=fichero[5];
            end;
          end;
      3,4,6,7,8:begin
        Control_Gloe(I);
        Identificador(FICHERO);
        Escribir(FICHERO,I,2);
        end;
    end;
  end;
  Detener(53,12);
  write(CHR(7));
  Proseguir(adelante);
  until(adelante);
  for A:= 1 to 8 do
  begin
    if a=1 then
      append(datos);
      write(datos,fichero[a];5);
    end;
    writeln(datos);
    close(datos);
  end;
end;

```

procedure Elemenviga(I:Integer);

var
X,Y:lugar;

```

PAL:palabra;

begin
  Ventana('Elemento Viga',24,1);
  textcolor(3);
  x[1]:=5;Y[1]:=5;PAL[1]:=Etapa de construcción de la
  viga:;
  X[2]:=5;Y[2]:=6;PAL[2]:=Material:;
  Enunciado(2,x,y,pal);
  Iluminar(I,X,Y,PAL);
end;

```

procedure Resulelemenviga;

var
INCIEV:array[1..50,1..4] of integer;
I,ECEV,MEV:integer;
tecla:char;
adelante:boolean;

```

begin
  clrscr;
  Margen;
  I:=1;
  Elemenviga(I);
  adelante:=false;
  repeat
  for I:= 1 to 2 do
  begin
    case I of
      1:begin
        Elemenviga(I);
        gotoxy(45,I+4);readln(ECEV);
        end;
      2:begin
        Elemenviga(I);
        gotoxy(45,5);write(ECEV);
        gotoxy(45,I+4);readln(MEV);
        Detener(53,12);
        end;
    end;
  end;
  Proseguir(adelante);
  until(adelante);
  for I:= 1 to EV do
  begin
    INCIEV[I,1]:=CVIGA[I];
    INCIEV[I,2]:=CVIGA[I+1];
    INCIEV[I,3]:=MEV;
    INCIEV[I,4]:=ECEV;
  end;
  append(datos1);
  for I:= 1 to EV do
  begin
    write(datos1,INCIEV[I,1];4);
    write(datos1,INCIEV[I,2];4);
    write(datos1,INCIEV[I,3];4);
    writeln(datos1,INCIEV[I,4];4);
  end;
end;

```

```

close(datos1);
end;

procedure Agregarc;

var
i:integer;

begin
for i:= 1 to w do
begin
append(datos);
write(datos, x[i]:5:2, y[i]:7:2);
writeln(datos, xs[i]:8, ys[i]:10,Ms[i]:12);
close(datos);
end;
assign(datosa,'coorf.dat');
rewrite(datosa);
for i:= 1 to w do
begin
writeln(datosa, x[i]:5:2, y[i]:7:2);
end;
close(datosa);
end;

```

```

{Programa principal de tarjetas}
begin
textmode(10);
clrscr;
Textbackground(7);
if COMIENZO = 1 then
Quedar(REGRESA)
else
begin
Resultitulo_Pro;
Resulglobal;
Resulinemo;
Agregarc;
Resulcargasc;
Resulcargasp;
Resulconi;
Resulglor(FICHEROA);
Materiales(FICHERO,FICHEROA);
if FICHEROG[3]=3 then
Resulemenviga;
end;
end;

```

procedure Limpiar;

```

begin
window(1,1,80,25);
textbackground(7);
textcolor(0);
end;

```

Procedure Marco;

```

begin
window(10,3,60,3);
textbackground(0);
clrscr;
end;

```

Procedure Mostrar(COLOR:ENTERO);

```

var
C,:integer;

begin
window(1,1,80,25);
textbackground(7);
clrscr;
textcolor(11);
for i:= 1 to 75 do
begin
gotoxy(2+i,1); write(chr(205));
gotoxy(3+i,25);write(chr(205));
if i < 24 then
begin
gotoxy(2,1+i);write(chr(186));
gotoxy(79,1+i); write(chr(186));
end;
end;
gotoxy(79,25);write(chr(188));
gotoxy(79,1);write(chr(187));
gotoxy(2,1);write(chr(201));
gotoxy(2,25);write(chr(200));
gotoxy(3,1);write(chr(175));
gotoxy(3,25);write(chr(175));
gotoxy(77,1);write(chr(174));
gotoxy(78,25);write(chr(174));
textcolor(11);
gotoxy(31,1);write('Generador de Mallas');
window(14,5,64,5);
textbackground(0);
clrscr;
Textcolor(1);
for C:= 1 to 50 do
begin
gotoxy(0+C,1);write('_');
end;
textcolor(11);
gotoxy(44,1);write('»Enter«');
window(15,7,65,21);
textbackground(8);
clrscr;
window(14,6,64,20);
textbackground(COLOR);
clrscr;
end;

```

procedure Mostrar1c;

```

begin
  window(13,10, 61,16);
  text(background(8));
  clrscr;
  window(12,9,60,15);
  text(background(1));
  clrscr;
end;

procedure Truncardib(XW,YW:CCAL);

var
  NUMERADORX,NUMERADORY:real;

begin
  for l:= 1 to w do
    begin
      numeradorX:=(xw[l] - xw[W]);
      XTM[L]:=round((numeradorX*590)/(xw[l]-
xw[W])+20);
      NUMERADORY:=YW[L] - YW[W];
      YTM[L]:=ROUND(390-
(NUMERADORY*380)/(YW[l]-YW[W]));
    end;
  end;
end;

```

procedure Imprimir(XW,YW:CCAL);

```

begin
  M:=M + 1;
  write(M);
  write(XW[L]:10:2);
  write(datos,XW[L]:5:2);
  write(YW[K]:12:2);
  write(datos,YW[K]:7:2);
end;

```

procedure CoorY(XA:CCAL);

```

var
  I,FINAL:integer;

```

```

begin
  ESPACIO:=((Y2-Y1) / (k2));
  U:=(L+1)/2;
  V:=trunc(U);
  if tipo in [5,6] then
    begin
      if (U < V) then
        begin
          for K:= 0 to K2 do
            begin
              YV[k]:=Y2-Espacio*k;
              Imprimir(XA,YV);
              W:=W+1;
              XF[W]:=XA[L];
              YF[W]:=YV[K];
            end;
          end;
        end;
      else
        begin
          for k:= 0 to k2 do
            begin
              YA[k]:=YV[k*2];
              imprimir(xa,yA);
              w:=w+1;
              xf[w]:=xv[l];
              yf[w]:=ya[k];
            end;
          end;
        end;
      end;
    end;
end;

```

```

end;
end
else
begin
  ESPACIO1:=ESPACIO*2;
  for K:= 0 to (K2 DIV 2) do
    begin
      YV[k]:=Y2 - Espacio1*k;
      Imprimir(XA,YV);
      W:=W+1;
      XF[W]:=XA[L];
      YF[W]:=YV[K];
    end;
  end;
end;
if tipo in [1,2] then
begin
  for k:= 0 to k2 do
    begin
      yv[k]:=y2 - espacio*k;
      imprimir(xa,yv);
      w:=w+1;
      xf[w]:=xa[l];
      yf[w]:=yv[k];
    end;
  end;
end;
end;
end;

```

procedure CoorYc(xa:CCAL);

```

var
  i:integer;

```

```

begin
  u:=(l+1)/2;
  v:=trunc(u);
  if (u < v) or (tipo in [1,2]) then
    begin
      for k:= 0 to k2 do
        begin
          imprimir(xa,yv);
          w:=w+1;
          xf[w]:=xv[l];
          yf[w]:=yv[k];
        end;
      end;
    end;
  else
    begin
      for k:= 0 to (k2 div 2) do
        begin
          YA[k]:=YV[k*2];
          imprimir(xa,yA);
          w:=w+1;
          xf[w]:=xv[l];
          yf[w]:=ya[k];
        end;
      end;
    end;
  end;
end;
end;

```

```
procedure CoorYs(XA,Yc:CCAL);
```

```
var
l:integer;

begin
U:=(L+1)/2;
V:=trunc(U);
if (U<>V) or (tipo in [1,2]) then
begin
for K:= 0 to K2 do
begin
YV[K]:=YC[K];
Imprimir(XA,YV);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
end;
end
else
begin
for K:= 0 to (K2 DIV 2) do
begin
YV[K]:=YC[K*2];
Imprimir(XA,YV);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
end;
end;
end;
```

{Genera y corrige incidencias}

```
procedure
Incidencias(ELEMENTOB,ELEMENTOM,ELEME
NTOI:INTEGER;
```

```
var CELDA:CUADRITO;var
cefin:integer);
```

```
var
XX,CONTADOR,CONTADOR2,CP,INC,ELE:integer;
```

```
procedure ochonodos;
```

```
var
ce:integer;

begin
CONTADOR:=0;
CONTADOR2:=1;
XX:=0;
Cefin:=(L2*K2) div 4);
ELE:=ELEMENTOB+ELEMENTOI+J1;
for cc:=1 to Cefin do
begin
if cc=1 then
contador:=0
```

```
else
contador:=contador+1;
if contador > (ELE-1) then
begin
contador:=0;
contador2:=contador2+ELE;
end;
if (cc<>1) and (cc=contador2) then
begin
INC:=(CC-1) div ELE;
xx:=(3*ELE+2)*INC;
end;
celda[cc,1]:=(3*ELE+3)+(contador*2)+xx;
celda[cc,2]:=celda[cc,1]+1;
celda[cc,3]:=celda[cc,2]+1;
celda[cc,4]:=celda[cc,3]-(3+(ELE-1))-contador;
celda[cc,5]:=celda[cc,4]-(2*ELE)+contador;
celda[cc,6]:=celda[cc,5]-1;
celda[cc,7]:=celda[cc,6]-1;
celda[cc,8]:=celda[cc,7]+(2*ELE+1)-contador;
end;
Limpiar;
end;
```

```
procedure cuatronodos;
```

```
var
ce:integer;

begin
contador:=0;
contador2:=0;
xx:=0;
cefin:=(L2*K2) ;
ele:=elementob+elementoi+j1;
for cc:= 1 to cefin do
begin
contador:=contador+1;
if contador > ele then
begin
contador:=1;
contador2:=contador2+1;
xx:=((ele+1)*contador2);
end;
celda[cc,1]:=ele+1+contador+xx;
celda[cc,2]:=celda[cc,1]+1;
celda[cc,3]:=celda[cc,2] - (ele+1);
celda[cc,4]:=celda[cc,2] - (ele+2);
end;
end;
```

```
begin
case tipo of
1,2:cuatronodos;
5,6:ochonodos;
end;
end;
```

```

Procedure
BorrarcoorY(NB1,IB1,RB1:INTEGER;xamb:CCAL
);

```

```

var
AUX,KB,I,K1:integer;
YB:CCAL;

```

```

begin
  U:=(L+1)/2;
  V:=trunc(U);
  KB:=0;
  K:=0;
  if tipo in [5,6] then
    begin
      if (U < V) then
        begin
          if L=0 then
            begin
              m:=0;
              w:=0;
              for I:=0 to 100 do
                YA[I]:=0;
                K2:=K2-2;
                AUX:=(NB1-2) - ((IB1-1)*2);
                for KB:= 0 to K2 do
                  begin
                    if KB <= AUX then
                      YA[KB]:=YV[KB];
                    if (AUX<KB) and (KB=AUX+1) then
                      YA[KB]:=((YV[KB+3] - YV[KB-1]) / 2) +
(YV[KB-1]);
                    if KB> AUX+1 then
                      YA[KB]:=YV[KB+2];
                    end;
                  end;
                for K:=K2+1 to 100 do
                  YV[K]:=0;
                for K:= 0 to K2 do
                  begin
                    YV[K]:=YA[K];
                    if IB1=RB1 then
                      begin
                        Imprimir(XAMB,YV);
                        writeln(datos);
                        W:=W+1;
                        XF[W]:=XV[L];
                        YF[W]:=YV[K];
                        readln;
                      end;
                    end;
                  end;
                if V=U then
                  begin
                    for KB:=0 to (K2 div 2) do
                      YB[KB]:=YA[KB*2];
                    for K:=0 to (K2 div 2) do
                      begin
                        if IB1=RB1 then
                          begin

```

