



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**DIVISION DE INGENIERIA CIVIL, TOPOGRAFICA
Y GEODESICA**

**SISTEMA EXPERTO PARA EL MANEJO DE
PROGRAMAS DE ANALISIS DINAMICO DE
PRESAS**

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO CIVIL

P R E S E N T A I

JESUS RAMIRO CRUZ SANCHEZ



México, D.F.

1993

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

CAPITULO I

	INTRODUCCION	1
1.1	GENERALIDADES	1
	1.1.1 Sistemas Expertos (SE)	1
	1.1.2 Análisis Dinámico en Presas de Tierra	3
1.2	OBJETIVOS	4
1.3	ALCANCES	5
1.4	CONSTRUCCION DEL SE	5
1.5	RESULTADOS OBTENIDOS	6

CAPITULO II

	NOCIONES SOBRE SISTEMAS EXPERTOS	7
2.1	ANTECEDENTES	7
	2.1.1 La Inteligencia Artificial y sus Aplicaciones	7
	2.1.2 Los Sistemas Expertos	8
	2.1.3 Propósitos de Construcción de los SE	8
2.2	CARACTERISTICAS GENERALES DE LOS SE	10
	2.2.1 Adquisición del Conocimiento	10
	2.2.2 Grado de Fiabilidad	10
	2.2.3 Area de Dominio del Conocimiento	10
	2.2.4 Procedimiento de Solución de Problemas	10
2.3	ESTRUCTURA BASICA DE LOS SISTEMAS EXPERTOS	11
	2.3.1 Motor de Inferencia	11
	2.3.2 Base del Conocimiento	11
	2.3.3 Base de Hechos	12
	2.3.4 Los Módulos de Comunicación	12
2.4	FUNCIONAMIENTO DEL MOTOR DE INFERENCIA	13
	2.4.1 Mecanismos de Búsqueda	13
	2.4.2 Elección del Conocimiento	14
	2.4.3 Metaconocimiento	15
	2.4.4 Lógica	15
	2.4.5 Evaluación del conocimiento	17
2.5	TECNICAS DE REPRESENTACION DEL CONOCIMIENTO	18
	2.5.1 Reglas de Producción	18
	2.5.2 Redes Semánticas	19
	2.5.3 Objetos Estructurados:(marcos, objetos y guiones)	20

2.6	METODOLOGIA Y CONSTRUCCION DE SE	22
2.6.1	Estudio de la Demanda	23
2.6.2	Análisis del Problema	23
2.6.3	Elección de la Fuente de Conocimiento	24
2.6.4	Preselección del Soporte	24
2.6.5	Obtención del Conocimiento	25
2.6.6	Selección del Soporte	25
2.6.7	Construcción del Prototipo	25
2.6.8	Validación del Prototipo	26
2.6.9	Construcción del SE	26
2.7	Lenguaje y Herramientas para SE	26
2.7.1	Lenguajes: Imperativo, Funcional, Orientados al objeto y Declarativos	27
2.7.2	Herramientas: Entornos desarrollados y Sistemas vacíos	31

CAPITULO III

PROCEDIMIENTO DE ANALISIS 32

3.1	ANTECEDENTES	32
3.1.1	Métodos Deterministas	33
3.1.2	Métodos Aleatorios	38
3.2	METODO DE ANALISIS	39
3.2.1	Selección de la Excitación	39
3.2.2	Respuesta Dinámica	40
3.2.3	Estabilidad del Talud	44
3.2.4	Pérdida de Bordo Libre	49
3.3	METODOLOGIA DE ANALISIS	52
3.3.1	Análisis de Respuesta Dinámica	52
3.3.2	Análisis de Estabilidad	53
3.3.3	Estimación de la Pérdida de Bordo Libre	59
3.3.4	Graficación de los Resultados	59

CAPITULO IV

SISTEMA EXPERTO DESARROLLADO 61

4.1	ESTUDIO DE LA DEMANDA	61
4.2	ANALISIS DEL PROBLEMA	62
4.2.1	Formulación general del problema	62
4.2.2	Objetivos del sistema	62

4.2.3	Alcances de desarrollo	63
4.2.4	Validez de la aplicación	63
4.2.5	Formulación particular de la solución	64
4.3	ELECCION DE LA FUENTE DE CONOCIMIENTO	65
4.4	PRESELECCION DEL SOPORTE	65
4.4.1	Soportes considerados	65
4.4.2	Evaluación de los soportes	67
4.4.3	Elección inicial del soporte	69
4.5	OBTENCION DEL CONOCIMIENTO	70
4.6	SELECCION DEL SOPORTE	72
4.7	CONSTRUCCION DEL PROTOTIPO	73
4.7.1	Módulo administrativo	74
4.7.2	Módulo de interfase	77
4.7.3	Módulo explicativo	78
4.7.3	Módulo maestro	79
4.8	EJEMPLO	80

CAPITULO V

	CONCLUSIONES	88
5.1	CONCLUSIONES	86
5.2	RECOMENDACIONES	87
	REFERENCIAS	89
	ANEXO I MANUAL DEL USUARIO DEL PROGRAMA PTLUSH	91
	ANEXO II MANUAL DEL USUARIO DEL PROGRAMA ANSTA	96
	ANEXO III MANUAL DE PROGRAMACION EN TURBO PASCAL	100
	ANEXO IV MODULO ADMINISTRATIVO	125
	ANEXO V MODULO DE INTERFASE	136
	ANEXO VI MODULO EXPLICATIVO	168
	ANEXO VII MODULO PRINCIPAL	184
	ANEXO VII GLOSARIO	190

CAPITULO I

INTRODUCCION

1.1 GENERALIDADES

La Inteligencia Artificial a contribuido para el logro grandes progresos a partir de la década de los cincuentas, es desde partir de este periodo cuando pedagogos y psicólogos se interesaron en encontrar métodos generales de resolución que junto con los avances en el área de la computación, permitieron desarrollar procesos casi automáticos para la resolución de determinados problemas.

El desarrollo de los Sistemas Expertos (SE) ha tenido una gran aceptación en diferentes áreas del conocimiento, por ejemplo en el área de la Ingeniería, la construcción de éstos es muy variada generalmente las aplicaciones pueden ser sistemas que permitan diagnosticar, reparar, controlar, simular o educar entre otros objetivos.

El SE es un conjunto de circuitos y programas capaces de simular parte de los procesos realizados por un Experto Humano* (EH) en el momento mismo de solucionar un problema. En realidad el SE sólo se comporta como EH de manera aproximada, puesto que a la fecha se desconocen gran parte de los procesos mentales y motrices del cerebro humano.

NOTA: las palabras marcadas con * pueden ser consultadas en el glosario

1.1.1 SISTEMAS EXPERTOS

Características generales del un SE

- Los SE no son capaces de iniciar por si solos el aprendizaje, ya que para solucionar un problema, necesariamente requieren se les introduzca una base conocimientos previos.
- El grado de fiabilidad de un SE esta en función de el grado de explicación de los resultados obtenidos.
- Al igual que un EH los SE sólo son expertos en una área finita de conocimiento. En el caso de los SE el limite es de tipo fisico: memoria disponible* y velocidad de procesado* entre otras.

Estructura básica de un SE

- Motor de inferencia*. Que tiene como funciones principales las de seleccionar, interpretar y explicar la base de conocimientos.
- Base de conocimientos*. Es la parte del SE donde se almacena el conocimiento.
- Base de hechos*. Es el conjunto de información que sirve como datos de entrada o condición para solucionar un determinado problema.
- Módulo de comunicación. Se divide en dos partes: a) módulo experto, que permite interactuar al programador con el sistema para fines de mantenimiento, depuración y reconfiguración*, b) módulo del usuario, que permite interactuar al usuario con el sistema.

Metodología de construcción de un SE

- Estudio de demanda: En este paso se estudian y clasifican las características del problema a resolver.
- Análisis del problema: En esta fase se determina si es posible resolver el problema por medio de un SE, de ser posible se propone la solución al mismo.

- Elección de la fuente de conocimiento: Es este punto se definen los posibles Expertos Humanos que puedan colaborar en la realización del proyecto.
- Preselección del soporte*: En este paso se elige en primera instancia la herramienta que se va a utilizar para construir el SE.
- Obtención del conocimiento: Esta etapa consume la mayor parte del tiempo programado para la construcción del SE pues es necesario familiarizarse con el marco teórico, que en muchos casos va desde el estudio de los comandos de la herramienta de construcción seleccionada hasta los conocimientos necesarios para entender el problema que se desea resolver por medio del SE.
- Selección del soporte: En este punto se reanaliza el soporte preseleccionado, evaluando las ventajas y desventajas del mismo, al contar con información detallada es posible determinar los problemas que el SE tendrá en el futuro, de resultar aprobatorio el análisis se pasa al siguiente punto de la metodología; de lo contrario es necesario regresar a preseleccionar otro soporte.
- Construcción del prototipo*: La construcción sirve para corroborar la validez del conocimiento, en muchos casos aquí termina el trabajo, pues lo que se busca es demostrar la viabilidad de los SE para determinada rama del conocimiento.
- Posteriormente se analiza la validez del prototipo, esto se logra mediante una etapa de pruebas, para finalmente pasar a la construcción del SE (en muchos casos con el fin de comercializarlo), es importante subrayar que el SE puede responder adecuadamente en el periodo de pruebas pero ello no significa que en el futuro lo hará con todos los problemas que se deseen resolver, de tal modo que es válido afirmar que el SE permanecerá siempre en un periodo de prueba lo que permite realizar modificaciones al mismo y conseguir actualizarlo constantemente.

Herramientas y lenguajes utilizados para construir SE

Es importante señalar que casi desde cualquier lenguaje* es posible construir SE, las ventajas de unos sobre otros radica fundamentalmente, en la complejidad que representa para el programador construir la lógica de funcionamiento del SE, en lo que respecta a la utilización de herramientas (sistemas vacíos o Shell's*) no es necesario realizar este proceso de construcción, justificación y explicación de la lógica utilizada lo cual resulta muy ventajoso para el programador. La desventaja fundamental de las herramientas es su rigidez, por lo que es necesario buscar la herramienta que se amolde adecuadamente para cada uno de los problemas a resolver.

1.1.2 ANALISIS DINAMICO EN PRESAS DE TIERRA

Los conocimientos sobre el comportamiento dinámico de presas de tierra han ido incrementándose con la documentación de varias presas de este tipo que han fallado, como son la presa Sheffield que se colapsó durante el sismo de Santa Bárbara de 1925 y la presa Mochi-Koshi de Japón que falló como consecuencia del sismo Izu-Oshima de 1979.

El incremento en la información permitió desarrollar nuevas metodologías de análisis, por el ejemplo la utilizada por Romo et al (1992) en los análisis de las presas El Infiernillo y La Villita, que al igual que otras permite comprobar que los elementos más importantes en el análisis dinámico son: la geometría de la cortina, la distribución y comportamiento no lineal de los materiales y por último las características con las cuales se considera actúa la excitación* inducida en la base de la cortina.

El método utilizado por Hernández, Magaña y Romo (1992), que da lugar al presente trabajo, considera que la fuerza sísmica que actúa en la base rígida de la cortina tiene características aleatorias.

Modelando la presa mediante una malla de elementos finitos tridimensionales y considerando la excitación como una función aleatoria (ec. 9, capítulo III) es posible determinar la respuesta sísmica de la presa y a partir de ésta, calcular la aceleración máxima esperada en cada uno de los puntos nodales* de la malla de elementos finitos, en que se discretizó la cortina.

Conocida la distribución de aceleraciones máximas en el cuerpo de la presa se procede a realizar un análisis de estabilidad de tipo Bishop modificado, con el cual se determina el factor de seguridad convencional del talud aguas abajo de la presa.

Para cuantificar la pérdida de bordo libre de la presa se utilizan las expresiones 25 y 26, capítulo III. El factor de seguridad real empleado se obtiene de la figura 3.11 a partir del factor de seguridad convencional calculado.

1.2 OBJETIVO

En el presente trabajo se desarrollará un SE para manejar programas relacionados con el análisis dinámico en presas de tierra, el SE debe cumplir con un objetivo básico: debe ser lo más autónomo posible del medio* donde se lleve a cabo su construcción.

Teniendo en cuenta el objetivo fundamental del SE, éste debe cumplir con los siguientes tres elementos:

- 1) Ser parte del proceso de entrenamiento de personal necesario en este tipo de análisis
- 2) Simular y automatizar parte del análisis, para servir de apoyo a los Expertos Humanos y
- 3) Constituir un medio de difusión del conocimiento.

1.3 ALCANCES

Los procesos necesarios para realizar el análisis dinámico en presas de tierra pueden ser divididos en dos grupos: los que se llevan a cabo en una work-station* (estaciones de trabajo) y los que se realizan en PC (computadora personal).

Los alcances de desarrollo del SE cubrirán los procedimientos llevados a cabo en la computadora personal, pero sentando las bases para que se puedan incorporar los demás procesos más adelante.

Para lograr estos alcances el SE debe cumplir además con los siguientes requisitos:

- 1) Que el sistema pueda ser desarrollado en base a módulos.
- 2) Que exista una relativa independencia entre los diferentes módulos
- 3) Que los módulos puedan ser utilizados desde el momento mismo que termine su etapa de construcción
- 4) Que los módulos puedan ser controlados por un módulo maestro o principal que permita ir incorporando nuevos módulos en futuras investigaciones.

1.4 CONSTRUCCION DEL SISTEMA EXPERTO

Para preseleccionar el soporte del SE se analizaron varias opciones, se realizaron pruebas con cada uno de los soportes para evaluar la adaptabilidad de los mismos a las características particulares del SE, considerando que el SE resultante debía ser lo más independiente del medio de construcción, se seleccionó el lenguaje Turbo Pascal versión 5.5, pues permite manejar gráficos, pantallas y crear interfases con otros lenguajes.

El prototipo se construyó en base a módulos independientes los que se ligaron por un módulo maestro o principal, lo que permitirá incorporar nuevos módulos en el futuro.

Finalmente la validación del mismo se llevó a cabo mediante la obtención de pérdidas de bordo libre por este procedimiento y compararlas contra las calculadas en los trabajos de Romo et al (1992).

1.5 RESULTADOS OBTENIDOS

- El prototipo ya construido demuestra que es viable la construcción de un SE utilizando Turbo Pascal como soporte.
- Uno de los objetivos fundamentales de este trabajo era poder lograr un SE lo más autónomo del entorno* de programación , esto se logró en un ciento por ciento.
- El prototipo constituye un medio idóneo para difundir nuevos conocimientos, además de realizar algunas actividades antes hechas por el investigador.
- El prototipo terminado es fácil de copiar a diferentes computadoras aumentando el grado de alcance del mismo.
- El SE logró automatizar casi la totalidad las tareas necesarias para estimar la estabilidad del talud, su pérdida de bordo libre, etc.
- El SE representa un elemento importante para desarrollar otras investigaciones relacionadas con el tema y las que tenían como principal limitante los tiempos que se consumía en la estimación de estabilidad y pérdida de bordo libre.
- Finalmente el SE puede ser transportado o exportado a una estación de trabajo (work-station), a fin de que se le incorporen nuevos módulos relacionados con el análisis de respuesta dinámica, esto es posible gracias a que Turbo Pascal es compatible con este tipo de máquinas.

CAPITULO II

NOCIONES SOBRE SISTEMAS EXPERTOS

2.1 ANTECEDENTES

2.1.1 LA INTELIGENCIA ARTIFICIAL Y SUS APLICACIONES

La Inteligencia Artificial (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.

CAMPOS DE APLICACION DE LA IA:

La Robótica:

Area en la que se desarrollan procesos repetitivos para la manipulación de objetos.

Los Sistemas Expertos:

Area en la que se simulan los procesos 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.

La Visión por Ordenador:

Area en la que se simula la percepción humana para la identificación, localización y verificación de objetos.

El Aprendizaje Automático:

Campo en el que se desarrollan programas ordenadores que puedan aprender automáticamente.

El Tratamiento Inteligente de la Información:

Campo que estudia y desarrolla diferentes mecanismos inteligentes para buscar y procesar información en bases de datos relativamente grandes.

La Programación Automática:

Desarrolla formas automáticas de reprogramación, resultado de solucionar un problema específico y necesaria para la resolución de nuevos problemas, en un determinado campo del conocimiento.

Los Juegos:

Desarrolla y resuelve problemas cuya solución no necesariamente debe tener una explicación lógica si no sólo basta con que se cumplan una serie de leyes o reglas de juego.

Difícilmente se desarrollan aplicaciones que puedan ser clasificables en sólo uno de los campos antes citados, en lo general la mayor parte de las aplicaciones de inteligencia artificial son una mezcla de todos los campos.

2.1.2 LOS SISTEMAS EXPERTOS (SE)

Un SE es un conjunto de circuitos (hardware) y/o programas (software) capaces de simular el comportamiento de un Experto Humano (EH), en el momento de resolver un problema en determinada área del conocimiento o del saber.

En realidad un SE sólo se comporta como un EH 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.

2.1.3 PROPOSITOS DE CONSTRUCCION DE LOS SE

Diagnóstico

Consiste en identificar las causas internas que provocan un problema o avería, a partir de la interpretación de datos previos sobre: el funcionamiento, la estructura del proceso y/o sistema a diagnosticar.

Reparación

Corrección o terapia: consiste en determinar las acciones concretas para la resolución de un problema.

Control	Consiste en conducir o guiar los procesos de manera secuencial (diagnóstico, reparación u otras tareas), según su tiempo de respuesta son de dos tipos: Un SE de control en tiempo real. Es aquel en el que se realiza el proceso de corrección permanentemente y el que antes de recibir el siguiente dato ya ha ejecutando acciones de manera correcta. Un SE de control en tiempo diferido. Es aquel en el que los datos de entrada son interpretados y corregidos según consideraciones del sistema, pudiendo transcurrir un periodo de tiempo sin que el sistema trabaje correctamente.
Simulación	Consiste en crear modelos basados en hechos reales con el propósito de estudiar su comportamiento, interpretando y analizando los datos de entrada y salida.
Planificación	Su propósito es el ordenamiento de tareas para cumplir con un objetivo global.
Diseño	Agilizar procesos iterativos donde se modifican los datos de entrada hasta que cumplan con las necesidades y/o restricciones del diseño.
Educación	Utilizados para crear medios lógicos de comunicación didácticos, que permitan interactuar al usuario con el sistema.
Prototipos	La construcción de prototipos tiene como objetivo principal el simular problemas complejos, en entornos adaptados, para tener un control sobre los posibles problemas que presentarán en la etapa de pruebas.

2.2 CARACTERISTICAS GENERALES DE LOS SE

2.2.1 ADQUISICION DEL CONOCIMIENTO

Los SE no son capaces de iniciar el aprendizaje por sí mismos, como lo hace un ser humano, para que los SE puedan automatizar la resolución de problemas es necesario que previamente se introduzca una base de conocimientos, la que regularmente se extrae de Expertos Humanos (EH).

2.2.2 GRADO DE FIABILIDAD DE LOS SE

En los programas tradicionales, la fiabilidad y explicación de los resultados está en función de la justificación de los algoritmos* utilizados en el programa, cuando los programas incluyen cálculos probabilísticos y/o estadísticos la explicación y la fiabilidad de los mismos puede ser determinada a través de la función de probabilidad asociada.

En el caso de los SE, el grado de fiabilidad puede determinarse con la capacidad del SE para explicar los procesos elegidos.

2.2.3 AREA DEL DOMINIO DEL CONOCIMIENTO

Al igual que un Experto Humano, (EH), los SE sólo dominan un área finita del conocimiento. Para un EH su límite es su propia creatividad, mientras para un SE su limitación es de carácter físico: memoria disponible, velocidad de procesamiento, la que esta relacionada con el tamaño de la base de datos necesaria y el tipo de máquina utilizada.

2.2.4 PROCEDIMIENTO DE SOLUCION DE PROBLEMAS

Para que un EH pueda resolver problemas dentro de un área determinada del conocimiento, es necesario que cumpla dos elementos: Una cantidad finita de reglas (relaciones dentro del conocimiento) y una estrategia que le permita abordar el problema.

En el SE la estrategia general de resolución constituye el control del sistema y recibe el nombre de Motor de Inferencia (MI) también denominada Estructura de Control.

Para mejorar la velocidad de respuesta de un SE, el MI puede ser dotado de diferentes tácticas que así lo permitan, la de mayor de las veces se recurre al empleo de metareglas como se explicará en posteriores apartados.

2.3 ESTRUCTURA BASICA DE LOS SE

2.3.1 MOTOR DE INFERENCIA

Las funciones básicas de el MI son las de seleccionar, interpretar y explicar una parte de la Base de Conocimiento en función de una Base de Hechos, con el fin de solucionar un problema planteado.

El MI se encarga de guiar al SE en la búsqueda de la solución a un determinado problema, para lograrlo hace uso de estrategias de búsqueda almacenadas como Metaconocimiento.

El Metaconocimiento se constituye por reglas que relacionan las diferentes partes de la Base de Conocimiento almacenado, estas reglas permiten construir alternativas de flujos de acuerdo con problemas particulares a resolver.

En resumen el metaconocimiento funciona como guía del MI para que éste realice las siguientes actividades:

- a) Búsqueda del conocimiento aplicable
- b) Elección del conocimiento
- c) Evaluación del conocimiento

2.3.2 LA BASE DE CONOCIMIENTO

La Base de Conocimiento, (BC), es la parte del SE que almacena el conocimiento para el cual éste es válido, su representación dentro del programa debe ser la más sencilla y compacta con el propósito que ocupe el menor espacio posible y pueda a la vez ser manipulada e interpretada con relativa facilidad, también puede ser representada en módulos independientes con el propósito de poder modificar o excluir alguno sin afectar al resto.

2.3.3 BASE DE HECHOS

La Base de Hechos (BH) es el conjunto de información introducida por el usuario o ya incorporada en el sistema, que sirve como datos de entrada y/o condición para solucionar un problema en un determinado dominio del conocimiento.

2.3.4 LOS MODULOS DE COMUNICACION

Dentro del SE existen dos tipos de módulos de comunicación:

- Con el programador. Conocido como Módulo Experto, que permite crear la interfase para el mantenimiento, configuración y depuración del SE.
- Con el usuario. Denominado Módulo del Usuario, que permite la interfase de diálogo (de manera sencilla, lo más aproximado al lenguaje natural) con el usuario, permitiendo que éste pueda tener cierto dominio del SE manipulando algunas variables como pueden ser: la entrada de datos, elección de procesado, justificación y explicación de resultados.

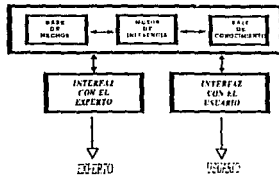


Fig. 2.1 Esquema de bloques de un SE.

Una representación gráfica de los componentes de un SE es la que se presenta en la ilustración (1), donde se puede observar la estrecha vinculación que existe entre la Base de Hechos, la Base de Conocimiento y el Motor de Inferencia.

2.4 FUNCIONAMIENTO DEL MOTOR DE INFERENCIA

2.4.1 MECANISMOS DE BUSQUEDA

Generalmente a los MI de los Se se les dota de una búsqueda de tipo ordenado.

La búsqueda de tipo ordenado se basa en la continuidad del conocimiento, a lo cual se denomina encadenamiento del conocimiento, una búsqueda de tipo ordenado se lleva a cabo cuando la evaluación lógica (cierto o falso) de una regla se convierte el antecedente para que se ejecute una regla subsecuente.

Los encadenamiento de tipo ordenado utilizados son:

Encadenamiento hacia delante:

Consiste en enlazar conocimiento a partir de la entrada de datos para así solucionar el problema, este encadenamiento se utiliza de forma común en cálculos de tipo numérico y tiene gran similitud con el método científico.

Encadenamiento hacia atrás:

Es de tipo inductivo y tiene gran similitud con el proceso de contrastación (del método científico), para utilizar este tipo de encadenamiento frecuentemente se debe conocer de antemano la solución a la que se desea llegar.

Encadenamiento mixto:

Es el que con más frecuencia se emplea en la construcción de los SE, consiste en seleccionar una respuesta de un conjunto de soluciones haciendo uso del encadenamiento hacia adelante, para así guiar la búsqueda y después verificar que ésta sea la correcta, utilizando el encadenamiento hacia atrás. El uso de encadenamiento mixto puede llevar a una búsqueda infinita, por lo que es necesario equipar al sistema con detectores de este tipo de falla.

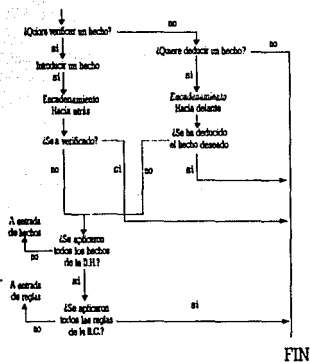


Fig. 2.2 Encadenamiento hacia atrás y hacia delante.

2.4.2 ELECCION DEL CONOCIMIENTO

Es posible crear una estructura de control a través de lenguajes de tipo procedimental (basic o pascal), si bien se sabe que en estos lenguajes una instrucción se ejecuta hasta que se ejecuta la anterior, es posible cambiar la posición de ejecución por medio de condiciones e identificadores (etiquetas*), en estos casos el mismo programa constituye la estructura de control, y es dependiente de los algoritmos del programa.

En lenguajes de tipo funcional las diferentes partes del programa se ejecutan en el momento que son necesarias, en este caso el programador no debe preocuparse por etiquetar las partes en que dividió el conocimiento, pues en el momento de la ejecución el propio programa recurrirá a la parte del conocimiento que sea necesaria, tantas veces como se requiera.

2.4.3 METACONOCIMIENTO

El metacocimiento es la parte del SE que permite cambiar de estrategia para solucionar un problema, en otras palabras el metacocimiento define, el o los lugares de la BC en la cual el MI debe buscar, elegir y aplicar el conocimiento para responder un cuestionamiento concreto.

El metacocimiento puede ser de 4 tipos:

- 1) Ciego y fijo. Este tipo de metacocimiento se encuentra inherente al MI y el usuario no tiene acceso al mismo.
- 2) Seleccionable antes de la ejecución del proceso. El usuario elige entre una serie de alternativas las más aproximadas a las características del problema a solucionar.
- 3) Seleccionable durante la ejecución del proceso. El sistema pregunta al usuario características secundarias del problema a medida que va obteniendo resultados parciales y de esta manera redireccionar la búsqueda si es necesario.
- 4) Externo. Cuando el usuario permanece en constante comunicació con el SE y realmente es el quíen se encarga de guiarlo.

2.4.4 LOGICA

Para lograr automatizar la toma de decisiones respecto al camino que debe seguir el SE, se utiliza de manera frecuente los siguientes tipos de lógica:

Según el tipo de variables:

- Lógica Bivaluada o Booleana, la que sólo admite dos tipos de valores: verdadero o falso.
- Lógica trivalente*, que es aquella en la que se admiten tres tipos de valores: falso, verdadero y otro intermedio.
- Lógica Multivalente, donde las variables pueden tomar muchos valores, pero de forma discreta.
- Lógica Difusa o Borrosa, es la generalización de la lógica multivalente, en este caso las variables pueden tomar valores del tipo continuo entre dos valores límite.

Según el tipo de conocimiento empleado:

- Lógica monótona. En la que los nuevos hechos generados no pueden entrar en contradicción con los que ya existen.
- Lógica no monótona, en la que los hechos se admiten en la BH aún siendo contradictorios, este tipo de lógica es muy importante si se recuerda que un SE no trabaja con conocimientos finales e infalibles.

Según la temporalidad:

- Lógica atemporal. En la que no se considera la influencia del tiempo.
- Lógica temporal donde el tiempo puede determinar el estado de un hecho.

Según el tipo de valores:

- Lógica de orden 0. La que admite proposiciones, es decir representaciones de condiciones del tipo Objeto-Valor.
Ejemplo:
SI Temperatura = Alta ENTONCES Sensación = Calor
- Lógica de orden 0+. En la que se puede tener representaciones del tipo Objeto-Atributo-Valor.
Ejemplo:
SI Temperatura >35 grados ENTONCES Sensación = Calor
donde:
Objeto = Temperatura
Atributo = Calor
Valor = 35
- Lógica de orden 1 o lógica de predicados. En la que se generaliza la lógica de orden 0+.
Ejemplo:
SI Temperatura (grados) Y Grados > 35 ENTONCES Sensación (Calor)

2.4.5 EVALUACION DEL CONOCIMIENTO

Cuando el MI a logrado determinar la parte de la Base del Conocimiento aplicable para solucionar un problema particular, pasa a verificar que éste sea el correcto.

Los tipos de evaluación del conocimiento son:

- a) Determinísticos cuando para su aplicación necesariamente deben cumplirse ciertas condiciones.
Ejemplo:
SI A Y B ENTONCES C
De manera particular:
SI patas (4) Y sonido (ladra) ENTONCES animal (perro)
- b) Probabilísticos cuando los parámetros de evaluación son hipótesis probables, es decir dependen de una probabilidad de ocurrencia de los hechos que los fundamentan, en este caso es necesario conocer de antemano tales probabilidades.
Ejemplo:
SI A Y B ENTONCES C(0.4) Y D(0.6)
De manera particular:
SI personal (edad laboral) ENTONCES situación laboral (paro(0.2)) Y situación laboral (activo(0.2))
- c) Aproximados cuando el grado de veracidad de los resultados esta en función de la veracidad de los hechos y datos empleados, en este caso para conocer la veracidad no se puede aplicar la teoría clásica de probabilidad y es necesario desarrollar nuevas maneras de evaluación aproximada.
Ejemplo:
SI A(a) Y B(b) ENTONCES sugiere C(c) Y excluye D(d)
De manera particular:
SI Nublado (n) Y Temperatura (t) ENTONCES nieve (a)
- d) Análogos cuando es posible suponer que las propiedades de un hecho las puede poseer otro por similitud.
Ejemplo:
SI A SIMILAR B ENTONCES TIENE PROPIEDADES DE B
- e) Hederitarias cuando un hecho adquiere o toma las propiedades del hecho que lo produce.
Ejemplo:
SI A HIJO DE B ENTONCES A TIENE LAS PROPIEDADES DE B

2.5 TECNICAS DE REPRESENTACION DEL CONOCIMIENTO

Para construir la Base de Conocimiento y ésta tenga funcionalidad al incorporar al SE, se utilizan diversas técnicas, pero generalmente se recurre a las siguientes:

2.5.1 REGLAS DE PRODUCCION

Es la técnica más usada, ya que es formulación mas inmediata del principio de causalidad (causa-efecto) y dada su gran sencillez de comprensión.

Una regla es la conjunción de una acción (efecto) que para su ejecución se debe cumplir una condición (causa).

La definición anterior se generaliza de la siguiente forma: Una regla es un conjunto de acciones (efectos) que necesariamente para su ejecución se debe cumplir un conjunto de condiciones (causas).

En resumen, un fenómeno puede ser representado por un conjunto de reglas de producción, lo que se logra mediante la unión de un conjunto de hechos ciertos y un conjunto de reglas de producción, que relaciona y generalizan los hechos.

La formulación general de una regla es:

SI <condiciones> **ENTONCES** <conclusiones o acciones>

donde:

- Acciones. Es la transformación de un hecho.
- Conclusiones. Es la creación de un nuevo hecho válido o la incorporación de una nueva característica al hecho.

Cuando se utilizan reglas en la construcción del metaconocimiento se les denomina metaregla. Las metareglas facilitan la resolución de problemas y disminuyen tiempos de procesamiento, ya que si la BC se desarrolla en forma modular, las metareglas sirven para dirigir la búsqueda a módulos donde se encuentra almacenada la información precisa.

Generalmente se utilizan dos tipos de Metarreglas:

Las de tipo ciego. Que contienen información sobre la sintaxis (estructura) de las reglas:

Ejemplos de metareglas de tipo ciego:

- Elegir la regla más corta
- Elegir la primer regla
- Elegir la regla mas probable

Las de tipo no ciego o inteligentes. Que contienen información sobre la semántica (contenido) de las reglas:

Un ejemplo de metarregla no ciega es:

Un árbol joven no es alto, por lo que no es posible aplicar la regla X

Las principales ventajas de las reglas de producción son su rápida representación del conocimiento y metaconocimiento, la independencia que existe entre ellas mismas permite la supresión o inclusión de reglas sin afectar relativamente a las demás.

Las desventajas más notorias del uso de esta técnica son: incurrir en el uso de metareglas para poder relacionar reglas elementales, utilizar un número amplio de reglas (lo que influye en la velocidad de procesamiento) y por último la posibilidad de incluir reglas repetidas o reglas contradictorias sin que el programador pueda percatarse fácilmente.

2.5.2 REDES SEMANTICAS

Esta técnica, también se le conoce como redes asociativas, sirve para representar el conocimiento mediante el uso de nudos o nodos (lugar donde se representan unidades de conocimiento) y ramas o arcos (que constituyen las relaciones entre las unidades).

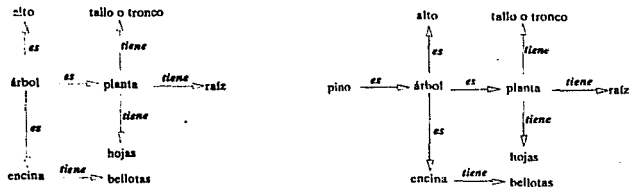


Fig. 2.3 Redes semánticas.

En la parte izquierda de la figura 2.3 se presenta un ejemplo de red semántica si a ésta se le incorpora un nuevo hecho (El pino es un árbol) se transforma en la red semántica contenida en la parte derecha de la misma figura.

De la primer red de pueden obtener los siguientes hechos:

- El árbol es una planta y tiene raíz
- El árbol es una planta y tiene tronco

Al incorporar el nuevo hecho se infiere que:

- El pino es un árbol; tiene raíz; tiene hojas...

Las ventajas de las redes semánticas radican en su alto grado de definición de relaciones y su gran adaptabilidad a procesos de tipo iterativo, las desventajas más importantes son su poca flexibilidad y complejidad de comprensión que se genera al representar una base de conocimiento relativamente grande. Por lo general esta técnica se utiliza conjuntamente con la de reglas de producción.