```

                        Imprimir(XAMB,YB);
                        writeln(datos);
                        W:=W+1;
                        XF[W]:=XV[L];
                        YF[W]:=YB[k];
                        readln;
                      end;
                    end;
                  end;
                if tipo in [1,2] then
                  begin
                    if L=0 then
                      begin
                        M:=0;
                        W:=0;
                        K2:=K2-1;
                        aux:=(NB1-2)-(IB1-1);
                        for Kb:= 0 to K2 do
                          begin
                            if KB <= aux then
                              YA[KB]:=YV[KB]
                            else
                              YA[KB]:=YV[KB+1];
                            end;
                          end;
                        for K:= 0 to K2 do
                          begin
                            YV[k]:=YA[k];
                            if IB1 = RB1 then
                              begin
                                Imprimir(Xamb, YV);
                                writeln(datos);
                                W:=W+1;
                                XF[W]:=XV[L];
                                YF[W]:=YV[K];
                                readln;
                              end;
                            end;
                          end;
                    end;
                  end;
                end;

```

```

Procedure BorrarY(var
ACTIVA,ELEMENTO:integer); (*LIN HORIZ*)

```

```

var
RB1,NB1,IB1:integer;
XAMB:CCAL;

begin
  clrscr;
  Mostrar(3);
  textcolor(8);
  gotoxy(9,2);write('BORRANDO LINEAS
HORIZONTALES');
  textcolor(15);
  assign(datos,'COORDE.dat');
  gotoxy(4,5);WRITE('No líneas a borrar (dirección "X"
:');

```

```

readln(RB1);
ELEMENTOB:=RB1+ELEMENTOB;
for IB1:= 1 to RB1 do
begin
gotoxy(4,7);write('Linea ' , IB1 , ' a borrar
');
readln(NB1);
writeln;
rewrite(datos);
for L:= 0 to L2 do
begin
Xamb[L]:=XV[L];
BorrarcoorY(NB1,IB1,RB1,xamb);
end;
end;

incidencias(elementob,elementom,elementoi,celda,cefin
);
Truncardib(XF,YF);
ACTIVA:=1;
Malla(XTM,YTM,cefin,1);
close(datos);
end;

{Procedimientos que modifican la geometría de los
elementos}

procedure BorrarX(B:CHAR; var
ACTIVA,ELEMENTOB:integer);

var
RB,IB,NB,AUX,LB:integer;

begin
clrscr;
Mostrar(3);
assign(datos,'COORDE.DAT');
textcolor(8);
gotoxy(9,2);
write('BORRANDO LINEAS VERTICALES');
textcolor(15);
gotoxy(4,5);write('No líneas a borrar (dirección "Y") :
');
readln(RB);
rewrite(datos);
for IB:= 1 to RB do
begin
gotoxy(4,7);write('Linea ' , IB , ' a borrar
');
readln(NB);
textcolor(15);
if tipo in [5,6] then
begin
AUX:=(NB-2)-((IB-1)*2);
L2:=L2 - 2;
if (IB<>1) then
begin
for L:= 0 to (L2*2) do
XV[L]:=XA[L];
end;
end;
end;
end;
end;

```

```

for LB:= 0 to L2 do
begin
if LB <= AUX then
XA[LB]:= XV[LB];
if (AUX < LB) and (LB=(AUX+1)) then
XA[LB]:= ((XV[LB+3] - XV[LB-1]) / 2) +
(XV[LB-1]);
if LB > AUX+1 then
XA[LB]:= XV[LB+2];
end;
end;
if tipo in [1,2] then
begin
aux:=(NB-2)-(IB-1);
L2:=L2-1;
for LB:= 0 to l2 do
begin
if LB<= aux then
XA[LB]:=XV[LB]
else
XA[LB]:=XV[LB+1];
end;
end;
for LB:= 0 to L2 do
XV[LB]:=XA[LB];
end;
M:=0;
w:=0;
if B='V' then
begin
for L:= 0 to L2 do
begin
XV[L]:=XA[L];
if ACTIVA=1 then
begin
CoorYc(XV);
ELEMENTOB:=ELEMENTOB;
end;
if ACTIVA=0 then
begin
CoorY(XV);
ELEMENTOB:=0;
end;
end;
end;
end;

Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Truncardib(XF,YF);
Malla(XTM,YTM,cefin,1);
close(datos);
end;
if B='A' then
begin
BorrarY(ACTIVA,ELEMENTOB);
end;
end;
end;

```

```

procedure
ModificarCoorY(nm1,IM1,RM1:integer;MO1:REA
L;xamb:CCAL);

```

```

var
km,I:integer;

```

```

begin
U:=(L+1)/2;
V:=trunc(U);
K2:=K2;
KM:=0;
if tipo in [5,6] then
begin
if (U > V) then
begin
if L=0 then
begin
W:=0;
M:=0;
end;
if (IM1 < 1) and (L=0) then
begin
for K:= 0 to (K2*2) do
YV[K]:=YA[K];
end;
for KM:= 0 to K2 do
begin
if (KM <= (nm1-2)) or ((NM1+2) <= KM) then
begin
YA[KM]:=YV[KM];
end;
if ((NM1-1)=KM) then
begin
YA[KM]:=YA[KM-1] - ((YA[KM-1]- MO1) / 2);
end;
if (NM1=KM) then
begin
YA[KM]:=MO1;
end;
if ((NM1+1)=KM) then
begin
YA[KM]:=((YV[KM+1]-YA[KM-1]) / 2) +
YA[KM-1];
end;
end;
end;
for K:=0 to K2 do
begin
YV[K]:=YA[K];
if IM1=RM1 then
begin
imprimir(XAMB,YA);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
readln;
end;
end;
end;

```

```

end;
if V=U then
begin
for K:= 0 to K2 div 2 do
begin
YA[K]:=YV[K*2];
if RM1=IM1 then
begin
imprimir(XAMB,YA);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YA[K];
readln;
end;
end;
end;
end;
if tipo in [1,2] then
begin
if l=0 then
begin
W:=0;
M:=0;
end;
for KM := 0 to K2 do
begin
if (KM <= (NM1-2)) or (NM1 <= KM) then
YA[KM]:=YV[KM];
if ((NM1-1) = KM) then
YA[KM]:= MO1;
end;
for K:= 0 to K2 do
begin
YV[K]:=YA[K];
if IM1 = RM1 then
begin
imprimir(Xamb,YA);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
readln;
end;
end;
end;
end;

```

```

procedure ModificarY(var
activa,ELEMENTOM:integer);

```

```

var
RM1,IM1,NM1:integer;
MO1:real;
XAMB:CCAL;

```

```

begin
Mostrar(3);
IM1:=0;

```

```

assign(datos,'COORDE.DAT');
textcolor(8);
gotoxy(9,2);
write('MODIFICANDO LINEAS HORIZONTALES');
textcolor(15);
gotoxy(4,5);write('No de líneas a modificar (dirección
"X") : ');
readln(RM1);
for IM1:= 1 to RM1 do
begin
gotoxy(4,7);write('Línea ', IM1 , ' a modificar
:');
readln(NM1);
gotoxy(4,9);write('Nuevo Valor
');
readln(MO1);
writeln;
writeln;
rewrite(datos);
for L:= 0 to L2 do
begin
Xamb[l]:=XV[l];
ModificarCoorY(NM1,IM1,RM1,MO1,Xamb);
end;
end;
Truncardib(XF,YF);
activa:=1;

```

```

Incencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Malla(XTM,YTM,cefin,1);
close(datos);
end;

```

```

procedure ModificarX(var MD:CHAR; var
activa,ELEMENTOM:integer);

```

```

var
RM,NM,IM,LM:integer;
MO:real;
xamb:CCAL;

```

```

begin
clrscr;
Mostrar(3);
assign(datos,'COORDE.DAT');
textcolor(8);
gotoxy(9,2);
write('MODIFICANDO LINEAS VERTICALES');
textcolor(15);
gotoxy(4,5);write('No de líneas a modificar (dirección
"Y") : ');
readln(RM);
rewrite(datos);
for IM:= 1 to RM do
begin
gotoxy(4,7);write('Línea ', IM , ' a modificar
:');
readln(NM);

```

```

gotoxy(4,9);write('Nuevo Valor
');
readln(MO);
if tipo in [5,6] then
begin
if IM<1 then
begin
for L:= 0 to (L2*2) do
XV[L]:=XA[L];
end;
for LM:= 0 to L2 do
begin
if (LM <= (NM-2) ) or ((NM+2) <= LM) then
XA[LM]:=XV[LM];
if ((NM-1)=LM) then
XA[LM]:=XA[LM-1] - ((XA[LM-1] - MO) / 2);
if (NM=LM) then
XA[LM]:=MO;
if ((NM+1) = LM) then
XA[LM]:=((XV[LM+1]-XA[LM-1]) / 2) +
XA[LM-1];
end;
end;
if tipo in [1,2] then
begin
for LM:= 0 to L2 do
begin
if (LM <= (NM-2) ) or ((NM) <= LM) then
XA[LM]:=XV[LM];
if ((NM-1) = LM) then
XA[LM]:=MO;
end;
end;
for LM:= 0 to L2 do
XV[LM]:=XA[LM];
end;
M:=0;
W:=0;
if MD='V' then
begin
for L:= 0 to L2 do
begin
XV[L]:=XA[L];
if activa = 1 then
CoorYc(XV);
if activa = 0 then
coorY(XV);
end;
Truncardib(XF,YF);

```

```

Incencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Malla(XTM,YTM,cefin,1);
close(datos);
end;
if MD='A' then
begin
ModificarY(ACTIVA,ELEMENTOM);
end;
end;

```

```

procedure
InsertarCoorY(RI1,II1:integer;NI1,LTE1,LTE2:rea
l;XAMB:CCAL);

```

```

var
KI,I,IC,NN,CON,AUX:integer,
YI:CCAL;

```

```

begin
U:=(L+1)/2;
V:=TRUNC(U);
if tipo in [5,6] then
begin
if (U<>V) then
begin
IC:=0;
if L=0 then
begin
W:=0;
M:=0;
for I:=0 to 100 do
begin
YA[I]:=0;
end;
K2:=K2+2;
IC:=0;
CON:=0;
for KI:= 0 to K2 do
begin
if KI=0 then
begin
YA[KI]:=YV[KI];
CON:=KI;
AUX:=0;
end;
if YV[KI+1] >= NI1 then
begin
YA[KI]:=YV[KI];
CON:=KI;
AUX:=0;
end;
if ((YV[KI+1] < NI1) and (KI<>0)) and (IC<2)
then
begin
YA[KI]:=YA[KI-1]+((YA[CON] - NI1)/2);
IC:=IC+1;
AUX:=KI;
end;
if (YV[KI+1] <= NI1) and ((CON+IC<KI) and
(IC<3)) then
begin
YA[KI]:=YA[KI-1] - ((YA[KI-1]-
YV[CON+IC])/2);
IC:=IC+1;
AUX:=KI;
end;
if (IC=3) and (AUX < KI) then
begin

```

```

YA[KI]:=YV[KI-2];
end;
end;
end;
for K:= K2+1 to 100 do
YV[K]:=0;
for K:= 0 to K2 do
begin
YV[K]:=YA[K];
if II1=RI1 then
begin
Imprimir(XAMB,YV);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
readln;
end;
end;
end;
if V=U then
begin
for KI:= 0 to (K2 div 2) do
begin
YI[KI]:=YA[KI*2];
end;
for K:= 0 to (K2 div 2) do
begin
if II1=RI1 then
begin
Imprimir(XAMB,YI);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YI[K];
readln;
end;
end;
end;
if tipo in [1,2] then
begin
if L=0 then
begin
K2:=K2+1;
ni:=0;
w:=0;
for KI:= 0 to K2 do
begin
if KI=0 then
begin
YA[KI]:=YV[KI];
aux:=0;
end;
if (YV[KI] <= NI1) and (aux=0) then
begin
YA[KI]:=NI1;
aux:=1;
end;
if (YV[KI] > NI1) and (aux=0) then

```

```

YA[KI]:=YV[KI];
if (YV[KI-1] < NI1) and (aux=1) then
YA[KI]:=YV[KI-1];
end;
end;
for K:= 0 to K2 do
begin
YV[K]:=YA[K];
if II=RI1 then
begin
Imprimir(xamb,yv);
writeln(datos);
W:=W+1;
XF[W]:=XV[L];
YF[W]:=YV[K];
readln;
end;
end;
end;
end;

```

procedure InsertarY(var
ACTIVA,ELEMENTOI:integer);