2.5.3 LOS OBJETOS ESTRUCTURADOS

Un objeto estructurado es una extensión de las redes semánticas, que además incluye características procedimentales, dentro de esta técnica de representación destacan tres tipos de objetos estructurados:

Marcos Los que normalmente conocemos como fichas, combina las características de las reglas y las redes semánticas, puede ser utilizada en forma de módulos y admiten la representación del conocimiento en forma declarativa y procedimental. Los marcos se forman por un nombre y una serie de ranuras donde se guardan las relaciones de herencia entre ellos, también se puede guardar procedimientos para calcular valores o reglas de condición particulares. Los marcos se utilizan en representación de estereotipos.

Ejemplos:

NOMBRE	encina
Tipo	árboles
Corteza	gris
Frutos	bellota
...	

NOMBRE	árbol
Tipo	planta
Tamaño	alto
...	

Objetos Son similares a los marcos, la diferencia radica en que en este caso la relación se define a través de mensajes.

ejemplo:

Sean las siguientes clases de objetos:

nombres pino, encina

árboles planta, perenne, alta, tronco desnudo

Y sea el mensaje:

Los <nombres> SON <árboles>

Se puede inferir que:

El pino es una planta perenne, alta y con tronco desnudo

La encina es una planta perenne, alta y con tronco desnudo

Guiones En los guiones las situaciones se describen mediante una sucesión de acciones de actores que las realizan (agentes) y unos objetos que la sufren (pacientes). Son muy apropiados para tratar situaciones descritas en lenguaje natural (humano).

Ejemplo:

actores: pino, encina

objeto: piñones, bellotas

suceso: el pino tiene piñones

2.6 METODOLOGIA DE CONSTRUCCION PARA LOS SE

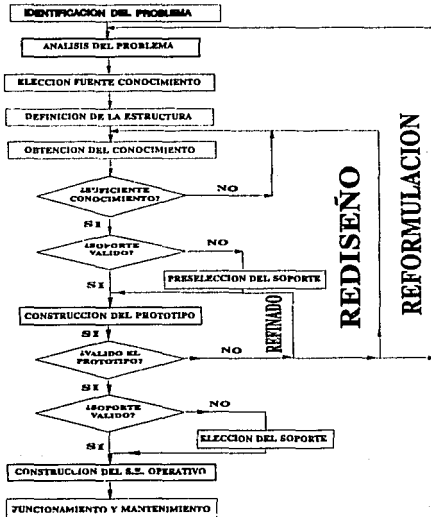


Fig. 2.4 Proceso general de construcción de un SE

Una metodología general para la construcción de un SE se muestra en la fig. (1.3), en la que se pueden observar los pasos significativos del proceso, así como los ciclos iterativos necesarios. Existen otras metodologías de construcción como la propuesta por Kemper (1991).

2.6.1 ESTUDIO DE LA DEMANDA

En esta primer etapa se deben tener claro los objetivos para los cuales se pretende crear el SE, éstos pueden ser de tres tipos:

- a) Para obtener experiencia, formar un grupo de especialistas, en determinado campo del conocimiento, y/o dar a conocer conocimientos recientes, sin la necesidad de estar presente el Experto Humano.
- b) La necesidad de automatizar la resolución de problemas en un área del conocimiento que por medio de las técnicas tradicionales no es posible representar, o las bases de datos que se ocupan son muy grandes y elevan la complejidad de la programación.
- c) Cubrir la demanda de usuarios (no expertos) en un área del conocimiento, en la cual necesitan resolver problemas.

2.6.2 ANALISIS DEL PROBLEMA

En esta fase se determina si es posible resolver una serie de problemas, por medio de la creación de un SE también se determinan los propósitos o tareas que cumplirá el SE en el momento de entrar en funcionamiento y por último, se determina y delimita el dominio de conocimiento que se utilizará.

Las condiciones para construir un SE son:

- El proceso de solución contiene condiciones para que se realicen procedimientos optativos, es decir no existe una solución única.
- El conocimiento empleado no se encuentra en libros o manuales, y si se encuentra, resulta difícil y lenta su utilización.
- No existen ecuaciones o algoritmos que definan totalmente su comportamiento, o solo existen algoritmos que modelan parte del fenómeno y es posible ensamblarlos por medio de un SE.
- Debe tratarse de un problema concreto para que se puedan establecer las fronteras de dominio en la BC, es decir que el conocimiento a utilizar tenga características finitas.

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.

TIPO	TAMAÑO	RESOLUCION	FUNCION	SOLUCION
Númérico	Pequeño	Algoritmo	Cálculo	Clásica
Númérico	Indiferente	Indiferente	Enseñanza	SE
Númérico	Grande	Probabilidades	Cálculo	Estadístico
Númérico	Infinito	Algoritmo	Indiferente	SE
Númérico	Indiferente	Se desconoce	Indiferente	SE
Simbólico	Indiferente	Indiferente	Indiferente	SE
Num/Sim	Indiferente	Indiferente	Indiferente	SE (Híbrido)

Tab. 2.1 Parámetros para validar la viabilidad de un SE.

Sánchez y Beltrán (1990), propusieron una serie de parámetros con los cuales validar la viabilidad de solución de un problema por medio de un SE, los que se resumen en la tabla 2.1, otra alternativa de validación es la propuesta por Kemper (1991).

Finalmente en esta etapa deben quedar bien definidos los objetivos y las herramientas necesarias para cumplirlos.

2.6.3 ELECCION DE LA FUENTE DE CONOCIMIENTO

En esta fase, es necesario definir y asegurar que un experto o un grupo de ellos, esté dispuesto a colaborar en la realización del SE, los expertos que intervendrán deben ser reconocidos por los futuros usuarios del sistema.

2.5.4 PRESELECCION DEL SOPORTE

En esta etapa, se eligen en primera instancia, la herramienta o lenguaje más apropiado para la construcción (ya en el computador) del SE. Se debe tomar en consideración en la evaluación del soporte, qué tipo de razonamiento se va a utilizar (motor de inferencia), también, qué tipo de representación del conocimiento es la más adecuada (reglas de producción, marcos.), y el tiempo de respuesta (real o diferido).

Es importante, en el momento de la preselección, considerar las facilidades de desarrollo, puesto que en la fase de implantación frecuentemente se deberán hacer modificaciones a la BC y otras partes del SE.

2.6.5 OBTENCION DEL CONOCIMIENTO

El conocimiento debe extraerse de un experto humano, formalizarse y representarse, primero en papel, para después representarlo en el soporte preseleccionado.

Este es el punto que consume mayor tiempo y consta de siete pasos generales:

- 1) Familiarización (marco teórico) con el conocimiento referente al problema que se desea solucionar.
- 2) Delimitación del tipo de problemas a solucionar, en esa área del conocimiento
- 3) Determinar que experto o expertos son los mas idóneos así como su disponibilidad de ayuda.
- 4) Motivar a los expertos para que participen en el desarrollo de el SE y colaboren hasta la entrada en operación del mismo.
- 5) Plantear la fórmula general de resolución del problema con las aportaciones del grupo de expertos.
- 6) Resolver dudas sobre el contenido de la BC, es decir obtener conocimiento al detalle.
- 7) Completar la BC del SE ya sobre el soporte, esto regularmente se realiza en un prototipo.

2.6.6 SELECCION DEL SOPORTE

En esta fase, se reanaliza el soporte que previamente se determinó como el más conveniente, evaluando las ventajas y desventajas del mismo, ya que al llegar a este punto se cuenta con información detallada sobre las características del SE, las que de manera general definen el tipo de MI requerido, el tamaño de la base de datos, las relaciones de interfase de entrada y salida con programas previos, el número de usuarios y las condiciones de trabajo con las que regularmente cuentan éstos, así como sus conocimientos sobre computación, por último qué tan incompleto es el conocimiento y por tanto qué tipos de cambio requerirá en el futuro.

2.6.7 CONSTRUCCION DEL PROTOTIPO DEL SE

La construcción del SE se realiza en un prototipo, con el objetivo de corroborar la validez del conocimiento.

En ocasiones aquí termina el trabajo, pues lo que se desea demostrar es la viabilidad de los SE, para que los Expertos Humanos sigan trabajando en este sentido.

2.6.8 VALIDACION DEL PROTOTIPO

En esta etapa se hacen pruebas con problemas concretos que son resueltos de manera simultánea por el SE y caminos paralelos, para finalmente comparar resultados.

Es importante recalcar que el SE puede responder adecuadamente a una serie de problemas de validación, pero esto no significa que lo hará en el futuro para otros problemas.

2.6.9 CONSTRUCCION DEL SE

En este paso se requerirá tal vez cambiar de soporte al SE, para que resulte mas rápida su operación, así como su regeneración y distribución.

2.7 LENGUAJES Y HERRAMIENTAS PARA SE

Un SE se define a partir de la estructura de programación utilizada y no por la herramienta o lenguaje utilizado, por otro lado también es importante considerar, que sea cuál sea la técnica de programación utilizada, en el momento de ejecutarse el programa éste es transformado a código de máquina y es ejecutado secuencialmente por el microprocesador, lo que significa que desde cualquier lenguaje se puede construir un SE.

2.7.1 LENGUAJES

Lenguajes Imperativos:

Lenguajes donde el control del programa pasa a la siguiente línea, a menos que se le ordene lo contrario (figura 2.5).

Ventajas:

Su gran flexibilidad y el conocimiento casi universal de sus comandos.

Desventajas:

Un desarrollo muy largo y complejo, el motor de inferencia es inherente a la base de conocimiento, la estructura del SE es difícil de interpretar además de no poder ser modificado fácilmente.

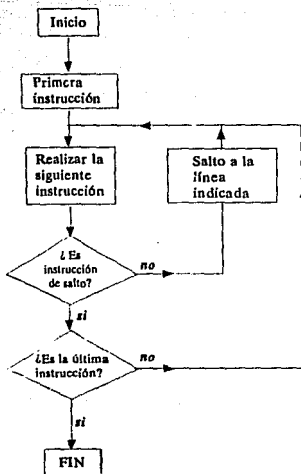


Fig. 2.5 Diagrama de bloques de un programa realizado en lenguaje imperativo

Los lenguajes de tipo imperativo que más se han utilizado son: BASIC, por su gran sencillez, PASCAL, por ser estructurado y el C, cuyo uso se está generalizando en muchos laboratorios de Inteligencia Artificial.

Lenguajes Funcionales:

También denominados aplicativos, éstos a diferencia de los de tipo imperativo, tienen un flujo de ejecución determinado por las necesidades que aparecen después de las etapas de valuación de una función o problema inicial, cuando se utiliza este tipo de lenguajes para la construcción de un programa no es necesario referir las posiciones de las diferentes condiciones y/o reglas, pues el control de tipo secuencial es sustituido por cuatro pasos de evaluación: 1) análisis de la función a evaluar, 2) búsqueda de la primera subfunción, 3) evaluación de la subfunción y 4) Retorno a la función anterior o valor final de la evaluación si se encuentra en la primera función.

Ventajas:

Un SE desarrollado en éste tipo de lenguaje nos permite omitir la declaración de relaciones entre las partes del programa, la sencillez de la construcción del MI y el rápido aprendizaje de los comandos.

Desventajas:

Poco eficaz para el tratamiento de problemas con arquitectura tradicional (necesidad de cálculos numéricos reiteradamente), su manejo es en entornos cerrados, por lo que es muy difícil que puedan manejar programas externos así como bases de datos, por último el desarrollo de este tipo de lenguajes es muy rápida de tal manera que difícilmente existen versiones del mismo lenguaje compatibles.

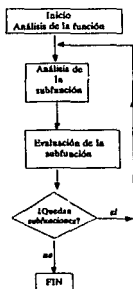


Fig. 2.6 Diagrama de bloques de un programa realizado en lenguaje funcional

Lenguajes Orientados al Objeto:

Su característica mas importante es la indistinción entre procedimientos y datos, pues los programas que se construyen con estos lenguajes se forman por los objetos (ejemplo: métodos, procedimientos, algoritmos) y los datos también son llamados datos locales o facetas*.

Un ejemplo de objeto es "pino" y puede heredar las propiedades de su antecesor que por ejemplo puede ser "árbol". Al grupo de propiedades de un objeto se le denomina atributos, un ejemplo de atributos de "pino" pueden ser: "fruto", "tronco", "raíz".

En resumen, el objeto es la particularización de una clase (grupo de objetos) de la que hereda sus propiedades. Por medio de este proceso generativo los objetos pueden ordenarse jerárquicamente en clases y subclases.

Cuando se ejecuta un programa en lenguaje orientado al objeto, no se desarrollan acciones como en los otros lenguajes; en este caso se generan mensajes que se envían entre sí los objetos definidos.

La programación por objetos puede explicarse en los siguientes pasos:

- Identificar a los objetos que aparecen a lo largo del problema y en la solución.
- Clasificar los objetos por sus semejanzas y diferencias.
- Redactar los mensajes que interrelacionan los objetos
- Implantar los métodos o procedimientos en los correspondientes objetos.

Lo que se resume en: introducir datos, proponer hipótesis, ordenarlas y verificarlas si se crean nuevas hipótesis se regresa a ordenarlas y verificarlas de lo contrario se propone una solución en base a las hipótesis válidas (ver figura 2.6).

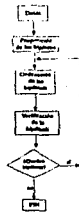


Fig. 2.6 Diagrama de bloques de un programa realizado en lenguaje orientado al objeto

Lenguajes Declarativos:

El método de programación consiste en indicar el objetivo que desea demostrar, especificando un universo de conocimientos (BC) con el cual debe ser demostrado y las condiciones para ello (que, como, cuando, donde, ...).

EL lenguaje más conocidos es PROLOG.

Las características principales de este tipo de programación son:

- La representación del conocimiento se hace en base al empleo de reglas
- El MI ya existe, en el caso de Prolog la lógica utilizada es de orden 1 con búsqueda en profundidad y con marcha hacia atrás
- Los lenguajes son pequeños y no necesitan que se declare la estructura de control

Los inconvenientes mas frecuentes son:

Las operaciones de entrada y salida son complejas de programar.

El MI es fijo y no parametrizable.

La forma de representación del conocimiento es única y es casi imposible crear interfases con otros entornos (relacionarse con otros procesos de programación).

Poco eficaz sobre arquitectura tradicionales.

Poco fiable al no aparecer explícito el comportamiento del programa, pues el MI (estructura de control) es implícita.

El elemento fundamental de PROLOG son los predicados (lo que se afirma de un sujeto), esta compuesto por un nombre y una serie de atributos, un ejemplo:

abuelo(A, B)

Lo que indica que el abuelo de A es B

Si el predicado es información inmutable se le denomina **hecho**, si por el contrario es variable, se le denomina **dato**.

Es posible estructurar una base de conocimientos en base a conjugar el empleo de reglas y predicados.

Por ejemplo:

abuelo(A, C) **IF** padre (A, B) **AND** padre (B, C) **OR** madre(A, B) **AND** madre (B, C).

Lo que indica que el abuelo de A es B **SI** el padre de A es B **Y** el padre de B es C **O** la madre de A es B **Y** la madre de B es C

2.7.2 Herramientas

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

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

La mayoría de los entornos se fabrica 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 ...).

Cuentan con:

- Editor, generalmente se manejan a través de ventanas y utilizando mouse.
- Subsistemas de detección de contradicciones, repeticiones o errores en la BC y BH.
- Subsistemas de explicación y justificación de procesos del MI.

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

Los sistemas vacíos (shell) o armazones, 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.

Al igual que los entornos, las ventajas de los sistemas vacíos, son su sencillez en lo que respecta al uso de los comandos y la rapidez que presentan en la construcción de SE.

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 SE.

La mayoría de estas herramienta esta programada en LIPS, lenguajes imperativos y un reducido grupo de ellos está desarrollado en PROLOG.

CAPITULO III

PROCEDIMIENTO DE ANALISIS

3.1 ANTECEDENTES

Los avances logrados en el estudio del comportamiento dinámico de presas de tierra y enrocamiento han mostrado que los aspectos más significativos del fenómeno son: a) la geometría de la presa, b) la distribución y comportamiento de los materiales no lineal de los mismos y c) las características con la que se considera actúa la excitación inducida en la base de la cortina.

Los diferentes métodos hasta hoy desarrollados, difieren en cuanto a las consideraciones hechas para establecer el tipo de relación existente entre los tres aspectos significativos ya mencionados.

Una clasificación de los diferentes métodos de análisis dinámico es de acuerdo con las hipótesis hechas para modelar los efectos sísmicos en la presa, (las de interés para este estudio):

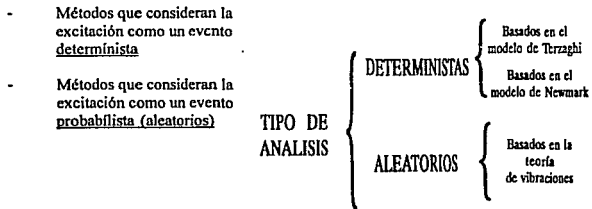


Fig. 3.1 Clasificación de los métodos de análisis dinámico.

3.1.1 METODOS DETERMINISTAS

Métodos Seudo-Estáticos

Los métodos seudo-estáticos se basan en el modelo propuesto por Terzaghi:

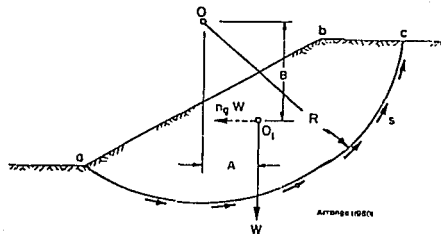


Fig. 3.2 Método de Seudo-Estático

Terzaghi propuso representar la energía generada por un sismo, como una fuerza horizontal equivalente actuante en el centro de gravedad de la masa potencial de falla. El valor de esta fuerza es igual al producto del peso de la masa potencial de falla por un coeficiente sísmico.

Las alternativas de selección del coeficiente van desde valores empíricos hasta los obtenidos de modelar la presa como un cuerpo viscoelástico. Una comparación de diferentes alternativas de selección del coeficiente sísmico se muestra en la fig. 3.3:

Por ejemplo, Freeman (1932) utilizó valores para el coeficiente sísmico que varían entre 0.10 y 0.50 donde a mayor energía del sismo mayor es el valor del coeficiente.

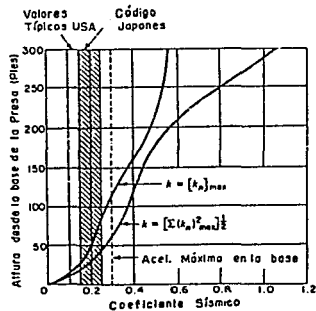


Fig. 3.3 Comparación de Coeficientes Sísmicos.

La teoría desarrollada por Terzaghi considera que la aceleración inducida por el sismo se distribuye uniforme en todo el cuerpo de la cortina.

El modelo es poco aproximado a lo que en realidad ocurre, puesto que se ha probado que el comportamiento de una presa de tierra ante solicitaciones sísmicas se asemeja más a la respuesta de un cuerpo flexible que a la de un cuerpo rígido.

Los factores de seguridad calculados por este método presentan dos desventajas importantes:

- a) Si se utiliza la aceleración máxima del sismo el factor de seguridad obtenido es muy conservador, puesto que la aceleración máxima sólo ocurre en un pequeño intervalo de la duración total del sismo.
- b) En ocasiones es posible obtener factores de seguridad menores a la unidad y sin embargo, el talud puede seguir en equilibrio, esto se debe principalmente a el valor del coeficiente sísmico utilizado.

Métodos Basados en la Teoría del Equilibrio Límite

Otros tipos de métodos deterministas son los que se basan en el modelo de Newmark (1965) y el concepto del equilibrio límite.

El principio fundamental de estos métodos es, determinar de antemano los desplazamientos máximos admisibles para una determinada superficie de falla antes que ésta llegue a la falla.

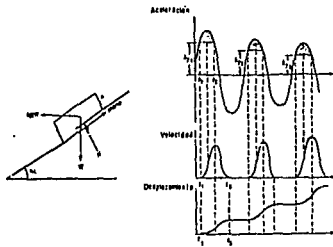


Fig. 3.4 Modelo de Newmark.

En general todas las metodologías que utilizan el modelo de Newmark como base, consideran que la resistencia al corte es rígido-plástica y por lo tanto que el deslizamiento ocurre hasta el instante en que se supera la resistencia de fluencia.

Sarma (1975, 1980) retomó estos conceptos para obtener las relaciones existentes entre la variación de las fuerzas inerciales y los incrementos de presión de poro con el valor del factor de seguridad, la aceleración de fluencia y los desplazamientos producidos.

Para lograrlo realizó pruebas de laboratorio para determinar la variación de la presión de poro en función del número de ciclos de aplicación de carga, tales pruebas demostraron que la presión de poro tiene gran influencia en la estabilidad dinámica del talud y por tanto afecta considerablemente el factor de seguridad y de la aceleración fluencia, en lo que respecta a los desplazamientos, concluyó que éstos están en función de tres parámetros: a) la aceleración máxima del sismo, b) el período natural de vibración y c) la relación existente entre la aceleración crítica en la superficie de falla y la aceleración máxima que se presenta en la cresta de la presa.

ANÁLISIS DINAMICO SEED-LEE-IDRISS

Con los avances registrados tanto en las técnicas de laboratorio como en las de campo utilizadas en la determinación de propiedades dinámicas en los suelos, y utilizando el método del elemento finito, Seed et al (1979) desarrollaron una metodología muy novedosa.

Los pasos que propone son:

- 1 - Determinar la o las secciones transversales para las cuales se desea realizar el análisis de estabilidad dinámico.
- 2 - Seleccionar el acelerograma máximo actuante en la base con el que se realizará el análisis.
- 3 - Calcular el estado de esfuerzos en condiciones estáticas.
- 4 - Determinar las propiedades dinámicas de los materiales (módulo de rigidez al corte, el amortiguamiento y la variación de éstos en relación con el nivel de deformaciones angulares.
- 5 - Calcular (utilizando elemento finito) los esfuerzos inducidos por la acción de la excitación seleccionada.
- 6 - Partiendo de los esfuerzos calculados (estáticos y dinámicos), determinar las deformaciones y la variación de las presiones de poro, para lo cual se deben realizar pruebas de laboratorio sobre muestras representativas de los materiales constitutivos de la cortina.
- 7 - Estimar el factor de seguridad de la presa durante y después del sismo, tomando en cuenta la distribución de las presiones de poro y las características de la deformabilidad de los materiales constitutivos (determinadas en el paso anterior).
- 8 - Finalmente si la presa es segura ante el sismo, entonces calcular las deformaciones permanentes ante un efecto combinado de cargas estáticas y dinámicas.

El utilizar la metodología descrita requiere de un conocimiento previo sobre las características de esfuerzo-deformación, resistencia al corte, comportamiento de los suelos típicos, fundamentos del método del elemento finito, así como contar con un acertado criterio ingenieril, puesto que es necesario juzgar el ajuste de los resultados a lo que realmente ocurre en la presa.

3.1.2 METODOS ALEATORIOS

Los métodos aleatorios (a diferencia de los deterministas) consideran la excitación como un evento probabilista la cual puede representarse mediante un espectro de potencia o bien un espectro de respuesta.

La excitación se considera una función aleatoria del tiempo cuyos valores específicos no se pueden predecir con exactitud, pero sí aproximadamente mediante una función de densidad de probabilidad que puede obtenerse a partir de un conjunto de excitaciones conocidas.

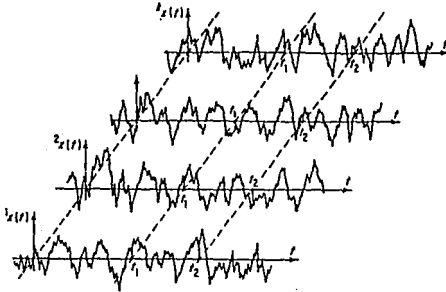


Fig. 3.6 Representación de un sismo como una función del tiempo.

Singh y Khatua (1978) estudiaron el comportamiento dinámico de una presa modelada como una malla de elementos finitos utilizando un sismo en terminos de espectro de potencia. La no linealidad de los suelos la introdujeron mediante un proceso iterativo de linealización y así lograron minimizar errores en la solución de la ecuación de movimiento, al considerar n propiedades del suelo.

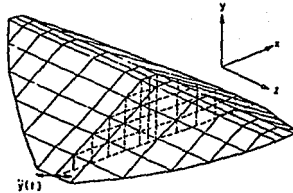


Fig. 3.7 Malla de elementos finitos para una presa ejemplo.

La estabilidad es evaluada por el criterio de Palmgran-Miler que supone una acumulación lineal del daño producido en el elemento finito a causa de los esfuerzos cíclicos ocasionados por el sismo, el factor de seguridad se determina dividiendo el esfuerzo necesario para llegar al falla entre el acumulado (por la acción del sismo).

3.2 METODO DE ANALISIS

En lo general el método de análisis dinámico aquí propuesto tiene como base los trabajos de Seed et al (1975) y Romo et al (1989).

Con respecto al método de Seed existen algunas diferencias en cuanto al número y el orden como se ejecutan los pasos por él propuestos.

Los trabajos de Romo y Villarraga son el elemento fundamental en este trabajo para conocer la respuesta sísmica de presas de tierra, aspecto esencial para poder desarrollar el análisis dinámico de ellas.

3.2.1 SELECCION DE LA EXCITACION

Para llevar acabo este paso es conveniente coordinar los estudios con especialistas del ramo (Ing. Geólogos y Sismológicos), a continuación se reseña (de manera general) algunos de los procedimientos utilizados con mayor frecuencia para la determinación de la excitación de control o diseño:

- a) Si se cuenta con registros sísmicos de algún acelerógrafo empotrado en la roca cercana al sitio donde se desea realizar el análisis dinámico, es posible determinar de éstos la excitación de control.
- b) Cuando la información que se tiene corresponde a puntos lejanos al lugar de análisis se obtiene la excitación utilizando la teoría de propagación de ondas sísmicas (Shnabel, 1972), teniendo en cuenta los efectos de atenuación del sismo.
- c) Si no se tiene un registro sísmico en roca pero se cuenta con uno obtenido en la superficie de lugar, es posible determinar el sismo en roca utilizando la teoría de propagación.
- d) Si no se cuenta con ninguno de los registros anteriores, se puede generar un histograma sintético a partir de un espectro de diseño basado en la sismicidad local.

- c) El procedimiento más común para obtener la excitación de control es modificando un sismo ocurrido en otro sitio, tomando en cuenta las características del lugar (fallas geológicas y sismicidad de la zona). Los parámetros que se adecuan al sitio de análisis son: a) la aceleración máxima, b) la duración y c) el período predominante.

3.2.2 RESPUESTA DINAMICA

La respuesta sísmica de una presa de tierra se evalúa de acuerdo con el modelo teórico empleado por Romo y Villarraga en (1989) con el cual es posible conocer la distribución de aceleraciones en la cortina.

A continuación se describe de manera general dicho modelo.

El modelo considera válidas las siguientes tres hipótesis:

- a) La presa puede ser modelada por elementos finitos tridimensionales, como el mostrado en la figura 3.7.
- b) El sismo puede ser considerado como una vibración aleatoria estacionaria, representada por un espectro potencia medio, obtenido a partir del espectro de respuesta de diseño (el método se clasifica dentro de los métodos aleatorios).
- c) La presa está desplantada sobre una base rígida que ante la acción del sismo se desplaza uniformemente.

Romo y Villarraga proponen que para un modelo no amortiguado de elementos finitos y una excitación en la base la ecuación de movimiento está dada por:

$$[M] \{\ddot{u}(t)\} + [K^*] \{u(t)\} = -[M] \{\ddot{x}\}(t) \quad (3)$$

donde

$\{u(t)\}$	son los desplazamientos nodales relativos a la base rígida
$\{\ddot{u}(t)\}$	son las aceleraciones nodales correspondientes
$[M]$	es la matriz de masa
$[K^*]$	es la matriz compleja de rigidez
$\{r\}$	es el vector de cargas que indica la dirección de la excitación
$\ddot{x}(t)$	es la aceleración de la excitación en la base rígida

Método de la Respuesta Compleja

Para solucionar la ecuación de movimiento, (3), se utiliza el método de la respuesta compleja que consiste en considerar que el sistema es lineal y por tanto es válido el principio de superposición de efectos y que la excitación inducida en la base puede ser modelada como una suma finita de armónicos es decir como una serie de Fourier truncada (4), donde N es el número de frecuencias especificadas en el espectro de respuesta e incluidas en el análisis.

$$\ddot{y}(t) = Re \left(\sum_{s=0}^N \ddot{y}_s e^{i\omega_s t} \right) \quad (4)$$

De la misma manera, la función que define los desplazamientos relativos puede representarse como una sumatoria de funciones armónicas, es decir:

$$u(t) = Re \left(\sum_{s=0}^N u_s e^{i\omega_s t} \right) \quad (5)$$

sustituyendo las ecuaciones (4) y (5) en la ecuación (3), ésta se transforma en:

$$[M] \{ \ddot{u}(t) \} + [K^*] Re \left(\sum_{s=0}^N u_s e^{i\omega_s t} \right) = -[M] \{ x \} Re \left(\sum_{s=0}^N \ddot{y}_s e^{i\omega_s t} \right) \quad (6)$$

Finalmente se obtiene un sistema de ecuaciones para cada una de las frecuencias ω_s ($s=0,1 \dots N$) consideradas en el análisis:

$$([K] - \omega_s^2 [M]) \{ u_s \} = [M] \{ x \} \ddot{y}_s \quad (7)$$

El sistema puede resolverse empleado algún método numérico, como puede ser el método de eliminación de Gauss. Así mismo puede calcularse la función de transferencia compleja de los desplazamientos, $\{u_s\}$, dada por:

$$H^d(\omega_p) = [[K^*] - \omega_p^2 [M]]^{-1} [M] \{r\} \quad (8)$$

de ésta puede obtenerse el vector de funciones de transferencia compleja para las velocidades y otro para las aceleraciones y a través de éste último es posible conocer la distribución de aceleraciones máximas en el cuerpo de la cortina.

Respuesta aleatoria del Sistema

Como ya se mencionó, el modelo considera una excitación con características aleatorias, para introducir tales características al modelo se utiliza la expresión propuesta por Romo y Villarraga, (1989); Romo, (1976):

$$P_y^j(\omega) = |H^j(\omega)|^2 P_x(\omega) \quad (9)$$

donde:

$P_y^j(\omega)$ es el espectro de potencia para los desplazamientos del nudo j.

$H^j(\omega)$ es la función de transferencia compleja de los desplazamientos en el nudo j, con respecto a las aceleraciones en la base.

$P_x(\omega)$ es el espectro de potencia de la excitación en la base rígida.

El valor medio cuadrático de los desplazamientos, velocidades, aceleraciones y deformaciones, es posible evaluarlo a partir de los espectros de potencia respectivos, usando la ecuación (9) y la expresión:

$$(\sigma_y^j)^2 = \sum_{\omega=0}^N P_y^j(\omega) \Delta\omega = \sum_{\omega=0}^N P_x(\omega) |H^j(\omega)|^2 \Delta\omega \quad (10)$$

Los valores máximos esperados se determinan como múltiplo del valor medio cuadrático, utilizando:

$$E_y^j = U_c(n) \sigma_y^j \quad (11)$$

donde $U_c(n)$ es el límite de confianza para C, con una duración T de la excitación (Vanmarke, 1976):

$$U_c(n) = [21n [2n [1 - \exp^{-q(\pi 2n)^{0.5}}]]]^{0.5} \quad (12)$$

donde:

$$n = \frac{-vT}{\ln C} \quad (13)$$

$$v = \frac{1}{2\pi} \frac{m_2^{0.5}}{m_0^{0.5}} \quad (14)$$

$$m_1 = \int_{-\infty}^{\infty} \omega^2 P(\omega) d\omega \quad (15)$$

m_1 , el momento espectral respecto al origen

$$q = \left[1 - \frac{m_1^2}{m_0 m_2} \right]^{0.5} \quad (16)$$

Para tener en cuenta los efectos no lineales del sismo en los materiales, se utiliza un proceso iterativo basado en el método equivalente de Seed e Idriss (1969), con el cual se logra una compatibilidad relativa entre las propiedades dinámicas (módulo de rigidez y amortiguamiento) y los niveles de deformación que el sismo produce.

Las deformaciones máximas de corte de cada uno de los elementos finitos se calculan con el tensor de deformaciones correspondiente y con el cual es posible obtener las deformaciones principales.

3.2.3 ESTABILIDAD DEL TALUD

La evaluación de estabilidad del talud se realiza en base al método de Bishop suponiendo que son válidas las siguientes hipótesis:

- a) El tipo de falla que se presenta puede modelarse como un superficie circular.
- b) La masa de suelo deslizante puede dividirse en dovelas.
- c) Se cumple la ley de Mohr-Coulomb.
- d) El factor de seguridad se define como el cociente de la división del Momento Resistente (debido a la resistencia promedio a lo largo de la falla) entre el Momento Motor (debido a los esfuerzos cortantes medios actuantes en dicha superficie).

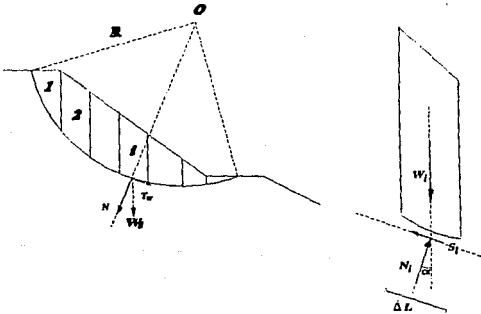


Fig. 3.8 Modelo teórico para cálculo de estabilidad con un solo estrato.

En la figura 3.8 se muestra un talud de un material cohesivo-friccionante. Para las condiciones ahí mostradas el factor de seguridad se calcula como sigue:

Procedimiento de Análisis

Los valores de las fuerzas de reacción normal, N_i , y tangencial, T_i , que actúa en la base de la dovela i , pueden obtenerse usando las ecuaciones 17 y 18 respectivamente:

$$N_i = W_i \cos \alpha_i \quad (17)$$

$$T_i = W_i \sin \alpha_i \quad (18)$$

Se acepta que la ecuación 19 es una buena aproximación al valor del esfuerzo normal actuante en la base de la dovela i , el que se considera actúa uniformemente:

$$\sigma_i = \frac{N_i}{\Delta l_i} = \frac{W_i}{\Delta L_i} \cos \alpha_i \quad (19)$$

Conocido el valor del esfuerzo y utilizando la ley de resistencia al esfuerzo cortante de Mohr Coulomb (ecuación 20), es posible determinar el valor de S_i , que se considera actúa uniformemente en la base de la dovela.

$$s_i = c + \sigma_i \tan \phi \quad (20)$$

La fuerza resistente en la dovela i es igual al producto que resulta de multiplicar el esfuerzo resistente calculado con la ecuación 20 por la longitud de la base:

$$F_i = S_i \Delta L \quad (21)$$

El momento resistente con respecto al centro del círculo de falla, M_R , es igual a la suma de los momentos producidos por las fuerzas resistentes en cada una de las dovelas:

$$M_R = \sum RF_i = \sum RS_i \Delta L_i \quad (22)$$

Por otra parte el momento motor con respecto al centro del círculo de falla se calcula como el giro que produce el peso de la superficie de falla, si el peso de cada dovela se descompone en sus componentes normal, W_{N_i} , y tangencial, W_{T_i} , como posible observar en la

figura 3.8 sólo la componente tangencial produce momento ya que la normal pasa por el punto O, la ecuación del momento motor, M_M , queda definido como:

$$M_M = \sum RW_i = \sum RW_i \operatorname{sen} \alpha_i \quad (23)$$

y el factor de seguridad se calcula con la siguiente ecuación:

$$F_s = \frac{M_R}{M_M} = \frac{\sum S_i \Delta L_i}{\sum W_i} = \frac{\sum (c \Delta L_i + W_i \cos \alpha_i \tan \phi)}{\sum W_i \operatorname{sen} \alpha_i} = \frac{\sum c \Delta L_i + N_i \tan \phi}{\sum W_i \operatorname{sen} \alpha_i} \quad (24)$$

el radio del círculo de falla sale de las sumatorias puesto que es constante y por tanto puede eliminarse.

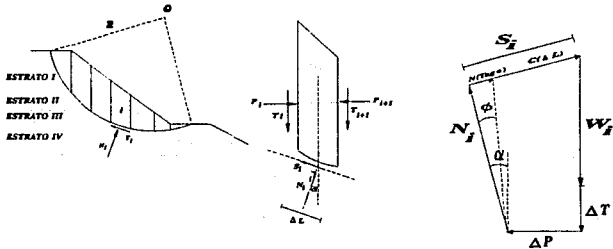


Fig. 3.9 Cálculo del Factor de Seguridad para varios estratos, fuerzas actuantes en los costados de cada dovela y diagrama de cuerpo libre para el modelo de Bishop.

Si se varían las condiciones de la figura 3.8 para introducir al modelo una estratificación en el talud, como el mostrado en la figura 3.9, las ecuaciones 17 y 18 siguen siendo válidas en lo general, solo que en este caso el peso se obtiene sumando el peso de cada uno los elementos de suelo contenidos en la dovela.

El valor de S_i se obtiene de acuerdo con la ley de resistencia del estrato en el que se encuentra la base de la dovela i y el factor de seguridad se obtiene según la ecuación 24.

Método de Bishop

Bishop refinó el método de las dovelas, puesto que en lo visto antes se ha supuesto que las fuerzas que se generan a los lados de las dovelas se contrarrestan entre sí, Bishop introduce al modelo los efectos que estas producen.

Retomando la expresión para calcular el factor de seguridad (ecuación 24) y trabajandola únicamente para la dovela i , se puede despejar W_i :

$$W_i = \frac{s_i \Delta L_i}{F_s} = \frac{N_i \tan \phi}{F_s} + \frac{c \Delta L_i}{F_s} \quad (25)$$

Por otra parte para la misma dovela i (en estudio) se muestra el polígono de fuerzas en la fig. 3.9 de donde al sumar las fuerzas en dirección vertical se obtiene que:

$$\begin{aligned} W_i + \Delta T &= N_i \cos \alpha_i + \left[\frac{N_i \tan \phi}{F_s} + \frac{c \Delta L_i}{F_s} \right] \text{sen} \alpha_i \\ \text{despejando } N_i : \\ W_i + \Delta T - \frac{c \Delta L_i \text{sen} \alpha_i}{F_s} & \\ N_i &= \frac{W_i + \Delta T - \frac{c \Delta L_i \text{sen} \alpha_i}{F_s}}{\cos \alpha_i + \frac{\tan \phi \text{sen} \alpha_i}{F_s}} \end{aligned} \quad (26)$$

El factor de seguridad se calcula conforme la ecuación 24, pero el valor de N_i se obtiene según la expresión 25, de lo que se puede observar que el factor de seguridad es función de sí mismo, así su valor se obtiene a través de un proceso de aproximaciones.

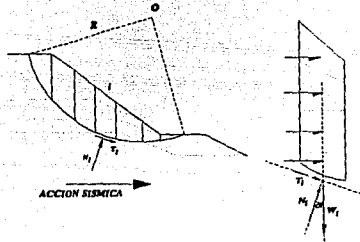


Fig 3.10 Cálculo de el Factor de Seguridad en condiciones dinámicas.

Para calcular el factor de seguridad en condiciones dinámicas se introduce al modelo de la figura 3.9 fuerzas actuantes en los centroides de cada elemento, como se muestra en la figura 3.10.

La fuerza se calcula como el producto de la masa del elemento por un coeficiente de aceleración cuya unidad base es el valor de la gravedad, obtenido a partir de un análisis de respuesta dinámica.

Al momento motor obtenido con la ecuación 23 deben sumarse los momento producidos por cada una de las fuerzas actuantes en los centroides de los elementos

En el caso que se desee trabajar con esfuerzos efectivos, a el valor obtenido de la ecuación 19 se debe restar la presión de poro, u , si esta existe en ese estrato.

3.2.4 PERDIDA DE BORDO LIBRE

La utilización de las ecuaciones (1) y (2) ha dado resultados confiables en presas homogéneas y de altura no muy importante, no así en el caso de presas de altura importante y con una estratificación de materiales (como es el caso de las presas de tierra y enrocamiento).

Los trabajos de Romo (1980) con presas estratificadas demuestran que la resistencia al corte, el módulo de rigidez máximo, E, y los esfuerzos cortante y confinante, presentan una variación con respecto a la profundidad, para estos casos se propuso la siguiente expresión:

$$\frac{\delta^{\max}}{H} = \frac{1}{4.65[(F-1)\frac{E}{\tau}] + 1.34[(F-1)\frac{E}{\tau}]^2 + 1.16[(F-1)\frac{E}{\tau}]^3} \quad (27)$$

donde:

- E es módulo de rigidez máximo
- τ es el esfuerzo cortante
- F es el factor de seguridad real obtenido a partir del factor convencional que se obtiene a través de análisis de estabilidad con el método del elemento finito y empleando la gráfica de la figura 3.11.

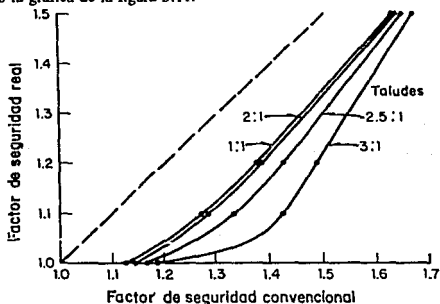


Fig 3.11 Relación entre F, real, obtenido con elementos finitos y el F, convencional, obtenido con el método de Bishop.

Para aplicar la ecuación 27 se considera que el terraplén descansa sobre un estrato muy resistente y la resistencia a la falla se incrementa con respecto a la profundidad, para estas condiciones se supone que en condiciones estáticas la superficie de falla mas critica es la que corresponde al círculo de falla tangente al estrato resistente, Romo et al (1980) y que para el caso de condiciones dinámicas la superficie mas critica se localiza a la mitad de la altura (o un poco mas abajo) de la superficie critica en condiciones estáticas Romo (1980).

La pérdida de bordo libre calculada según la ecuación (1) esta en función de círculo de falla seleccionado, lo que puede llevar a resultados equivocados, un método modificado fue propuesto por Romo (1992) teniendo como hipótesis:

- a) El terraplén se divide en rebanadas horizontales como se indica en la figura 9 (es suficiente con tres o cuatro)
- b) El factor de seguridad utilizado es el obtenido de análisis de estabilidad utilizando elementos finitos, el cual es posible conocer a través de un análisis de tipo Bishop y utilizando la figura 3.11, obtenida por Seed et.
- c) Cuando se conocen los factores de seguridad de las cuñas utilizadas, se evalúa δ_{max}/H utilizando la ecuación número 27, donde H es la distancia que existe, entre la corona de la presa y la primera línea horizontal que define la primer rebanada y subsecuentemente es la distancia entre las diferentes rebanadas.
- d) La pérdida de bordo libre, L, se calcula sumando los efectos de cada una de las rebanas consideradas con la expresión 28 y conforme la figura 3.12.

$$L = \sum \frac{H_i}{B_i + b_i} \{ [\delta_{max}^i]_a + [\delta_{max}^i]_d \} \quad (28)$$

donde:

- | | |
|------------------|--|
| L_i | es la pérdida de bordo libre en la rebanada i |
| H_i | es la altura de la rebanada i |
| b_i | es ancho de la corona de la rebanada i |
| B_i | es el ancho de la base de la rebanada i |
| δ_{max}^i | es el valor de la deformación máxima para la rebanada i |
| N | es el número de rebanadas en que se dividió el terraplén |

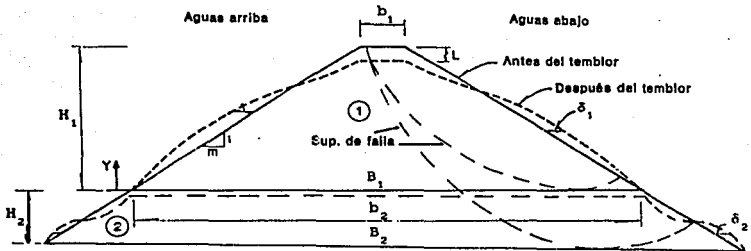


Fig 3.12 Modelo modificado de Romo.

3.3 METODOLOGIA DE ANALISIS

3.3.1 ANALISIS DE RESPUESTA DINAMICA

Para determinar el comportamiento dinámico de la presa, se utiliza el programa **PTLUSH**, con el cual es posible conocer la aceleración máxima en cada uno de los puntos nodales de la malla de elementos finitos tridimensionales utilizada.

Programa empleado

El programa **PTLUSH** resuelve la ecuación de movimiento bajo las hipótesis planteadas en el punto 3.2.2.

Para resolver la ecuación de movimiento el programa utiliza la teoría de la respuesta compleja, las características aleatorias de la respuesta se modelan con la expresión propuesta por (Romo y Villarraga, 1989; Romo, 1976) (ecuación 9).

Resultados del Programa

El resultado de correr este programa es un archivo que contiene los siguientes datos por renglón: N(I), X(I), Y(I), Z(I), AMAX(I), AMED(I), AMIN(I), donde:

N(I)	Número de nodo (identificador)
X(I)	Coordenada X del nodo i
Y(I)	Coordenada Y del nodo i
Z(I)	Coordenada Z del nodo i
AMAX(I)	Aceleración máxima Calculada
AMED(I)	Aceleración media Calculada
AMIN(I)	Aceleración mínima Calculada

donde N(I) es adimensional, X(I), Y(I), Z(I) estas en pies y AMAX(I), AMED(I), AMIN(I) estas en gal's (0.00981 m/s²).

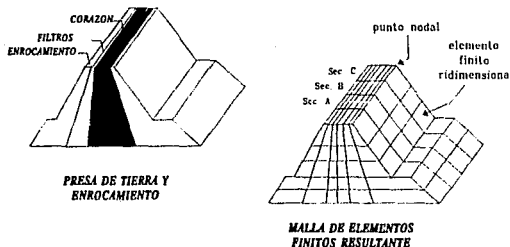


Fig. 3.13 Modelo de elementos finitos de una presa de tierra ejemplo.

3.3.2 ANALISIS DE ESTABILIDAD

El análisis de estabilidad se apoya en el método de Bishop modificado y se realiza para cualquier sección transversal de la presa, siempre y cuando se trate de un plano de frontera entre elementos finitos tridimensionales (utilizados en el análisis de respuesta dinámica), como se muestra en la figura:

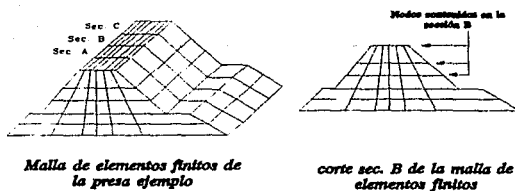


Fig 3.14 Elección de la sección para realizar el análisis de estabilidad.

Procedimiento general

Reconstrucción gráfica de la sección de análisis

La reconstrucción gráfica de la sección a analizar se hace en base a algún archivo de resultados de algún Análisis de Respuesta Dinámica realizado previamente, como ya se mencionó el archivo contiene los identificadores de los puntos nodales y las coordenadas (X,Y,Z) de los mismos.

Las coordenadas de los nudos pertenecientes a la sección a analizar se vacían en un papel milimétrico a una escala conveniente, se procede a unirlos según el corte utilizado en el análisis dinámico, a continuación se sobrepone un papel albanene lo suficientemente transparente para calcar la parte externa de la presa y sobre este papel se dibuja la estratificación (propuesta u obtenida de registros de campo) de los materiales constitutivos de la cortina.

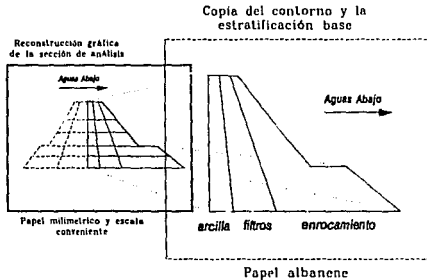


Fig. 3.15 Procedimiento de reconstrucción gráfica.

Estratificación y endovelado de la sección a analizar

Para definir los estratos contenidos en la sección de análisis, se debe tomar en cuenta que los elementos finitos tridimensionales (utilizados en el análisis de la respuesta dinámica) por lo general se definen en las fronteras entre diferentes tipos de suelo y/o entre las propiedades de los mismos, por lo que seguramente existen fronteras de estratos que coinciden con alguna(s) frontera(s) de los elementos tridimensionales.

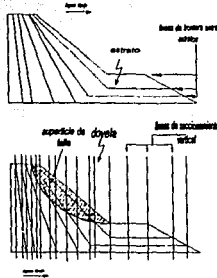


Fig. 3.16 Endovelado y estratificación.

Para llevar a cabo el proceso de estratificación se deben contemplar las siguientes reglas:

- 1) Un estrato queda definido por dos líneas llamadas líneas de frontera, entonces el número de estratos considerados en el análisis es igual al número de líneas de frontera menos uno.
La línea de frontera superior define el contorno de la sección mientras que la línea de frontera inferior define el contacto de la sección y la base rígida.
- 2) Una dovela queda definida por dos líneas de seccionamiento vertical (el programa ANSTA acepta hasta 40 dovelas suficientes para un buen análisis), entonces el número de dovelas utilizadas en el análisis es igual a el número de líneas de seccionamiento vertical menos uno.
- 3) Un elemento bidimensional queda definido por el área contenida entre dos líneas de frontera y dos líneas de seccionamiento vertical, entonces el número de elementos bidimensionales considerados es igual a la multiplicación de el número de estratos por el número de dovelas.

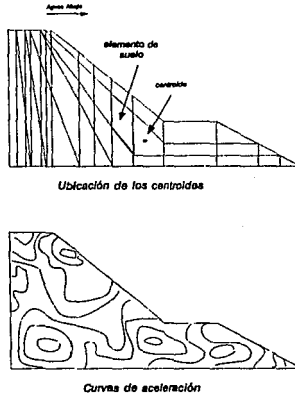


Fig. 3.17 Ubicación geométrica de los centroides y curvas de aceleración para interpolar la aceleración en los centroides.

Edición de la Base de Datos

Archivo de Intersecciones

Este archivo contiene las coordenadas de intersección de las líneas de frontera y las líneas de seccionamiento vertical, contiene además las cotas de profundidad de las grietas de tensión superficial junto con las cotas de la línea superior de flujo (si éstos dos aspectos se consideran en el análisis), para la creación de el archivo se procede conforme al manual del programa ANSTA.

Archivo de las propiedades de los materiales

El archivo se edita de acuerdo con el manual del programa ANSTA.

Cálculo de las Aceleraciones en los Centroides

Una manera sencilla para calcular la aceleración a la que se encuentran sujetos los centroides es dibujando las curvas de aceleración como se muestra en la figura 3.17 a la misma escala que se a venido trabajando y en papel lo suficientemente transparente; para luego sobreponerlo en el dibujo donde se encuentran los centroides y así poder interpolar los valores de la aceleración, un procedimiento alterno es utilizar el modulo de interfase (MODINT), que se explicará en el siguiente capítulo.

Evaluación de la Estabilidad de la Sección

La evaluación de la estabilidad de la sección se realiza utilizando el programa ANSTA que realiza el análisis de estabilidad utilizando el método de BISHOP modificado, el programa analiza la estabilidad del talud para una superficie potencial de falla propuesta por el usuario

Este programa ANSTA es el resultado de una serie de modificaciones hechas al programa SLOAF con el propósito de modificar los procesos de entrada y salida para lograr una mayor automatización en el mismo, se le incorporo una rutina mas que permite generar un archivo de salida especial donde se guardan los resultados que son necesarios para la estimación de la Pérdida de Bordo Libre.

Archivo de Datos y Operación del Programa ANSTA

Para utilizar el programa se debe tener listos los siguientes archivos de estrada:

- I) Archivo controlador
- II) Archivo DASTA1
- III) Archivo de datos
- IV) Archivo RESPALDO

La edición de estos archivos se explica en el anexo 2 (manual de usuario del programa ANSTA).

Resultados Obtenidos

Los resultados de la ejecución del programa ANSTA, se guardan en el archivo de resultados definido desde el archivo controlador, en este archivo se guarda información necesaria para calcular la pérdida de bordo libre y los datos que se guardan son:

Los resultados se guardan conforme el siguiente formato:

desde INICIO	hasta FINAL			
FSC	EEP	TTP	ESFNP	ESFTP
FSC inicial				
...				
FSC final				

donde:

- Inicio es el primer valor que toma el ángulo de fricción de los materiales donde éste es variable
- Final es el último valor que toma el ángulo de fricción
- FSC es el factor de seguridad convencional calculado por el método de Bishop
- EEP es el valor del módulo de elasticidad promedio
- TTP es el valor del esfuerzo tangencial promedio
- ESFNP es el valor del esfuerzo normal promedio
- ESFTP es el valor del esfuerzo tangencial promedio

Variación de la superficie de falla

Los análisis de estabilidad se realizan para los círculos de falla correspondientes a las rebanadas en que se dividió el talud, esto se logra modificando el archivo de datos de la primer superficie.

Variación de las Aceleraciones Inducidas

Es posible utilizar parte de los archivos ya creados para realizar análisis adicionales bajo diferentes condiciones sísmicas, esto se logra realizando un nuevo análisis de respuesta dinámica e intercambiando los nuevos resultados por el archivo utilizado inicialmente para interpolar las aceleraciones en los elementos de la sección de análisis.

Variación de la Sección de Análisis

Para utilizar parte de la base de datos con la que se cuenta, se considera que la estratificación de la presa no varía entre una sección y otra, como se muestra en la figura 3.14, si este es el caso, entonces es posible utilizar parte de la base de datos ya generada pero interpolando las aceleraciones con los nudos de la sección que ahora se desea analizar.

3.3.3 ESTIMACION LA PERDIDA DE BORDO LIBRE

Para la estimación de la pérdida de Bordo Libre se utilizan las expresiones de Romo descritas en la sección 3.2.5.

Una parte de los datos se obtiene de los resultados del programa ANSTA (de acuerdo con la tabla mostrada anteriormente) y la otra parte es información adicional: altura de la rebanada así como el valor de su base mayor y menor (esta información puede obtenerse de reconstrucción gráfica de la sección hecha previamente), la estimación se puede llevar a cabo con una calculadora programable o a través de un programa para PC.

Es posible conocer la variación de la pérdida de bordo libre con respecto a la variación del ángulo de fricción debido a que el programa ANSTA obtiene Factores de seguridad para diferentes ángulos de fricción automáticamente, lo que finalmente es muy importante si se desea conocer la variación de la pérdida de bordo libre con respecto a la degradación de los materiales; por otra parte para conocer la pérdida de bordo es necesario realizar pruebas de laboratorio sobre muestras representativas de la cortina y de esta forma determinar en ángulo de fricción de los materiales, esto en ocasiones resulta muy complejo de realizar y los resultados obtenidos llevan inherentes un cierto grado de incertidumbre por lo que es necesario contar con gráficas de variación de pérdida de bordo libre, para así determinar el valor de la pérdida dentro de un intervalo de confianza.

3.3.4- GRAFICACION DE LOS RESULTADOS

Existen varios tipos de gráficas y la variedad de las mismas depende de la cantidad de análisis realizados.

- 1) **Angulo de fricción vs Pérdida de bordo libre.** Es la gráfica básica y se construye a partir de analizar la estabilidad de una superficie potencial de falla variando el ángulo de fricción de los materiales (lo que se lo logra automáticamente en el programa ANSTA).
- 2) **Angulo de fricción vs Pérdida de bordo libre (para varias superficies de falla).** La gráfica se construye cuando se analizan varias rebanadas en la misma sección.

Es posible construir otro tipo de gráficas adicionales, por ejemplo cuando se realizan análisis utilizando varias rebanadas, en dos o tres secciones diferentes de la presa o modificando condiciones sísmicas a la que se somete la cortina.

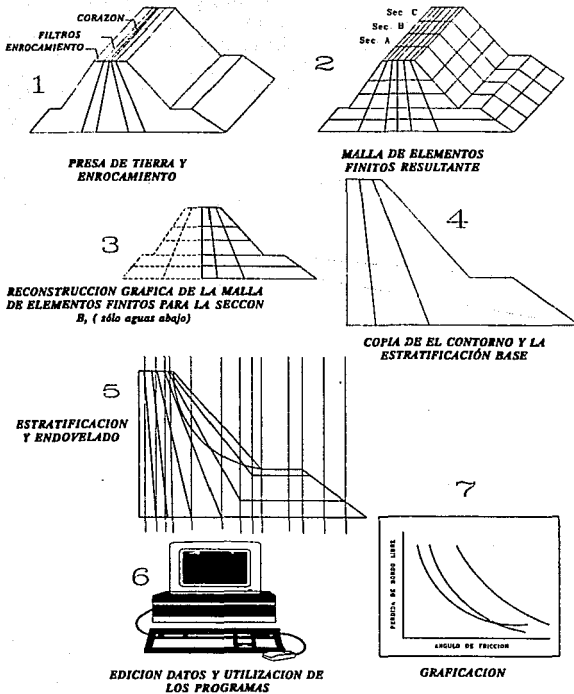


Fig. 3.18 Pasos generales para llevar a cabo un análisis dinámico.

CAPITULO IV

SISTEMA EXPERTO DESARROLLADO

El desarrollo del SE aquí propuesto se hizo en base a la metodología descrita en el capítulo II del presente trabajo, cabe destacar que existen otro tipo de metodologías como la propuesta por Kemper (1991).

4.1 ESTUDIO DE LA DEMANDA

Un manejo eficiente de la metodología, referente al análisis dinámico de presas de tierra y enrocamiento, implica un proceso de entrenamiento previo a fin de contar con el personal competente en ésta rama del conocimiento, este proceso de entrenamiento generalmente significa inversión de tiempo por parte del Experto Humano lo que en ocasiones interfiere con sus actividades diarias.

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

El medio automático (en este caso un programa de computadora) permitiría además automatizar parte del proceso de análisis de estabilidad y por otra parte podría ser un medio idóneo para comunicar los avances logrados por el Experto en el campo que domina.

La construcción de un sistema en esta área del conocimiento cubriría la demanda de usuarios no expertos que necesariamente requieren resolver problemas en esta área.

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

4.2 ANALISIS DEL PROBLEMA

4.2.1 Formulación general del problema

El problema planteado en el apartado anterior es posible resolverlo mediante la creación de un Sistema Experto, que por un lado permita la formación de Nuevos Expertos Humanos y a la vez pueda ser un medio de comunicación de nuevos conocimiento.

Finalmente el problema en sí es la elaboración de un Sistema Experto en el campo de la educación, que pueda conjugar la enseñanza y la simulación, para por un lado explicar al usuario los temas referentes a el análisis dinámico y por otra parte le permita simular y resolver problemas en dicho campo.

La elaboración del Sistema se propone se acompañe de un medio de comunicación lógico (interfase) que permita la interacción del usuario con el sistema.

4.2.2 Objetivos del Sistema

- 1) Desarrollar un medio de comunicación que permita la socialización y difusión de nuevos conocimientos, el cual incorpore características novedosas para que resulte mas atractivo ante los usuarios potenciales.
- 2) Puesto que se trata de un medio de socialización de conocimiento, el sistema resultante debe ser autónomo, en la medida de lo posible, del ambiente en el cual se desarrolló, esto es con el propósito de dotarlo de la mayor transportabilidad posible, tomando en cuenta dos consideraciones básicas: la necesidad de una rápida reproducción del sistema y su utilización en computadoras con características lo mas variadas posibles.
- 3) El sistema debe funcionar como asistente del usuario para la toma de decisiones, por tanto debe permitir consultas específicas sobre un tema o procedimiento en particular.
- 4) El sistema debe además contar con una base de datos, que permita ordenar los diferentes archivos de datos y resultados obtenidos durante los análisis realizados, con el fin de contar con un medio automático de búsqueda de dicha información.

- 5) El sistema debe automatizar por lo menos parte de la metodología utilizada en el análisis dinámico de presas, reduciendo sustancialmente los tiempos requeridos para realizar un análisis.
- 6) El sistema debe trabajar en interacción con programas creados con anterioridad, permitiendo unir diferentes etapas del análisis dinámico que se ha logrado automatizar.

4.2.3 Alcances de desarrollo

Siendo que el sistema a desarrollar se encuentra en los campos educativo, administrativo y de simulación, la solución deberá cumplir con los siguientes alcances mínimos :

- 1) Que el sistema pueda ser desarrollado en base a módulos.
- 2) Que exista una relativa independencia entre los módulos, con el propósito de poder utilizarlos desde el momento mismo en que concluya su etapa de construcción.
- 3) Que los procesos lógicos desarrollados puedan ser reutilizados en la construcción de otros Sistemas Expertos en lo futuro.
- 4) Que el sistema pueda ponerse en marcha sin la necesidad de contar con el ambiente de programación y utilizando esencialmente los comandos del MS-DOS.

4.2.4 Validez de la aplicación

Se considera cumple con la validez de aplicación, tal afirmación se basa en el punto 2.6.3 del capítulo dos del presente trabajo, puesto que el conocimiento empleado aún no se encuentra en manuales o libros específicos y la aplicación del mismo es complicada.

4.2.5 Formulación particular de la solución

La construcción del sistema se desarrollará en base a módulos los cuales se ligaran por medio un menú principal, los módulos propuestos son:

Módulo Explicativo

Con el cual se pretende explicar los diferentes conceptos vertidos en el capítulo precedente y el cual se divide en cuatro vertientes:

- 1) Explicación general del marco teórico que sustenta el método utilizado
- 2) Explicación del método propuesto
- 3) Explicación de los procesos utilizados en la metodología aquí propuesta.
- 4) Una explicación animada de la metodología propuesta.

Módulo Administrativo

El módulo permitirá mantener un control sobre la información generada durante los análisis. En este modulo se pretende ordenar y clasificar el nombre de todos los archivos generados durante los análisis realizados y de esta forma permitir una rápida consulta.

Módulo Interfase

El módulo debe permitir controlar programas previos (codificados en Fortran 77), el objetivo del modulo es demostrar la viabilidad en cuanto al manejo de programas desde un entorno amigable.

4.3 ELECCION DE LA FUENTE DE CONOCIMIENTO

Para definir las principales fuentes del conocimiento, inicialmente se realizó una lista de los posibles Expertos Humanos que posiblemente contribuirían en la construcción del sistema, por otra parte, se recabó información sobre la bibliografía existente de los temas relacionados con el proyecto.

Cabe destacar que las fuentes del conocimiento relacionadas van desde los conocimientos sobre el análisis dinámico en presas de tierra hasta las metodologías y técnicas utilizadas en la construcción de sistemas expertos.

La elección de los Expertos Humanos que intervendrían en el proyecto se hizo en base a la fiabilidad de contar con ellos durante todo el proceso de construcción del prototipo.

4.4 PRESELECCION DEL SOPORTE

4.4.1 Soportes considerados

Para la preselección del soporte se consideraron dos opciones básicas: herramientas desarrolladas (shell's) y lenguajes.

Herramientas

La herramienta considerada es el paquete EXSYS Profesional (1988), utilizada en los trabajos de Kermer(1992) con la que obtuvo magníficos resultados.

Se incluyó en la evaluación considerando que:

- a) La representación del conocimiento es en base a reglas.
- b) La rapidez de ejecución es muy alta.
- c) Puede manejar hasta 5000 reglas.
- d) Puede realizar cálculos numéricos.
- e) El razonamiento utilizado es de tipo combinado (búsqueda hacia adelante y hacia atrás).

- f) Cuenta con opciones para crear interfaces con el usuario con la posibilidad de desplegar hipertextos y gráficos.
- h) El uso del editor y comandos de operación es relativamente sencillo.
- i) Software no es muy costoso (1000 dólares, al año 1990), debido que este paquete fue utilizado previamente por Kermer, no fue necesario adquirirlo.

Lenguajes:

Se consideraron dos lenguajes (Prolog y Turbo Pascal):

- 1) PROLOG (de tipo declarativo) considerado básicamente por utilizar lógica de orden 1 y búsqueda en profundidad, lo cual se amolda a parte de los objetivos del proyecto.
- 2) TURBO PASCAL ver 5.5 (de tipo estructurado) se incluyó en la evaluación debido a la gran variedad de comandos que maneja, estos permiten manejar desde mecanismos de presentación (ventanas) y gráficos, hasta comandos del MS-DOS como son los de visualizar, borrar, renombrar, ejecutar, modificar atributos de archivos entre otros comandos:

Se incluyó en la evaluación considerando que:

- a) La programación es de tipo modular.
- b) Es posible manejar lógica booleana.
- c) El producto final sólo está limitado por la capacidad física de la PC utilizada, el compilador de Turbo Pascal permite ligar diferentes programas.
- d) Resulta relativamente sencillo crear interfaces de comunicación con el usuario.
- e) Puede controlar pantallas y pueden crearse ventanas.
- f) Es capaz de utilizar gráficos y animar imágenes.
- h) El ambiente de programación y el de compilar pueden trabajar de manera conjunta, reduciendo drásticamente los tiempos necesarios para compilar y depurar programas.
- i) El software es fácil de conseguir.

- j) Es posible crear archivos ejecutables e independientes del entorno de programación.

4.4.2 Evaluación de los soportes

Para la evaluación del soporte más idóneo para este Sistema Experto en particular se consideraron las características particulares de cada uno de los módulos en que se divide la solución y los objetivos generales del proyecto.

Motor de Inferencia recomendable:

Mecanismos de Búsqueda

En lo que respecta al módulo de explicativo se requiere realizar una búsqueda en profundidad y encadenamiento hacia adelante, mientras que para el módulo administrativo se requiere utilizar encadenamiento hacia adelante y hacia atrás, el uso de éstos está en función del tipo de consulta que se desee hacer.

La representación del conocimiento

Se propone sea mediante reglas de producción (causa-efecto), el manejo de las reglas requiere utilizar lógica de tipo booleana, que por las características propias del problema es de tipo atemporal y en primera instancia de orden 0.

Pruebas realizadas con EXEYS

Las pruebas demostraron:

- a) Su gran facilidad para construir Sistemas Expertos con poco conocimiento de programación
- b) Que se amolda perfectamente a la solución del problema en cuanto a la representación del conocimiento en base a reglas y la lógica que se desea emplear es la mínima comparada con la que puede utilizar este paquete.

- c) Se concluyó que sí se utiliza el paquete, la creación del motor de inferencia y la lógica de búsqueda no representa ningún problema, puesto que están incluidos en el paquete.
- d) Durante la etapa de pruebas el shell EXSYS no resultó satisfactorio en lo que respecta a la interfase con programas externos, considerando que los programas fuente utilizados en el análisis dinámico están codificados en lenguaje Fortran77 no se encontró un método que permitiera controlar este tipo de programas desde EXSYS.
- f) Finalmente no es posible obtener SE ejecutables utilizando EXSYS.

Pruebas realizadas con Prolog

Resultados:

- a) Los trabajos realizados con Prolog resultaron satisfactorios en lo que respecta a la utilización de reglas de producción, con algunas deficiencias cuando se requiere mas de un efecto para una misma causa.
- b) Resulta difícil su programación y mas difícil aún es comprender y modificar la estructura de un programa.
- c) En lo que respecta a la interfase con programas codificados en Fortran77, ésta no es posible o al menos no se encontró información al respecto.
- d) Para que el usuario pueda utilizar programas codificados en Prolog, indiscutiblemente antes debe comprender los comandos.

Pruebas realizadas con Turbo Pascal vc: 5.5

Resultados:

- a) Las pruebas realizadas resultan satisfactorias en lo que respecta a la facilidad para construir interfases con programas codificados en otro lenguaje.
- b) Desde Turbo Pascal es posible invocar comandos del MC-DOS lo que permite en última instancia simular interfases ejecutando programas ya compilados.

- c) La gran variedad de herramientas para manejo de pantalla y teclado, permite hacer mas atractivo el sistema, el lenguaje cuenta con algunas características de programación denominada orientada al objeto, lo que permitira más adelante convertir los programas fuente a ese tipo de lenguaje.

Es importante señalar que de utilizar este lenguaje es necesario construir el motor de inferencia a sabiendas que quedará inmerso en el cuerpo del sistema y muy difícilmente podrá ser recuperado para ser reutilizado en otro sistema.