```

var
RI1,II1:integer;
NI1,LTE1,LTE2:real;
xamb:CCAL;

begin
clrscr;
Mostrar(3);
Textcolor(8);
gotoxy(9,2);
write('INSERTANDO LINEAS HORIZONTALES');
textcolor(15);
assign(datos,'COORDE.DAT');
gotoxy(4,5);write('No de líneas a insertar (dirección
"X"): ');
readln(RI1);
ELEMENTOI:=RI1+ELEMENTOI;
rewrite(datos);
for II1:= 1 to RI1 do
begin
gotoxy(4,7);write('Valor de la línea ', II1 , ' a
insertar : ');
readln(NI1);
gotoxy(4,9);
writeln;
for L:= 0 to L2 do
begin
Xamb[l]:=XV[l];
InsertarCoorY(RI1,II1,NI1,LTE1,LTE2,xamb);
end;
end;
Truncardib(XF,YF);
ACTIVA:=1;

```

```

Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Malla(XTM,YTM,ccfin,1);
close(datos);
end;

```

procedure InsertarX(Z:char; var
ACTIVA,ELEMENTOI:integer);

```

var
RI,II,LI,IC,NC,NN,CON,AUX:integer;
NI,LTE1,LTE2:real;
xamb:CCAL;

begin
clrscr;
Mostrar(3);
assign(datos,'COORDE.DAT');
textcolor(8);
gotoxy(9,2);
write('INSERTANDO LINEAS VERTICALES ');
textcolor(15);
gotoxy(2,5);write('No de líneas a insertar (dirección
"Y"): ');
readln(RI);
rewrite(datos);
for II:= 1 to RI do
begin
gotoxy(2,7);write('Valor de la línea ', II , ' a insertar
: ');
readln(NI);
if tipo in [5,6] then
begin
L2:=L2+2;
for L:= 0 to 100 do
XA[L]:=0;
IC:=0;
for LI:= 0 to L2 do
begin
if LI=0 then
begin
XA[LI]:=XV[LI];
CON:=LI;
AUX:=0;
end;
if XV[LI+1] >= NI then
begin
XA[LI]:=XV[LI];
CON:=LI;
AUX:=0;
end;
if ((XV[LI+1] < NI) and (LI<0)) and (IC<2) then
begin
XA[LI]:= XA[LI-1] - ((XA[CON] - NI) / 2);
IC:=IC+1;
AUX:=LI;
end;
end;

```

```

    if (XV[LI+1] <= NI) and ((CON+IC<LI) and
(IC<3)) then
    begin
        XA[LI]:=XA[LI-1]-((XA[LI-1]-XV[CON+IC])/2);
        IC:=IC+1;
        AUX:=LI;
    end;
    if (IC=3) and (AUX<LI) then
    begin
        XA[LI]:=XV[LI-2];
    end;
end;
if tipo in [1,2] then
begin
    L2:=L2+1;
    for LI:= 0 to l2 do
    begin
        if LI=0 then
        begin
            XA[LI]:=XV[LI];
            aux:=0;
        end;
        if (XV[LI] <= NI) and (aux=0) then
        begin
            XA[LI]:=NI;
            aux:=1;
        end;
        if (XV[LI] > NI) and (aux=0) then
            XA[LI]:=XV[LI];
        if (XV[LI-1] < NI) and (aux=1) then
            XA[LI]:= XV[LI-1];
        end;
    end;
    for LI:= 0 to L2 do
        XV[LI]:=XA[LI];
    for LI:= L2+1 to 100 do
        XV[LI]:=0;
    end;
    M:=0;
    W:=0;
    if Z='V' then
    begin
        for L:=0 to L2 do
        begin
            XV[L]:=XA[L];
            if ACTIVA = 1 then
            begin
                CoorYc(XV);
                ELEMENTOI:=ELEMENTOI;
            end;
            if ACTIVA = 0 then
            begin
                CoorY(XV);
                ELEMENTOI:=0;
            end;
        end;
        Truncardib(XF,YF);

```

```

Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Malla(XTM,YTM,cefin,1);
close(datos);
end;
if Z='A' then
begin
    InsertarY(ACTIVA,ELEMENTOI);
end;
end;

```

procedure Insertar(var
ACTIVA,ELEMENTOI:integer);

```

var
Z:char;

begin
    clrscr;
    MostrarIC;
    repeat
        textmode(10);
        textbackground(7);
        clrscr;
        MostrarIC;
        textcolor(7);
        gotoxy(4,3);write('INSERTAR LINEA HORZ, VERT,
AMBAS O SALIR? ');
        Z:=UpCase(readkey);
        case Z of
            'V': InsertarX(Z,ACTIVA,ELEMENTOI);
            'H': InsertarY(ACTIVA,ELEMENTOI);
            'A': InsertarX(Z,ACTIVA,ELEMENTOI);
        end;
    until Z = 'S';
end;

```

procedure Modificar(var
ACTIVA,ELEMENTOM:integer);

```

var
MD:char;

begin
    clrscr;
    MostrarIC;
    repeat
        textmode(10);
        textbackground(7);
        clrscr;
        MostrarIC;
        textcolor(7);
        gotoxy(4,3);write('MODIFICAR LINEA HORZ, VERT,
AMBAS O SALIR? ');
        MD:=UpCase(readkey);
        case MD of
            'V': ModificarX(MD,activa,ELEMENTOM);

```

```

    'F': ModificarY(activa,ELEMENTOM);
    'A': ModificarX(MD,activa,ELEMENTOM);
end;
until MD = 'S';
end;

```

```

procedure Borrar(var
ACTIVA,ELEMENTOB:integer);

```

```

var
B:char;

```

```

begin
  clrscr;
  Mostrarlc;
  repeat
    textmode(10);
    textbackground(7);
    clrscr;
    Mostrarlc;
    textcolor(7);
    gotoxy(4,3); write('BORRAR LINEAS
HONZ.,VERT.,AMBAS,SALIR ? ');
    B:=UpCase(readkey);
    clrscr;
    Limpiar;
    case B of
      'V': BorrarX(B,ACTIVA,ELEMENTOB);
      'F': BorrarY(ACTIVA,ELEMENTOB);
      'A': BorrarX(B,ACTIVA,ELEMENTOB);
    end;
  until B='S';
end;

```

```

procedure Arreglo;

```

```

var
OPCION:char;
ACTIVA:integer;

```

```

begin
  textmode(10);
  textbackground(7);
  clrscr;
  ACTIVA:=0;
  ELEMENTOB:=0;
  ELEMENTOM:=0;
  ELEMENTOI:=0;
  repeat
    Mostrar(1);
    Textcolor(11);
    gotoxy(14,4);write('1) MODIFICAR LINEAS' );
    gotoxy(14,6);write('2) INSERTAR LINEAS' );
    gotoxy(14,8);write('3) BORRAR LINEAS' );
    textcolor(3);gotoxy(14,10);write('4) SIN CAMBIOS' );
    textcolor(15+128);
    gotoxy(20,14);write('ELIJA UNA OPCION:');
  repeat

```

```

    OPCION:=readkey
  until OPCION in ['1'..'4'];
  case OPCION of
    '1': Modificar(ACTIVA,ELEMENTOM);
    '2': Insertar(ACTIVA,ELEMENTOI);
    '3': Borrar(ACTIVA,ELEMENTOB);
  end;
  until OPCION = '4'
end;

```

```

procedure Grados;

```

```

type
  cadena=string;
  posicion = integer;

```

```

var

```

```

LIM,DX,DY,DT,dt1,dt2,PN,Nn,xg,yg,ct,nodales,pnv,i,
E,secciones:integer;
ACUMU,Nn1,seccion:integer;
RES,RES1,res2,vigas:char;
GRADOX,aux:array[1..600] of integer;
gradoxr,gradoY,gradoT:array[1..100] of integer;

```

```

procedure esquema(cadenat:cadena;x,y:posicion);

```

```

var

```

```

q:integer;

```

```

begin
  Mostrar(1);
  textcolor(7);
  gotoxy(x,y);write(cadenat);
  textcolor(11);
  for q:= 1 to 26 do
  begin
    gotoxy(12+q,6);write(chr(205));
    gotoxy(12+q,10);write(chr(205));
  end;
  for q:= 1 to 3 do
  begin
    gotoxy(12,6+q);write(chr(186));
    gotoxy(39,6+q);write(chr(186));
  end;
  gotoxy(12,6);write(chr(201));
  gotoxy(39,6);write(chr(187));
  gotoxy(12,10);write(chr(200));
  gotoxy(39,10);write(chr(188));
  gotoxy(18,7);write('No Punto');
end;

```

```

procedure libre;

```

```

var
dx,dt:integer;

```

```

begin
repeat;
clrscr;
textcolor(7);
gotoxy(2,3);write(' Puntos Nodales Fijos en X:? (S/n)
');
res:= upcase(readkey);
until res in ['S','N'];
if RES ='S' then
begin
gotoxy(2,5);writeln('Introduce los puntos que definen
la frontera');
for dx:= 1 to 4 do
begin
esquema('Introducir los puntos que definen la
frontera',2,3);
textcolor(7);
gotoxy(18,9);write(dx);
gotoxy(30,9);readln(gradoxr[dx]);
end;
for dx:= gradoxr[1] to gradoxr[2] do
begin
gradox[dx]:=dx;
end;
for dx:= gradoxr[3] to gradoxr[4] do
begin
gradox[dx]:=dx;
end;
end;
repeat;
mostrar(1){(1,1,'')};
textcolor(7);
gotoxy(2,4);write('Puntos nodales totalmante fijos
varian:? (S/N): ');
res2:=upcase(readkey);
until res2 in ['S','N'];
if res2 = 'S' then
begin
mostrar(1);
textcolor(15);
for dt:= dt1 to dt2 do
begin
esquema('Introducir los puntos ',12,3);
textcolor(7);
gotoxy(18,9);write(dt);
gotoxy(30,9);readln(gradot[dt]);
end;
end
else
begin
case tipo of
1,2: begin
for dt:= dt1 to dt2 do
begin
ct:=ct+(j1+1);
gradot[dt]:=ct;
end;
end;
5,6: begin
for dt:= dt1 to dt2 do

```

```

begin
u:=(dt)/2;
v:=trunc(u);
if u<>v then
begin
ct:=ct+(j1*2)+1);
gradot[dt]:=ct;
end
else
begin
ct:=ct+(j1+1);
gradot[dt]:=ct;
end;
end;
end;
end;
end;

procedure Libret;

var
np:integer;

begin
n:=0;
for np:= Nn1 to Nn do
begin
n:=n+1;
case tipo of
1,2: begin
if np <= (ACUMU+(j1+1)) then
pn:=np;
end;
5,6: begin
if np <= (ACUMU+(j1*2)+1)) then
pn:=np;
end;
end;
if gradox[np]=np then
begin
xs[m]:=1;
ys[m]:=0;
end
else if gradoy[np]=np then
begin
xs[m]:=0;
ys[m]:=1;
end
else if gradot[pn]=np then
begin
xs[m]:=1;
ys[m]:=1;
pn:=pn+1;
end
else
begin
xs[m]:=0;
ys[m]:=0;

```



```

readln;
Tarjetas(COMIENZO);
Mostrar(1);
textcolor(7);
gotoxy(8,2);write('Tipo de Elemento :');
gotoxy(27,2);readln(TIPO);
assign(datos,'COORDE.DAT');
textcolor(11);
gotoxy(8,4);write('Dar las coordenadas de frontera');
gotoxy(10,6);write('Valor de X1:= ');
gotoxy(10,7);write('Valor de X2:= ');
gotoxy(10,8);write('Valor de Y1:= ');
gotoxy(10,9);write('Valor de Y2:= ');
gotoxy(26,6);readln(X1);
gotoxy(26,7);readln(X2);
gotoxy(26,8);readln(Y1);
gotoxy(26,9);readln(Y2);
gotoxy(4,11);write('Número de elementos (en direc x)
EX:= ');
gotoxy(4,12);write('Número de elementos (en direc y)
EY:= ');
gotoxy(42,11);readln(J);
gotoxy(42,12);readln(J1);
ELEMENTOB:=0;
ELEMENTOI:=0;
M:=0;
W:=0;
X:=X2;
if tipo in [1,2] then
begin
L2:=j;
K2:=j1;
end
else
begin
L2:=J*2;
K2:=J1*2;
end;
FACE:=1;
textcolor(7);
rewrite(datos);
for L:=0 to L2 do
begin
ESPACIO:=((X2-X1)/(L2));
XV[l]:=X2-Espacio*l;
writeln;
CoorY(XV);
end;

Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,cefin);
Truncardib(XF,YF);
Malla(XTM,YTM,cefin,FACE);
readln;
close(datos);
end;

```

procedure CoorMalla;

```

type
CADENA = string[30];

var
X,Y,XC,YC:CCAL;
CADEN:CADENA;
comienzo:integer;

procedure Letrero(CADEN:CADENA);

begin
window(14,14,63,14);
textbackground(7);
textcolor(15);
gotoxy(13,1);write(CADEN);
end;

begin
textbackground(7);
clrscr;
Mostrar(1);
assign(datos,'COORDE.DAT');
textcolor(7);
COMIENZO:=1;
gotoxy(8,8);write('Enter para mostrar tipos de
elementos');
readln;
Tarjetas(COMIENZO);
Mostrar(1);
textcolor(7);
gotoxy(8,2);write('Tipo de elemento :');
gotoxy(27,2);readln(TIPO);
gotoxy(8,5);write('Número de elementos (direcc x): ');
gotoxy(40,5);readln(J);
gotoxy(8,7);write('Número de elementos (direcc y): ');
gotoxy(40,7);readln(J1);
Letrero('COORDENADAS EN "X"');
Textcolor(7);
for L:= 0 to J do
begin
window(20,17,30,17);
textbackground(1);
gotoxy(4,1);write('X',L+1);
window(38,17,43,17);
textbackground(0);
clrscr;
if tipo in [1,2] then
readln(X[l]);
if tipo in [5,6] then
readln(X[L*2]);
end;
Letrero('COORDENADAS EN "Y"');
textcolor(7);
for K:= 0 to J1 do
begin
window(20,17,30,17);
textbackground(1);
clrscr;

```

```

gotoxy(4,1);write('Y:',K+1);
window(38,17,43,17);
textbackground(0);
clrscr;
if tipo in [1,2] then
begin
  readln(Y[k]);
  L2:=J;
  K2:=J1;
  y2:=y[0];
  y1:=y[j1];
end;
if tipo in [5,6] then
begin
  L2:=J*2;
  K2:=J1*2;
  readln(Y{K*2});
  y1:=y[k*2];
  y2:=y[0];
end;
end;
M:=0;
ELEMENTOB:=0;
ELEMENTOI:=0;
if tipo in [5,6] then
begin
  for L:= 0 to J*2 do
  begin
    U:=(L+1)/2;
    V:=trunc(U);
    if U<>V Then
    begin
      Xc[L]:=X[L];
    end
    else
      Xc[L]:=X[L-1] - (X[L-1] - X[L+1])/2;
    end;
  for K:= 0 to j1*2 do
  begin
    U:=(K+1)/2;
    V:=trunc(U);
    if U<>V then
    begin
      Yc[K]:=Y[K];
    end
    else
      Yc[K]:= Y[K-1] - ((Y[K-1] - Y[K+1])/2);
    end;
  end;
end;
if tipo in [1,2] then
begin
  for L:= 0 to j do
  xc[l]:=x[l];
  for k:= 0 to j1 do
  yc[k]:=y[k];
end;
rewrite(datos);
M:=0;

```

```

for L:= 0 to L2 do
begin
  XV[L]:=Xc[L];
  CoorYs(XV,YC);
end;

Incidenciaa(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,CEFIN);
Truncardib(XF,YF);
Malla(XTM,YTM,CEFIN,1);
readln;
close(datos);
end;

procedure CoorArch;

var
ARCHIVO:string[25];
numep,Error,nm,cy,cx,clec,aux,comienzo:integer;
lectura:text;

begin
  Elementob:=0;
  Elementoi:=0;
  clrscr;
  Mostrar(1);
  comienzo:=1;
  textcolor(7);
  gotoxy(8,8);write(" Enter para mostrar tipos de
  elementos ");
  readln;
  tarjetas(comienzo);
  repeat
  clrscr;
  Mostrar(1);
  textcolor(7);
  gotoxy(3,4);write('Nombre del archivo ');
  {$I-}
  gotoxy(25,4);readln(archivo);
  assign(lectura, archivo);
  reset(lectura);
  {$I+}
  Error:=IOResult;
  if Error <> 0 then
  begin
    gotoxy(3,6);write(archivo, 'no se puede encontrar');
    gotoxy(3,8);write('Pruebe de nuevo');
    readln;
  end;
  until Error = 0;
  gotoxy(3,6);write('Numero de puntos nodales : ');
  readln(numep);
  gotoxy(3,8);write('Tipo de Elemento :');
  readln(tipo);
  nm:=0;
  cy:=0;
  cx:=0;
  aux:=0;
  while not eof(lectura) do

```

```

begin
  nm:=nm+1;
  readln(lectura, xf[nm],yf[nm]);
end;
for clec:= 1 to numep do
begin
  if (xf[clec] = xf[clec+1]) and (Aux=0) then
  begin
    yv[cy]:=yf[clec];
    cy:=cy+1;
  end;
  if xf[clec] <> xf[clec+1] then
  begin
    if aux=0 then
    begin
      yv[cy]:=yf[clec];
      aux:=1;
    end;
    xv[cx]:=xf[clec];
    cx:=cx+1;
  end;
  if clec=numep then
  xv[cx]:=xf[clec];
end;
case tipo of
  1,2:begin
    j:=cx;
    j1:=cy;
  end;
  5,6:begin
    j:= cx div 2;
    j1:= cy div 2;
  end;
end;
L2:=cx;
K2:=cy;
w:=numep;

```

```

Incidencias(elementob,elementom,elementoi,celda,cefin
);
  Truncardib(xf,yf);
  Malla(xtm,ytm,cefin,1);
  readln;
end;

```

procedure Restaurar;

```

begin
  textmode(10);
  clrscr;
  textbackground(7);
  clrscr;
end;

```

procedure Elegir(var OPCION:integer);

```

var
  I:integer;

```

```

begin
  window(19,10,58,17);
  textbackground(7);
  clrscr;
  textcolor(8);
  for l:= 1 to 35 do
  begin
    gotoxy(3+l,1);write(chr(205));
    gotoxy(3+l,8);write(chr(205));
  end;
  for l:= 1 to 6 do
  begin
    gotoxy(3,l+1);write(chr(186));
    gotoxy(38,l+1);write(chr(186));
  end;
  gotoxy(3,1);write(chr(201));
  gotoxy(3,8);write(chr(200));
  gotoxy(38,1);write(chr(187));
  gotoxy(38,8);write(chr(188));
  repeat
    textcolor(15);
    gotoxy(8,3);write('1) Teclar datos ');
    gotoxy(8,5);write('2) Lectura de archivo');
    textcolor(1);
    gotoxy(31,7);write('» «');
    gotoxy(33,7);readln(opcion);
    until OPCION in [1..2];
  end;

```

procedure Generator;

```

var
  J,K,L2,K2,n,XT1,XT2,YT1,YT2,pne:integer;

```

procedure Leer(var n:integer; var centrox,centroy:coorg; var celda:cuadrto);

```

type
  item = real;
  nombred = string;

```

```

var
  lectura:text;
  nlume,nluna,ne, nma,nx1 ,nx2,ny1 ,ny2:item;
  j,puntos,Error:integer;
  archivo,archivo1:nombred;

```

procedure Nombre(var archivo:nombred);

```

begin
  Mostrar(1);
  repeat
  clrscr;
  Mostrar(1);
  textcolor(7);
  gotoxy(3,4);write('Archivo de coordenadas: ');

```

```

{!-}
gotoxy(28,4);readln(archivo);
assign(lectura, archivo);
reset(lectura);
{!+}
Error:=IOResult;
if Error <> 0 then
begin
gotoxy(3,6);write(archivo, 'no se puede encontrar');
gotoxy(3,8);write('Pruebe de nuevo');
readln;
end;
until Error = 0;
end;

```

procedure Nombre1(var archivol:nombred);

```

begin
mostrar(1);
repeat
clrscr;
Mostrar(1);
textcolor(7);
gotoxy(3,4);write('Archivo de incidencias: ');
{!-}
gotoxy(28,4);readln(archivol);
assign(lectura, archivol);
reset(lectura);
{!+}
Error:=IOResult;
if Error <> 0 then
begin
gotoxy(3,6);write(archivo, 'no se puede encontrar');
gotoxy(3,8);write('Pruebe de nuevo');
readln;
end;
until Error = 0;
textcolor(7);
gotoxy(3,6);write('Puntos nodales del elemento:');
gotoxy(33,6);readln(pne);
case pne of
8: tipo:=5;
4: tipo:=1;
end;
end;
end;

```

**procedure Leer1(var j:integer;var
xf,yf:CCAL;archivo:nombred);**

```

begin
repeat
assign(lectura, archivo);
{!-}
reset(lectura);
{!+}
until IOResult = 0;
j:=0;
while not eof(lectura) do

```

```

begin
j:=j+1;
read(lectura, x[f[j],y[f[j]]);
end;
w:=j;
close(lectura);
end;

```

**procedure Leer2(var n:integer;var
celda:cuadrilo;archivol:nombred);**

```

var
k:integer;

begin
repeat
assign(lectura,archivol);
{!-}
reset(lectura);
{!+}
until IOResult = 0;
n:=0;
while not eof(lectura) do
begin
n:=n+1;
for k:= 1 to pne do
begin
read(lectura,celda[n,k]);
end;
end;
cefin:=n;
close(lectura);
end;

```

**procedure Ordenar(var valor:CCAL; j:integer; var
nme,nma:real);**

```

var
i,n:integer;

```

procedure Intercambio(var valor,d:item);

```

var
aux:item;

```

```

begin
aux:=valor;
valor:=d;
d:=aux;
end;

```

```

begin
for i:= 1 to j-1 do
for n:= i+1 to j do
if valor[i] > valor[n] then
Intercambio(valor[i], valor[n]);
nme:=valor[1];
nma:=valor[j];

```

```

end;

procedure Ajustarx(n1,n2:real;p:CCAL;j:integer;
var centrox:coogr);

var
paso,relacion,aux:real;
i,gener:integer;

begin
paso:=n2-n1;
for i:=1 to j do
begin
relacion:=p[i]-nme;
aux:=(relacion*580)/paso)+20;
centrox[i]:=round(aux);
end;
end;

procedure Ajustary(n1,n2:real;p:CCAL;j:integer;
var centroy:coogr);

var
paso,relacion,aux:real;
i:integer;

begin
paso:=n2-n1;
for i:=1 to j do
begin
relacion:=p[i]-nme;
aux:=380-(relacion*380)/paso;
centroy[i]:=round(aux);
end;
end;

begin
Nombre(archivo);
Nombre1(archivo1);
Leer1(j,xf,yf,archivo);
Ordenar(xf,j,nme,nma);
nx1:=nme;
nx2:=nma;
Ordenar(yf,j,nme,nma);
ny1:=nme;
ny2:=nma;
Leer1(j,xf,yf,archivo);
Ajustarx(nx1,nx2,xf,j,centrox);
Ajustary(ny1,ny2,yf,j,centroy);
Leer2(n,celda,archivo1);
write(chr(7));
textcolor(11);
gotoxy(18,12);write("Fin de lectura");
readln;
end;

```