4.4.3 Elección inicial del soportes

Tomando en cuenta los objetivos que dieron pie a este proyecto y las tareas que implica utilizar cualquiera de las opciones analizadas se optó por preseleccionar Turbo Pascal 5.5, bajo los siguientes considerados:

- Una parte de los objetivos de la Construcción de este Sistema Experto, y tal vez la de mayor peso, se refieren a lograr la mayor independencia del sistema del entorno de programación, lo que fundamentalmente define el tipo de soporte a emplear.
- La dirección de crecimiento del SE indudablemente involucra crear interfaces con programas de tipo tradicional, que debido al grado de complejidad de la lógica empleada en ellos se requiere utilizar un lenguaje o paquete lo mas flexible posible.
- Cualquiera de las opciones implicaba iniciar desde el principio el estudio del manejo de los comandos empleados, del cual resulta lógico optar por el que pueda ser utilizado no sólo en la elaboración de este trabajo, si no en la mayor parte de los trabajos que impliquen programación en el futuro.

4.5 OBTENCION DEL CONOCIMIENTO

Para llevar a la práctica la elaboración del sistema fue necesario consultar Expertos Humanos en las dos áreas del conocimiento involucradas, por un lado los que se dedican a la construcción de Sistemas Expertos y por otro los relacionados con los análisis dinámicos de presas de tierra.

Puesto que la creación del sistema esta orientado al tratamiento de análisis dinámico de presas de tierra y enrocamiento, fue necesario realizar una serie de análisis de este tipo, para así delimitar el área de dominio del sistema.

Para definir las características generales de dicho análisis se siguieron los siguientes pasos:

- 1) Identificar las características generales y específicas de dicho análisis con el propósito de definir perfectamente los módulos y su alcance.
- 2) Seleccionar las etapas que se desea se desempeñe el Sistema Experto.

Fuentes y niveles de conocimiento:

En lo que se refiere a los elementos teóricos en los que se basa el análisis dinámico las fuentes utilizadas son:

- a) De tipo escrito, artículos aparecidos en memorias de pasados simposios, libros sobre el tema e informes de investigaciones.
- b) De expertos humanos donde se recurrió a colaborar en las investigaciones realizadas por ellos relacionadas con el tema así como al planeamiento de cuestionarios y entrevistas sobre la metodología propuesta.

El marco teórico que susienta el desarrollo del Sistema Experto, ya en la práctica, se encuentra contenido en los dos capítulos precedentes, es claro que los conceptos vertidos en dichos capítulos representan aspectos muy generales, lo que principalmente se debe a dos hechos: la experiencia personal en el manejo de estos conceptos y los alcances mínimos fijados desde el comienzo, por estos motivos se decidió desarrollar de manera mas amplia la parte que más se domina y que se refieren básicamente al análisis de estabilidad dinámico y el cálculo de la pérdida de bordo libre.

Propuesta de construcción del Prototipo del SE

La base de conocimiento adquirido hasta este punto, tanto en la rama de la geotecnia como en la de sistemas permitió proponer el diagrama del prototipo base y comenzar la etapa de pruebas para la construcción del mismo.

En resumen las características generales del prototipo son:

- Los módulos se desarrollan de forma independiente, permitiendo que el usuario pueda trabajar con uno en particular sin tener la necesidad de que los demás módulos estén cargados en memoria de la computadora.
- Los módulos se ligan por un menú principal, que funciona como metaconocimiento base y el cual permite manipular cada uno de los módulos desde un ambiente amigable.
- Los módulos del SE pueden ser invocados por separado utilizando comandos del MS-DOS.

Las características de cada módulo son:

Módulo Explicativo

para lograr una explicación mas eficiente y para tener un mejor control de la información en el momento de programarla se decidió trabajar con tres archivos de textos y una base de datos para cumplir con:

- La explicación general del marco teórico que sustenta el método utilizado
- La explicación del método propuesto
- La explicación de los procesos utilizados en la metodología aquí propuesta.
- La explicación animada de la metodología propuesta.

Módulo Administrativo el módulo almacenará el nombre del archivo, el tipo de archivo del que se trata y características particulares de éste, como por ejemplo, la presa a la que pertenece. En éste módulo el SE será capaz de buscar en su base de datos la ubicación o lugar donde se guardó el archivo que el usuario solicite, también deberá ser capaz de rastrear el nombre de un archivo a partir de una o un grupo de características que el usuario conozca.

Módulo Interfase El módulo constituye la interfase del programa denominado ANSTA (programa para calcular el factor de seguridad de un talud sujeto a fuerzas sísmicas, codificado en lenguaje Fortran), la interfase debe contar con editores de datos que permitan capturar la información necesaria para utilizar ANSTA, debe además contar con los procesos necesarios para preparar el o los archivos de datos que el usuario requiera para ejecutar ANSTA, finalmente será capaz de reconocer los resultados de dicho programa y calcular la pérdida de bordo libre guardando los resultados en un formato lo suficientemente entendible para ser graficado.

Módulo Maestro EL módulo se propone como el elemento de unión entre los demás módulos y que cuente con la capacidad de incorporar nuevos módulos en el momento que estos se desarrollen.

4.6 SELECCION DEL SOPORTE

La selección del soporte se hizo a partir de sostener al lenguaje preseleccionado como el idóneo y compararlo contra las ventajas y desventajas que ofrecen otros procedimientos de construcción.

- a) Es claro que el lenguaje no cuenta con un motor de inferencia predefinido, como el de las denominadas herramientas y sistemas vacíos, por lo se debe crear tal elemento a sabiendas que quedará insertado al sistema desarrollado y será muy difícil recuperarlo para usos posteriores.

- b) No se detectó herramienta alguna que permitiera desarrollar SE y que pudiera ser utilizado con independencia del ambiente de programación, por otra parte sí se detectaron otros lenguajes que cumplen tal objetivo, pero se optó por continuar trabajando con Turbo Pascal puesto que ya se domina parte de la estructura del lenguaje.
- c) En lo que respecta a la base de conocimiento necesaria para construir el prototipo del SE, es posible manejarla bajo el lenguaje preseleccionado debido a sus características de modularidad, pudiendo finalmente ensamblar los programas desarrollados bajo un programa principal.
- d) El lenguaje preseleccionado cumple con las expectativas relacionadas con el manejo desde el SE de programas realizados en otros lenguajes (caso específico Fortran 77).
- e) En lo que toca a los objetivos de transportabilidad del SE, Turbo Pascal resulta muy noble puesto que después de compilar programas fuente, éste puede operarse en cualquier tipo computadora compatible con la IBM 38086 o posteriores, ampliando así el grado de difusión del mismo.
- f) El tipo de estructura del lenguaje permite que los módulos del sistema se vaya perfeccionando con el tiempo y/o se vaya ampliando el número de estos.

4.7 CONSTRUCCION DEL PROTOTIPO

El nombre identificador del prototipo es **SEANDI** (Sistema Experto para Análisis Dinámico)

La construcción del prototipo se realizó en base a módulos independientes con el propósito de que estos, pudieran ser operativos desde el momento mismo de ser terminados.

En general, en todos los módulos se utilizan estructuras de control selectivo, iterativo y repetitivo de acuerdo a las necesidades de cada uno de los módulos, de la misma manera se utilizaron pantallas, ventanas y teclas especiales con el propósito de hacer más atractivo el SE, para el tratamiento de archivos de texto y archivos de datos se utilizaron una amplia gama de tipos de conjuntos de datos estándar y definidos por el programador, en la medida de cumplir con las características particulares de cada uno de los módulos del SE.

4.7.1 Modulo Administrativo

El modulo administrativo se compone de cinco alternativas de trabajo y una mas para salir del modulo, el diagrama de bloques se muestra en la figura 4.1 y el programa fuente se presenta en el anexo IV.

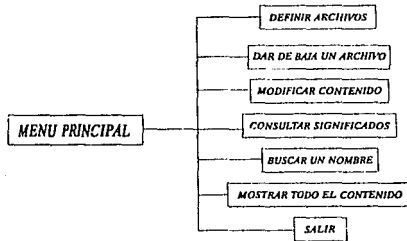


Fig. 4.1 Diagrama de bloques del módulo administrativo

Las opciones de trabajo y sus principales características de funcionamiento se presentan a continuación:

1) DEFINIR UN NUEVO ARCHIVO

La opción sirve para definir el nombre de un nuevo archivo generado durante el análisis, así como sus principales características.

Cuando se selecciona esta alternativa se activa la ventana de entrada de datos, en la que se despliegan el tipo de datos que requiere el SE para clasificar el archivo, si el archivo a clasificar no tiene una de las características especificadas es posible continuar trabajando oprimiendo ENTER, el-SE no considera esa característica en futuras consultas.

2) DAR DE BAJA UN ARCHIVO

Esta opción permite desactivar el nombre de un archivo, con lo cual el SE entiende que el nombre no será consultado en adelante y por tanto no lo considerará para futuras evaluación de búsqueda de información, se recomienda utilizar la opción solo en casos extremos.

3) MODIFICAR EL CONTENIDO

La opción permite buscar el nombre de un archivo de datos para modificar su contenido e incluso el nombre del archivo, el SE busca el nombre deseado, si lo encuentra despliega su contenido y se prepara para la entrada de los nuevos datos, estos datos los guarda el lugar de los que ya contenía, con este procedimiento se logra minimizar el número de nombres declarados, dotando al SE con la capacidad de depuración de información.

4) CONSULTAR SIGNIFICADOS

La opción permite buscar las características de un archivo así como el lugar donde se almaceno dicha información, el SE está dotado de un elemento de búsqueda que permite encontrar el nombre buscado y el lugar donde se almacenó.

5) BUSCAR UN NOMBRE

La opción permite invocar un procedimiento de rastreo a partir de las características de un archivo, desplegando por medio de una ventana el archivo que satisfaga los datos iniciales; el procedimiento de rastreo con que cuenta el SE permite iniciar la búsqueda con sólo las características que conoce el usuario dejando indeterminadas las otras, cuando esto sucede, el SE entiende que debe buscar y desplegar los archivos que contengan las características conocidas por el usuario.

6) MOSTRAR TODOS LOS CONTENIDOS

Cuando se selecciona esta opción el SE activa el procedimiento de despliegue con lo que logra mostrar todos los nombres y características de los archivos almacenados en su base de datos.

7) SALIR

La opción permite salir del módulo y trabaja de dos formas:

- a) Si el módulo se está utilizando a través de el menú principal, el control es transferido a dicho menú.
- b) Si el módulo se invocó desde el sistema operativo, el control regresa a éste después de desplegar una pantalla de despedida.

Recomendaciones de uso general

El módulo cuenta con su propios sistemas de detección de errores, unas de las detecciones con la que fue dotado son las siguientes:

- a) En el momento de definir un nombre, primero busca que no éste contenido en la base de datos, de estarlo despliega un letrero informándolo y el control regresa a esperar se introduzca un nuevo nombre de archivo.
- b) Cuando se busca un nombre o características que previamente no han sido definidas despliega una ventanas con mensajes que así lo hacen saber
- c) En ocasiones los procesos de búsqueda caen en ciclos repetitivos, estos procesos es posible detenerlos utilizando la tecla especial ESC, cuando esta tecla es válida se despliega una ventana que así lo informa.

Una recomendación final y la mas importante se refiere a la lógica inicial del módulo:

Es importante revisar que el archivo de datos esté presente en el directorio en el cuál se cargó el SE, el nombre identificador del archivo es **ALMACEN.DAT**, de no estar presente e invocar el módulo administrativo, el SE entiende que se trata de una nueva base de datos y procederá a crear el archivo iniciándolo desde el principio.

4.7.2 Módulo de interfase

El módulo de interfase permite capturar, modificar y preparar la información previa para ejecutar el programa ANSTA, con este módulo también es posible reconocer los archivos de resultado del programa antes mencionado, cuenta con cinco alternativas de trabajo y una más para salir del módulo, el diagrama de bloques correspondiente se puede observar en la fig. 4.2 y el programa fuente se presenta en el anexo V.

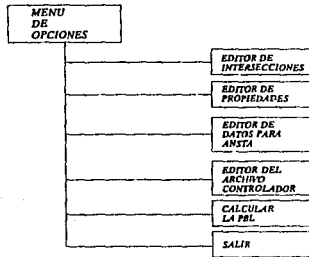


Fig 4.2 Diagrama de bloques del módulo de interfase.

Las opciones de trabajo y sus principales características de funcionamiento se presentan a continuación:

EDITOR DE ARCHIVO DE INTERSECCIONES

La opción esta diseñada para capturar los datos del archivo de intersecciones descrito en la sección 3.2.2 de este trabajo, el SE es capaz de reconocer archivos de datos capturados previamente e incluso cuenta con opciones para modificar su contenido.

EDITOR DE ARCHIVO DE PROPIEDADES

Si se selecciona esta alternativa de trabajo es posible capturar las propiedades de los materiales involucrados en el análisis, al igual que en

la opción anterior, el SE puede reconocer archivos creados previamente y modificar su contenido

EDITOR DE ARCHIVO DE DATOS PARA EL PROGRAMA ANSTA

Esta es la opción que da cuerpo a la interfase con el programa ANSTA, pues a través de ella es posible construir uno o varios archivos de datos, a partir de un archivo de aceleraciones otro de intersecciones y una más de propiedades, esta alternativa sustituye casi en su totalidad los procedimientos descritos en la sección 3.2.2 del capítulo anterior.

EDITOR DEL ARCHIVO CONTROLADOR

El SE es capaz de preparar el archivo controlador del programa ANSTA, al activar esta opción se activarán procedimientos que permiten al SE saber si los archivos que se están declarando existen y así hacerlo saber al usuario.

CALCULAR LA PERDIDA DE BORDO LIBRE

Los procedimientos de esta opción dotan al SE de capacidad para reconocer archivos de resultados generados por ANSTA y que junto con información adicional proporcionada por el usuario es posible determinar la pérdida de bordo libre.

4.7.3 Módulo Explicativo

El módulo explicativo permite que el usuario consulte una base de información cargada en el SE, con la cual es posible acceder a diferentes explicaciones de los diferentes aspectos que se relacionan con el análisis dinámico de presas de tierra.

Para lograr este objetivo el módulo maneja un archivo de texto en ASCII, cuando se invoca esta opción el SE activa un menú principal con el cual se controla el sistema al usuario para que éste pueda ir invocando las diferentes partes del texto según sus necesidades de consulta, el menú principal queda activo durante todo el tiempo que dure la consulta y puede ser llamado a través de la tecla especial ESC, por otra parte para acceder a mayor información sólo se utilizan las teclas especiales que dan movimiento al cursor (las flechas) para iluminar la palabra elegida, para desplegar mayor información se oprime la tecla ENTER, el SE es capaz de retomar el control de manera automática cuando no encuentre mayor información, cuando esto sucede despliega el menú principal del módulo.

Sistema Experto Desarrollado

El módulo también invoca una serie de imágenes animadas que explican al usuario los procedimientos principales para realizar un análisis dinámico.

En la fig. 4.3 muestra el diagrama de bloques correspondiente y en el anexo VI de presenta el programa fuente.

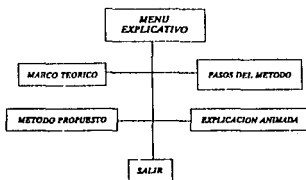


Fig. 4.3 Diagrama de bloques del módulo explicativo

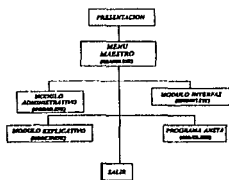


Fig 4.4 Diagrama de bloques del menu maestro.

4.7.4 Módulo Maestro o Principal

El módulo está diseñado para ligar los diferentes módulos antes descritos, junto con el programa ANSTA, el programa fuente está contenido en el anexo VII y su diagrama de bloques se muestra en la figura 4.4.

4.8 EJEMPLO

Para fines de ejemplo, a continuación se describe la manera como se evaluó la Pérdida de Bordo Libre (PBL), para la sección media, (A), de la presa El Infiernillo durante la acción del sismo ocurrido el 14 de marzo de 1979.

La instrumentación instalada en la presa El Infiernillo se muestra en la figura 4.5, a continuación en la figura 4.6 en su parte superior se pueden observar los principales componentes de la presa y en su parte inferior se muestran las principales características geológicas del lugar donde se construyó la obra.

Para realizar el análisis de respuesta dinámica se utilizó la malla de elementos finitos contenida en la figura 4.7 (parte superior), la distribución de aceleraciones máximas, en la sección media, correspondientes a las calculadas con el programa PTLUSH, para el sismo ocurrido el 14 de marzo de 1979 se muestran en la figura 4.7 (parte media) y finalmente la estratificación y edovelamiento (utilizado en este ejemplo) para el cálculo de estabilidad de la sección media se puede observar en la parte inferior de dicha figura.

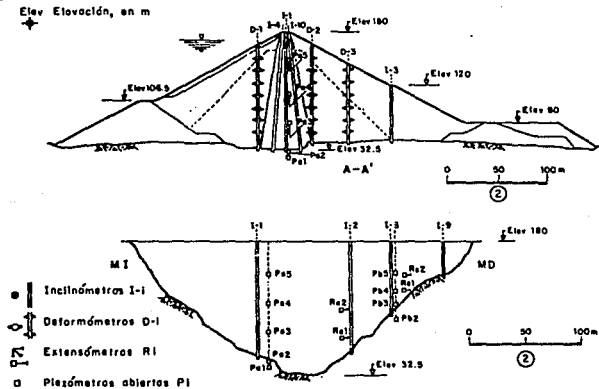


Fig. 4.5 Instrumentación en la presa El Infiernillo.

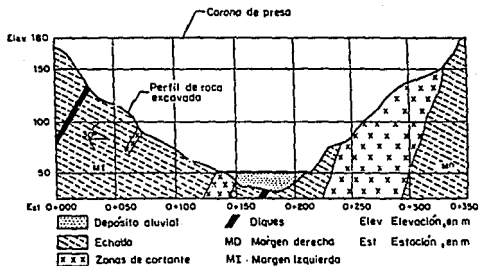
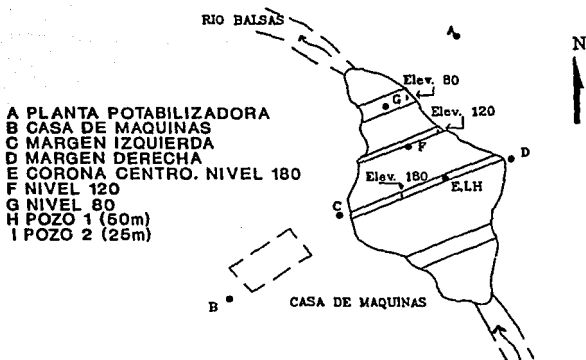


Fig. 4.6 Instalaciones en la presa El Infiernillo y Perfil.

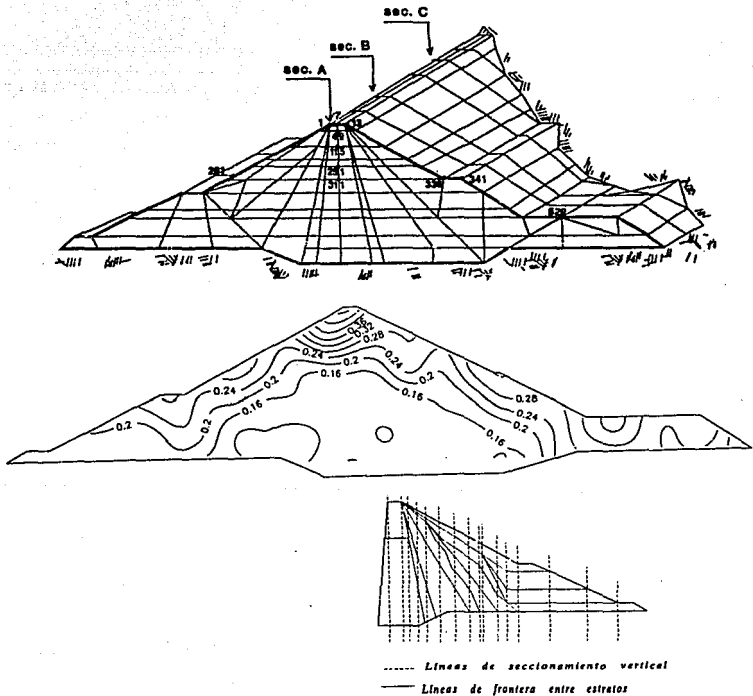


Fig. 4.7 Malla de elementos finitos, distribución de aceleraciones y envolvelado ejemplo para la sección media (a).

Para conocer la PBL con las condiciones antes descritas se procedió como sigue:

- 1) Se corre el módulo de interfase con ANSTA. ya sea desde el menú maestro o desde el sistema operativo teclado MODINT.
- 2) Del primer menú que aparece se selecciona la opción para editar archivos de datos para ANSTA.
- 3) A continuación el programa pregunta por el archivo de aceleraciones a utilizar, si se trabaja con el disco flexible entonces debe escribir:
A:\EJEMPLO\ACEL79.INF
si se trabaja desde el disco duro debe especificar la unidad y el directorio donde cargo el archivo de aceleraciones: ACEL79.INF.
- 4) En seguida el programa pregunta por el archivo de intersecciones a utilizar, en este caso se teclea:
A:\EJEMPLO\INTER.INF
- 5) Después que el programa haya interpolado las aceleraciones, pregunta por el archivo de propiedades, en este caso se utilizó el archivo:
A:\EJEMPLO\PROP.INF
- 6) Si el formato del archivo de propiedades es compatible con el número de estratos considerados en el análisis, entonces el programa pide se introduzca las valores de las coordenadas X y Y del centro y punto obligado del círculo de falla utilizados, en este caso se utiliza información adicional, la que se encuentra contenida en la tabla 4.1
- 7) Finalmente el programa pregunta por el nombre con el cual se va a grabar el archivo, en este caso se utilizó el nombre ANSTA.DAT.
- 8) A continuación se sale de esa opción y se pasa a la correspondiente al editor del controlador de archivos para ANSTA, dentro de esta opción se debe definir el número de archivos de datos (en este caso uno), el ángulo de fricción mínimo (en este caso se utilizó 20), el ángulo de fricción máximo (en este caso 60); a continuación se deben introducir el nombre del archivo de datos (ANSTA.DAT) y el nombre del archivo de resultados: ANSTA.RES.
- 9) Finalmente se sale del módulo y se corre el programa ANSTA.EXE ya sea desde el menú maestro o sistema operativo.

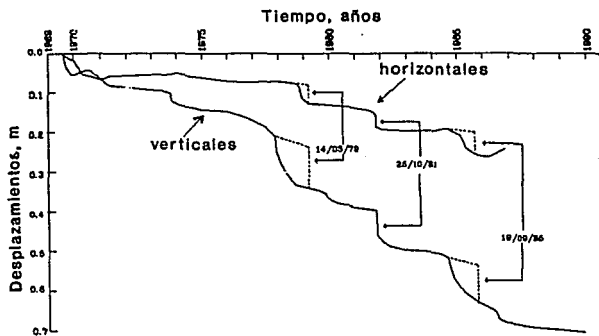
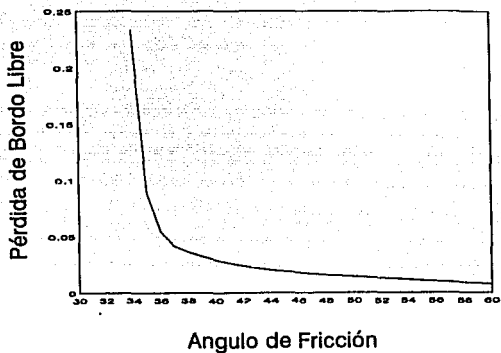


Fig. 4.8 PBL calculada y PBL medida en campo.

- 10) Si se desea calcular la pérdida de bordo libre se vuelve a entrar al módulo de interfase y se entra a la opción correspondiente, el archivo de datos es el archivo de resultados obtenidos del programa ANSTA.EXE, que en este caso es ANSTA.RES y otros datos adicionales (también contenidos en la tabla 4.1) que son: altura base menor y base mayor de la rebanada.
- 11) Finalmente se pide introducir el nombre del archivo donde se van a guardar los resultados, en este caso se le nombró como ANSTA:PBL y los resultados pueden observarse en la parte superior de la figura 4.8.

COORD.	CENTRO	COORD.	Pt.OBLI	DATOS	REBANDA	(m)
X(ft)	Y(ft)	X(ft)	Y(ft)	ALTURA	INFERIOR	SUPERIOR
345	667	0	590.5	84.12	326.74	10

Tabla 4.1 Datos adicionales para obtener la PBL.

De pruebas de laboratorio se sabe que el ángulo de fricción promedio en el talud analizado es 48 grados, si comparamos la pérdida de bordo libre calculada es parecida a la registrada en campo (fig. 4.8 parte inferior).

La tolerancia entre lo medido y lo calculado debe tomar en cuenta que sólo se trabajó con una rebanada (para fines de ejemplo), que los resultados del programa ANSTA pueden variar en función del número de dovelas utilizadas en el análisis (aún con la misma aceleración) y por tanto el factor de seguridad calculado varía a ese respecto y como se sabe la cc. 27 es significativamente sensible a los valores que toma el factor de seguridad, por lo que la pérdida de bordo libre calculada, puede variar significativamente cambiando ligeramente las condiciones de análisis.

Para fines de validación de esta parte del prototipo se retomaron los trabajos de Romo et al (1992), utilizando la misma distribución de materiales y el mismo número de dovelas (40) obteniendo resultados semejantes.

CAPITULO V

CONCLUSIONES

5.1 CONCLUSIONES

- La herramienta y técnica de solución se seleccionan considerando las características del problema a resolver así como las necesidades posteriores de crecimiento del SE.
- La metodología y la técnica utilizada para la construcción de este SE permite enfrentar una gran diversidad de problemas en diferentes ramas del conocimiento de manera muy flexible.
- Los programas fuentes de los módulos en los que se dividió la solución es posible transformarlos posteriormente a lenguaje orientado al objeto.
- El SE desarrollado automatiza gran parte de los procesos utilizados en el análisis dinámico de presas reduciendo significativamente los tiempos necesarios para solucionar problemas en ese campo del conocimiento y dando mayor flexibilidad a los datos de entrada.
- Las pruebas hechas demuestran que cumple con su función de realizar algunas tareas del Experto Humano.
- El SE ya se utiliza como parte del proceso de entrenamiento de nuevo personal en el área de los análisis dinámico de presas.
- Se logró total independencia del ambiente de programación utilizado.
- Se demuestra que puede constituir un vehículo idóneo para la difusión de nuevos conocimientos.
- El prototipo terminado es muy fácil de transportar y copiar, aumentando así el grado de alcance del mismo.

Conclusiones

- El principal problema que presento el SE se refiere al programa de interfase ANSTA, debido a que su programa fuente está en lenguaje Fortran 77, por lo que muchas veces requiere de una recompilación para las características de la máquina.
- Es posible recuperar parte de la lógica empleada para ser utilizada en la construcción de otros SE.

5.2 RECOMENDACIONES

El siguiente paso de desarrollo del SE es incorporar el programa PTLUS, reseñado de manera general en el capítulo 3, como ya se dijo el programa solo puede ser ejecutado en estaciones de trabajo (work-station) es entonces que para lograr acoplar el programa PTLUS al SE éste tenga que incorporarse a una estación de trabajo, esto es posible gracias a que el lenguaje de programación utilizado es compatible con este tipo de máquinas.

De acuerdo con las características del presente trabajo y las ramas que se involucraron para poder concretarlo resulta que hay dos líneas fundamentales a las que se pueden enfocar futuras investigaciones:

En el área de Sistemas

- Desarrollo de procedimientos que permitan la interacción del módulo explicativo y los demás módulos, con el propósito de lograr invocar el modulo explicativo desde cualquier punto del SE.
- Aunque como ya se mencionó el SE desarrollado permite recuperar parte de su lógica para usos posteriores, resulta estratégico lograr desarrollar lógicas puras, es decir desarrollar Shell's propios que se ajusten a las necesidades propias de la geotecnia.

En el área de la geotecnia

- Gracias a que el SE desarrollado permitió automatizar gran parte de los pasos necesarios en el análisis dinámico de presas, es posible desarrollar otras líneas de investigación relacionadas con el tema, como son:
- La influencia de la línea superior de presión de poro en los resultados de estabilidad
- La variación de los resultados al utilizar mayor o menor número de dovelas y estratos.
- La variación de los resultados al incorporar grietas de tensión a diferentes profundidades.
- El aspecto tal vez más importante de investigación que se abre a partir de este SE, es poder calcular la pérdida de bordo libre respecto de la variación de la aceleración como función del tiempo.

REFERENCIAS

- Hernández, J. R; "Respuesta Dinámica de Presas de Boquilla Triangular"; Tesis de licenciatura, Universidad Nacional Autónoma de México; México; 1990.
- Jackson, Peter; "Introduction to Expert Systems"; McDonnell Douglas Research Laboratories; Saint Louis, Missouri, The United States of America; 1990.
- Joyanes, Luis; "Programación en Turbo Pascal (versiones 4.0, 5.0 y 5.5)"; Ed. McGraw-Hill; España; 1990.
- Keller, Robert; "Expert System Technology Development and Application"; Prentice-Hall; The United States of America; 1987.
- Magaña, Roberto; "Análisis Tridimensional Dinámico de Presas"; Revista Latinoamericana de Geotecnia, No 2, Vol. III; Caracas, Venezuela; 1976
- Magaña, R.; "Fórmula Semiempírica para Estimar la Frecuencia Fundamental de Presas de Tierra y Enrocamiento"; Memorias del VI Congreso Nacional de Ingeniería Sísmica, pag 23-26; México; 1983.
- Magaña, R.; Romo, M. P.; "Evaluación de la Respuesta Sísmica y Seguridad de las Presas El Infiernillo y La Villita"; Informe para CFE; Instituto de Ingeniería, Universidad Nacional Autónoma de México; México; 1992.
- Marsal, R. J.; "Mechanical Properties of Rockfills"; Embankment Dam Engineering, Casagrande Volume; J. Wiley, New York, The United States of America; 1967.
- Newmark, N. M.; "Effects of Earthquakes on Dams and Embankment"; Géotechnique, No. 15, pag 139-160; 1965.
- Payne, Edmundo C.; McArthur, Robert; "Developing Expert Systems"; Ed. John Wiley Sons, The United States of America; 1990.
- Reséndiz, D.; Rosenblueth E.; Mendoza, E.-; "Diseño Sísmico de Presas de Tierra y Enrocamiento, Estado del Arte"; Informe Interno No. 300; Instituto de Ingeniería, Universidad Nacional Autónoma de México; México; 1972.

- Reséndiz, D.; Romo, M. P.; "Presas de Tierra y Enrocamiento"; Ed. Limusa; México; 1975.
- Romo, M. P.; "Soil-Structure Interaction in a Random Environment"; Dissertation Submitted in Partial Satisfaction on the Requirements for the degree of Doctor of Philosophy; University of California, Berkeley; California, The United States of America; 1976.
- Romo, M. P.; Chen J. H.; Lysmer J., Seed, H. B.; "PLUSH: A Computer Program for Probabilistic Finite Element Analysis of Seismic Soil-Structure Interaction"; Report No. EERC-77/01; University of California, Berkeley; California, The United States of America; 1977.
- Romo, M. P.; Aynala, G.; Reséndiz D.; Diaz, C.; "Respuesta Dinámica de las Presas El Infiernillo y La Villita"; Informe para CFE; Instituto de Ingeniería, Universidad Nacional Autónoma de México; México; 1980.
- Romo, M. P.; Villarraga, R.; "Modelo Teórico del Comportamiento Sísmico de Presas: El Infiernillo"; Informe Interno No. 518; Instituto de Ingeniería, Universidad Nacional Autónoma de México; México; 1989.
- Sánchez y Beltrán, J.P.; "Sistemas Expertos (Una metodología de programación)"; Ed. Microbit; México; 1990.
- Seed, H. B.; Martin, G. R.; "Soil Moduli and Damping Factor for Dynamic Response Analysis"; Report No. UCB/EERC-70/10, University of California, Berkeley; California, The United States of America; 1970.
- Seed, H. B.; Idriss, I. M.; "Influence of Soil Condition on Ground Motion During Earthquakes"; J.S.M.F.D.; ASCE, vol. 95, No. SM1; 1969.
- Villarraga, R.; "Respuesta Aleatoria Tridimensional de Presas de Tierra"; Tesis de Maestría, Universidad Nacional Autónoma de México; México; 1987.

ANEXO I

MANUAL DEL USUARIO DEL PROGRAMA PTLUSH

OPERACION DEL PROGRAMA

El programa puede ser operado de tres maneras diferentes (a las que se denomina modos).

- El modo uno** Se utiliza para un primer análisis de respuesta dinámica, en este modo se especifican los puntos nodales considerados en el problema, así como las características del movimiento inducido, el amortiguamiento del mismo y se especifica en que puntos nodales se requiere conocer la solución del problema (la historia de aceleraciones que produce en los puntos nodales como consecuencia de la excitación el la base).
- El modo dos** Se utiliza para realizar nuevas iteraciones de los resultados obtenidos en el modo uno, en este modo es posible variar los puntos nodales donde se requiere la salida.
- El modo tres** Se utiliza para purgar el archivo de resultados y seleccionar solo una parte de éstos, los que se almacenan en un archivo de menor dimensión y por tanto con mayor facilidad de manejo.

ARMADO DE LA BASE DE DATOS PARA TODOS LOS MODOS

0.- Control de operación (I5, I10)

- 1-5 NOPT Si es = a 0 el programa solo lee los datos
 1 se ejecuta el primer modo
 2 se ejecuta el segundo modo
 3 se ejecuta el tercer modo
 4 chequea los datos los datos de la corrida
- 6-15 NSCM determina el numero de decimales que se va considerar en la corrida.