```

procedure
Malla1(XT1,XT2,YT1,YT2,L2,R2,n:integer;centrox,
centroy:coogr);

type
pasar = integer;

var
R,l:integer;
numero,finalx,finaly,NUMERO1,NUMERO2,CENTX,
CENTY:pasar;
PUNTO:string;

begin
cleardevice;
setbkcolor(1);
setfillstyle(0,1);
floodfill(3,1,7);
setcolor(2);
rectangle(1,1,getmaxx-1,getmaxy);
setcolor(10);
SetViewPort(20,40,600,400,false);
for l:=1 to n do
begin
for R:= 1 to (pne+1) do
begin
numero:=celda[i,r];
if r=1 then
begin
moveto(centrox[numero],centroy[numero]);
finalx:=centrox[numero];
finaly:=centroy[numero];
pausa;
end
else if (r>1) and (r < (pne+1)) then
begin
lineto(centrox[numero],centroy[numero]);
end
else
begin
lineto(finalx,finaly);
end;
end;
end;
end;

(*****Progra Prncipal del Generador*****)

begin
Leer(n,centrox,centroy,celda);
Detectar;
Malla1(XT1,XT2,YT1,YT2,L2,K2,n,centrox,centroy);
Pausa;
closegraph;
end;

```

procedure Entrada;

```
var
PAL: array[0..5] of string[35];
X1,Y1:array[0..5] of integer;
I,OPCION:integer;
LLAMADA:string;
```

procedure Cargar;

```
begin
  X1[0]:=16;Y1[0]:=2; PAL[0]:='
  X1[1]:=12;Y1[1]:=6; PAL[1]:='Rectangulares y
  Proporcionales';
  X1[2]:=12;Y1[2]:=8; PAL[2]:='Rectangulares no
  proporcionales';
  X1[3]:=12;Y1[3]:=10;PAL[3]:='Generador (Elementos
  diferentes)';
  X1[4]:=12;Y1[4]:=12;PAL[4]:='Continuar';
end;
```

procedure Desplegar;

```
var I:integer;
begin
  TextBackground(1);
  Mostrar(1);
  clrscr;
  textcolor(11);
  for I:=0 to 4 do
  begin
    if I=0 then
      textcolor(3)
    else
      textcolor(11);
    gotoxy(X1[I],Y1[I]);write(PAL[I])
  end;
end;
```

procedure Activar(I:Integer);

```
begin
  TextBackground(3);
  textcolor(9);
  gotoxy(X1[I],Y1[I]);write(PAL[I])
end;
```

procedure Desactivar(var I:integer);

```
begin
  TextBackground(1);
  textcolor(11);
  gotoxy(X1[I],Y1[I]);write(PAL[I])
end;
```

procedure Adelante(var I:Integer);

```
begin
  Desactivar(I);
  if I=4 then I:=1 else I:=I+1;
  Activar(I);
end;
```

procedure Regresa(var I:integer);

```
begin
  Desactivar(I);
  if I=1 then I:=4 else I:=I-1;
  Activar(I);
end;
```

procedure Tareas(var I:integer);

```
begin
  case I of
    1:begin
      CoorMalla;
      Restaurar;
      Desplegar,I:=1;
      Activar(I);
    end;
    2:begin
      Elegir(OPCION);
      case OPCION of
        1:begin
          CoorMalla;
          Restaurar;
          Desplegar,I:=1;
          Activar(I);
        end;
        2:begin
          CoorArch;
          Restaurar;
          Desplegar,I:=1;
          Activar(I);
        end;
      end;
    end;
    3:begin
      Generador;
      genero:=1;
      readln;
      Restaurar;
      Desplegar,I:=1;
      Activar(I);
    end;
  end;
end;
```

procedure Menu;

```
const
SELECCION:set of char =[#72,#80,#13];
```

```

var TECLA:char;
    SALIR:boolean;

begin
    SALIR:=FALSE;
    repeat
    repeat
    TECLA:=readkey;
    until(TECLA in SELECCION);
    case TECLA of
        #80: Adelante(1);
        #72: Regresa(1);
        #13: if l= 4 then SALIR:=TRUE else
TAREAS(1);
    end;
    until(SALIR);
end;

begin
    genero:=0;
    Mostrar(1);
    Cargar;
    Desplegar;
    l:=1;Activar(1);
    Menu;
end;

procedure Agregari;

var
    i,h,final:integer;

begin
    if genero = 0 then
    begin
Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT
OI,CELDA,cefin);
        Truncardi(XF,YF);
        Malla(XTM,YTM,cefin,0);
    end;
    if genero = 1 then
    begin
        Malla(centrox,centroy,cefin,0);
    end;
    case tipo of
        1,2: final:=6;
        3,6: final:=10;
    end;
    for i:= 1 to cefin do
    begin
        append(datos);
        for h:= 1 to final do
            write(datos,celda[i,h]:5);
            writeln(datos);
        end;
        assign(datos,'inciden.dat');
        rewrite(datos);
        for i:= 1 to cefin do
            begin
                for h:= 1 to (final-2) do
                    write(datos,celda[i,h]:5);
                    writeln(datos);
                end;
                close(datos);
                close(datos);
            end;

procedure Transferencia;

var
    get:integer;
    linea:string;
    datost,f3:text;

procedure Archivo1;

begin
    assign(datos,'datossa.dat');
    reset(datos);
    assign(datost, salidarch);
    rewrite(datost);
    while not eof(datos) do
    begin
        readln(datos,linea);
        writeln(datost,linea);
    end;
    close(datos);
    close(datost);
end;

procedure archivo2;

begin
    get:=1;
    assign(datos,'datossa.dat');
    reset(datos);
    assign(datost, salidarch);
    rewrite(datost);
    assign(datos1,'datos2e.dat');
    reset(datos1);
    assign(datost, salidarch);
    rewrite(datost);
    while not eof(datos) do
    begin
        readln(datos,linea);
        writeln(datost,linea);
    end;
    while not eof(datos1) do
    begin
        readln(datos1,linea);
        writeln(datost,linea);
    end;
    close(datos);
    close(datos1);
    close(datost);
end;

```

```
end;  
begin  
  if envio = 1 then  
    archivo2;  
  if envio = 0 then  
    archivo1;  
end;
```

procedure Procesarm;

```
begin  
  if genero = 0 then  
    begin  
      arreglo;  
      grados;
```

```
Incidencias(ELEMENTOB,ELEMENTOM,ELEMENT  
OI,CELDA,cefin);  
  Truncardib(XF,YF);  
  Malla(XTM,YTM,cefin,1);  
  readln;  
end;  
  if genero = 1 then  
    begin  
      Grados;  
    end;  
end;
```

{***PROGRAMA PRINCIPAL*****}**

```
begin  
  Entrada;  
  procesarm;  
  Tarjetas(COMIENZO);  
  Agregari;  
  transferencia;  
end.
```

ANEXO IV

MÓDULO DE GRAFICACIÓN

Lista de procedimientos utilizados en este módulo por orden alfabético

- Acomodo:** Procedimiento para acomodar los vectores de coordenadas desplazadas.
- Adelantar:** Ajusta el decimal del número calculado para la malla deformada.
- Ajuste:** Cierra el decimal del número calculado.
- Calculo_Coorde:** Calcula las coordenadas desplazadas para dibujar la malla deformada.
- Colocación:** Procedimiento que calcula el punto (x, y) en la escala gráfica.
- Crear_Imagenes:** Crea las imágenes que representarán los puntos de los valores colocados.
- Deformada:** Procedimiento que permite crear la malla deformada.
- Detectar:** Inicializa los gráficos y determina el tipo de monitor que se va a utilizar.
- Ejes:** Procedimiento que dibuja y ajusta la escala de los ejes coordenados.
- Entragraf:** Inicia la presentación del módulo.
- Graficar:** Procedimiento que encierra todo el proceso para generar las gráficas.
- Horizontal:** Dibuja el eje horizontal.
- Intercambio:** Procedimiento auxiliar del procedimiento ordenar, que permite ordenar los valores analizados.
- Infoentra:** Genera la ventana de trabajo.
- Leer_Datos:** Procedimiento para leer el archivo de datos.
- LecturaDesplazos:** Calcula las coordenadas desplazadas para la malla deformada.
- Limpiar:** Limpia la ventana de trabajo.
- Opcion:** Procedimiento que permite elegir la opción a graficar.
- Ordenar_Grupo:** Crea vectores auxiliares para buscar el número menor y mayor de cada grupo de valores.
- Ordenar_Valor:** Procedimiento que ordena los valores de menor a mayor.

Pausa:	Procedimiento que crea una pausa en los gráficos hasta que el usuario oprima cualquier tecla.
Puntos:	Procedimiento que recupera las imágenes creadas y el valor correspondiente de cada punto.
PosicionX:	Ajusta las coordenadas en dirección X para modo gráfico.
PosicionY:	Ajusta las coordenadas en dirección Y para modo gráfico.
Titulos:	Procedimiento que configura los títulos dados por el usuario a modo gráfico.
Titulos1:	Títulos para indicar al usuario la secuencia de entrada de datos. También (Titulos2).
Seleccion	Selecciona la longitud de la escala calculada.
Ventana:	Procedimiento para crear la ventana de trabajo.
Vertical:	Dibuja el eje vertical.

```
Program GRAFICO(DATOS,OUTPUT);
{$M 50000,0,40000}
```

```
uses crt,dos,graph;
```

```
type
nom = array[1..6] of string[50];
```

```
var
nme,nma:real;
EjePrin:char;
GRUPO:integer;
cadena: nom;
```

```
procedure Detectar;
```

```
var
D,M,MAXCOLORS:integer;
```

```
begin
D:=Detect;
Initgraph(D,M,'bp\bgi');
if Graphresult<>Grok then halt(1);
Maxcolors:=getmaxcolor+1;
end;
```

```
procedure Pausa;
```

```
var
CH:char;
begin
while keypressed do CH := readkey;
CH:= readkey;
end;
```

```
procedure Entragraf;
```

```
var
fichero:text;
caracter:string;
i,m,b:integer;
```

```
begin
setbkcolor(0);
setfillstyle(11,1);
floodfill(1,1,1);
setcolor(11);
settextstyle(1,0,0);
```

```
outtextxy(100,200,'GRAFICACION DE
RESULTADOS');
```

```
readln;
cleardevice;
setbkcolor(0);
setfillstyle(11,1);
floodfill(1,1,1);
setcolor(11);
rectangle(118,148,482,352);
rectangle(120,150,480,350);
setbkcolor(0);
setfillstyle(9,1);
floodfill(180,100,11);
settextstyle(0,0,0);
assign(fichero,'inicio.tex');
reset(fichero);
m:=0;
b:=0;
for i:= 1 to 400 do
begin
m:=m+1;
b:=b+10;
if m=3 then
begin
m:=1;
end;
readln(fichero,caracter);
outtextxy(120+m,156+b,caracter);
end;
close(fichero);
readln;
end;
```

```
procedure Infoentra;
```

```
var
i:integer;
```

```

begin
  textbackground(7);
  clrscr;
  textcolor(9);
  for i:= 1 to 15 do
  begin
    gotoxy(5,5+i);
    write(chr(186));
  end;
  for i:= 1 to 15 do
  begin
    gotoxy(70,5+i);
    write(chr(186));
  end;
  for i:= 1 to 65 do
  begin
    gotoxy(5+i,5);
    write(chr(205));
  end;
  for i:= 1 to 65 do
  begin
    gotoxy(5+i,20);
    write(chr(205));
  end;
  gotoxy(6,5);write(chr(175));
  gotoxy(6,20);write(chr(175));
  gotoxy(69,5);write(chr(174));
  gotoxy(69,20);write(chr(174));
  gotoxy(5,5);write(chr(201));
  gotoxy(70,5);write(chr(187));
  gotoxy(5,20);write(chr(200));
  gotoxy(70,20);write(chr(188));
  gotoxy(7,5);write('Enter');
  window(12,8,63,8);
  textbackground(11);
  clrscr;
  window(12,9,63,17);
  textbackground(1);
  clrscr;
end;

procedure Titulos1;

var
tit: array[1..6] of string[50];

begin
  Infoentra;
  textcolor(7);
  gotoxy(4,3);write('GRAFICAR:');
  gotoxy(18,2);write('Asentamiento Colindancia');
  gotoxy(18,3);write('Desplazamiento Lateral');
  gotoxy(18,4);write('Movimiento De Fondo');
  gotoxy(4,6);write('No De Etapas a Analizar');
  textcolor(3);
  gotoxy(18,2);write('A');
  gotoxy(18,3);write('D');
  gotoxy(18,4);write('M');

  textcolor(9);
  gotoxy(46,4);write('n ');
  gotoxy(46,7);write('n ');
  textcolor(3);
  gotoxy(48,4);readln(EjePrin);
  EjePrin:=UpCase(EjePrin);
  gotoxy(48,7);readln(GRUPO);
end;

procedure Titulos2;

var
i:integer;
tit: array[1..6] of string[50];

begin
  Infoentra;
  textcolor(9);
  for i:= 0 to 8 do
  begin
    gotoxy(22,1+i);
    write(chr(179));
  end;
  textcolor(7);
  gotoxy(4,2);tit[1]:= 'Titulo';
  gotoxy(4,3);tit[2]:= 'Eje Vertical';
  gotoxy(4,4);tit[3]:= 'Eje Horizontal';
  gotoxy(4,5);tit[4]:= 'Serie 1';
  gotoxy(4,6);tit[5]:= 'Serie 2';
  gotoxy(4,7);tit[6]:= 'Serie 3';
  case GRUPO of
    1:begin
      for i:= 1 to 4 do
      begin
        gotoxy(4,1+i);write(tit[i]);
      end;
      for i:= 1 to 4 do
      begin
        gotoxy(24,1+i);readln(cadena[i]);
      end;
    end;
    2:begin
      for i:= 1 to 5 do
      begin
        gotoxy(4,1+i);write(tit[i]);
      end;
      for i:= 1 to 5 do
      begin
        gotoxy(24,1+i);readln(cadena[i]);
      end;
    end;
    3:begin
      for i:= 1 to 6 do
      begin
        gotoxy(4,1+i);write(tit[i]);
      end;
      for i:= 1 to 6 do
      begin
        gotoxy(24,1+i);readln(cadena[i]);
      end;
    end;
  end;
end;

```

```

    end;
    end;
end;
readln;
end;

Procedure Deformada;

type
    Desplaza = array [1..500,1..6] of real;
var
    CDespla,CDespla1:Desplaza;
    J:integer;
    lectura:text;

procedure LecturaDesplazos(var
CDespla: Desplaza);

var
    NombreDes:string;
    Error,I,K,L,Aux:integer;

begin
    repeat
        Infoentra;
        textcolor(7);
        gotoxy(4,2);write('Archivo de Desplazamientos ');
        {$I-}
        gotoxy(34,2);readln(nombreDes);
        assign(lectura, NombreDes);
        reset(lectura);
        {$I+}
        Error:=IoResult;
        if Error < 0 then
            begin
                textcolor(5);
                gotoxy(4,4);write(NombreDes, 'no se puede
                encontrar');
                gotoxy(4,6);writeln('pruebe de nuevo');
                readln;
            end;
        until Error = 0;
        J:=0;
        while not eof(lectura) do
            begin
                J:=J+1;
                for I:= 1 to 6 do
                    begin
                        read(lectura,CDespla[J,I]);
                    end;
                end;
                write(chr(7));
                readln;
                close(lectura);
            end;

procedure CalculoCoorde;

var
    CXD,CYD: array [1..500] of real;
    A,L1,L,K,factor:integer;
    nombre:string;

begin
    Infoentra;
    textcolor(7);
    gotoxy(4,3);write('Archivo a generar ');
    gotoxy(4,5);write('factor de ampliación ');
    gotoxy(34,3);readln(nombre);
    gotoxy(34,5);readln(factor);
    assign(lectura,nombre);
    rewrite(lectura);
    A:=0;
    L1:=round(CDespla[J,1]);
    for L:= 1 to J do
        begin
            repeat
                A:=A+1;
                if A < CDespla[l,1] then
                    begin
                        for k:= 2 to 6 do
                            CDespla1[A,k]:=0;
                        end;
                        until A = CDespla[L,1];
                        for K:= 2 to 6 do
                            CDespla1[A,K]:= CDespla[L,K];
                        end;
                    for L:= 1 to a do
                        begin
                            CXD[L]:= CDespla1[L,2] - (CDespla1[L,4]*factor);
                            CYD[L]:= CDespla1[L,3] - (CDespla1[L,5]*factor);
                            writeln(lectura,CXD[L]:5:6,CYD[L]:10:6);
                        end;
                    close(lectura);
                    readln;
                end;

begin
    LecturaDesplazos(CDespla);
    Calculocoorde;
    Exec('a:generado.exe', *);
end;

procedure Graficar;

type
    item = real;
    lista= array[1..100] of real;

var
    grimage:array[1..4] of pointer;
    lec:text;
    dis,des1,des2,des3,valor,s,vec0,vec1,vec2,vec3,ng:lista;
    nhme,nhuma,nvme,nvma:real;
    j,nd,fronterah,frontera v:integer;

```

```
Procedure Leer_Datos(var J:integer;var
dis,des1,des2,des3:llista);
```

```
var
vector:array[1..150,1..6] of real;
Error,numero,m,n,ancho,c:integer;
cxmax,cymax:real;
NombreArch:string[25];
respuesta:char;

begin
  Infoentra;
  textcolor(7);
  gotoxy(6,2);write('Seleccionar datos para graficar
(S/N):');
  gotoxy(45,2);readln(Respuesta);
  Respuesta:=Ucase(Respuesta);
  repeat
  Infoentra;
  textcolor(7);
  gotoxy(6,4);write('Nombre de Archivo');
  {$I-}
  gotoxy(28,4);readln(nombreach);
  assign(lec, NombreArch);
  reset(lec);
  {$I+}
  Error:= IOResult;
  if Error < 0 then
  begin
    textcolor(5);
    gotoxy(6,6);write(NombreArch, ' no se puede
encontrar');
    textcolor(3);
    gotoxy(6,8);writeln('pruebe de nuevo');
    readln;
  end;
  until Error = 0;
  gotoxy(46,8);write('lectura');
  j:=0;
  if Respuesta= 'N' then
  begin
    while not eof(lec) do
    begin
      j:=j+1;
      read(lec, dis[j]);
      if grupo=1 then
      begin
        readln(lec,des1[j]);
      end;
      if grupo=2 then
      begin
        read(lec,des1[j]);
        readln(lec,des2[j]);
      end;
      if grupo=3 then
      begin
        read(lec,des1[j]);
        read(lec,des2[j]);
        readln(lec,des3[j]);
```

```

      end;
    end;
    write(chr(7));
    readln;
    close(lec);
    Titulos2;
  end;
  if Respuesta = 'S' then
  begin
    gotoxy(6,6);write('No de puntos nodales analizar: ');
    readln(numero);
    j:=numero;
    m:=0;
    while not eof(lec) do
    begin
      m:=m+1;
      for n:= 1 to 6 do
      begin
        read(lec,vector[m,n]);
      end;
    end;
    write(chr(7));
    readln;
    close(lec);
    if EjePrin = 'A' then
    begin
      gotoxy(6,8);write('Ancho de la excavación ');
      readln(Ancho);
      if vector[1,2] >= vector[numero,2] then
        cxmax:= vector[1,2]
      else
        cxmax:= vector[numero,2];
      for c:= 1 to numero do
      begin
        dis[c]:= 80-Ancho-vector[c,2]; {VECTO{1,2} Para
q' ya n}
        des1[c]:= vector[c,5];
        des2[c]:= vector[(numero)+c,5];
        des3[c]:= vector[(numero*2)+c,5];
      end;
      Titulos2;
    end;
    if EjePrin = 'M' then
    begin
      for c:= 1 to numero do
      begin
        dis[c]:=vector[c,2] - vector[1,2];
        des1[c]:=vector[c,5];
        des2[c]:=vector[numero+c,5];
        des3[c]:=vector[(numero*2)+c,5];
      end;
      Titulos2;
    end;
    if EjePrin = 'D' then
    begin
      if vector[1,3] >= vector[numero,3] then
        cymax:=vector[1,3]
      else
        cymax:=vector[numero,3];
      for c:=1 to numero do
```

```

begin
  dis[c]:= vector[c,3] - cymax;
  des1[c]:= vector[c,4];
  des2[c]:= vector[(numero)+c,4];
  des3[c]:= vector[(numero*2)+c,4];
end;
Titulos2;
end;
end;
end;

```

```

procedure
Intercambio(j:integer;dis,des1,des2,des3:lista; var
vec0,vec1,vec2,vec3:lista);

```

```

var
k:integer;

```

```

begin
  for k:= 1 to j do
    begin
      vec0[k]:=dis[k];
      vec1[k]:=des1[k];
      vec2[k]:=des2[k];
      vec3[k]:=des3[k];
    end;
  end;
end;

```

```

procedure Ordenar_Valor(var valor:lista; j:integer;
var nme,nma:real);

```

```

var
i,n:integer;

```

```

procedure Intercambio(var valor,d:item);

```

```

var
aux:item;

```

```

begin
  aux:=valor;
  valor:=d;
  d:=aux;
end;

```

```

begin
  for i:= 1 to j-1 do
    for n:= i+1 to j do
      if valor[i] > valor[n] then
        Intercambio(valor[i], valor[n]);
        nme:=valor[1];
        nma:=valor[j];
    end;
  end;

```

```

procedure Ordenar_Grupo(var s:lista; var
nme,nma:real;j:integer; des1,des2,des3:lista);

```

```

var
i:integer;

```

```

begin
  i:=0;
  case grupo of
    1: begin
      Ordenar_Valor(des1,j,nme,nma);
      s[1]:=nme;
      s[2]:=nma;
      i:=2+i;
    end;
    2: begin
      Ordenar_Valor(des1,j,nme,nma);
      s[1]:=nme;
      s[2]:=nma;
      i:=2+i;
      Ordenar_Valor(des2,j,nme,nma);
      s[3]:=nme;
      s[4]:=nma;
      i:=2+i;
    end;
    3: begin
      Ordenar_Valor(des1,j,nme,nma);
      s[1]:=nme;
      s[2]:=nma;
      i:=2+i;
      Ordenar_Valor(des2,j,nme,nma);
      s[3]:=nme;
      s[4]:=nma;
      i:=2+i;
      Ordenar_Valor(des3,j,nme,nma);
      s[5]:=nme;
      s[6]:=nma;
      i:=2+i;
    end;
  end;
  Ordenar_Valor(s,i,nme,nma);
end;

```

```

procedure Ventana;

```

```

begin
  setcolor(7);
  rectangle(4,26,628,470);
  rectangle(0,22,632,474);
  line(4,450,628,450);
end;

```

```

procedure Ejes(j:integer; var
nhme,nhma,nvme,nvma:real;
dis,des1,des2,des3:lista;
var fronterah,fronterav:integer);

```

```

var
H,R,L,I,C,par,parte:integer;
decremneto,inc2,anme,anma,partida:real;
buffer:char;

```

```

procedure Ajuste(var inc2:real;decre:real; var
long:Integer);

var
incremento:real;
inc,h:integer;

begin
  if (0.1 <= abs(decre)) and (abs(decre) <= 0.9) then
  begin
    incremento:=decre*10;
    inc:= trunc(incremento);
    inc2:= inc/10;
    long:=1;
  end;
  if (0.01 <= abs(decre)) and (abs(decre) <= 0.09) then
  begin
    incremento:=decre*100;
    inc:=trunc(incremento);
    inc2:= inc/100;
    long:=2;
  end;
  if (0.001 <= abs(decre)) and (abs(decre) <= 0.009)
then
  begin
    incremento:=decre*1000;
    inc:=trunc(incremento);
    inc2:=inc/1000;
    long:=3;
  end;
  if (1 <= (abs(decre))) or (decre = 0) then
  begin
    inc:=trunc(decre);
    inc2:=inc;
    long:=0;
  end;
end;

```

```

procedure Adelantar(var final:real; numero:real;
g:Integer);

```

```

var
cambio,numerofi:integer;
inter:real;

begin
  if (0.1 <= abs(numero)) and (abs(numero) <= 0.9)
then
  begin
    inter:=numero*10;
    cambio:= trunc(inter);
    if g=1 then
      numerofi:=succ(cambio);
    if g=0 then
      numerofi:=pred(cambio);
    final:=numerofi / 10;
  end;

```