Para el modo 1:

1. Identificadores del problema (12A6, I8)

- 1-72 TITLE Nombre identificador del problema (tetras)
73-80 INTER Nombre identificador de la iteración

ANEXO I Manual de Usuario del Programa Ptlush

2. Condiciones generales del análisis (815)

1-5 NELM	Número total de elementos tridimensionales en que se discretizó la presa
6-10 NDPT	Número total de puntos nodales
11-15 NB1	Primer punto nodal sobre la base rígida (el programa asume que todos lo demás puntos nodales por arriba de esta coordenadas también se encuentran por arriba de la frontera rígida
16-20 NBP	Es el número de desplazamientos condicionales
21-25 NUMBER	Número máximo de iteraciones para lograr una compatibilidad relativa entre las propiedades de los materiales propuestas y las calculadas por el propio programa (el programa realiza una iteración de autemano)
26-30 MATYP	Total de los materiales donde existe dependencia con los esfuerzos
31-35 KMATYP	Si es igual a cero entonces el programa lee la curva de comportamiento del material del archivo de datos, por el contrario si es igual a uno el programa utiliza una curva estándar definida previamente en el programa.
36-40 NFORM	Si su valor es igual a cero solo lee datos, si es igual a uno lo escribe en el archivo denominado "Tape5"

2.A Control de las unidades para el análisis (F10.0, A2, 2(A2), 5A5, 2(A6))

1-10 GRAV	Valor de la gravedad a utilizar
11-12 ULONG	Unidad de longitud a utilizar
12-20 UNCEL	Unidad de la aceleración a utilizar
36-40 UG	Unidad de la gravedad a utilizar
41-45 UPSP	Unidades del espectro de aceleración

3. Características generales de la excitación inducida (215, 4F0.4)

1-5 MSPEC	Cuando el espectro de entrada es un espectro de potencia el valor de MSPEC es igual a cero, si su valor es igual a uno entonces se trata de un espectro de respuesta de aceleraciones
6-10 NFRE	Es el número de frecuencias consideradas
11-20 TLEN	Tiempo de duración de las frecuencias consideradas
21-30 CONF	Confianza (en decimales)
31-40 FACTS	Factor de escalamiento del movimiento inducido, (el espacio se deja en blanco si se desea que el control del programa lo tenga la variable AGMAX)
41-45 AGMAX	Aceleración máxima considerada (si deja en blanco el control del programa esta en FACTS)
46-50 NCUT	Número de frecuencias para las cuales se requiere se resuelva el sistema.

4. Características particulares de la excitación inducida (3F10.4)

Es posible modificar el valor de la aceleración en cada una de las direcciones, con el fin de atenuar o amplificar su valor.

1-10 HORX	Factor de corrección en el eje X
11-20 VERT	Factor de corrección en el eje Y
21-30 HORZ	Factor de corrección en el eje Z

ANEXO I Manual de Usuario del Programa Ptlush

5. Datos de control para las frecuencias de análisis (2F10.3)

1-10 FIRFR	Primer frecuencia considerada en el análisis (Hz)
11-20 TOTFR	Máxima frecuencia utiliza en el análisis, solo sirve para configurar la memoria de la computadora, las frecuencias utilizadas se declaran conforme el punto 9

6 Datos de control para la salida del programa (7I5)

1-5 KDISP	Si se le da el valor de uno gravan los datos leídos en el archivo denominado "Tape1"
6-10 KSTRN	Si se le asigna el valor de cero entonces el programa asume que los esfuerzos son independientes de las propiedades
11-15 KPNCH	Si su valor es uno el programa recalcula el valor de las propiedades y las utiliza en la siguiente iteración
16-20 NOUT	Total de puntos nodales donde se desea conocer resultados
21-25 ND	Número de amortiguamiento evaluados para el espectro de respuesta

7. Datos que controlan la matriz de masa y el esfuerzo efectivo (2F10.4)

1-10 RATIA	Tolerancia para de consistencia de la matriz de masa, para la deformación de misma.
11-20 FCT	RadioTolerancia esfuerzo efectivo máximo, usado para determinar el esfuerzo compatible con las propiedades del suelo.

8. Datos que controlan las características del amortiguamiento (F10.4)

1-10 SDAMP	Si el amortiguamiento utilizado actúa uniformemente en todos los elementos, el espacio se deja en blanco
------------	--

9. Datos de control para la frecuencia calculada en la iteración (8F10.4)

1-10 STEP(1)	Máxima frecuencia para el análisis durante la primera iteración.
21-30 STEP(2)	
...	
STEP(N)	Máxima frecuencia para el análisis durante la última iteración; donde N es el número de iteración (NUMBRE) según el punto 2 de este mismo modo.

10. Amortiguamiento considerado (8F10.4)

1-10 DAMP(1)	Valor de la primera evaluación del amortiguamiento en el espectro de respuesta
1-11 DAMP(2)	
...	
DAMP(N)	Valor de la última evaluación del amortiguamiento en el espectro de respuesta; donde N es en número de evaluaciones (AND) según el punto 6 de este mismo modo.

ANEXO I Manual de Usuario del Programa Plush

11. Datos que controlan la salida de información (415)

1-5 M	Total de punto donde se requiere conocer resultados
6-10 KEYSPEC(3M-2)	Control de salida en la dirección X
11-15 KEYSPEC(3M-1)	Control de salida en la dirección Y
16-20 KEYSPEC(3M)	Control de salida en la dirección Z

12. Declaración de los elementos finitos (914, 312, F4.3, 3F10.4, F4.3)

1-4 N	Número de elementos finitos considerados en el análisis
5-8 NP1(I)	Punto nodal 1 del elemento finito, I, (el elemento finito tridimensional se construye con 8 puntos nodales que equivales al número de aristas del mismo)
9-12 NP2(I)	Punto nodal 2 del elemento finito
13-16 NP3(I)	Punto nodal 3 del elemento finito
17-20 NP4(I)	Punto nodal 4 del elemento finito
21-24 NP5(I)	Punto nodal 5 del elemento finito
25-28 NP6(I)	Punto nodal 6 del elemento finito
29-32 NP7(I)	Punto nodal 7 del elemento finito
33-36 NP8(I)	Punto nodal 8 del elemento finito
37-38 MTYPE(N)	Número que identifica el tipo de material del que esta constituido el elemento N. Puede tomar valor de cero para materiales donde el esfuerzo es independiente de la propiedades, uno para las arcillas y dos para las arenas
39-40 LX	Incremento del esfuerzo
41-42 IDEL(N)	Número identificador del elemento finito
43-46 PO(N)	Relación de Poisson para el elemento N
47-56 DENS(N)	Densidad del elemento N, en unidades de peso
57-66 S3(N)	Máximo módulo de corte para el elemento N
67-76 G(N)	Valor inicial del módulo de corte
77-80 XL(N)	Valor inicial del amortiguamiento.

La declaración de los datos anteriores se debe hacer para todos los elementos finitos considerados.

13. Modificación de la curva de material, solo se aplica para los elementos constituidos de material tipo 2, para los demás no se declaran los siguientes datos:

13.A Identificación de los materiales (15, 12A6)

1-5 (N)	Tipo de material
6-77 TTL	Nombre identificador del material, (con letras)

14. Datos para cada punto nodal considerado en el análisis (15, 3F10.4, 15, F10.4, 15)

1-5 M	Total de puntos nodales considerados
6-15 XORD(I)	Coordenada en la dirección X del punto nodal I
16-25 YORD(I)	Coordenada en la dirección Y del punto nodal I
26-35 ZORD(I)	Coordenada en la dirección Z del punto nodal I
36-40 KEYBC(I)	Condición de desplazamientos del punto nodal. La variable puede tomar los siguientes valores: 0 si existe desplazamiento en todas las direcciones 1 si no hay desplazamiento en la dirección X 2 si no hay desplazamiento en la dirección Y

ANEXO I Manual de Usuario del Programa PlusH

3 si no hay desplazamiento en la dirección Z
4 si no hay desplazamiento en la dirección X o Y
5 si no hay desplazamiento en la dirección Y o Z
6 si no hay desplazamiento en la dirección X o Z
Peso en el punto nodal 1
Incremento

41-50 SMAS(1)
51-55 NG

15. Control del tipo de excitación (12A6, F10.0)

1-72 EQN(12)	Nombre identificador del tipo de excitación (con letras)
PST(1)	Valor de la aceleración máxima en unidades G
SPT(1)	Entrada del espectro de respuesta (en unidades G), donde l es el número de puntos considerados en el espectro
FRE(1)	Entrada de frecuencias
KSOL(1)	Control de frecuencias para las cuales el sistema se resuelve

Para el modo 2:

1. Controladores de datos (15, I10)

1-5 KDISP	Ver el punto 6 del modo 1
5-10 KSTRN	Ver el punto 6 del modo 1
11-15 KPNCB	Ver el punto 6 del modo 1
16-20 NCU7	Ver el punto 6 del modo 1
21-25 ND	Ver el punto 6 del modo 1
26-30 NUMBER	Ver el punto 2 del modo 1
31-35 NFORM	Ver el punto 2 del modo 1

2. Frecuencia de análisis (2F10.4)

1-19 FIRFRN	Nuevo valor de TOTFR (ver el punto 5 del modo 1)
1-19 TOTFRN	Nuevo valor de KINTS (ver el punto 5 del modo 1)

3. Nuevos datos para las frecuencias donde se desea realizar el análisis (8F10.4)

1-10 STEP(1)	Ver el punto 5 del modo 1
--------------	---------------------------

4. Amortiguamiento (8F10.4)

21-25 ND	Ver el punto 6 del modo 1
----------	---------------------------

5. Nuevo control de salida (415)

Ver el punto 12 del modo 1

Para el modo 3:

1. Control de datos (715)

1-5 KPNCB	Ver el punto 6 del modo 1
6-10 NOUT	Ver el punto 6 del modo 1
11-15 ND	Ver el punto 6 del modo 1
16-20 KSTRN	Ver el punto 6 del modo 1

2. Control de amortiguamiento (8F10.4)

Ver el punto 10 del modo 1

3. Nuevo control de salida (415)

Ver el punto 11 del modo 1

ANEXO II

MANUAL DE USUARIO DEL PROGRAMA ANSTA

El programa ANSTA es el resultado de las modificaciones hechas al programa SLOAF, realiza el análisis de estabilidad utilizando el método de BISHOP modificado, para una superficie potencial de falla propuesta por el usuario, es importante recordar que el ángulo de fricción varía con respecto a la profundidad, por el efecto del esfuerzo de confinamiento para modelar esta variación el programa cuenta con la posibilidad de realizar una variación automática de dicho ángulo dentro de límites propuestos por el mismo usuario.

Operación del programa

El programa utiliza esencialmente cuatro archivos de datos: a) Archivo controlador, b) Archivo DASTAI, c) Archivo de datos y d) Archivo RESPALDO.

ARCHIVO CONTROLADOR

Este archivo es el que controla la ejecución del programa, contiene el número de análisis que se desea realizar, el límite inferior y superior del ángulo de fricción, para su edición se procede como sigue:

Puede ser nombrado de cualquier manera (se recomienda utilizar la extensión CON que significa controlador con el fin de diferenciarlo de los otros archivos).

En el primer renglón del archivo se almacena el número de análisis que se desea realizar, dejando un espacio se escribe el límite inferior que se desea tome el ángulo de fricción y partir del cual comenzará la variación, dejando otro espacio se escribe el valor del límite superior hasta el cual se desea varíe el ángulo de fricción.

En el segundo renglón y empezando en la primera columna se escribe el nombre del archivo donde se leen los datos y a partir de la columna 16 de ese mismo renglón se escribe el nombre del archivo donde se desea guardar los resultados de ese análisis, los demás nombres de los archivos de datos y resultados se almacenan de igual manera en los siguientes renglones hasta llegar al número de análisis que especificados en el primer renglón.

Ejemplo:

2,25,60	
ARCHIVO1.DAT	ARCHIVO1.RES
ARCHIVO2.DAT	ARCHIVO2.RES

ANEXO II Manual de Usuario del Programa Ansta

ARCHIVO DASTAI

El archivo DATSTAI debe estar declarado antes de cualquier corrida, para su edición se siguen los siguientes pasos:

En el primer renglón y primera columna se escribe el número 1, que sirve para dar continuidad al programa después, dejando un espacio se escribe el nombre de la presa o cualquier otro identificador del problema (con un máximo de 15 caracteres), en el siguiente renglón se escribe el número cero y dejando un espacio se repite el identificador que se utilizó en el renglón anterior. El archivo DATSTAI sólo se compone de estos dos renglones.

Ejemplo:

```
1 PROBLEMA PRESA INFIERNILLO
0 PROBLEMA PRESA VILLITA
```

ARCHIVO DE DATOS

Este archivo es el más importante, puesto que contiene la información necesaria para evaluar la estabilidad del talud.

1. Datos controladores del programa (todos los datos se almacenan separados por una coma o dejando un espacio en blanco)

En el primer renglón se almacenan AMOD, NINAX, TOL, NCIRC (F5.4, I3, F2.5, I3) donde:

AMOD	es la aceleración promedio de la superficie de falla, en unidades G
NINAX	es el número de iteraciones máximas (para el análisis de estabilidad se puede asignar cualquier entero positivo por ejemplo 10)
TOL	es la tolerancia para el error absoluto (se emplea cuando se realiza un análisis de aceleración de fluencia, para el análisis de estabilidad se supone un valor de 0.5)
NCIRC	es el número de círculos considerados en el análisis (para análisis de estabilidad su valor es 1, puesto que sólo se analiza una superficie de falla por cada archivo).

En el segundo renglón se especifica NSPEC, NLEVEL, NSECT, NSTRAT, LIGNE, NPTPU, A1, A2, GAM (6I3, 2F5.4, F3.3) donde:

NSPEC	es el número de círculos especificados además del inicial (se le asigna el valor de cero para el caso de análisis de estabilidad)
NLEVEL	es el número de puntos tangentes para cada una de los círculos de NSPEC (para el análisis de estabilidad se le asigna el valor de cero)
NSECT	es el número de líneas de seccionamiento vertical
NSTRAT	es el número líneas de frontera entre materiales utilizadas
LIGNE	es el número de líneas de presión de poro especificadas (se le asigna el valor de cero)
NPTCU	es el número de puntos que definen la variación de la presión de poro (se le da el valor de cero para el análisis de estabilidad)
A1	es el valor de la aceleración promedio en la dirección horizontal (el valor de AMOD)
A2	es el valor de la aceleración promedio en la dirección vertical (se le da el valor de cero puesto que no se considera aceleración en esta dirección)
GAM	Es el peso volumétrico del agua (64.2 libras por pulgada cúbica)

ANEXO II Manual de Usuario del Programa Ansta

2. Coeficientes sísmicos asociados a los centroides de los elementos (formato libre) (los datos se almacenan separados de una coma o dejando un espacio en blanco).

La Matriz de aceleraciones se construye considerando renglones a los estratos y columnas a las dovela, como el programa puede simular aceleraciones en las dos direcciones, el número de columnas se multiplica por dos de manera que para una misma dovela existan dos renglones en el primero se almacenan los coeficientes en la dirección horizontal y en el segundo los coeficientes en la dirección vertical (cuando el análisis de estabilidad no considera la componente vertical del sismo ha este renglón se le asignan ceros)

3.- Datos que definen la superficie circular de falla

Las variables utilizadas son XI, YI, DC, XTOE y YTOE (datos en Ft (pies) y en formato libre) donde:

XI	es la coordenada X del centro del círculo
YI	es la coordenada Y del centro del círculo
DC	es la variable que controla la variación de la ubicación del centro (para los análisis de estabilidad se le asigna el valor de cero)
XTOE	es la coordenada X del punto obligado por el que se desea pase el círculo
YTOE	es la coordenada Y del punto obligado

4.- Datos que definen la geometría de la sección de análisis

Para cada una de las líneas de seccionamiento vertical se edita un renglón en el que se almacenan los datos X(L), YFI(L), YSURF(L), Y(L,J) (datos en Ft (pies) y en formato libre), donde:

X(L)	es el valor de la intersección de la línea de seccionamiento vertical, L, con el eje de las abscisas
YFI(L)	es el valor de la coordenada Y hasta donde llega la profundidad de la grieta de tensión
YSURF(L)	es el valor de la coordenada Y donde se intersecta la línea de seccionamiento vertical, L, con la línea superior del flujo
Y(L,J)	es el valor de la intersección de la línea de seccionamiento, L, con las líneas de frontera entre estratos, J; (donde J=1 hasta J=NSTRAT), es decir el número de datos de tipo Y(L,J) se repite tantas veces como líneas de frontera se consideren en el análisis

NOTA: En los análisis realizados no se consideraron grietas de tensión y la línea superior de flujo se consideró igual a la primer línea de frontera: $YFI(L) = YSURF(L) = Y(L,J)$; cuando $J=1$
Se debe tener especial cuidado cuando se construye esta parte del archivo porque el programa ANSTA considera la dirección positiva del eje Y hacia abajo, mientras que el programa PTLUSH lo considera hacia arriba, cuando se edita esta parte del archivo se tiene que multiplicar por -1 todas las coordenadas Y, se debe proceder de igual forma para las coordenadas del círculo de falla y su punto obligado.

5. Datos que definen las propiedades y características de los estratos considerados

Las variables utilizadas son J, JES, CU(J), FRICT(J), WTOTAL(J), FRI0(J), VARA(J), CONFO(J) (213, 8F10.4) (las variables se almacenan separadas de una coma o dejando un espacio en blanco entre una y otra, para los datos de cada estrato se utiliza un

ANEXO II Manual de Usuario del Programa Ansta

renglón), donde:

J	es número identificador del estrato (los estratos se enumera del mas superficial hacia abajo)
JES	es la variable que permite la variación σ no variación del ángulo de fricción (cero para no variarlo y 1 para variarlo)
CU(J)	es la cohesión del material J
FRICT(J)	ángulo de fricción
WTOTAL(J)	es el peso volumetrico total en toneladas por metro cuadrado
FRI0(J)	es el valor de el ángulo de fricción inicial para el confinamiento, según la ec. 1
VARA(J)	es la variación de presión de confinamiento, según ec. 1
CONF0(J)	valor de la presión atmosférica
EK(J)	el valor de K para el estrato J, según la ec. 2
EN(J)	el valor de N para el estrato J, según la ec. 2

$$\hat{\phi} = \phi_0 - \Delta\phi \cdot 10g \frac{\sigma_1}{P_a} \quad (1)$$

$$E = K \left\{ \frac{\sigma}{P_a} \right\}^n \quad (2)$$

donde:

P_a	es la presión de poro
Φ_0	es el valor de Θ para la presión de poro P_a
σ_1	es el esfuerzo de confinamiento en la parte media de la cortina y el incremento de Φ es un parámetro que se define, en base a pruebas, para cada material.

ARCHIVO RESPALDO

El archivo RESPALDO debe ser declarado antes de ejecutar el programa, en el archivo se guardan todos los cálculos hechos por el programa y se utiliza como memoria de cálculo, lo que facilita encontrar los errores cuando el programa falla. Cada vez que el programa es ejecutado, los datos contenidos en el archivo RESPALDO se pierden, para declarar este archivo solo es necesario cambiar el nombre de algún archivo que no sirva por el nombre RESPALDO.

ANEXO III

MANUAL DE PROGRAMACION EN TURBO PASCAL (versión 5.5)

El manual se diseñó para un lector con conocimientos previos de programación (ya sea en Fortran 77 o Basic) y que desee introducirse en el manejo de Turbo Pascal. Los comandos a los que se hace referencia pertenecen a la versión 5.5 y son totalmente compatibles con la versión 6 del mismo lenguaje*1.

El manual se compone de los siguientes apartados:

- Descripción y explicación de los componentes de la estructura de programación utilizada en Turbo Pascal.
- Descripción de las estructuras de tipo selectivas e iterativas que contiene el lenguaje.
- Uso de los comandos básicos para controlar la pantalla y el teclado.
- Se reseñan además algunas estrategias básicas para el tratamiento de archivos
- Por último se anexan dos aplicaciones prácticas del lenguaje como ejemplo de los alcances del mismo.

Para una mejor explicación de las estructuras y los comandos, el anexo se complementa con una serie de programas denominados DEMOS por medio de los cuales es posible apreciar una aplicación práctica del lenguaje. Se propone al lector utilice este manual junto con una PC y el lenguaje Turbo Pascal cargado, con el propósito de poder visualizar lo que se va explicando, ya que los DEMOS se concibieron como una explicación concreta y manipulable para que el usuario pueda modificarlos y observar los cambios que ocurren en el mismo. Por otra parte, es importante remarcar que los programas DEMO posteriores incorporan conocimientos del DEMO anterior, y la explicación de los mismos sólo se limita a comentar los conocimientos nuevos. Por otra parte los comandos de mayor uso se describen en una sección del glosario, (anexo VIII).

En este anexo sólo se explican las estructuras básicas las cuales son suficientes para iniciarse en la programación en lenguaje Turbo Pascal y desarrollar programas tan amplios o variados como la propia creatividad del programador, claro que se hará necesario recurrir a textos más profundos esto en función de las necesidades de cada usuario.

Ambiente de programación

Turbo Pascal puede utilizarse de dos maneras diferentes:

- a) utilizando el entorno de programación (método usual en una PC) y

* Las palabras marcadas con * pueden ser consultadas en el Glosario, Anexo VIII.

b) utilizando sólo el compilador (método usual para estaciones de trabajo).

Turbo Pascal, cuenta con su propio ambiente de programación* lo que hace más fácil su aprendizaje, el editor incluido minimiza las tareas normales en cuanto a la edición y compilación de programas.

Si Turbo Pascal se usa junto con su ambiente de programación es posible que el programa en edición pueda compilarse en memoria activa*, pudiendo detectar los errores rápidamente y no esperar a compilar por separado como es el caso de Fortran 77.

Para iniciar Turbo Pascal, desde el MS-DOS* se teclea la palabra TURBO y a continuación se oprime la tecla ENTER. Si la versión 5.5 esta completa y sin modificaciones la primer pantalla que se despliega contiene información referente a la versión del lenguaje, oprimiendo nuevamente ENTER, se inicia el editor; si por el contrario el paquete ha sido modificado no despliega ninguna pantalla previa y automáticamente se inicia el editor.

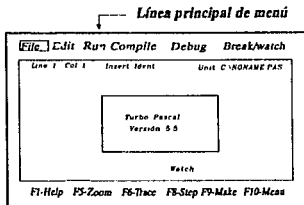


Fig. 1 Pantalla de inicialización de Turbo Pascal ver. 5.5

Desde el entorno de programación* es posible cargar, guardar, editar, compilar y ejecutar un programa fuente; para llevar a cabo las acciones antes señaladas se utiliza la línea superior de menú ubicada en la parte superior de la pantalla.

LINEA PRINCIPAL DE MENÚ

(Para entrar y salir de la línea de menú se oprime la tecla F-10)

Para elegir un menú particular se oprime la primer letra que lo identifica o se emplean las flechas derecha e izquierda para colocarse sobre la palabra de interés, a continuación se oprime ENTER para desplegar la ventana de ordenes asociadas, para salir de la dicha ventana se utiliza la tecla Esc.

Las ordenes que se usan con mayor frecuencia están asociadas a un camino de atajo (combinación de teclas), el que permite invocar y ejecutar directamente una orden determinada sin la necesidad de entrar a la línea superior de menú por ejemplo la tecla F2 sirve para archivar el texto en pantalla; cuando una orden tiene asociado un atajo puede conocerse en el momento de desplegar las ordenes vía ventanas (generalmente los caminos de atajo aparecen al lado derecho de la orden y entre paréntesis).

A continuación se describen las ordenes de mayor uso, si estas tienen un camino de atajo asociado éste

ANEXO III Manual de Programación en Turbo Pascal

se muestra entre paréntesis:

FILE (ALT-F):

Permite cargar, guardar y crear archivos, así como elegir, cambiar y visualizar directorios, de este menú es posible salir ya sea temporalmente o definitivamente al sistema operativo.

Las ordenes mas frecuentes son:

Load (F3)	Carga un archivo desde disco
Pick(Alt-F3)	Permite cargar un archivo entre los 8 últimos editados
New	Limpia la memoria de edición para iniciar otro programa
Save(F2)	Guarda el archivo en edición con el nombre con el que se cargó
Write to	Guarda el archivo en edición con un nuevo nombre definido por el usuario
Directory	Muestra los archivos contenidos en el directorio activo
Change Dir	Cambia el directorio activo
Os Shell	Suspende momentáneamente Turbo Pascal y regresa a DOS, para restablecer el ambiente se teclea EXIT
Quit (Alt-X)	Suspende definitivamente Turbo Pascal, regresando a DOS en el directorio activo.

EDIT (ALT-E):

Activa el editor del entorno de programación, los comandos válidos en esta opción sólo pueden ser invocados utilizando una combinación de teclas, a continuación los mas útiles:

Ctrl-X o ↓	Desplazamiento hacia abajo
Ctrl-S o ←	Desplazamiento hacia la izquierda
Ctrl-D o →	Desplazamiento hacia la derecha
Ctrl-E o ↑	Desplazamiento hacia arriba
Ctrl ←	Desplazamiento hacia el inicio de la línea
Ctrl →	Desplazamiento hacia el final de la línea
PgDn	Desplazamiento una página hacia abajo
PgUp	Desplazamiento una página hacia arriba
Home	Desplazamiento al principio de la página
End	Desplazamiento al final de la página
Ctrl-Home	Desplazamiento al principio del programa
Ctrl-End	Desplazamiento al final del programa
Ctrl-Y	Eliminación de la línea donde se encuentra el cursor
Ctrl-KB o F7	Marcar principio de bloque
Ctrl-KK o F8	Marcar fin de bloque
Ctrl-KC	Copiar un bloque
Ctrl-KY	Borrar un bloque
Ctrl-KP	Imprimir (si el bloque esta activo, sólo se imprime éste)

RUN (ALT-R)

Controla los modos de ejecución del programa:

Run (Ctrl-F9)	Ejecuta el programa cargado en el editor, de ser necesario Turbo Pascal compila automáticamente el programa
Program Reset(Ctrl-F2)	Reinicia la compilación
Go To Cursor (F4)	Ejecuta el programa deteniéndose en la línea donde se encuentre el cursor
Trace Into (F7)	Ejecuta todo el cuerpo del programa línea por línea
Step Over (F8)	Ejecuta línea por línea sólo la parte principal.
User Screen (Alt-F5)	Muestra la pantalla de salida del programa.

MENU COMPILE (ALT-C)	Controla el tipo de compilación:
Compile (Alt-F9)	Compila el programa en edición y muestra los resultados en una ventana especial
Make (F9)	Verifica el programa en edición y las unidades utilizadas, si existe algún cambio desde la última compilación Turbo Pascal procede a realizar una recompilación
Destination	Opción para modificar los principales parámetros de compilación, el mas importante es <i>destination</i> con el cual es posible cambiar el direccionamiento de la compilación pasando de compilación en memoria activa a compilación en secundaria (disco) y viceversa.

Los demás menús requieren de una previa familiarización con el entorno de programación, para determinar sus aplicaciones se recomienda recurrir a la ayuda en línea que ofrece Turbo Pascal.

AYUDA EN LINEA

Es posible obtener ayuda en línea siempre y cuando el lenguaje este completo (es decir se hayan cargado todos los archivos), existen cuatro maneras de acceder a la ayuda en línea:

- a) Para desplegar ayuda referente al uso del editor de Turbo Pascal se oprime la tecla F1.
- b) Para desplegar ayuda referente a procedimientos y funciones predefinidas así como palabras clave se oprime dos veces la tecla F1 o el camino de atajo Ctrl-F1.
- c) Para desplegar información específica sobre una palabra clave primero se lleva al cursor cerca de la palabra que se desea investigar y entonces se oprime Ctrl-F1.
- d) Para restablecer la última ventana de ayuda se oprime Alt-F1

Estructura de un programa en Turbo Pascal

Un programa fuente hecho en lenguaje Turbo Pascal se compone de cinco secciones de declaración y la cláusula USES (mostradas en la figura 2), estas secciones no necesariamente deben aparecer en todos los programas, ni necesariamente en el orden mostrado, Turbo Pascal ver. 5.5 es flexible al respecto.

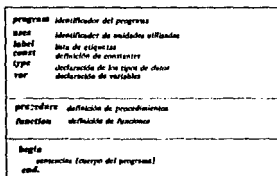


Fig. 2 Estructura de un programa en Turbo Pascal ver. 5.5

La explicación de la estructura se apoya en el programa DEMO1 que se encuentra en la siguiente página. En lo siguiente se describen brevemente los conceptos mencionados:

- PROGRAM** Utilizado para identificar al programa. Para el caso del programa DEMO1 se utilizó dem01.
- USES** En esta sección se declaran las unidades (librerías*) predefinidas por el propio lenguaje o definidas por el usuario. Cuando se utilizan unidades es posible invocar rutinas predefinidas que facilitan el manejo de pantallas, archivos y gráficos, entre otras posibilidades de Turbo Pascal, en DEMO1 sólo se utiliza la Unidad identificada por Ctrl, (con lo que se logra acceder a comandos para limpiar la pantalla y desplazar el cursor).
- LABEL** En esta sección se definen las etiquetas (campos de liga entre sentencias, parecidas a las utilizadas en lenguajes Fortran y Basic), en lo general no se utiliza esta sección ya que Turbo Pascal cuenta con otras opciones para redirigir la ejecución de un programa.
- CONST** Sección para asignar valores a una literal o grupo de ellas (su uso es similar las memorias de una calculadora de bolsillo), en DEMO1 se definió constante matemática π , con cuatro decimales.
- TYPE** Sección en la cual el usuario puede definir conjuntos de datos. Por ejemplo un conjunto predefinido en Turbo Pascal es el de los enteros el usuario puede definir el conjunto de los enteros positivos o el de los enteros del uno al diez. En DEMO1 no es necesario su uso, la utilidad de esta sección se explicará en ejemplos posteriores.
- VAR** En esta sección se declaran el tipo de variables a utilizar, estas pueden tomar valores dentro de un conjunto dado por ejemplo las pueden pertenecer a los conjuntos de tipo: REAL (real), INTEGER (entero), CHAR (carácter), STRIN (cadena Alfanumérica) o definidos por el propio usuario, en DEMO1 se utilizan dos variables de tipo real: A, R.

PROCEDURE En esta sección se definen los procedimientos, parecidos a las subrutina utilizada por Fortran, después de ser definidas pueden invocarse desde cualquier parte del programa. En DEMO1 no se utiliza esta sección (el propio programa DEMO1 constituye un procedimiento como se verá mas adelante).

FUNCTION Son parecidos a los procedimientos y trabajan a manera de función (se les alimenta con una serie de valores, realiza una evaluación y regresa un resultado).

Declaración de sentencias

Una sentencia* describe acciones algorítmicas*, para iniciar la declaración de sentencias se antepone la palabra **BEGIN**, para marcar donde acaba una y comienza otra se utiliza el carácter ";" y para marcar el final de la declaración de un grupo de sentencias se utiliza la palabra **END**

Declaración de variables:

Para declarar una variable o grupo de ellas se utiliza la sección **VAR** como sigue:

VAR

```
variable1 : tipo1;  
variable2 : tipo2;  
.  
.  
variableN : tipoN;  
variablen1, variableN2:TipoN2;
```

Donde *variable1* en el identificador o nombre de la variable y *tipo1* es el tipo de conjunto al que pertenece (ejemplo: CHAR, STRING, INTEGER), para declarar un grupo de variables del mismo tipo se utiliza una coma "," como en *variablen1, variableN2 : tipoN1*, en DEMO1 se utilizan dos variables de tipo real.

Operaciones de entrada, salida y matemáticas:

La operación de entrada se realiza hasta después de oprimir **ENTER** y ésta puede ser de dos tipos:

Read	El cursor se permanece inmediatamente después del último carácter introducido
ReadLn	El cursor pasa al principio de la siguiente línea.

La operación de salida también es de dos tipos:

Write	Utilizada para que el cursor permanezca después del último carácter desplegado.
WriteLn	El cursor pasa al principio de la siguiente línea.

Movimiento del cursor:

GotoXY(A,B)	El cursores de desplaza A columnas y B renglones, donde la esquina superior izquierda es la posición 1,1. En monitores GVA y CGA el número de columnas es 80 y de renglones es 25. Para utilizar el comando se requiere este declarada unidad Crt.
--------------------	--

Los operadores matemáticos son similares a los utilizados en otros lenguajes y al igual que en ellos se respeta la jerarquía entre operadores.

EJEMPLO PROGRAMA DEMO1

Una descripción práctica de la estructura del programación, declaración de variables y utilización de los comandos de entrada y salida se muestra en DEMO1, el cual obtiene el área de un círculo a partir de un radio definido por el usuario:

<i>Program demo1;</i>	<i>(declaración del nombre del programa)</i>
<i>uses</i>	
<i>cr; crt;</i>	<i>(declaración de las unidades a utilizar)</i>
<i>Const</i>	
<i>PI = 3.1416;</i>	<i>(declaración de las constantes)</i>
<i>Var</i>	
<i>A, R: real;</i>	<i>(Definición de Variables)</i>
<i>Begin</i>	<i>(Marca de inicio de Programa)</i>
<i>Clrscr;</i>	<i>(Limpia la Pantalla)</i>
<i>GotoXY(10,5);</i>	<i>(Desplaza el Cursor, donde X es columna y Y es renglón)</i>
<i>WriteLn('Programa para calcular el Area de un Círculo');</i>	<i>(Muestra texto)</i>
<i>GotoXY(14,8);</i>	
<i>Write('Introduzca el Valor del Radio : ');</i>	
<i>ReadLn(R);</i>	<i>(Lee Datos de Pantalla)</i>
<i>A := PI * R * R;</i>	
<i>GotoXY(10,14);</i>	
<i>WriteLn('El Area del Círculo con Radio = ',R:4:3,' es : ',A:4:3);</i>	<i>(:4:3 se utiliza para declarar el formato de salida, en este caso 4 enteros y dos decimales)</i>
<i>GotoXY(10,22);</i>	
<i>WriteLn(' Para terminar presione <ENTER> ');</i>	
<i>ReadLn;</i>	<i>(orden para detener el programa)</i>
<i>end.</i>	<i>(Marca de fin de programa)</i>

Como puede observarse en DEMO1 el uso de mayúsculas y minúsculas es indiscriminado, esto es posible en Turbo Pascal, el tipo de letra se emplea según se desee.

El principio del programa se marca con la palabra *begin*, el final de cada sentencia se marca con el carácter ";" y el final del programa se marca con la palabra *end.*, cuando la sentencia es la última puede omitirse la marca ";".