```

  if (0.01 <= abs(numero)) and (abs(numero) <= 0.09)
then
  begin
    inter:=numero*100;
    cambio:= trunc(inter);
    if g=1 then
      numerofi:=succ(cambio);
    if g=0 then
      numerofi:=pred(cambio);
    final:= numerofi/100;
  end;
  if (0.001 <= abs(numero)) and (abs(numero) <=
0.009) then
  begin
    inter:=numero*1000;
    cambio:= trunc(inter);
    if g=1 then
      numerofi:=succ(cambio);
    if g=0 then
      numerofi:=pred(cambio);
    final:=numerofi/1000;
  end;
  if (1 <= (abs(numero))) then
  begin
    cambio:=trunc(numero);
    if g=1 then
      numerofi:=succ(cambio);
    if g=0 then
      numerofi:=pred(cambio);
    final:=numerofi;
  end;
end;

```

```

procedure Seleccion(var
parte,long,long1:integer;var
decremento,anme,anma,partida:real;
nme,nma:real);

```

```

var
paso,salida,g:integer;
decre,varia,valorf,final:real;
caracter:string;

begin
  if j <= 10 then
  begin
    parte:=j;
    decre:=((nma-nme) / (j-1));
    ajuste(inc2,decre,long);
    decremento:=abs(inc2);
    long1:=long;
  end;
  if (j>10) and (j<=30) then
  begin
    parte:=11;
    decre:=((nma-nme) / parte);
    ajuste(inc2,decre,long);
    decremento:=abs(inc2);
    long1:=long;
  end;

```

```

end;
if 30<j then
begin
  parte:=14;
  decre:=((nma-nme)/parte);
  ajuste(inc2,decre,long);
  decremento:=abs(inc2);
  longl:=long;
end;
if (nme < 0) and (nma > 0) then
begin
  ajuste(inc2,nma,long);
  anma:=inc2;
  if anma < nma then
  begin
    g:=1;
    adelantar(final,anma,g);
    anma:=final;
    valorf:= anma - (decremento*parte);
  end;
  if anma=nma then
  valorf:= anma - (decremento*parte);
  if valorf > nme then
  begin
    decremento:=decremento*2;
    parte:=(parte div 2) + 1;
    parte:=trunc(parte);
    repeat
      parte:=parte+1;
      valorf:=anma - (decremento*parte);
    until valorf < nme
  end;
  if valorf < nme then
  begin
    decremento:=decremento;
    parte:=parte;
    partida:=valorf;
    anma:=anma;
    anme:=valorf;
  end;
  end;
  if (nme < 0) and (nma <= 0) then
  begin
    if (decremento<1) and (abs(nme) > 1) then (*q lo
haga solo para x*)
    anme:=nme
    else
    begin (*para el 10%*)
      ajuste(inc2,nme,long);
      anme:=inc2;
    end;
    if anme > nme then
    begin
      g:=0;
      adelantar(final,anme,g);
      anme:=final;
      valorf:=anme + (decremento*parte);
    end;
    if anme=nme then
    valorf:=anme+(decremento*parte);
  
```

```

if valorf < nma then
begin
  decremento:=decremento*2;
  parte:=(parte div 2)+1;
  parte:=trunc(parte);
  repeat
    parte:=parte+1;
    valorf:= anme+ (decremento*parte);
  until valorf >= nma
end;
if nma <= valorf then
begin
  decremento:=decremento;
  parte:=parte;
  partida:=anme;
  anme:=anme;
  anma:=valorf;
end;
end;
if (nme >= 0) and (nma > 0) then
begin
  if (decremento < 1 ) and (abs(nme) > 1) then
  anme:=nme
  else
  begin
    ajuste(inc2,nme,long);
    anme:=inc2;
    valorf:= anme + (decremento * parte);
  end;
  if valorf < nma then
  begin
    decremento:=decremento*2;
    parte:=(parte div 2) + 1;
    parte:=trunc(parte);
    repeat
      parte:=parte+1;
      valorf:= anme + (decremento*parte);
    until nma < valorf
  end;
  if valorf >= nma then
  begin
    decremento:=decremento;
    parte:=parte;
    partida:=anme;
    anme:=anme;
    anma:=valorf;
  end;
end;
end;
end;

```

**procedure Horizontal(var nhme,nhma:real;var
fronterah:integer);**

```

var
absi,parte,salida,paso,long,longl:integer;
decremento,varia:real;
caracter:string;

```

```

begin

```

```

if (EjePrin = 'A') or (EjePrin = 'M') then
begin
  Ordenar_Valor(dis,j,nme,nma);
end;
if EjePrin = 'D' then
begin
  ordenar_Grupo(s,nme,nma,j,des1,des2,des3);
end;

seleccion(parte,long,long1,decremento,anme,anma,partida,nme,nma);
nhme:=anme;
nhma:=anma;
variav:= 490 / parte;
salidav:=trunc(variav);
fronterah:=salidav*parte;
line(70,420,fronterah+70,420);
for paso:= 1 to parte do
begin
  absi:=(70+salidav*paso);
  line(absi,415,absi,420);
  str((partida+(decremento*paso)):4:long1,caracterv);
  (**verificar valor[nd]**)
  outtextxy((50+(salidav*paso)),422,caracterv);
end;
delay(1000);
end;

```

```

procedure Vertical(var nvme,nvma:real; var
fronterav:integer);

```

```

var
pasov,salidav,partev,orde,i,long,long1:integer;
decrev,decrementov,variav,frontera:real;
caracterv:string;
s:lista;

```

```

begin
if (EjePrin = 'A') or (EjePrin = 'M') then
begin
  ordenar_Grupo(s,nme,nma,j,des1,des2,des3);
end;
if EjePrin = 'D' then
begin
  Ordenar_Valor(dis,j,nme,nma);
end;

```

```

seleccion(partev,long,long1,decrementov,anme,anma,partida,nme,nma);
nvme:=anme;
nvma:=anma;
variav:= 350/partev;
salidav:=trunc(variav);
fronterav:=salidav*partev;
line(70,420,70,420-fronterav);
line(fronterah+70,420-fronterav,fronterah+70,420);
line(70,420-fronterav,fronterah+70,420-fronterav);
i:=0;
repeat

```

```

frontera:=partida+(decrementov*i);
i:=i+1;
if i>j then
begin
  frontera:=0;
  i:=1;
end;
until frontera = 0;
orde:=(420-salidav*(i-1));
line(70,orde,fronterah+70,orde);
for pasov:=0 to partev do
begin
  orde:= (420 - salidav*pasov);
  line(75,orde,70,orde);
  str((partida+decrementov*pasov):4:long1,caracterv);
  outtextxy(26-long1,420-salidav*pasov,caracterv);
end;
end;

```

```

begin
Setbkcolor(0);
Setfillstyle(0,1);
Floodfill(3,1,7);
Ventana;
Setcolor(7);
Horizontal(nhme,nhma,fronterah);
Vertical(nvme,nvma,fronterav);
end;

```

```

procedure
Colocacion(J:integer;DIS,DES1,DES2,DES3,VALOR:LISTA);

```

```

type
vector = array[1..50] of integer;
vecto = array[1..100] of real;
var
colocarx,colocary:vector;

```

```

procedure Crear_Imagenes;

```

```

const
  figura:array[1..16] of integer =
(0,5,15,5,12,10,17,10,10,
18,12,13,8,13,10,5);
  vec: array[1..4] of integer = (20,40,60,80);

```

```

var
lj:integer;

```

```

begin
setfillstyle(1,0);
if maxcolors > 4 then setcolor(10);
for i:=0 to 5 do
pieslice(5,5,i*60,(i*60)+60,5);
getmem(grimage[1],imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[1]^);
putimage(0,0,grimage[1]^,xorput);

```

```

if maxcolors > 4 then setcolor(11);
rectangle(0,0,5,5);
getmem(grimage[2], imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[2]^);
putimage(0,0,grimage[2]^,xorput);
if maxcolors > 4 then setcolor(12);
arc(10,10,105,360,8);
ellipse(0,50,300,90,8,3);
ellipse(3,10,0,90,16,8);
setfillstyle(solidfill,getcolor);
floodfill(10,10,getcolor);
getmem(Grimage[3], imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[3]^);
putimage(0,0,grimage[3]^, xorput);
if maxcolors > 4 then setcolor(13);
bar3d(0,5,5,8,3,true);
getmem(grimage[4], imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[4]^);
putimage(0,0,grimage[4]^, xorput);
end;

```

procedure Titulos;

```

begin
  Crear_Imagenes;
  setcolor(7);
  settxtstyle(1,horizdir,1);
  settxtjustify(lefttext,topleft);
  outtextxy(180,40, cadena[1]);
  settxtstyle(4,vertdir,1);
  settxtjustify(centertext,centertext);
  outtextxy(595,240, cadena[2]);
  settxtstyle(4,horizdir,1);
  settxtjustify(centertext,centertext);
  outtextxy(300,444, cadena[3]);
  settxtjustify(lefttext,centertext);
  settxtstyle(2,horizdir,4);
  case GRUPO of
    1:begin
      setcolor(13);
      outtextxy(36,460, cadena[4]);
      putimage(20,460,grimage[4]^,xorput);
      end;
    2:begin
      setcolor(13);
      outtextxy(36,460, cadena[4]);
      putimage(20,460,grimage[4]^,xorput);
      setcolor(11);
      outtextxy(136,460, cadena[5]);
      putimage(120,460,grimage[2]^,xorput);
      end;
    3: begin
      setcolor(13);
      outtextxy(36,460, cadena[4]);
      putimage(20,460,grimage[4]^,xorput);
      setcolor(11);
      outtextxy(136,460, cadena[5]);
      putimage(120,460,grimage[2]^,xorput);
      setcolor(10);

```

```

      outtextxy(236,460, cadena[6]);
      putimage(220,460,grimage[1]^,xorput);
      end;
    end;
  end;
end;

```

procedure Acomodo(var dis,des1,des2,des3,vec0:lista);

```

var
  i,k:integer;
begin
  Intercambio(j,dis,des1,des2,des3,vec0,vec1,vec2,vec3);
  Ordenar_valor(dis,j,nme,nma);
  for i:= 1 to j do
    begin
      for k:= 1 to j do
        begin
          if vec0[k]=dis[i] then
            begin
              des1[i]:=vec1[k];
              des2[i]:=vec2[k];
              des3[i]:=vec3[k];
            end;
          end;
        end;
      end;
    end;
end;

```

Procedure PosicionX(ng:llista;var colocax:vector);

```

var
  arriba,bajo,origene,escp,auxx,auxy:real;
  l:integer;
begin
  for l:= 1 to j do
    begin
      arriba:=ng[l]-nhune;
      bajo:=nhma-nhune;
      auxx:=((fronterah*arriba)/bajo)+70;
      colocax[l]:=round(auxx);
    end;
  end;
end;

```

procedure posicionY(ng:llista;var colocary:vector);

```

var
  auxy,arri,bajo,origene,escp:real;
  l:integer;
begin
  for l:= 1 to j do
    begin
      arri:=ng[l] - nvme;
      bajo:=nvma-nvme;

```

```

    auxy:=420-((fronterav*arri)/bajo);
    colocary[1]:=round(auxy);
end;
end;

procedure Puntos;

const
    opcion:set of char = [#13];

type
    vecto = array [1..50] of real;

var
    k:integer;
    i:integer;
    ngx,ngy:lista;
    tecla:char;
    adelante:boolean;

begin
    Detectar;