Para escribir comentarios se utilizan las llaves: (comentario) y para lograr detener el flujo del programa se utiliza el comando *ReadLn*.

Cuando se utilizan operadores matemáticos para asignar un valor a una variable se utilizan los caracteres ":=".

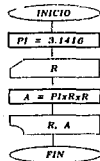


Fig. 3 Diagrama de flujo de DEMO1

Estructuras Selectivas

Las expresiones de tipo selectivo permiten orientar el flujo* del programa a partir de realizar una evaluación lógica (expresión que puede ser verdadera o falsa), su incorporación en programas permite reorientar el flujo del mismo en el momento de la ejecución. En Turbo Pascal existen dos tipos de sentencias de este tipo: **IF** y **CASE**.

Sentencia IF Su estructura es parecida a la utilizada en otros lenguajes (Fortran y Basic) y se compone por **if-then-else**.

EJEMPLO PROGRAMA DEMO2

El programa DEMO2 incorpora el uso de procedimientos y de sentencias de tipo IF, la explicación del mismo se encuentra la final del listado:

```

Program demo2;
uses
  crt;
VAR
  OPCION:CHAR;
PROCEDURE CIRCULO;
  Const
    Pi = 3.1416;
  Var
    A,R:real;
  Begin
    Clrscr;
    GotoXY(10,5);
    WriteLn('Programa para calcular el Area de un Circulo');
    GotoXY(14,8);
    Write ('Introduzca el Valor del Radio : ');
    ReadLn(R);
    A:=Pi*R*R;
    GotoXY(10,10);
    Write(' El Area del Circulo con Radio = ',R:4:3,' es : ',A:4:3);
  End;
PROCEDURE CUADRADO;
  Var
    A,L:real;
  Begin
    Clrscr;
    GotoXY(10,5);
    WriteLn('Programa para calcular el Area de un Cuadrado');
    GotoXY(14,8);
    Write ('Introduzca el Valor de cualquier lado : ');
    ReadLn(L);
    A:=L*L;
    GotoXY(10,10);
    Write(' El Area del Cuadrado con lado = ',L:4:3,' es : ',A:4:3);
  End;
BEGIN

```

(Utilización de Estructuras selectivas y Procedimientos)

(DEFINICION DEL PROCEDIMIENTO CIRCULO)

(MARCA DE INICIO DE PROCEDIMIENTO)

(MARCA DE FIN DE PROCEDIMIENTO, en este caso el end lleva el carácter ; esto ocurre cuando no es el último)

(DEFINICION DEL PROCEDIMIENTO CUADRADO)

(MARCA DE FIN DE PROCEDIMIENTO)

(INICIO DEL PROGRAMA PRINCIPAL)


```

CLRSR;
GotoXy(15,4);
Write(' ELJA EL TIPO DE CALCULO :');
GotoXy(10,7);
Write('1) Calcular el área de un circunferencia. ');
GotoXy(10,9);
Write('2) Calcular el área de un cuadrado. ');
OPCION := READKEY;
    
```

(comando especial que permite leer una tecla sin presionar enter)

```

IF OPCION = '1' THEN CIRCULO;
IF OPCION = '2' THEN CUADRADO;
IF (OPCION <> '2') AND (OPCION <> '1') THEN
  BEGIN
    GOTOXY(15,11);
    WRITE('LA OPCION NO EXISTE');
  END;
    
```

(INICIO DE TAREA)

(FIN DE TAREA)

```

GOTOXY(10,14);
WRITE('PRESIONE ENTER PARA CONTINUAR');
READLN
END.
    
```

(FIN DEL PROGRAMA, END, significa fin del programa)

DEMO2 calcula el área de un círculo o un cuadrado dependiendo de la selección del usuario. Como se puede apreciarse en el diagrama de bloques, figura 4, el programa DEMO2 se compone de dos procedimientos y una estructura de control utilizando sentencias IF. Cada procedimiento se construye conforme la estructura general del Turbo Pascal, de lo que es posible afirmar que cada procedimiento constituye un programa elemental (como es el caso de DEMO1 que aquí aparece como un procedimiento). La variables declaradas dentro de un procedimiento se activan en el momento en que se invoca éste, en tanto que cuando se abandona la memoria utilizada es liberada.

Como se observa en el listado las sentencias If pueden ser declarada de manera parcial, es decir sólo la parte IF-THEN, también es posible ligar dos sentencias IF utilizando operadores (AND) y (OR) como en el tercer IF de DEMO2. Cuando la tarea a realizar después de IF se compone por más de una orden, es posible programarla utilizando un begin y un end; como marcas de principio y fin de la tarea.

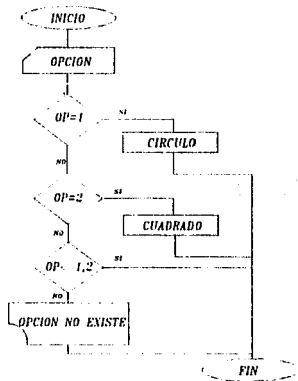


Fig. 4 Diagrama de bloques de DEMO2

Finalmente, en DEMO2 se incorpora el uso de nuevos comandos como es el caso de READKEY que permite leer una tecla sin la necesidad de oprimir ENTER, los demás comandos pueden ser consultados en el glosario, y notese que para marcar el fin de un procedimiento se utiliza la marca END; (el carácter ; es parte

de la marca).

Sentencia CASE

La sentencia CASE se utiliza para programar más de dos alternativas de flujo. Las alternativas pueden ser compuestas (tarear) o simples. Como en el caso de la sentencia IF, es opcional el uso de la palabra ELSE.

Las alternativas de flujo definidas ya sea por IF o CASE pueden contener a su vez otras sentencias de tipo IF o CASE internamente, con lo que se logra construir árboles de decisión casi ilimitados. A este tipo de estructura de programación se le conoce como Anidamiento de estructuras de selección.

Un ejemplo de las sentencias IF-THEN-ELSE es:

```
IF A = "A"  
  THEN WRITELN(' USTED PULSO LA TECLA A');  
  ELSE WRITELN(' USTED PULSO UNA TECLA DIFERENTE DE A');
```

En el programa DEMO3 se incorpora el uso de la sentencia CASE (el objetivo de su uso se explicará en su momento).

Estructuras Repetitivas

Estas sentencias permiten repetir una orden o tarea hasta que se cumpla una condición dada por el usuario o generada a partir de los resultados parciales que va arrojando el programa cuando éste se ejecuta. Las sentencias de este tipo son: FOR, WHILE, REPEAT.

Sentencia FOR

La sentencia FOR es el ciclo iterativo más elemental su uso requiere se conozca o determine de antemano el número de iteraciones a realizar.

Su estructura es:

	For c:=1 to N Do
donde: For	es la palabra clave que define al contador
c	es la variable entera a incrementar (en este caso desde 1)
N	es el valor último de c
do	es la palabra clave que define el ciclo.

Al igual que en las sentencias selectivas es posible definir un ciclo iterativo compuesto (tarea) utilizando las marcas **begin** y **end**;

Sentencia WHILE

La estructura repetitiva WHILE (mientras) se utiliza frecuentemente cuando el número de iteraciones a realizar no se conoce de antemano, un ciclo definido de esta manera se le denomina condicional.

Cuando el camino de ejecución* del programa encuentra una sentencia WHILE se sucede una evaluación lógica si ésta resulta verdadera se ejecuta el interior de la sentencia, la evaluación se repite hasta que el resultado de la evaluación lógica sea falso. Su utilidad queda expresada en el Programa DEMO3.

Sentencia REPEAT

La sentencia **repeat** es una variante de la sentencia **WHILE** a diferencia de ésta, donde la evaluación lógica se hace al principio de cada iteración, en la sentencia **REPEAT** la evaluación se realiza al final.

La sentencia **REPEAT** se compone de dos palabras clave: **REPEAT** y **UNTIL**, éstas sirven para marcar el comienzo y el fin del ciclo respectivamente. El uso de la sentencia puede ser apreciado en el programa **DEMO3**.

Tratamiento de archivos

Las estructuras descritas hasta el momento trabajan con memoria central o activa, para conservar y manipular información almacenada en memoria secundaria o externa (disco o cinta), es necesario se definan procedimientos adicionales que así lo permitan.

Los archivos pueden ser utilizados para almacenar datos, guardar resultados, conservar programas y almacenar textos en general.

Existen dos modalidades para acceder a un archivo de datos: de manera secuencial y de manera directa.

El acceso secuencial realiza una lectura dato por dato comenzando por el primero mientras que el acceso directo permite acceder a datos específico a partir de conocer su posición dentro del archivo de almacenamiento.

Declaración de un archivo

Un archivo se declara de igual modo que cualquier otro tipo de datos definidos por el usuario (en la sección de tipos, **type**), el formato de declaración es:

Nombre - file of tipo de datos

donde nombre es el identificador que se asigna como nombre del archivo y tipo de datos es el tipo de conjunto al que pertenecen los datos (ejemplo: integer, real...).

Declaración de variables para manejar archivos

Para declarar una variable de tipo file se procede al igual que para las demás variables, sólo que en este caso se agregan las palabras **of file**, un ejemplo es:

```
VAR  Variable1 : real  
      archivo1 : file of real
```

Control de pantalla y teclado

Para el tratamiento de archivos se utiliza la unidad Crt que contiene los procedimientos necesarios para el manejo eficiente de una PC, para tener acceso a las palabras claves es necesario declararla en la sección USES (esto se ha estado haciendo desde DEMO1). La tabla 1 presenta los comandos para llamar a los procedimientos contenidos en la unidad Crt, algunos de ellos ya fueron utilizados, en los demos anteriores, dejado al lector investigue por cuenta propia la utilidad de los demás utilizando la ayuda de Turbo Pascal, es posible consultar algunos en el glosario.

PROCEDIMIENTOS	FUNCIONES	COLORES	
AssignCrt	WhereX	negro	0
ClrEol	WhereY	azul	1
Delay	Window	verde	2
Deline		cian	3
GotoXY		rojo	4
HighVideo		magenta	5
InsLine		marrón	6
KeyPressed		gris claro	7
LowVideo		gris oscuro	8
NormVideo		azul claro	9
ReadKey		verde claro	10
Sound		cian claro	11
TextBackground		rojo claro	12
TextColor		magenta claro	13
TextForeground		amarillo	14
TextMode		blanco	15
		parpadeo	128

Tabla 1 Procedimientos y funciones contenidas en la unidad Crt y constantes de color válidas.

EJEMPLO PROGRAMA DEMO3

Un ejemplo práctico de la utilidad de las estructuras hasta aquí vistas lo constituye DEMO3:

El programa DEMO3 permite construir, buscar y ampliar una base de datos. El programa se forma de los siguientes procedimientos:

ACTIVAR

El procedimiento se utiliza para verificar la existencia del archivo de datos (DEMO.DAT) si éste no existe procede a crearlo, para lograrlo se desactiva el detector de errores para que continúe la ejecución en caso de error entonces se abre el archivo si el programa marca error entonces se procede a crear dicho archivo.

AMPLIAR

Permite ampliar la base de datos, primero abre el archivo después utilizando el comando SEEK conoce el tamaño del archivo en seguida lee los nuevos datos de pantalla y finalmente los almacena al final del archivo.

ANEXO III Manual de Programación en Turbo Pascal

CONSULTAR Permite consultar algún datos almacenado, primero lee del teclado el dato a consultar, después inicia el rastreo del mismo, si lo encuentra lo despliega en la pantalla de lo contrario despliega un aviso de no encontrado.

LISTADOTOTAL Despliega el contenido de la base de datos en su totalidad.

Para dimensionar la base de datos que se va a utilizar se utiliza la sección **TYPE**, como se muestra en el listado de DEMO3, donde **ARNOMBRE** es el tipo de vector que se utiliza y **FICHERO** es el archivo que contiene a **ARNOMBRE**. Se utilizan básicamente dos variables: **F** para definir el nombre del archivo como variable y **DATO** para definir al vector como variable, como puede observarse en este caso se utilizan variables definidas por el usuario en la sección **TYPE**.

Los procedimientos se ligan a través de un programa principal (menú de selección) que se encuentra dentro de una estructura de tipo **REPEAT**, para salir de ella se utiliza la tecla "4" para hacer verdadera la sentencia **UNTIL** de dicha estructura. El diagrama de bloques de DEMO3 se presenta después listado del programa fuente de éste:

```
PROGRAM DEMO3;                                     {MANEJO DE REGISTROS,
                                                    ESTRUCTURAS REPEAT, WHILE,
CASE y COLORES}
USES
  CRT;                                             {*****DEFINICION DE REGISTRO****}

TYPE
  ARNOMBRE = RECORD
    ARCHIVO :STRING(10);
    PRESA   :STRING(20);
    TIPO    :STRING(20) ;
  END;
  FICHERO = FILE OF ARNOMBRE;                    {VARIABLE PARA ARNOMBRE}

VAR
  F: FICHERO;
  DATO:ARNOMBRE;

PROCEDURE ACTIVAR;                                {PROCEDIMIENTO DE DETECCION DE]
VAR                                              EXISTENCIA DE ARCHIVOS)
  RESULTADO : INTEGER;
BEGIN
  {#-1}
  RESET (F);
  RESULTADO := IORESULT;
  {#+1}
  IF RESULTADO <> 0
  THEN REWRITE (F);
  CLOSE (F)
  END;

PROCEDURE AMPLIAR;                                {emplia la base de datos}
VAR
  I : INTEGER;
BEGIN
  RESET (F);
  TEXTCOLOR(11);
  CtrScr;
  GOTOXY(2,2);WRITELN ('OPCION DE ALTAS');
  I:=FILESIZE (F);                               {PARA CONOCER EL TAMAÑO DE F}
```

ANEXO III Manual de Programación en Turbo Pascal

```
SEEK (F,I);                                     {POSICION DE LECTURA}
GOTOXY(10,5);WRITE('ARCHIVO : ');READLN(DATO.ARCHIVO);
GOTOXY(10,7);WRITE('TIPO : ');READLN(DATO.TIPO);
GOTOXY(10,9);WRITE('PRESA : ');READLN(DATO.PRESA);
WRITE (F,DATO);                                 {ESCRIBE LOS DATOS EN F}
CLOSE (F)
END;
```

```
PROCEDURE CONSULTAR;                           {PROCEDIMIENTO DE CONSULTA}
VAR
N : STRING(10);
ESTADO.BOOLEAN;
S:CHAR;
BEGIN
  RESET(F);
  ESTADO:= FALSE;
  CLRSCR;
  TEXTCOLOR(4);
  GOTOXY (5,3); WRITE('INTRODUZCA EL NOMBRE DEL REGISTRO : ');
  READLN(N);
  WHILE NOT EOF(F) DO                            {"Mientras NO termine el archivo"}
  BEGIN
    READ(F,DATO);
    IF DATO.ARCHIVO=N THEN
      BEGIN
        ESTADO:= TRUE;
        TEXTCOLOR(0);
        GOTOXY(5,5);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.ARCHIVO);
        GOTOXY(5,7);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.TIPO);
        GOTOXY(5,9);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.PRESA);
      END;
    END;
  IF NOT ESTADO THEN                             {ESTRUCTURA IF NEGADA}
  BEGIN
    GOTOXY(10,11);
    WRITELN ('NO EXISTE EL NOMBRE BUSCADO');
  END;
  TEXTCOLOR(11 + 128);                           {color parpadeante}
  GOTOXY(5,15);WRITELN('CUALQUIER TECLA PARA CONTINUAR');
  S:= READKEY;
  CLOSE (F)
END;
```

```
PROCEDURE LISTADOTAL;                           {VISUALIZACION TOTAL}
VAR
S:CHAR;
BEGIN
  RESET (F);
  WHILE NOT EOF(F) DO
  BEGIN
    ClrScr;
    READ(F,DATO);
    TEXTCOLOR(0);
    GOTOXY(5,5);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.ARCHIVO);
    GOTOXY(5,7);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.TIPO);
    GOTOXY(5,9);WRITELN('NOMBRE DEL ARCHIVO : ',DATO.PRESA);
    TEXTCOLOR(4);
    GOTOXY(5,12);WRITELN('CUALQUIER TECLA PARA CONTINUAR');
```

```

S:=READKEY;
END;
CLOSE (F)
END;

VAR                                {INICIA EL PROGRAMA PRINCIPAL}
OPCION : CHAR;                    {DEFINICION DE VARIABLE VALIDA}
BEGIN                              {EN TODO EL PROGRAMA}
  TextBackground(2);
  ClrScr;
  TEXTCOLOR(1);
  ASSIGN (F, 'C:DEMO3.DAT');      {ASIGNACION DEL ARCHIVO }
  ACTIVAR;                        {LLAMADO A UN PROCEDIMIENTO}
  REPEAT                            {ESTRUCTURA REPETITIVA}
    CLRSCR;TEXTCOLOR(1);
    GOTOXY (10,2); WRITELN (' MENU DE OPCIONES');
    GOTOXY (10,4); WRITELN ('1) DEFINIR UN NUEVO ARCHIVO');
    GOTOXY (10,6); WRITELN ('2) CONSULTAR SIGNIFICADOS');
    GOTOXY (10,8); WRITELN ('3) MOSTRAR TODOS LOS CONTENIDOS');
    TEXTCOLOR(4); GOTOXY (10,10); WRITELN ('4) SALIDA');
    TEXTCOLOR(10+128);
    GOTOXY (20,14); WRITE('ELIJA UNA OPCION : ');
  REPEAT
    OPCION := READKEY
  UNTIL OPCION IN '1'..'4';      {SE REPITE HASTA PULSAR LAS}
  CLRSCR;                        {TECLAS VALIDAS}
  CASE OPCION OF
    '1':AMPLIAR;                 {TOMA DOS POSIBLES CAMINOS}
    '2':CONSULTAR;               {POSIBLES CAMINOS LIGADOS A }
    '3':LISTADOTOTAL;           {LLAMAR PROCEDIMIENTOS}
  END;
  UNTIL OPCION = '4'             {TERMINA ESTRUCTURA REPEAT}
END.

```

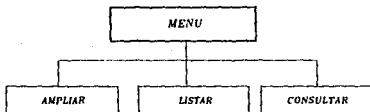


Fig. 5 Diagrama de bloques de programa DEMO3.

Tratamiento de archivos de texto

Las Pc cuentan con dos modos de presentación en vídeo: el modo texto y el modo gráfico.

Modo Texto: El modo texto se declara con el procedimiento Textmode (su uso práctico puede investigarse via Ayuda en línea de Turbo Pascal), al utilizar Pascal la declaración del modo texto es automática. Un ejemplo de control de pantalla y teclado se muestra en DEMO4.

Modo Gráfico: Para trabajar en modo gráfico es necesario incluir en la sección USES la unidad Graph, el uso de las los procedimientos de salida y entrada elementales se muestran en el programa DEMO5 así como algunas utilidades de la unidad Graph, se deja al lector investigue por cuenta propia las demás utilidades de Graph mediante el uso de Ayuda en línea.

EJEMPLO PROGRAMA DEMO4

En el programa DEMO4 permite leer un archivo en código ASCII*, en este caso se utiliza el archivo DEMO4.DAT, pero puede ser cualquier otro, se utiliza la unidad Crt, y tres variables: F, para el archivo, carácter, para leer los caracteres contenidos en el archivo y S, para declarar una pausa dentro del programa.

Cuando se utilizan variables para archivos de tipo TEXTO no es necesario declararlos en la sección TYPE, se puede notar la diferencia al comparar los programas DEMO3 y DEMO4.

```

PROGRAM DEMO4;           { VISUALIZAR ARCHIVOS DE TEXTOS }
USES CRT;

var
  f: text;               { VARIABLE PARA ABRIR UN ARCHIVO DE TIPO TEXTO }
  S,carácter: Char;     { INICIO DEL PROGRAMA }
begin
  TEXTCOLOR(!);        { ASIGNACION DEL NOMBRE DE UN ARCHIVO A UNA VARIABLE }
  Assign(f, 'demo4.dat'); { ABRIENDO EL ARCHIVO }
  Reset(f);             { ESTRUCTURA MIENTRAS NO ACABE EN ARCHIVO F }
  while not Eof(f) do  { COMIENZO DE LA TAREA }
  begin
    Read(f,CARACTER);  { LECTURA DEL ARCHIVO CARACTER POR CARACTER }
    Write(CARACTER);   { ESCRITURA DEL MISMO MODO }
  end;                 { TERMINACION DE LA TAREA }
  close(f);            { CERRANDO EL ARCHIVO }
  S:=READKEY           { PAUSA EN LA EJECUCION }
end.                  { FINAL DEL PROGRAMA }
    
```

En la tabla 2 se muestran los procedimientos contenidos en las unidades Crt y Dos de Turbo Pascal útiles para el tratamiento de archivos, algunos de ellos se han utilizado en DEMO4, si desea investigar sobre el uso de los demás se recomienda acudir a la ayuda en línea o al glosario de este trabajo.

TODOS LOS ARCHIVOS	ARCHIVOS DE TEXTO	ARCHIVOS SIN TIPO
Assign	Append	BlockRead
ChDir	Flush	Blockwrite
Close	Read	
Erase	Readln	
GetDir	SetTextBuf	
MkDir	Write	
Rename	Writeln	
Reset		
Rewrite		
RmDir		
Eof	Eofn	
IORResult	SeekEof	
FilePos	SeekEofn	
FileSize		
Seek		
Truncate		

Tabla 2 Procedimientos y funciones válidos en el tratamiento de archivos.

EJEMPLO PROGRAMA DEMOS

El programa DEMOS constituye un ejemplo del manejo de gráficos desde Turbo Pascal, el cual esta compuesto por los siguientes procedimientos:

- | | |
|---------------------|--|
| DETECTAR | Procedimiento para amoldar el programa a las características del tipo de monitor de la computadora. |
| PRESENTACION | Despliega el texto "BIENVENIDOS", y lo mueve en dirección inclinada, para lograrlo el texto se encuentra dentro de una estructura FOR. |
| DIBUJO | Dibuja un cubo y desde uno de sus vértices despliega líneas en diferentes dirección esto se logra mediante una estructura FOR. |

PROGRAM DEMOS; (USOS ELEMENTALES DE GRAFICOS)

```
uses
  Graph,CRT; {DECLARACION DE UNIDADES}
var
  Gd, Gm : Integer;
  Radius : Integer;
```

PROCEDURE DETECTAR;

{DETECTA EL TIPO DE MONITOR ACTIVO}**

```
VAR
  D,M:Integer;
BEGIN
  D:=DETECT;
  INITGRAPH(D,M,'');
  IF GraphResult <> grOk then Halt(1);
END;
```

PROCEDURE PRESENTACION;

```
VAR
  I:INTEGER;
```

BEGIN

```
SETBKCOLOR(1);
SETTEXTSTYLE(1,0,5);
FOR I:=1 TO 50 DO
  BEGIN
```

**{SELECCION DEL COLOR DE FONDO}
{SELECCIÓN DEL TIPO DE LETRA Y TAMAÑO}
{DESPLAZAMIENTO DEL TEXTO}
{LOS DOS PRIMEROS DATOS DE OUTTEXTXY SON LA
COORDENADAS, donde X son columnas y Y es
renglones}**

```
    SETCOLOR(I);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
    SETCOLOR(0);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
  END;
  SETCOLOR(1);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
  SETTEXTSTYLE(1,0,3);
  SETCOLOR(14);OUTTEXTXY(80,300,'ESTE ES UN PRODUCTO MAS DEL II');
  FOR Radius:=1 to 5 do
    BEGIN
      Circle(100,100,Radius*10);
      Circle(450,100,Radius*10);
    END;
  DELAY(2500);
END;
```

PROCEDURE DIBUJO;

```

VAR I:INTEGER;
BEGIN
  SetFillStyle(0,1);           (ESTILO DE RELLENO)
  FloodFill(3,1,7);          (RELLENAR LA SUPERFICIE DE UN COLOR DEFINIDO)
  SETCOLOR(10);              (DIBUJA UN CAJA)
  BAR3D(200,200,300,300,20,TRUE);
  DELAY(400);
  FOR I:= 1 TO 60 DO
  BEGIN
    DELAY(180);
    LINE(300,200,I* 10,400);  (DIBUJAR LINEAS)
  END;
END;

BEGIN
  DETECTAR;                   (PROGRAMA PRINCIPAL)
  PRESENTACION;
  DIBUJO;
  DELAY(250);
  CloseGraph;                 (REESTABLECER PANTALLA DE TEXTO)
END.

```

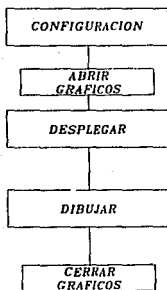


Fig.6 Diagrama de bloques del programa DEMOS

EJEMPLO PROGRAMA DEMO6

El programa DEMO6 despliega palabras y usando las teclas flecha arriba y flecha abajo es posible iluminar una a la vez, el programa principal está dentro de una estructura REPEAT, la cual se rompe cuando se oprime la tecla ENTER. Para lograrlo se utilizan tres arreglos vectoriales: PAL, vector de tipo cadena de 25 caracteres; X y Y, vectores de valores enteros.

Los procedimientos que contiene son:

CARGAR	Sirve para declarar en contenido de los vectores
DESPLEGAR	Para desplegar las palabras en la pantalla
ACTIVAR	Para iluminar la primer palabra
DESACTIVAR	Para regresar al color original la palabra
ADELANTE	Para iluminar la siguiente palabra
REGRESAR	Para iluminar la palabra anterior
MENU	Liga los procedimientos anteriores

```
PROGRAM DEMO6;                                {DECLARACION DE MENUS}
USES CRT;
VAR
PAL: ARRAY [1..5] OF STRING[25];            {DECLARACION DE VECTORES}
X,Y:ARRAY [1..5] OF INTEGER;
VAR I:INTEGER;

PROCEDURE CARGAR;                              {DECLARACION DE CONTENIDO}
BEGIN
X[1]:= 25;Y[1]:= 6; PAL[1]:= 'VER LA PRESENTACION';
X[2]:= 25;Y[2]:= 9; PAL[2]:= 'REALIZAR CALCULOS DE AREA';
X[3]:= 25;Y[3]:= 12;PAL[3]:= 'LEER UN TEXTO';
X[4]:= 25;Y[4]:= 15;PAL[4]:= 'TRABAJAR CON REGISTROS';
X[5]:= 25;Y[5]:= 18;PAL[5]:= 'SALIR';

END;

PROCEDURE DESPLEGAR;                          {DESPLEGAR LOS VECTORES}
BEGIN
  TextBackground(1);
  CLRSCR;
  TEXTCOLOR(14);
  FOR I:= 1 TO 5 DO
    BEGIN
      GOTOXY(X[I],Y[I]);WRITE(PAL[I])
    END;
END;
```

ANEXO III Manual de Programación en Turbo Pascal

```
PROCEDURE ACTIVAR(I:INTEGER);           {ILUMINAR EL ELEMENTO I DEL VECTOR}
BEGIN
  TextBackground(3);
  TEXTCOLOR(9);
  GOTOXY(X(I), Y(I));WRITE(PAL(I))
END;

PROCEDURE DESACTIVAR(I:INTEGER);        {APAGAR EL ELEMENTO I}
BEGIN
  TextBackground(1);
  TEXTCOLOR(14);
  GOTOXY(X(I), Y(I));WRITE(PAL(I))
END;

PROCEDURE ADELANTE(I:INTEGER);          {ADELANTAR UNA POSICION EL VECTOR}
BEGIN
  DESACTIVAR(I);
  IF I = 5 THEN I := 1 ELSE I := I + 1;
  ACTIVAR(I);
END;

PROCEDURE REGRESA(I:INTEGER);           {DISMINUIR EL VECTOR UNA POSICION}
BEGIN
  DESACTIVAR(I);
  IF I = 1 THEN I := 5 ELSE I := I - 1;
  ACTIVAR(I);
END;

PROCEDURE MENU;                         {SELECCION DE LA PALABRA}
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: REGRESA(I);
    #13: SALIR := TRUE;
  END;
UNTIL(SALIR);
END;

{*****PROGRAMA PRINCIPAL*****}
BEGIN
  CARGAR;
  DESPLEGAR;
  I := 1;ACTIVAR(I);
  MENU;
END;
```

Para que puedan ser utilizadas las teclas especiales, en este caso Flecha arriba y Flecha abajo estas deben ser declaradas en código ASCII, como puede observarse el DEMO6 se utiliza el símbolo # seguido del número ASCII correspondiente a las teclas que se desea declarar.

Las tareas que realiza el programa son:

- 1) Desplegar un vector de palabras en coordenadas dadas en forma de vector también.
- 2) Inicializar el contador en 1 y iluminar (activar) la palabra guardada en la posición 1 del vector de palabras.
- 3) Esperar a oprimir una de las tres teclas posibles, el camino de flujo depende de la tecla oprimida
 - 3.1) Si se oprime la tecla Flecha arriba, entonces se regresa al color original (desactiva) la palabra iluminada, se disminuye el contador en una posición y se ilumina la palabra contenida en esa posición en el vector de palabras.
 - 3.2) Si se oprime la tecla Flecha abajo entonces se desactiva la palabra actual, se aumenta en una posición el contador y se ilumina la palabra contenida en esa posición en el vector de palabras.
 - 3.3) Si se oprime la tecla ENTER se le asigna a la variable Salir el valor de verdadero con lo que termina el ciclo.
 - 3.4) Si se oprime cualquier otra tecla se regresa a esperar oprimir otra tecla.

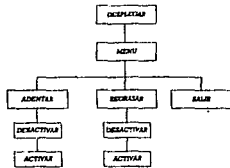


Fig. 7 Diagrama de bloque del programa DEMO6

Programas como DEMO6 se utilizan frecuentemente como estructuras de menú para programas mucho mayores, un ejemplo es el programa DEMO7.

EJEMPLO PROGRAMA DEMO7

El programa DEMO7 ensambla algunos de los DEMOS anteriores utilizando como soporte al DEMO6, en este caso la tecla ENTER esta asociada a el procedimiento TAREAS.

TAREAS El procedimiento sirve para redirigir el flujo del programa a otros procedimientos mediante una estructura de tipo CASE.

```

PROGRAM DEMO7;                                     {ENSAMBLE DE ALGUNOS DEMOS
                                                    MEDIANTE EL USO DE UN MENU}

USES
  Graph,CRT;

PROCEDURE DEMOS;                                   {*****Presentación y TAREA #1}
var
  Gd, Gm : Integer;
  rAdius : Integer;
  PROCEDURE DETECTAR;
  VAR
    D,M:Integer;
  BEGIN
    D:=DETECT;
    INITGRAPHID,M,"";
    IF GraphResult<>grOk then Halt(1);
  END;
  PROCEDURE PRESENTACION;
  VAR
    I,RADIO:INTEGER;
  BEGIN
    SETBKCOLOR(1);
    SETTEXTSTYLE(1,0,5);
    FOR I:= 1 TO 50 DO
      BEGIN
        SETCOLOR(1);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
        SETCOLOR(0);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
      END;
    SETCOLOR(11);OUTTEXTXY(I*3,I*4,'BIENVENIDO');
    SETTEXTSTYLE(1,0,3);
    SETCOLOR(14);OUTTEXTXY(80,300,'ESTE ES UN PRODUCTO MAS DEL I');
    FOR RADIO:= 1 to 5 do
      BEGIN
        Circle(100, 100, RADIO*10);
        Circle(450, 100, RADIO*10);
      END;
    DELAY(2500);
  END;

BEGIN
  DETECTAR;
  PRESENTACION;
  DELAY(250);
  CloseGraph;
END;
PROCEDURE DEMO1;                                   {*****TAREA #2*****}
  Const  Pi = 3.1416;
  Var    A,r:real;

```

ANEXO III Manual de Programación en Turbo Pascal

```
Begin
  TextBackground(3);
  TEXTCOLOR(4);
  Clscr;
  WriteLn('Programa para calcular el Area de un Circulo');
  GotoXY(5,4);
  Write ('Introduzca el Valor del Radio : ');
  ReadLn(R);
  A:= Pi*R*R;
  GotoXY(5,6);
  Write(' El Area es : ',A;4:3);
  GotoXY(5,9);
  Write(' Para Continuar Presione ENTER ');
  ReadLn;
end;
```

```
PROCEDURE INDEFINIDO;
```

{***PARA LAS DEMAS OPCIONES}

```
BEGIN
  WINDOW(20,8,60,14);
  TextBackground(7);
  TEXTCOLOR(10+ 128);
  Clscr;
  GotoXY(5,4);
  WriteLn('); TAREA INDEFINIDA II');
  DELAY(1400);
  WINDOW(1,1,80,25);
END;
```

```
PROCEDURE PRINCIPAL;
```

{PROCEDIMIENTG PRINCIPAL}

```
VAR
  PAL: ARRAY [1..5] OF STRING(25);
  X1,Y1:ARRAY [1..5] OF INTEGER;
  I:INTEGER;
```

```
PROCEDURE CARGAR;
```

```
BEGIN
  X1[1]:= 25;Y1[1]:= 6; PAL[1]:= 'VER LA PRESENTACION ';
  X1[2]:= 25;Y1[2]:= 9; PAL[2]:= 'REALIZAR CALCULOS DE AREA';
  X1[3]:= 25;Y1[3]:= 12;PAL[3]:= 'LEER UN TEXTO';
  X1[4]:= 25;Y1[4]:= 15;PAL[4]:= 'TRABAJAR CON REGISTROS';
  X1[5]:= 25;Y1[5]:= 18;PAL[5]:= 'SALIR';
END;
```

```
PROCEDURE DESPLEGAR;
```

```
VAR I:INTEGER;
BEGIN
  TextBackground(1);
  CLRSCR;
  TEXTCOLOR(14);
  FOR I:= 1 TO 5 DO
    BEGIN
      GOTOXY(X1[I],Y1[I]);WRITE(PAL[I])
    END;
  END;
```

```
PROCEDURE ACTIVAR(I:INTEGER);
```

ANEXO III Manual de Programación en Turbo Pascal

```
BEGIN
  TextBackground(3);
  TEXTCOLOR(9);
  GOTOXY(X1(I), Y1(I));WRITE(PAL(I))
END;

PROCEDURE DESACTIVAR(VAR I:INTEGER);
BEGIN
  TextBackground(1);
  TEXTCOLOR(14);
  GOTOXY(X1(I), Y1(I));WRITE(PAL(I))
END;

PROCEDURE ADELANTE(VAR I:INTEGER);
BEGIN
  DESACTIVAR(I);
  IF I=5 THEN I:=1 ELSE I:=I+1;
  ACTIVAR(I);
END;

PROCEDURE REGRESA(VAR I:INTEGER);
BEGIN
  DESACTIVAR(I);
  IF I=1 THEN I:=5 ELSE I:=I-1;
  ACTIVAR(I);
END;

PROCEDURE TAREAS(VAR I:INTEGER);
BEGIN
  CASE I of
    2:BEGIN
      WINDOW(15,5,60,15);
      DEMO1;
      WINDOW(1,1,80,25);
      DESPLEGAR:I:=1;
      ACTIVAR(I);
    END;
    1:BEGIN
      DEMOS;
      DESPLEGAR:I:=1;
      ACTIVAR(I);
    END;
    3,4:BEGIN
      INDEFINIDO;
      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(I);
  #72: REGRESA(I);
  #13: IF I = 5 THEN SALIR:= TRUE ELSE TAREAS(I);
END;
UNTIL(SALIR);
END;

```

```

BEGIN
CARGAR;
DESPLEGAR;
I:= 1;ACTIVAR(I);
MENU;
END;

```

```

BEGIN
demo5;
PRINCIPAL;
END.

```

{PROGRAMA PRINCIPAL DE MENU}

{*****FIN DEL PROCEDIMIENTO MENU}
{PROGRAMA PRINCIPAL}

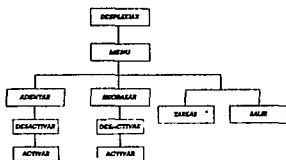


Fig. 8 Diagrama de bloques del programa DEMO7

Se propone como ejercicio final la modificación del programa DEMO7, para incorporar los DEMOS que faltan.

Se recuerda que los comandos utilizados en este ANEXO se explican en el ANEXO VIII

ANEXO IV

MODULO ADMINISTRATIVO

DESCRIPCION DE LOS PROCEDIMIENTO UTILIZADOS DENTRO DEL MODULO¹

En orden numérico:

- 1 CONTINUAR *Después de elegir una opción detiene la ejecución del programa y pide al usuario confirme la opción si la confirma sigue adelante de lo contrario regresa al menú principal.*
- 2 LIMPIAR *El procedimiento desactiva cualquier ventana en uso y limpia toda la pantalla.*
- 3 TRABAJANDO *Activa ventana de comunicación con el usuario donde se le informa que la computadora esta procesando información.*
- 4 HACIENDO *Activa ventana en la parte superior derecha de la pantalla, para ser utilizada como salida de la opción en la que se encuentre trabajando el programa.*
- 5 MOSTRAR *Activa ventana para capturar y mostrar el contenido de la base de datos.*
- 6 MOSTRARI *Activa ventana para desplegar las características del nombre que se le ordene.*
- 7 ELEGIR *Activa ventana para desplegar menú*
- 8 COMENTARIO *Activa ventana para desplegar la ubicación del archivo en disco.*
- 9 ACTIVAR *Permite saber si existe el archivo de almacenamiento, de no existir dicho archivo el programa procede a crearlo.*

¹ **LOS NOMBRES QUE IDENTIFICAN A LOS PROCEDIMIENTOS UTILIZADOS, EN LO GENERAL SON PALABRAS COMPLETAS O COMPUESTAS (TODAS EN ESPAÑOL) LO QUE HACE RELATIVAMENTE FACIL DIFERENCIAR LOS PROCEDIMIENTOS DE LOS COMANDOS PROPIOS DE TURBO PASCAL.**

- 10 POSICION** *La función permite conocer el lugar dentro del archivo de almacenamiento donde se encuentra ubicado el archivo que busca.*
- 11 FORMATO** *Despliega un formato especial para capturar nueva información.*
- 12 LEERREGISTRO** *Permite capturar nueva información.*
- 13 VISUALIZAR** *Permite desplegar la información contenida en un lugar determinado dentro del archivo de almacenamiento y seleccionado con la función posición.*
- 14 LISTADOTOTAL** *Visualiza la totalidad de la base de datos, para salir del procedimiento se utiliza la tecla Esc.*
- 15 AMPLIAR** *Permite ampliar la base de datos*
- 16 BORRAR** *procedimiento para dar de baja el nombre de un archivo*
- 17 MODIFICAR** *Permite modificar el nombre y las características de un archivo*
- 18 CONSULTAR** *Permite buscar las características de un archivo a partir de conocer el nombre del mismo.*
- 19 BUSCAR** *Permite rastrear el nombre de un archivo a partir de conocer una o un grupo de características del mismo.*
- 20 ENTRARADTOS** *Es un procedimiento interno de 19 y permite capturar los datos para iniciar el rastreo.*
- 21 DESPLEGAR** *Es un procedimiento interno de 19 y permite desplegar el nombre y las características del archivo rastreado.*
- 22 MENU** *Procedimiento que activa el menú de opción principal.*

En orden alfabético:

- 9 ACTIVAR** *Permite saber si existe el archivo de almacenamiento, de no existir dicho archivo el programa procede a crearlo.*
- 15 AMPLIAR** *Permite ampliar la base de datos*
- 16 BORRAR** *Procedimiento para dar de baja el nombre de un archivo*
- 19 BUSCAR** *Permite rastrear el nombre de un archivo a partir de conocer una o un grupo de características del mismo.*
- 8 COMENTARIO** *Activa ventana para desplegar la ubicación del archivo en disco.*

ANEXO IV Módulo Administrativo

- 18 CONSULTAR** *Permite buscar las características de un archivo a partir de conocer el nombre del mismo.*
- 1 CONTINUAR** *Después de elegir una opción detiene la ejecución del programa y pide al usuario confirme la opción si la confirma sigue adelante de lo contrario regresa al menú principal.*
- 21 DESPLEGAR** *Es un procedimiento interno de 19 y permite desplegar el nombre y las características del archivo rastreado.*
- 7 ELEGIR** *Activa ventana para desplegar menú*
- 20 ENTRARADTOS** *Es un procedimiento interno de 19 y permite capturar los datos para iniciar el rastreo.*
- 11 FORMATO** *Despliega un formato especial para capturar nueva información.*
- 4 HACIENDO** *Activa ventana en la parte superior derecha de la pantalla, para ser utilizada como salida de la opción en la que se encuentre trabajando el programa.*
- 12 LEERREGISTRO** *Permite capturar nueva información.*
- 2 LIMPIAR** *El procedimiento desactiva cualquier ventana en uso y limpia toda la pantalla.*
- 14 LISTADOTOTAL** *Visualiza la totalidad de la base de datos, para salir del procedimiento se utiliza la tecla Esc.*
- 22 MENU** *Procedimiento que activa el menú de opción principal.*
- 17 MODIFICAR** *Permite modificar el nombre y las características de un archivo*
- 5 MOSTRAR** *Activa ventana para capturar y mostrar el contenido de la base de datos.*
- 6 MOSTRARI** *Activa ventana para desplegar las características del nombre que se le ordene.*
- 10 POSICION** *La función permite conocer el lugar dentro del archivo de almacenamiento donde se encuentra ubicado el archivo que busca.*
- 3 TRABAJANDO** *Activa ventana de comunicación con el usuario donde se le informa que la computadora esta procesando información.*
- 13 VISUALIZAR** *Permite desplegar la información contenida en un lugar determinado dentro del archivo de almacenamiento y seleccionado con la función posición.*

Listado del Programa:

PROGRAM BUSCA_INFORMACION;

USES CRT;

[DECLARACION DEL CAMPO DE CLASIFICACION,
con lo cual es posible definir los lugares o espacios en el
vector así como el tipo de elemento que se va a
abstronar en cada uno de éstos]

TYPE

CADENA=STRING[10];

CADENA2=STRING[20];

ARNOMBRE = RECORD

ARCHIVO :STRING[10];

TIPO :CADENA2;

PRESA :CADENA2;

SISMO :CADENA2;

SECCION :CADENA2;

CUNA :CADENA2;

COMENT :STRING[80];

SW :BOOLEAN;

END;

FICHERO = FILE OF ARNOMBRE;

VAR

ARCH : FICHERO;

A :INTEGER;

[1 CONFIRMACION DE LA ACCION A REALIZAR]

PROCEDURE CONTINUAR;

BEGIN

WINDOW(19,10,62,14);

TextBackground(0);

ClrScr;

WINDOW(17,9,60,13);

TextBackground(10);

ClrScr;

TEXTCOLOR(1);

GOTOXY(5,3);

WRITELN('DESEA SEGUIR EN LA OPCION: (S) N');

TEXTCOLOR(4+128);

GOTOXY(36,3);WRITELN('S');

END;

[2 BORRA EL CONTENIDO DE LA PANTALLA]

PROCEDURE LIMPIAR;

BEGIN

WINDOW(1,1,80,25);

TextBackground(1);

TEXTCOLOR(0);

ClrScr;

END;

[3 AVISO DE TRABAJO]

PROCEDURE TRABAJANDO;

VAR A:INTEGER;

BEGIN

WINDOW(25,8,57,14);

TEXTBACKGROUND(0);

CLRSCR;

WINDOW(24,7,55,13);

TEXTBACKGROUND(4);

CLRSCR;

TEXTCOLOR(11);

FOR A:=2 TO 30 DO

BEGIN

GOTOXY(A,1); WRITE('==');

GOTOXY(A,7);WRITE('==');

IF A<7 THEN

BEGIN

GOTOXY(1,A);WRITE(' | ');

GOTOXY(31,A);WRITE(' | ');

END;

END;

GOTOXY(1,1); WRITE(' ==');

GOTOXY(1,7);WRITE(' ==');

GOTOXY(31,7);WRITE(' ==');

GOTOXY(31,7);WRITE(' ==');

GOTOXY(10,4);

TEXTCOLOR(14+128);

WRITE('TRABAJANDO...');

END;

[4 VENTANA PARA AVISO DE OPCION]

PROCEDURE HACIENDO;

BEGIN

WINDOW(62,7,9,4);

TextBackground(0);

CLRSCR;

WINDOW(60,3,77,5);

TextBackground(2);

CLRSCR;

TEXTCOLOR(9+128);

GOTOXY(4,2);

END;

(5) VENTANA PARA MOSTRAR LA INFORMACION

```
PROCEDURE MOSTRAR;
BEGIN
  WINDOW(12,4,47,21);
  TextBackground(0);
  CLRSCR;
  TEXTCOLOR(9+128);
  WINDOW(10,3,45,20);
  TextBackground(3);
  CLRSCR;
END;
```

(6) OPCIONES PARA DESPLEGAR INFORMACION

```
PROCEDURE MOSTRAR1;
BEGIN
  WINDOW(55,15,77,18);
  TextBackground(0);
  CLRSCR;
  WINDOW(53,14,75,17);
  TextBackground(7);
  CLRSCR;
  TEXTCOLOR(9+128);
  GOTOXY(7,2); TEXTCOLOR(12);
  WRITELN('M) PARA MAYOR INFORMACION');
  WINDOW(55,21,77,24);
  TextBackground(0);
  CLRSCR;
  WINDOW(53,20,75,23);
  TextBackground(7); CLRSCR;
  GOTOXY(7,2);
  TEXTCOLOR(12);
  WRITELN('(Esc) SALIR');
  WINDOW(55,9,77,12);
  TextBackground(0); CLRSCR;
  WINDOW(53,8,75,11);
  TextBackground(4); CLRSCR;
  TEXTCOLOR(9+128); GOTOXY(2,2);
  TEXTCOLOR(11);
  WRITE('ENTER) LEER OTRO REGISTRO');
END;
```

(7) VENTANA PARA MENU

```
PROCEDURE ELEGIR;
BEGIN;
  WINDOW(12,9,62,13);
  TextBackground(0);
  ClrScr;
  WINDOW(10,8,60,12);
  TextBackground(3);
  ClrScr;
  TEXTCOLOR(5);
END;
```

(8) VENTANA PARA DESPLEGAR LUGAR EN DISCO

```
PROCEDURE COMENTARIO;
BEGIN
  WINDOW(10,19,70,24);
  TextBackground(0);
  ClrScr;
  WINDOW(8,18,68,23);
  TEXTBackground(4);
  ClrScr;
END;
```

(9) PERMITE CONOCER INICIALIZAR O LEER LA BASE DE DATOS

```
PROCEDURE ACTIVAR (VAR F: FICHERO);
VAR
  RESULTADO : INTEGER;
BEGIN
  CLRSCR;
  (31-1)
  RESET (F);
  RESULTADO := IORESULT;
  (31+1)
  IF RESULTADO <> 0 THEN REWRITE (F);
  CLOSE (F)
END;
```

(10) PERMITE CONOCER EL LUGAR DONDE SE ENCUENTRA EL DATO BUSCADO

```
FUNCTION POSICION (N : CADENA; VAR F: FICHERO) : INTEGER;
VAR
  REGISTRO : ARNOMBRE;
  HALLADO : BOOLEAN;
BEGIN
  HALLADO := FALSE;
  SEEK(F,0);
  WHILE NOT EOF(F) AND NOT HALLADO DO
  BEGIN
    READ (F,REGISTRO);
    HALLADO := REGISTRO.ARCHIVO = N
  END;
  IF HALLADO THEN
    POSICION := FILEPOS(F) - 1
  ELSE
    POSICION := - 1
  END;
```

(11) FORMATO PARA CAPTURAR INFORMACION

```
PROCEDURE FORMATO;
BEGIN
  TEXTCOLOR(14);
  GOTOXY(4,4); WRITELN('ARCHIVO :');
```

```

GOTOXY(4,6); WRITELN(TIPO :);
GOTOXY(4,8); WRITELN(PRESA :);
GOTOXY(4,10); WRITELN(SISMO :);
GOTOXY(4,12); WRITELN(SECCION :);
GOTOXY(4,14); WRITELN(CUÑA :);
END;

```

12 PROCEDIMIENTO PARA CAPTURAR INFORMACION

PROCEDURE LEBERREGISTRO (VAR E: ARNOMBRE);

```

BEGIN
WITH E DO
BEGIN
GOTOXY(15,4); READLN(ARCHIVO);
GOTOXY(15,6); READLN(TIPO);
GOTOXY(15,8); READLN(PRESA);
GOTOXY(15,10); READLN(SISMO);
GOTOXY(15,12); READLN(SECCION);
GOTOXY(15,14); READLN(CUNA);
COMENTARIO.TEXTCOLOR(0); WRITE(' ESCRIBA SU
UBICACION EN DISCO ');
WINDOW(14,20,60,22);
READLN(COMENT);
SW := TRUE;
END;
END;

```

13 PROCEDIMIENTO PARA VIZUALIZAR INFORMACION SELECCIONADA

PROCEDURE VISUALIZAR (E : ARNOMBRE);

```

BEGIN
WITH E DO
IF E.SW THEN
BEGIN
TEXTBackground(3); CLRSCR;
TEXTCOLOR(10); GOTOXY(3,4);
WRITELN(ARCHIVO : :ARCHIVO);
IF TIP = 'S' THEN BEGIN GOTOXY(3,6);
WRITELN(ES DE : :TIPO);
END;
IF PRESA = 'P' THEN
BEGIN
GOTOXY(3,8);
WRITELN(DE LA PRESA: :PRESA);
END;
IF SISMO = 'S' THEN
BEGIN
GOTOXY(3,10);
WRITELN(DEL SISMO : :SISMO);
END;
IF SECCION = 'S' THEN
BEGIN
GOTOXY(3,12);

```

```

WRITELN(SECCION : :SECCION);
END;
IF CUNA = 'C' THEN
BEGIN
GOTOXY(3,14);
WRITELN(EN LA CUÑA : :CUNA);
END;
END;
END;

```

14 PROCEDIMIENTO CON EL CUAL SE LOGRA VISUALIZAR TODA LA BASE DE DATOS

PROCEDURE LISTADOTOTAL (VAR F: FICHERO);

```

VAR
E: ARNOMBRE;
SALIR: CHAR;
BEGIN
RESET(F);
WHILE NOT EOF(F) DO
BEGIN
LIMPIAR;
HACIENDO;
WRITELN('VISUALIZANDO');
MOSTRAR;
MOSTRAR;
CLRSCR;
TEXTCOLOR(10);
GOTOXY(5,1);
WRITELN(REGISTRO: : FILEPOS(F)+1:1);
READ(F,F);
IF E.SW THEN
BEGIN
VISUALIZAR(F);
SALIR := READKEY;
END
ELSE
BEGIN
TEXTCOLOR(5+128);
GOTOXY(10,9);
WRITELN('i REGISTRO VACIO i');
SALIR := READKEY;
END;
CASE SALIR OF
#27:
BEGIN
CLOSE(F);
EXIT;
END;
'M', 'm':
BEGIN
COMENTARIO;
WRITE(UBICACION DEL ARCHIVO: :
E.ARCHIVO);

```

```

WINDOW(14,20,60,22);
TEXTCOLOR(0);
WRITE(E.COMENT);
SALIR:=READKEY;
END;
END;
END;
CLOSE (F)
END;

```

(15) PROCEDIMIENTO PARA AMPLIAR LA BASE DE

```

DATOS
PROCEDURE AMPLIAR (VAR F : FICHERO);
VAR
R,E: ARNOMBRE;
I : INTEGER;
SI:CIAR;
SALI:BOOLEAN;
BEGIN
RESET (F);
REPEAT
BEGIN
LIMPIAR;
CONTINUAR;
GOTOXY(15,4);WRITE('AMPLIANDO...');
SALI:=FALSE;
SI:=READKEY;
CASE SI OF
'N', 'n':
BEGIN;
SALI:=TRUE;
CLOSE(F);
EXIT;
END;
END;
LIMPIAR;
HACIENDO;
WRITE('AMPLIANDO');
MOSTRAR;
TEXTCOLOR(4);GOTOXY(8,1);
Writeln('INTRODUZCA LOS DATOS');
TEXTCOLOR(12);
FORMATO;
TEXTCOLOR(4);
LEERREGISTRO (F);
I:= POSICION (E.ARCHIVO, F);
IF I = -1 THEN
BEGIN
I := FILESIZE (F);
SEEK (F,0);
WRITE (F,E);
END
ELSE;

```

```

BEGIN
SEEK (F,0);
WRITE (F,E);
IF R.SW THEN
BEGIN
GOTOXY(10,16);
Writeln ('EL REGISTRO YA EXISTE');
DELAY(700);
END
ELSE
WRITE (F,F)
END;
END;
END;
UNTIL(SALI);
CLOSE (F)
END;

```

(16) PROCEDIMIENTO QUE PERMITE BORRAR EL NOMBRE DE UN ARCHIVO)

```

PROCEDURE BORRAR (VAR F : FICHERO);
VAR
E : ARNOMBRE;
N :CADENA;
I : INTEGER;
SALI:CIAR;
SALVER:BOOLEAN;
BEGIN
SALVER:=FALSE;
RESET (F);
REPEAT
LIMPIAR;
HACIENDO;
Writeln ('BORRANDO');
RESET (F);
ELEGIR;
GOTOXY (3,2);
Writeln ('INTRODUZCA EL NOMBRE DEL
REGISTRO A BORRAR');
GOTOXY (10,4);
READLN (N);
LIMPIAR;
I:= POSICION (N,F);
IF I = -1 THEN
BEGIN
GOTOXY (13,12);
Writeln ('NO EXISTE EL NOMBRE !!!')
END
ELSE
BEGIN
SEEK(F,0);
READ(F,E);
IF E.SW THEN
BEGIN

```



```

MOSTRAR;
FORMATO;
VISUALIZAR (E);
E.SW := FALSE;
I := FILEPOS (F) -1;
SEEK (F,I);
WRITE (F,E)
END;
TEXTCOLOR(128+1);
GOTOXY(5,17);
WRITELN('¡¡ DADO DE BAJA !!');
END;
DELAY(1200);
LIMPIAR;
CONTINUAR;
TEXTCOLOR(4+128);
WRITE(' BORRANDO...');
SAL2:=READKEY;
IF SAL2='n' THEN SALVER:=TRUE;
IF SAL2='N' THEN SALVER:=TRUE;
UNTIL SALVER,
CLOSE (F)
END;

[17 PROCEDIMIENTO QUE PERMITE MODIFICAR
INFORMACION YA DECLARADA]
PROCEDURE MODIFICAR (VAR F: FICHERO);
VAR
E: ARNOMBRE;
N: CADENA;
I: INTEGER;
S21,S22: CHAR;
BEGIN
RESET(F);
REPEAT
LIMPIAR;
HACIENDO;
CLRSCR;
GOTOXY(5,2);WRITE('MODIFICANDO');
ELEGIR;
GOTOXY(10,2);
WRITE('INTRODUZCA EL NOMBRE DEL
REGISTRO');
GOTOXY(15,3);
READLN (N);
MOSTRAR;
CLRSCR;
I:= POSICION (N,F);
IF I = -1 THEN
BEGIN
GOTOXY(5,9);TEXTCOLOR(6);
WRITELN ('NO EXISTE EL NOMBRE BUSCADO');
S22:=READKEY;

```

```

END
ELSE
BEGIN
SEEK(F,I);
READ(F,E);
IF E.SW THEN
BEGIN
TEXTCOLOR(0);GOTOXY(5,2);
WRITELN('INTRODUZCA LO NUEVOS DATOS...');
VISUALIZAR (E);
S221:=READKEY;
CLRSCR;
FORMATO;
LEERREGISTRO (E);
I := FILEPOS (F) -1;
SEEK(F,I);
WRITE(F,E);
TEXTCOLOR(14);
GOTOXY(5,18);
WRITE('REGISTRO MODIFICADO');
END
ELSE
BEGIN
TEXTCOLOR(14);
GOTOXY(5,8);
WRITELN('EL REGISTRO FUE DADO DE BAJA');
END;
END;
DELAY(900);
LIMPIAR;
CONTINUAR;
WRITELN(' MODIFICANDO... ');
UNTIL UPCASE (READKEY) = 'N';
CLOSE (F)
END;

```

```

[18 PROCEDIMIENTO PARA CONSULTAR
INFORMACION A PARTIR DE CONOCER EL
NOMBRE DEL ARCHIVO]
PROCEDURE CONSULTAR (VAR F: FICHERO);
VAR
E: ARNOMBRE;
N: CADENA;
I: INTEGER;
S1,S21: CHAR;
BEGIN
RESET (F);
REPEAT
LIMPIAR;

```

```

HACIENDO:
GOTOXY(5,2);WRITE('CONSULTANDO');
ELEGIR;
GOTOXY(10,2);
WRITE('INTRODUZCA EL NOMBRE DEL
ARCHIVO');
GOTOXY(13,3);
READLN(N);
MOSTRAR;
P:= POSICION (N,F);
IF I = -1 THEN
BEGIN
GOTOXY(5,9);TEXTCOLOR(4);
WRITELN('NO EXISTE EL NOMBRE BUSCADO');
END
ELSE
BEGIN
SEEK(F,0);
READ(F,E);
IF E.SW THEN
BEGIN
GOTOXY(5,9);TEXTCOLOR(4);
VISUALIZAR (E);
MOSTRAR1;
S12:=-READKEY;
CASE S121 OF
#27:
BEGIN
CLOSE(F);
EXIT;
END;
'M',m':
BEGIN
COMENTARIO;
WRITE('UBICACION DEL ARCHIVO ':
E.ARCHIVO);
WINDOW(14,20,60,22);
TEXTCOLOR(0);
WRITE(E.COMENT);
END;
END;
END;
END;
S12:=-READKEY;
CONTINUAR;
WRITE(' CONSULTANDO...');
UNTIL UPCASE (READKEY) = 'N';
CLOSE (F)
END;

```

```

PROCEDURE BUSCAR(VAR F:FICHERO);
VAR
NOENCONTRADO,SAL23:BOOLEAN;
R1,R2,R3,R4,R5:CADENA2;
SS:CHAR;

```

**(20 PROCEDURE PARA CAPTURAR DATOS DE
BUSQUEDA INTERNO EN 19)**
PROCEDURE ENTRARDATOS;

```

BEGIN
WINDOW(40,7,77,22);
TextBackground(0);
ClrScr;
WINDOW(38,6,75,21);
TextBackground(6);
ClrScr;
TEXTCOLOR(15);
GOTOXY(4,3);
WRITE('INTRODUZCA LOS DATOS QUE CONOCE');
TEXTCOLOR(9);
GOTOXY(4,6);
WRITE('TIPO DE ARCHIVO : ');
READLN(R1);
GOTOXY(4,8);
WRITE('DE QUE PRESA : ');
READLN(R2);
GOTOXY(4,10);
WRITE('FECHA DEL SISMO : ');
READLN(R3);
GOTOXY(4,12);
WRITE('EN QUE SECCION : ');
READLN(R4);
GOTOXY(4,14);
WRITE('PARA QUE CUÑA : ');
READLN(R5);
END;

```

**(19 PROCEDIMIENTO PARA RASTREAR
INFORMACION A PARTIR DE CONOCER PARTE DE
LOS DATOS DECLARADOS**

¿21 DESPLIEGA LA INFORMACION REFERENTE A EL TIPO DE ARCHIVO Y SU UBICACION, SI ES QUE ESTE EXISTE INTERNO EN 19

PROCEDURE DESPLEGAR;

VAR

CONTENIDO: ARNOMBRE;

HAY1, HAY2, HAY3, HAY4, HAY5: BOOLEAN;

POS: INTEGER;

Q1: CHAR;

BEGIN

NOENCONTRADO:=TRUE;

POS:=1;

SEEK(F,0);

WHILE NOT EOF(F) DO

BEGIN

HAY1:=FALSE;

HAY2:=FALSE;

HAY3:=FALSE;

HAY4:=FALSE;

HAY5:=FALSE;

READ(F,CONTENIDO);

IF R1="" THEN HAY1:=TRUE

ELSE HAY1:=CONTENIDO.TIPO=R1;

IF HAY1 THEN

BEGIN

IF R2="" THEN HAY2:=TRUE

ELSE HAY2:=CONTENIDO.PRESA=R2;

END;

IF HAY2 THEN

BEGIN

IF R3="" THEN HAY3:=TRUE

ELSE HAY3:=CONTENIDO.SISMO=R3;

END;

IF HAY3 THEN

BEGIN

IF R4="" THEN HAY4:=TRUE

ELSE HAY4:=CONTENIDO.SECCION=R4;

END;

IF HAY4 THEN

BEGIN

IF R5="" THEN HAY5:=TRUE

ELSE HAY5:=CONTENIDO.CUNA=R5;

END;

IF HAY5 THEN

BEGIN

LIMPIAR;

HACIENDO;WRITE('BUSCANDO...');

MOSTRAR;

MOSTRAR;

VISUALIZAR(CONTENIDO);

NOENCONTRADO:=FALSE;

Q1:=READKEY;

CASE Q1 OF

'M', 'm':

BEGIN

COMENTARIO;

WRITE('UBICACION DEL ARCHIVO:');

CONTENIDO.ARCHIVO);

WINDOW(14,20,60,22);

TEXTCOLOR(0);

WRITE(CONTENIDO.COMENT);

Q1:=READKEY;

END;

#27:

BEGIN

CLOSE(F);

EXIT;

END;

END;

END;

END;

END;

(PROGRAMA PRINCIPAL DEL PROCEDIMIENTO

BUSCAR (19))

BEGIN

RESET(F);

SAL23:=FALSE;

REPEAT;

LIMPIAR;

ENTRARDATOS;

DESFLEGAR;

IF NOENCONTRADO THEN

BEGIN

MOSTRAR;

TEXTCOLOR(4+128);

GOTOXY(3,9);

WRITELN ('|| DATO(S) NO ENCONTRADO(S) ||');

READLN;

END;

LIMPIAR;

CONTINUAR;

WRITE(' BUSCANDO...');

SS:=READKEY;

IF SS='n' THEN SAL23:=TRUE;

IF SS='N' THEN SAL23:=TRUE;

UNTIL(SAL23);

CLOSE(F);

END;

(22 PROCEDIMIENTO QUE DESPLIEGA EL MENU DE OPCION DE TRABAJO)

PROCEDURE MENU (VAR F:FICHERO);

VAR

OPCION : CHAR;

BEGIN

REPEAT

LIMPIAR;

WINDOW(12,5,71,24);

TextBackground(0);

ClrScr;

WINDOW(10,5,69,23);

TextBackground(3);

ClrScr;

TEXTCOLOR(5);

FOR A:=2 TO 59 DO

BEGIN

GOTOXY(A,1); WRITE(█);

GOTOXY(A,19); WRITE(█);

END;

FOR A:=1 TO 19 DO

BEGIN

GOTOXY(2,A); WRITE(█);

GOTOXY(59,A); WRITE(█);

END;

TEXTCOLOR(11);

GOTOXY(5,19);

WRITE(' INSTITUTO DE INGENIERIA ');

GOTOXY(48,19);

WRITE(' U N A M ');

TEXTCOLOR(1);

GOTOXY (10,2);

WRITELN(' MENU PRINCIPAL ');

GOTOXY (10,4);

WRITELN ('(1) DEFINIR UN NUEVO ARCHIVO');

GOTOXY (10,6);

WRITELN ('(2) DAR DE BAJA UN ARCHIVO');

GOTOXY (10,8);

WRITELN ('(3) MODIFICAR EL CONTENIDO');

GOTOXY (10,10);

WRITELN ('(4) CONSULTAR SIGNIFICADOS');

GOTOXY (10,12);

WRITELN ('(5) BUSCAR UN ARCHIVO');

GOTOXY (10,14);

WRITELN ('(6) MOSTRAR TODO EL CONTENIDO');

TEXTCOLOR(4);

GOTOXY (10,16);

WRITELN ('(7) SALIR');

TEXTCOLOR(10+128);

GOTOXY (20,17);

WRITE('ELIJA UNA OPCION : ');

REPEAT

OPCION :=READKEY

UNTIL OPCION IN ['1..7'];

LIMPIAR;

TRABAJANDO;

CASE OPCION OF

1:AMPLIAR (F);

2:BORRAR (F);

3:MODIFICAR (F);

4:CONSULTAR (F);

5:BUSCAR(F);

6:LISTADOTOTAL (F);

END;

UNTIL OPCION = '7'

END;

(PROGRAMA PRINCIPAL DEL MODULO ADMINISTRATIVO)

BEGIN

TextBackground(2);

ClrScr;

TEXTCOLOR(1);

ASSIGN (ARCH, 'ARCHIVI.REG');

ACTIVAR (ARCH);

MENU (ARCH);

LIMPIAR;

END.

ANEXO V

MODULO DE INTERFASE

DESCRIPCION DE LOS PROCEDIMIENTO UTILIZADOS DENTRO DEL MODULO¹

En orden numérico:

- | | |
|-------------------------------------|--|
| 1 RESTAURAR | Procedimiento para desactivar cualquier ventana activa y limpiar la pantalla. |
| 2 ALERTA | Genera un sonido para avisar que existe algún error |
| 3 AVISAR | Activa ventana para desplegar información para el usuario. |
| 4 NOEXISTE | Avisa al usuario que el archivo que pidió no fue encontrado. |
| 5 NOCAMINO | Avisa al usuario que el directorio tecleado no fue encontrado. |
| 6 TRABAJANDO | Avisa al usuario que la computadora se Encuentra realizando algún cálculo. |
| 7 FORMATOINCO | Avisa al usuario que el archivo de donde esta leyendo datos no corresponde al formato de lectura que necesita el programa. |
| 8 ENTRAR_ARCHIVOS | Activa ventana para que el usuario teclee el nombre del archivo ya sea para leer o guardar datos. |
| 9 EDITOR_DEL_ARCHIVO_INTERSECCIONES | Permite crear o reeditar un archivo de propiedades |
| 10 VERIFICAR_PROP | Verifica si el archivo que se desea abrir existe, procedimiento interno en 9. |
| 11 LEER_INTER | Si se trata de un archivo nuevo se activa el procedimiento para capturar la nueva información. |

¹ LOS NOMBRES QUE IDENTIFICAN A LOS PROCEDIMIENTOS UTILIZADOS, EN LO GENERAL SON PALABRAS COMPLETAS O COMPUESTAS (TODAS EN ESPAÑOL) LO QUE HACE RELATIVAMENTE FACIL DIFERENCIAR LOS PROCEDIMIENTOS DE LOS COMANDOS PROPIOS DE TURBO PASCAL.

ANEXO V Módulo de Interfase

- 12 GUARDAR_PROP** Cuando se termina la captura o edición del archivo de propiedades el programa activa este procedimiento para guardar los nuevos datos en disco, procedimiento interno en 9.
- 13 CARGAR_PROP** Si el archivo que se desea abrir existe se activa este procedimiento que sirve para leer los datos contenidos en el archivo, procedimiento interno en 9.
- 14 DESPLEGAR_PROP** Cuando se a terminado de captar los datos del nuevo archivo o se ha cargado el archivo de disco el programa activa este procedimiento que sirve para visualizar en pantalla los datos del archivo, procedimiento interno en 9.
- 15 CARGAR_DISCO_TECLA** Procedimiento general para cargar las intersecciones ya sea de un archivo que ya existe o desde el teclado, interno en 9.
- 16 MODIFICAR** Procedimiento para modificar el archivo de intersecciones, interno en 9, interno en 16.
- 17 CAMBIARDOY** Procedimiento que permite modificar el lugar de la dovela, interno en 16.
- 18 SUPRIMIRDOY** Procedimiento para suprimir una dovela, interno en 16.
- 19 AMPLIARDOY** Procedimiento para declarar una dovela más, interno en 16.
- 20 MODIFICARLIN** Permite modificar las intersecciones de las líneas de frontera entre materiales y las líneas de seccionamiento vertical, interno en 16.
- 21 BORRARLIN** Permite suprimir una línea de frontera entre materiales, interno en 16.
- 22 AMPLIARLIN** Permite insertar una nueva línea de frontera entre materiales, interno en 16.
- 23 EDITOR_DEL_ARCHIVO_PROPIEDADES** Procedimiento general para construir el archivo de propiedades.
- 24 VERIFICAR_PROP** Procedimiento para detectar la existencia del archivo de propiedades tecleada por el usuario, interno en 23.
- 25 LEER_PROP** Procedimiento para capturar las propiedades de los materiales, interno en 23.
- 26 GUARDAR_PROP** Procedimiento para guardar el archivo de propiedades en disco, interno en 23.
- 7 CARGAR_PROP** Si el archivo existe entonces se activa este procedimiento que permite leer las propiedades de el archivo en disco, interno en 23.
- 28 DESPLEGAR_PROP** Después de cargar las propiedades ya sea desde disco o teclado se activa este procedimiento que permite visualizar las propiedades, interno en 23.
- 29 CARGAR_DISCO_TECLA** Procedimiento general para cargar las propiedades, interno en 23.