    Ejes(j,nhune,nhuma,nvme,nvma,dis,des1,des2,des3,frontera,
    rah,fronterav);
    Titulos;
    Acomodo(dis,des1,des2,des3,vecto);
    Crear_Imagenes;
    for k:= 1 to grupo do
    begin
        case k of
            1: begin
                for i:= 1 to j do
                begin
                    ngx[i]:=dis[i];
                    ngy[i]:=des1[i];
                end;
                IF (EjePrin = 'A') or (EjePrin = 'M') then
                begin
                    PosicionX(ngx,colocarx);
                    PosicionY(ngy,colocary);
                end;
                If EjePrin = 'D' then
                begin
                    PosicionY(ngx,colocary);
                    PosicionX(ngy,colocarx);
                end;
                moveto(colocarx[1],colocary[1]);
                for i:= 1 to j-1 do
                begin
                    PutImage(getx-3, gety-3 , grimage[4]^,
xorput);
                    lineto(colocarx[i+1],colocary[i+1]);
                    end;
                    PutImage(getx-3, gety-3 , grimage[4]^ , xorput);
                end;
                2: begin
                    if (EjePrin = 'A') or (EjePrin = 'M') then
                    begin
                        PosicionX(dis,colocarx);
                        PosicionY(des2,colocary);
                    end;
                    if EjePrin = 'D' then
                    begin
                        PosicionY(dis,colocary);
                        PosicionX(des2,colocarx);
                    end;
                    moveto(colocarx[1], colocary[1]);
                    for i:= 1 to j-1 do
                    begin
                        setcolor(11);
                        PutImage(getx-2, gety-2, grimage[2]^, xorput);
                        lineto(colocarx[i+1], colocary[i+1]);
                    end;
                        putImage(getx-2, gety-2, GrImage[2]^ , xorput);
                        outtextxy(colocarx[j],colocary[j],'+');
                        outtextxy(colocarx[j],colocary[j], '/');
                    end;
                    3: begin
                        if (EjePrin = 'A') or (EjePrin = 'M') then
                        begin
                            PosicionX(dis,colocarx);
                            PosicionY(des3,colocary);
                        end;
                        If EjePrin = 'D' then
                        begin
                            PosicionY(dis,colocary);
                            PosicionX(des3,colocarx);
                        end;
                        moveto(colocarx[1], colocary[1]);
                        for i:= 1 to j-1 do
                        begin
                            setcolor(10);
                            putImage(colocarx[i]-2,colocary[i]-2,
grimage[1]^, xorput);
                            lineto(colocarx[i+1], colocary[i+1]);
                        end;
                            PutImage(getx-2, gety-2, GrImage[1]^, xorput);
                        end;
                        end;
                        end;
                        repeat
                        tecla:=readkey;
                        if not(tecla in opcion) then
                        write(chr(7));
                        until (tecla in opcion);
                        if tecla= #13 then
                        adelante:=true;
                        until(adelante);
                    end;
                    end;
                    textnode(10);
                    Titulos1;
                    Leer_Datos(j,dis,des1,des2,des3);
                    Puntos;
                end;
            end;
        end
    end

```

```
{** Programa Principal (Graficar) **}
```

```
begin  
  Detectar;  
  Colocacion(j,dis,des1,des2,des3,valor);  
  Cleardevice;  
end;
```

```
procedure Limpiar;
```

```
begin  
  window(1,1,80,25);  
  Textbackground(7);  
  Textcolor(0);  
end;
```

```
procedure Opcion;
```

```
var  
  resul:char;  
  
begin  
  Detectar;  
  Entragraf;  
  repeat  
    textmode(10);  
    Infoentra;  
    textcolor(7);  
    gotoxy(15,3);write('1) Desplazamientos');  
    gotoxy(15,5);write('2) Malla Deformada');  
    gotoxy(15,7);write('3) Salir');  
    textcolor(3);  
    gotoxy(44,9);write('« «');  
    gotoxy(46,9);write('');  
    repeat  
      resul:=readkey  
    until resul in ['1'..'3'];  
    case resul of  
      '1': Graficar;  
      '2': Deformada;  
    end;  
  until resul = '3'  
end;
```

```
{*****PROGRAMA PRINCIPAL*****}
```

```
begin  
  Opcion;  
  Closegraph;  
end.
```

7

ANEXO IV

MÓDULO EXPLICATIVO

Lista de procedimientos utilizados en este módulo por orden alfabético

- | | |
|--------------------|---|
| Abrir: | Despliega la información contenida en archivo de texto. |
| Antes: | Permite desplegar la página anterior |
| Despues: | Permite desplegar la página siguiente. |
| Detectar: | Inicializa los gráficos y determina el tipo de monitor a utilizar. |
| Texto: | Relaciona la tecla accionada con el procedimiento a ejecutar. |
| Visualizar: | Contiene los procedimientos para visualizar la información. |

```

Program Explicativo;

Uses
  Crt,Dos,Graph;

procedure Detectar;

var
  D,M,ErrorCode:integer;

begin
  D:=DETECT;
  InitGraph(D,M,\BP^BGT);
  ErrorCode:=GraphResult;
  if ErrorCode <> Grok then
  begin
    writeln('Error en el sistema de
    gráficos:',GraphErrorMsg(ErrorCode));
    readln;
    halt(1);
  end;
end;

procedure Presentacion;

begin
  Detectar;
  setbkcolor(0);
  setfillstyle(11,1);
  floodfill(1,1,1);
  setcolor(11);
  settextstyle(1,0,0);
  outtextxy(150,230,'MODULO EXPLICATIVO');
  readln;
end;

procedure Visualizar;

var
  limitea,limiteb:integer;
  adelante:boolean;

procedure Abrir;

var
  l,m,b:integer;
  caracter:string;
  fichero:text;

begin
  cleardevice;
  setbkcolor(0);
  setfillstyle(9,1);
  floodfill(1,1,1);
  setcolor(15);
  rectangle(1,1,638,478);
  rectangle(27,17,603,463);
  rectangle(30,20,600,460);
  setfillstyle(11,1);
  floodfill(275,285,15);
  settextstyle(0,0,0);
  setcolor(11);
  outtextxy(360,35,' Desplegar: Re Pág Av Pág <');
  outtextxy(40,35,' Salir:Eac <');
  setcolor(15);
  assign(fichero,'explica.tex');
  reset(fichero);
  m:=0;
  b:=0;
  for l:= 1 to 1000 do
  begin
    readln(fichero,caracter);
    if l >= limitea then
    if l <= limiteb then
    begin
      m:=m+1;
      b:=b+12;
      if m=3 then
      m:=1;
      outtextxy(30+m,50+b,caracter);
    end
    else
    begin
      close(fichero);
      exit;
    end;
  end;
  if eof(fichero) then
  begin
    close(fichero);
    exit;
  end;
end;

procedure Antes;

begin
  if limitea=1 then
  begin
    limitea:=limitea;
    limiteb:=limiteb;
  end
  else
  begin
    limitea:=limitea-27;
    limiteb:=limiteb-27;
  end;
end;

```

```
procedure Despues;

begin
  limitea:=limitea+27;
  limiteb:=limiteb+27;
end;

procedure Texto(var adelante:boolean);

const
  seleccion: set of char = {#0, #27};
  mover : set of char = {#73, #81, #119};
var
  tecla:char;

begin
  limitea:=1;
  limiteb:=27;
  Abrir;
  adelante:=false;
  repeat
    repeat
      tecla:=upcase(readkey);
    until(tecla in seleccion);
    case tecla of
      #0: begin
        tecla:=readkey;
        if tecla in mover then
          case tecla of
            #81: begin
              Despues;
              Abrir;
            end;
            #73: begin
              Antes;
              Abrir;
            end;
          end;
        else
          write(chr(7));
        end;
      #27: adelante:=true;
    end;
  until(adelante);
end;

begin
  Detectar;
  Presentacion;
  Texto(adelante);
end;

(** Programa Principal **)

begin
  Visualizar;
end.
```

ANEXO V
MÓDULO PRINCIPAL

Lista de procedimientos en orden alfabético utilizados en este módulo.

- Activar:** Procedimiento que activa la palabra iluminada.
- Adelante:** Activa la palabra siguiente.
- Atras:** Regresa el cursor a la palabra anterior.
- Cargar:** Declara los letreros utilizados en este módulo.
- Desactivar:** Desactiva la palabra iluminada
- Despedir:** Avisa que el módulo a finalizado.
- Detectar:** Inicializa los gráficos y el tipo de monitor a utilizar.
- Escribir:** Despliega en pantalla los letreros utilizados.
- Información:** Lee y presenta en pantalla información general
- Iniciar:** Procedimiento que permite elegir opción
- Menu:** Procedimiento para relacionar las palabras iluminadas con los módulos con que cuenta el SE.
- Modulos:** Procedimiento que ejecuta el módulo relacionado con la palabra iluminada.
- Ventana:** Crea la ventana de trabajo.

Program Entrada;
(\$M 65520,0,40000)

Uses
Crt,Dos,Graph;

Procedure Detectar;

```

Var
D,M,ErrorCode:Integer;

Begin
D:=Detect;
InitGraph(D,M,^BP^BGF);
ErrorCode:=GraphResult;
if ErrorCode <> Grok then
begin
writeln('Error en el sistema de gráficos',
GraphErrorMsg(ErrorCode));
readln;
halt(1);
end;
end;

```

Procedure Informacion;

```

var
m,b,i:integer;
caracter:string;
Fichero:Text;

begin
Detectar;
setbkcolor(0);
setfillstyle(9,1);
floodfill(1,1,1);
setcolor(15);
rectangle(1,1,638,478);
setcolor(15);
setttextstyle(1,0,0);
rectangle(27,17,603,463);
rectangle(30,20,600,460);
setfillstyle(11,1);
floodfill(275,285,15);
setttextstyle(0,0,0);
assign(fichero, 'a:informa.tex');
reset(fichero);

```

```

m:=0;
b:=0;
for i:= 1 to 800 do
begin
m:=m+1;
b:=b+12;
if m=3 then
m:=1;
readln(fichero,caracter);
outtextxy(30+m,50+b,caracter);
end;
close(fichero);
readln;
end;

```

procedure Ventana;

```

begin
cleardevice;
setfillstyle(11,1);
floodfill(1,1,1);
setcolor(15);
rectangle(118,148,482,352);
rectangle(120,150,480,350);
setfillstyle(9,1);
floodfill(180,100,15);
end;

```

procedure Despedir;

```

begin
Ventana;
setcolor(9);
setttextstyle(1,0,1);
outtextxy(180,250,FIN DEL MODULO);
readln;
setttextstyle(0,0,0); {recupera la letra normal}
end;

```

procedure Iniciar;

```

var
PAL: array [1..4] of string[50];
X,Y: array [1..5] of integer;
i:integer;

```

procedure Cargar;

```

begin
X[1]:=200;Y[1]:=170; PAL[1]:= 'MODULO
EXPLICATIVO';
X[2]:=200;Y[2]:=220; PAL[2]:= 'MODULO DE
ALIMENTACION';
X[3]:=200;Y[3]:=270; PAL[3]:= 'MODULO DE
GRAFICACION';
X[4]:=200;Y[4]:=320; PAL[4]:= 'SALIDA';

```

```

end;

procedure Escribir;

var
I:integer;

begin
Ventana;
setcolor(15);
for I:= 1 to 4 do
begin
if I=1 then
setcolor(3)
else
setcolor(15);
outtextxy(X[I],Y[I],PAL[I]);
end;
end;

procedure Activar(I:Integer);

begin
setcolor(9);
outtextxy(X[I],Y[I],PAL[I]);
end;

procedure Desactivar(var I:Integer);

begin
setcolor(15);
outtextxy(X[I],Y[I],PAL[I]);
end;

procedure Adelante(var I:Integer);

begin
Desactivar(I);
if I=4 then I:=1
else I:=I+1;
Activar(I);
end;

procedure Atras(var I:Integer);

begin
Desactivar(I);
if i=1 then I:=4
else I:=I-1;
Activar(I);
end;

procedure Modulos(var I:Integer);

```

```

begin
case I of
1:begin
Exec('a:explica.exe');
Despedir;
Escribir;
I:=1;
Activar(I);
end;
2:begin
Exec('a:mallad.exe');
Detectar;
Despedir;
Escribir;
I:=1;
Activar(I);
end;
3:begin
Exec('grades.exe');
Detectar;
Despedir;
Escribir;
I:=1;
Activar(I);
end;
end;
end;

procedure Menu;

const
seleccion: set of char = [#72, #80, #13];

var
tecla:char;
salir:boolean;

begin
salir:=false;
repeat
repeat
tecla:=readkey;
until(tecla in seleccion);
case tecla of
#80: Adelante(I);
#72: Atras(I);
#13: if I=4 then salir:=true
else Modulos(I);
end;
until(salir);
end;

begin
Ventana;
Cargar;
Escribir;
i:=1;
Activar(I);
Menu;

```

```
Ventana;  
settextstyle(7,0,1);  
outtextxy(170,240,'TERMINACION DEL PROCESO');  
readln;  
end;
```

```
{** Programa Principal **}
```

```
begin  
Informacion;  
Iniciar;  
Closegraph;  
end.
```