30 MODIFICAR	<i>Procedimiento general para modificar el archivo de propiedades en edición, interno en 23.</i>
31 CAMBIAR	<i>Procedimiento para modificar las propiedades de un estrato, interno en 30.</i>
32 SUPRIMIR	<i>Procedimiento para suprimir las propiedades de un material, interno en 30.</i>
33 AMPLIAR	<i>Procedimiento para insertar las propiedades de un nuevo estrato, interno en 23.</i>
34 CALCULAR_Y_EDITAR_LOS_ARCHIVOS_DE_ANSTA	<i>Procedimiento general para calcular los centroides, los coeficientes sísmicos y demás datos necesarios para crear el archivo de datos para el programa ANSTA.</i>
35 VERIFICAR_EXISTENCIA	<i>Procedimiento para verificar que existen los archivos señalados por el usuario, interno en 35.</i>
36 COEFICIENTES_SISMICOS	<i>Procedimiento para calcular los coeficientes sísmicos, interno en 35.</i>
37 CARGARDATOS	<i>Procedimiento para leer los datos de los archivos en disco, interno en 35.</i>
38 CALCULO CENTRO	<i>Procedimiento para calcular el centroide del elemento de suelo, interno en 35.</i>
39 CENTROIDE1	<i>Permite conocer el centroide del elemento superior (el elemento de suelo se divide en dos elementos triangulares, donde el elemento superior corresponde al triángulo superior), interno en 39.</i>
40 CENTROIDE2	<i>Procedimiento para calcular el centroide del elemento inferior, interno en 39.</i>
41 RECTA	<i>calcula en centroide de una recta, interno en 39.</i>
42 TRISUPER	<i>Determina el centroide cuando el elemento total de suelo es igual al elemento superior (según el procedimiento centroide1), interno en 39.</i>
43 TRIINFER	<i>Determina el centroide cuando el elemento total de suelo es igual al elemento inferior (según el procedimiento centroide2), interno en 39.</i>
44 COMPLETO	<i>Calcula el centroide del elemento de suelo cuando contiene los dos tipos de elementos en que se dividió (centroide1 y centroide2), interno en 39.</i>
45 NODOCERCA	<i>Procedimiento para determinar los nudos, de la malla de elementos finitos, mas cercanos.</i>
46 CUADRO1	<i>Procedimiento para determinar el nudo más cercano en el primer cuadrante (según un plano X-Y), interno en 45.</i>
47 CUADRO2	<i>Procedimiento para determinar el nudo más cercano en el segundo cuadrante (según</i>

- un plano X-Y), interno en 45.
- 48 CUADROS** Procedimiento para determinar el nudo más cercano en el tercer cuadrante (según un plano X-Y), interno en 45.
- 49 CUADRO4** Procedimiento para determinar el nudo más cercano en el cuarto cuadrante (según un plano X-Y), interno en 43.
- 50 IMPRICOEF** Procedimiento para guardar los coeficiente sísmicas en el archivo de datos de ANSTA, interno en 34.
- 51 COEFL_PROMEDIO** Procedimiento para conocer el coeficiente promedio actuante en la superficie de falla propuesita por el usuario, interno en 34.
- 52 LEERCIRCULO** Procedimiento para declarar la superficie de falla que se va a utilizar, interno en 32.
- 53 PROPIEDADES** Procedimiento para declarar el archivo de propiedades que va ser utilizado para consultar el archivo de datos del programa ANSTA, interno en 34.
- 54 EDITAR_ANSTA** Procedimiento general para editar el archivo de datos de ANSTA, interno en 34.
- 55 MATRIZ_COEFI** Procedimiento para introducir los coeficientes sísmicas al archivo de datos de ANSTA, interno en 34.
- 56 CIRC_FALLA** Procedimiento para introducir los datos referentes a la superficie de falla a utilizar, dentro del archivo de datos de ANSTA, interno en 34.
- 57 INTER_DOV_ESTR** Procedimiento para introducir las intersecciones de las líneas de frontera y las dovelas, dentro del archivo de datos de ANSTA, interno en 34.
- 58 GRAVAR_PROPIEDADES** Procedimiento para guardar el archivo de datos para el programa ANSTA en disco, interno en 34.
- 59 EDITAR_CONTROL_ANSTA_EXE** Procedimiento para editar el archivo controlador del programa ANSTA.
- 60 AVISO1** Aviso para el usuario sobre la no existencia del archivo de datos que desea declarar, interno en 39.
- 61 AVISO2** Aviso para el usuario sobre la existencia del archivo de resultados que desea declarar.
- 62 BORRARARCHIVO** Procedimiento para borrar archivos de resultados del programa ANSTA si así lo decide el usuario, interno en 39.
- 63 EXISTE** Procedimiento para verificar si existe el archivo que el usuario esta declarando, cualquier resultado es mostrada al usuario por medio de una ventana, interno en 39.

- 64 CONTROL** Procedimiento general para editar el archivo controlador del programa ANSTA, interno en 59.
- 65 PERDIDA_DE_BORDO_LIBRE** Procedimiento para calcular el borde libre perdido a partir de un archivo de resultados generado por el programa ANSTA.
- 66 LEER** Procedimiento para reconocer y leer el archivo de resultados de ANSTA, interno en 65.
- 67 LEER_DATOS_PANTALLA** Procedimiento para leer de pantalla los datos adicionales, interno en 65.
- 68 CORRECTFR** Procedimiento para corregir el factor de seguridad, interno en 65.
- 69 DATABASE** Procedimiento para establecer el factor de seguridad mínimo, interno en 65.
- 70 COREEP** Procedimiento para corregir el módulo de elasticidad, interno en 65.
- 71 CALPBL** Cálculo del borde libre perdido, para cada una de los factores de seguridad obtenidos con el programa ANSTA, interno en 65.
- 72 GUARDAR** Procedimiento para guardar el disco los resultados obtenidos, interno en 65.
- 73 CARGAR** Procedimiento utilizado para declarar el vector de palabras que se iluminara desde el menú principal.
- 74 DESPLEGAR** Procedimiento para desplegar las palabras en el menú principal.
- 75 ACTIVAR** Procedimiento para iluminar una palabra.
- 76 DESACTIVAR** Procedimiento para regresar a su color original la palabra iluminada.
- 77 ADELANTE** Procedimiento para iluminar la siguiente palabra.
- 78 REGRESA** Procedimiento para iluminar la palabra anterior.
- 79 INICIAR** Procedimiento para inicial el menú principal.
- 80 ACTIVIDAD** Procedimiento que relaciona las palabras del menú principal con las diferentes opciones con que cuenta el programa.
- 81 MENU2** Declaración del menú principal.

En orden alfabético:

75 ACTIVAR	<i>Procedimiento para iluminar una palabra.</i>
77 ADELANTE	<i>Procedimiento para iluminar la siguiente palabra.</i>
2 ALERTA	<i>Genera un sonido para visar que existe algún error</i>
33 AMPLIAR	<i>Procedimiento para insertar las propiedades de un nuevo estrato, interno en 23.</i>
19 AMPLIARDOV	<i>Procedimiento para declarar una davela más, interno en 16.</i>
22 AMPLIARLIN	<i>Permite insertar una nueva línea de frontera entre materiales, interno en 16.</i>
3 AVISAR	<i>Activa ventana para desplegar información para el usuario.</i>
60 AVISO1	<i>Aviso para el usuario sobre la no existencia del archivo de datos que desea declarar, interno en 59.</i>
61 AVISO2	<i>Aviso para el usuario sobre la existencia del archivo de resultados que desea declarar.</i>
62 BORRARARCHIVO	<i>Procedimiento para borrar archivos de resultados del programa ANSTA si así lo decide el usuario, interno en 59.</i>
21 BORRARLIN	<i>Permite suprimir una línea de frontera entre materiales, interno en 16.</i>
34 CALCULAR_Y_EDITAR_LOS_ARCHIVOS_DE_ANSTA	<i>Procedimiento general para calcular los centroides, los coeficientes sísmicos y demás datos necesarios para crear el archivo de datos para el programa ANSTA.</i>
38 CALCULOCENTRO	<i>Procedimiento para calcular el centroide del elemento de suelo, interno en 35.</i>
71 CALFBL	<i>Cálculo del bordo libre perdido, para cada una de los factores de seguridad obtenidos con el programa ANSTA, interno en 65.</i>
31 CAMBIAR	<i>Procedimiento para modificar las propiedades de un estrato, interno en 30.</i>
32 SUPRIMIR	<i>Procedimiento para suprimir las propiedades de un material, interno en 30.</i>
17 CAMBIARDOV	<i>Procedimiento que permite modificar el lugar de la davela, interno en 16.</i>
73 CARGAR	<i>Procedimiento utilizado para declarar el vector de palabras que se iluminara desde el menú principal.</i>
15 CARGAR_DISCO_TECLA	<i>Procedimiento general para cargar las intersecciones ya sea de un archivo que ya existe o desde el teclado, interno en 9.</i>

29 CARGAR_DISCO_TECLA	<i>Procedimiento general para cargar las propiedades, interno en 23.</i>
7 CARGAR_PROP	<i>Si el archivo existe entonces se activa este procedimiento que permite leer las propiedades de el archivo en disco, interno en 23.</i>
13 CARGAR_PROP	<i>Si el archivo que se desea abrir existe se activa este procedimiento que sirve para leer los datos contenidos en el archivo, procedimiento interno en 9.</i>
14 DESPLEGAR_PROP	<i>Cuando se a terminado de capturar los datos del nuevo archivo o se ha cargado el archivo de disco el programa activa este procedimiento que sirve para visualizar en pantalla los datos del archivo, procedimiento interno en 9.</i>
37 CARGARDATOS	<i>Procedimiento para leer los datos de los archivos en disco, interno en 35.</i>
39 CENTROIDE1	<i>Permite conocer el centroide del elemento superior (el elemento de suelo se divide en dos elementos triangulares, donde el elemento superior corresponde al triángulo superior), interno en 39.</i>
40 CENTROIDE2	<i>Procedimiento para calcular el centroide del elemento inferior, interno en 39.</i>
41 RECTA	<i>calcula en centroide de una recta, interno en 39.</i>
56 CIRC_FALLA	<i>Procedimiento para introducir los datos referentes a la superficie de falla a utilizar, dentro del archivo de datos de ANSTA, interno en 54.</i>
36 COEFICIENTES_SISMICOS	<i>Procedimiento para calcular los coeficientes sísmicos, interno en 35.</i>
44 COMPLETO	<i>Calcula el centroide del elemento de suelo cuando contiene los dos tipos de elementos en que se dividió (centroide1 y centroide2), interno en 39.</i>
64 CONTROL	<i>Procedimiento general para editar el archivo controlador del programa ANSTA, interno en 59.</i>
70 COREEP	<i>Procedimiento para corregir el módulo de elasticidad, interno en 65.</i>
68 CORRECPR	<i>Procedimiento para corregir el factor de seguridad, interno en 63.</i>
46 CUADRO1	<i>Procedimiento para determinar el nudo más cercano en el primer cuadrante (según un plano X-Y), interno en 45.</i>
47 CUADRO2	<i>Procedimiento para determinar el nudo más cercano en el segundo cuadrante (según un plano X-Y), interno en 45.</i>
48 CUADRO3	<i>Procedimiento para determinar el nudo más cercano en el tercer cuadrante (según un plano X-Y), interno en 45.</i>

ANEXO V Módulo de Interfase

- 49 CUADROS** Procedimiento para determinar el nudo más cercano en el cuarto cuadrante (según un plano X-Y), interno en 45.
- 69 DATABASE** Procedimiento para establecer el factor de seguridad mínima, interno en 65.
- 76 DESACTIVAR** Procedimiento para regresar a su color original la palabra iluminada.
- 74 DESPLEGAR** Procedimiento para desplegar las palabras en el menú principal.
- 28 DESPLEGAR_PROP** Después de cargar las propiedades ya sea desde disco o teclado se activa este procedimiento que permite visualizar las propiedades, interno en 23.
- 54 EDITAR_ANSTA** Procedimiento general para editar el archivo de datos de ANSTA, interno en 34.
- 59 EDITAR_CONTROL_ANSTA_EXE** Procedimiento para editar el archivo controlador del programa ANSTA.
- 23 EDITOR_DEL_ARCHIVO_PROPIEDADES** Procedimiento general para construir el archivo de propiedades.
- 63 EXISTE** Procedimiento para verificar si existe el archivo que el usuario está declarando, cualquier resultado es mostrado al usuario por medio de una ventana, interno en 59.
- 7 FORMATOINCO** Avisa al usuario que el archivo de donde está leyendo datos no corresponde al formato de lectura que necesita el programa.
- 8 ENTRAR_ARCHIVOS** Activa ventana para que el usuario teclee el nombre del archivo ya sea para leer o guardar datos.
- 9 EDITOR_DEL_ARCHIVO_INTERSECCIONES** Permite crear o reeditar un archivo de propiedades.
- 58 GRAVAR_PROPIEDADES** Procedimiento para guardar el archivo de datos para el programa ANSTA en disco, interno en 54.
- 72 GUARDAR** Procedimiento para guardar el disco los resultados obtenidos, interno en 65.
- 12 GUARDAR_PROP** Cuando se termina la captura o edición del archivo de propiedades el programa activa este procedimiento para guardar los nuevos datos en disco, procedimiento interno en 9.
- 26 GUARDAR_PROP** Procedimiento para guardar el archivo de propiedades en disco, interno en 23.
- 50 IMPRICOEF** Procedimiento para guardar los coeficiente sísmicos en el archivo de datos de ANSTA, interno en 34.
- 51 COEFI_PROMEDIO** Procedimiento para conocer el coeficiente promedio actuante en la superficie de falla propuesta por el usuario, interno en 34.
- 79 INICIAR** Procedimiento para inicial el menú principal.

80 ACTIVIDAD	<i>Procedimiento que relaciona las palabras del menú principal con las diferentes opciones con que cuenta el programa.</i>
57 INTER_DOV_ESTR	<i>Procedimiento para introducir las intersecciones de las líneas de frontera y las davelas, dentro del archivo de datos de ANSTA, interno en 54.</i>
66 LEER	<i>Procedimiento para reconocer y leer el archivo de resultados de ANSTA, interno en 65.</i>
67 LEER_DATOS_PANTALLA	<i>Procedimiento para leer de pantalla los datos adicionales, interno en 65.</i>
11 LEER_INTER	<i>Si se trata de un archivo nuevo se activa el procedimiento para capturar la nueva información.</i>
25 LEER_PROP	<i>Procedimiento para capturar las propiedades de los materiales, interno en 23.</i>
52 LEERCIRCULO	<i>Procedimiento para declarar la superficie de falla que se va a utilizar, interno en 52.</i>
55 MATRIZ_COEFI	<i>Procedimiento para introducir los coeficientes sísmicos al archivo de datos de ANSTA, interno en 54.</i>
81 MENU2	<i>Declaración del menú principal.</i>
16 MODIFICAR	<i>Procedimiento para modificar el archivo de intersecciones, interno en 9, interno en 16.</i>
30 MODIFICAR	<i>Procedimiento general para modificar el archivo de propiedades en edición, interno es 23.</i>
20 MODIFICARLIN	<i>Permite modificar las intersecciones de las líneas de frontera entre materiales y las líneas de seccionamiento vertical, interno en 16.</i>
5 NOCAMINO	<i>Avisa al usuario que el directorio tecleado no fue encontrado.</i>
45 NODOCERCA	<i>Procedimiento para determinar los nudos, de la malla de elementos finitos, mas cercanos.</i>
4 NOEXISTE	<i>Avisa al usuario que el archivo que pidió no fue encontrado.</i>
65 PERDIDA_DE_BORDO_LIBRE	<i>Procedimiento para calcular el bordo libre perdido a partir de un archivo de resultados generado por el programa ANSTA.</i>
53 PROPIEDADES	<i>Procedimiento para declarar el archivo de propiedades que va ser utilizado para construir el archivo de datos del programa ANSTA, interno en 34.</i>

ANEXO V Módulo de Interfase

78 REGRESA	<i>Procedimiento para iluminar la palabra anterior.</i>
1 RESTAURAR	<i>Procedimiento para desactivar cualquier ventana activa y limpiar la pantalla.</i>
18 SUPRIMIRDOY	<i>Procedimiento para suprimir una dovela, interno en 16.</i>
6 TRABAJANDO	<i>Avisa al usuario que la computadora se Encuentra realizando algún cálculo.</i>
43 TRIUNFER	<i>Determina el centroide cuando el elemento total de suelo es igual al elemento inferior (según el procedimiento centroide2), interno en 39.</i>
42 TRISUPER	<i>Determina el centroide cuando el elemento total de suelo es igual al elemento superior (según el procedimiento centroide1), interno en 39.</i>
35 VERIFICAR_EXISTENCIA	<i>Procedimiento para verificar que existen los archivos señalados por el usuario, interno en 35.</i>
10 VERIFICAR_PROP	<i>Verifica si el archivo que se desea abrir existe, procedimiento interno en 9.</i>
24 VERIFICAR_PROP	<i>Procedimiento para detectar la existencia del archivo de propiedades tecleada por el usuario, interno en 23.</i>

```

PROGRAM INTERFAS_CON_ANSTA;
USES CRT,DOS;
(DEFINICION DE LAS VARIABLES VALIDAS EN
TODO EL CUERPO DEL PROGRAMA)
VAR
PAL: ARRAY [1..6] OF STRING[45];
X,Y:ARRAY [1..6] OF INTEGER;
VAR LDURACION:INTEGER;

```

1 PROCEDURE RESTAURAR;

```

BEGIN
WINDOW(1,1,80,25);
TEXTBACKGROUND(7);
CLRSCR;
END;

```

2 PROCEDURE ALERTA;

```

VAR TIEMPO,TIEMPO1:INTEGER;
BEGIN
FOR TIEMPO1:=1 TO 10 DO
FOR TIEMPO:=1 TO 30 DO
BEGIN
SOUND(300+TIEMPO*10);
DELAY(10);
END;
FOR TIEMPO:=1 TO 200 DO
BEGIN
SOUND(3*0+TIEMPO);
DELAY(10);
END;
NOSOUND;
END;

```

3 PROCEDURE AVISAR;

```

VAR A:INTEGER;
BEGIN
WINDOW(26,9,57,14);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(24,7,55,13);
TEXTBACKGROUND(5);
CLRSCR;
TEXTCOLOR(10);
FOR A:=2 TO 30 DO
BEGIN
GOTOXY(A,1); WRITE('=');
GOTOXY(A,7);WRITE('=');
IF A<7 THEN
BEGIN
GOTOXY(1,A);WRITE(' ');
GOTOXY(31,A);WRITE(' ');
END;

```

```

END;
GOTOXY(1,1); WRITE('=');
GOTOXY(1,7);WRITE('=');
GOTOXY(31,1);WRITE(' ');
GOTOXY(31,7);WRITE(' ');
END;

```

4 PROCEDURE NOEXISTE;

```

BEGIN
AVISAR;
TEXTCOLOR(15);
GOTOXY(3,3);
WRITELN(' ARCHIVO NO ENCONTRADO !');
GOTOXY(3,4);
WRITELN(' DIRECTORIO O NOMBRE');
GOTOXY(6,5);
WRITE('INCORRECTO...');
ALERTA;
END;

```

5 PROCEDURE NOCAMINO;

```

BEGIN
AVISAR;
GOTOXY(6,5);
WRITE(' CAMINO INCORRECTO...');
ALERTA;
END;

```

6 PROCEDURE TRABAJANDO;

```
BEGIN
  AVISAR;
  TEXTCOLOR(14+128);
  GOTOXY(10,4);
  WRITE('TRABAJANDO... ');
END;
```

7 PROCEDURE FORMATOINCO;

```
BEGIN
  AVISAR;
  TEXTCOLOR(10+128);
  GOTOXY(3,4);
  WRITE('!!! FORMATO INCORRECTO !!! ');
  FOR DURACION:=1 TO 2 DO
    ALERTA;
  END;
```

8 PROCEDURE ENTRAR_ARCHIVOS;

```
BEGIN
  RESTAURAR;
  WINDOW(12,9,72,11);
  TEXTBACKGROUND(0);
  CLRSCR;
  WINDOW(10,8,70,10);
  TEXTBACKGROUND(1);
  TEXTCOLOR(12);
  CLRSCR;
END;
```

(CREA O REEDITA ARCHIVOS DE INTERSECCIONES)

9 PROCEDURE

```
EDITOR_DEL_ARCHIVO_INTERSECCIONES;
VAR
  PROP:TEXT;
  P:PATHSTR;
  EXISTE:BOOLEAN;
  ESTRATOS,DOVELAS:INTEGER;
  CORDX,SURF,GRIET:ARRAY [1..40] OF REAL;
  CORDY:ARRAY[1..40,1..40] OF REAL;
```

10 PROCEDURE VERIFICAR_PROP(var

```
existe:boolean);
VAR
  D: DirStr;
  N: NameStr;
  E: ExStr;
  BIENPROP:BOOLEAN;
  RESULTPROP:INTEGER;
BEGIN
```

```
BIENPROP:=FALSE;
RESTAURAR;
ENTRAR_ARCHIVOS;
TEXTCOLOR(15);
GOTOXY(2,2);
DELLINE;
WRITE('NOMBRE DEL ARCHIVO (INTER.DAT): ');
READLN(P);
FSP(LIT(P, D, N, E));
IF N="" THEN N:='INTER';
IF E="" THEN E:='.DAT';
P := D + N + E;
ASSIGN(PROP:P);
RESULTPROP:=0;
EXISTE:=FALSE;
(1-1)
RESET (PROP);
RESULTPROP := IORESULT;
(1+1)
WINDOW(22,9,72,15);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(20,8,70,14);
TEXTBACKGROUND(4);
CLRSCR;
TEXTCOLOR(11+128);
GOTOXY(4,4);
IF RESULTPROP=0
  THEN
  BEGIN
    WRITELN('...REEDITANDO ARCHIVO : ',P,
    ...!!');
    EXISTE:=TRUE;
  END
  ELSE
  BEGIN
    WRITELN('...CREANDO ARCHIVO : ',P, ...!!');
    EXISTE:=FALSE;
    REWRITE (PROP);
  END;
  CLOSE(PROP);
  DELAY(1500);
END;
```

(PROCEDIMIENTO PARA INTRODUCIR LAS CORDENADAS DESDE EL TECLADO CUENTA CON DETECTOR DE ERROR**)

11 PROCEDURE LEER_INTER(VAR I:INTEGEN;

```
VAR J:INTEGEN;
BEGIN
  WINDOW(20,4,62,22);
  TEXTBACKGROUND(0);
```



```

CLRSCR;
WINDOW(18,3,60,20);
TEXTBACKGROUND(3);
CLRSCR;
TEXTCOLOR(4);
WRITE(' DATOS EN LA DOVELA : ');
TEXTCOLOR(5);
|$|-1|
REPEAT
  GOTOXY(7,3);
  DELLINE;
  WRITE('INTERSECCION CON EL EJ: X : ');
  READLN(CORDX(I));
  UNTIL (IORESULT=0);
REPEAT
  GOTOXY(7,5);
  DELLINE;
  WRITE('COTA DE FLUJO : ');
  READLN(SURF(I));
  UNTIL (IORESULT=0);
REPEAT
  GOTOXY(7,7);
  DELLINE;
  WRITE('COTA GRIETA : ');
  READLN(GRIET(I));
  UNTIL (IORESULT=0);
  TEXTCOLOR(4);
  GOTOXY(4,10);
  WRITE('INTERSECCIONES CON LOS
        ESTRATOS');
  TEXTCOLOR(1);
  FOR J:=1 TO ESTRATOS DO
  BEGIN
    REPEAT
      GOTOXY(2,5);
      DELLINE;
      WRITE('INTERSECCION CON ESTRATO
            (',J,') = '); READLN(CORDY(I,J));
      UNTIL (IORESULT=0);
    END;
  |$|+1|
  END;

```

['**GRABA EN DISCO LAS PROPIEDADES**']

12 PROCEDURE GUARDAR_PROP;

```

VAR
  JIII: INTEGER;
BEGIN
  REWRITE(PROP);
  WRITELN(PROP,DOVELAS,' ESTRATOS);
  FOR III:=1 TO DOVELAS DO
  BEGIN

```

```

WRITE(PROP,CORDX[III]:3:3,' SURF[III]:3:2,'
      ',GRIET[III]:3:2);
  FOR J:=1 TO ESTRATOS DO
    WRITE(PROP,' CORDY[III,J]:3:2);
    WRITELN(PROP);
  END;
  CLOSE(PROP);
END;

```

['**LEE DE DISCO LAS PROPIEDADES**']

13 PROCEDURE CARGAR_PROP;

```

VAR
  IJ: INTEGER;
BEGIN
  RESET(PROP);
  READLN(PROP,DOVELAS,ESTRATOS);
  FOR I:=1 TO DOVELAS DO
  BEGIN;
    READ(PROP,CORDX(I),SURF(I),GRIET(I));
    BEGIN
      FOR J:=1 TO ESTRATOS DO
        READ(PROP, CORDY[I,J])
      END;
    END;
  END;
  CLOSE(PROP);
END;

```

['**DESPLIEGA EN PANTALLA LAS LINEAS ACTIVAS**']

14 PROCEDURE DESPLEGAR_PROP;

```

VAR
  LJ: INTEGER;
BEGIN
  WINDOW(1,3,80,24);
  TEXTBACKGROUND(7);
  CLRSCR;
  TEXTCOLOR(4);
  WRITELN('# de líneas entre ESTRATOS : 'ESTRATOS,'
          '# de líneas de secc. Vertical : 'DOVELAS);
  WRITELN('X', Línea Sup. de flujo, Cota de la
          Grieta, INTERSECCIONES ....);
  TEXTCOLOR(1);
  FOR I:=1 TO DOVELAS DO
  BEGIN
    GOTOXY(1,I+2);
    WRITE( CORDX[I]:3:3, ' ', SURF[I]:3:2, ' ',
          GRIET[I]:3:2);
    FOR J:=1 TO ESTRATOS DO
      WRITE(' ',CORDY[I,J]:3:2);WRITELN;
    END;
  END;

```

[PROCEDIMIENTO PARA EDITAR LAS PROPIEDADES]

15 PROCEDURE CARGAR_DISCO_TECLA;

```

VAR
L:INTEGER;
BEGIN
  VERIFICAR_PROP(EXISTE);
  RESET(PROP);
  IF NOT EXISTE
  THEN
  BEGIN
    CLRSCR;
    TEXTCOLOR(14);
    WRITELN('INICIANDO ARCHIVO : 'P, ' ');
    TEXTCOLOR(13);
    (S1-1)
    REPEAT
      GOTOXY(2,3);
      DELLINE;
      WRITE('LINEAS DE FRONTERA ENTRE
ESTRATOS : ');
      READLN(ESTRATOS);
      UNTIL (IORESULT=0);
      (S1+1)
    (S1-1)
    REPEAT
      GOTOXY(2,5);
      DELLINE;
      WRITE('LINEAS DE SECCIONAMIENTO
VERTICAL : ');
      READLN(DOVELAS);
      UNTIL (IORESULT=0);
      (S1+1)
      FOR I:=1 TO DOVELAS DO
        LEER_INTER(I);
      END
    ELSE
      BEGIN
        CARGAR_PROP;
        END;
      DESPLEGAR_PROP;
      END;
  
```

[EL SIGUIENTE PROCEDIMIENTO CONTIENE INTERNAMENTE LOS PROCEDIMIENTOS DEL 17 al 22]

16 PROCEDURE MODIFICAR;

```

VAR CAR:CHAR;
SALYA:BOOLEAN;
I3:INTEGER;

```

17 PROCEDURE CAMBIARDOV;

```

VAR LUG:INTEGER;
BEGIN
  (S1-1)
  REPEAT
    WRITE(' NUMERO DE DOVELA A MODIFICAR
: ');
    READLN(LUG);
    UNTIL (IORESULT=0);
    (S1+1)
    LEER_INTER(LUG);
  END;

```

18 PROCEDURE SUPR:MIRDOV;

```

VAR
LIN,I,J:INTEGER;
BEGIN
  (S1-1)
  REPEAT
    WRITE(' QUE DOVELA DESEA BORRAR ');
    READLN(LIN);
    UNTIL (IORESULT=0);
    (S1+1)
    FOR II:=LIN TO DOVELAS DO
      BEGIN
        CORDX[II]:=-CORDX[II+1];
        SURF[II]:=-SURF[II+1];
        GRIET[II]:=-GRIET[II+1];
        FOR J:=1 TO ESTRATOS DO
          CORDY[I,J]:=-CORDY[II+1,J];
        END;
        DOVELAS:=DOVELAS-1
      END;

```

19 PROCEDURE AMPLIARDOV;

```

VAR
SCORDX,SSURF,SGRIET:ARRAY [1..40] OF REAL;
SCORDY:ARRAY [1..40,1..40] OF REAL;
LJ,DOV:INTEGER;
BEGIN
  (S1-1)
  REPEAT
    WRITE(' QUE DOVELA VA A INSERTAR ');
    READLN(DOV);
    UNTIL (IORESULT=0);
    (S1+1)
    FOR I:=DOV TO DOVELAS DO
      BEGIN
        SCORDX[I]:=-CORDX[I];
        SSURF[I]:=-SURF[I];
        SGRIET[I]:=-GRIET[I];
        FOR J:=1 TO ESTRATOS DO
          SCORDY[I,J]:=-CORDY[I,J];
        END;

```

```

END;
LEER_INTER(DOV);
FOR I:=DOV TO DOVELAS DO
BEGIN
  CORDX[I+1]:=SCORDX[I];
  SURF[I+1]:=SSURF[I];
  GRIET[I+1]:=SGRIET[I];
  FOR J:=1 TO ESTRATOS DO
  CORDY[I+1,J]:=SCORDY[I,J];
END;
DOVELAS:=DOVELAS+1;
END;

20 PROCEDURE MODIFICARLIN;
VAR
  I,J,LIN:INTEGER;
BEGIN
  ($I-1)
  REPEAT
    WRITE(' QUE LINEA DE FRONTERA VA A
MODIFICAR ');
    READLN(LIN);
    UNTIL (IORESULT=0);
    ($I+1)
    FOR I:=1 TO DOVELAS DO
    BEGIN
      WRITE('INTERSECCION CON LA DOVELA ('I,
- ');
      READLN(CORDY[LLIN]);
      END;
      FOR I:=1 TO DOVELAS DO
      FOR J:=LIN TO ESTRATOS DO
      CORDY[I,J+1]:=SCORDY[I,J];
      ESTRATOS:=ESTRATOS+1;
      END;
    END;
  END;
END;

21 PROCEDURE BORRARLIN;
VAR
  I,J,LIN:INTEGER;
BEGIN
  ($I-1)
  REPEAT
    WRITE(' QUE LINEA DE FRONTERA VA A
ELIMINAR ');
    READLN(LIN);
    UNTIL (IORESULT=0);
    ($I+1)
    FOR I:=1 TO DOVELAS DO
    FOR J:=LIN TO ESTRATOS-1 DO
    CORDY[I,J]:=CORDY[I,J+1];
    ESTRATOS:=ESTRATOS-1;
    END;
  END;
END;

22 PROCEDURE AMPLIARLIN;
VAR
  SCORDY:ARRAY [1..40,1..40] OF REAL;
  LLIN:INTEGER;
BEGIN
  ($I-1)
  REPEAT
    WRITE(' QUE LINEA DE FRONTERA VA
INSERTAR ');
    READLN(LIN);
    UNTIL (IORESULT=0);
    ($I+1)
    FOR I:=1 TO DOVELAS DO
    FOR J:=LIN TO ESTRATOS DO
    SCORDY[I,J]:=CORDY[I,J];
    FOR I:=1 TO DOVELAS DO
    BEGIN
      WRITE('INTERSECCION CON LA DOVELA ('I,
- ');
      READLN(CORDY[LLIN]);
      END;
      FOR I:=1 TO DOVELAS DO
      FOR J:=LIN TO ESTRATOS DO
      CORDY[I,J+1]:=SCORDY[I,J];
      ESTRATOS:=ESTRATOS+1;
      END;
    END;
  END;
END;

(PROGRAMA PRINCIPAL DEL PROCEDIMIENTO
MODIFICAR (número 16) )
BEGIN
  $ALYA:=FALSE;
  REPEAT
    #WINDOW(1,1,80,2);
    TEXTBACKGROUND(1);
    CLRSCR;
    TEXTCOLOR(11);
    WRITELN('LINEA DE SECCIONAMIENTO
VERTICAL: 1) MODIFICAR 2) BORRAR 3)
AMPLIAR');
    WRITE ('LINEA DE FRONTERA ENTRE
ESTRATOS: 4) MODIFICAR 5) BORRAR 6)
AMPLIAR');
    WINDOW(1,25,80,25);
    TEXTBACKGROUND(3);
    CLRSCR;
    TEXTCOLOR(4);
    WRITE('Editando Archivo:','P. ...');
    GOTOXY(49,1);
    WRITE('CUALQUIER TECLA PARA TERNINAR');
    CAR:=READKEY;
    WRITELN;
    CASE CAR OF
      '1': CAMBIARDOV;
      '2': SUPRIMIRDOV;
      '3': AMPLIARDOV;
      '4': MODIFICARLIN;
    END;
  END;
END;

```

```

3: BORRARLN;
6: AMPLIARLN;
ELSE SALVA:=TRUE;
END;
DESPLEGAR_PROP;
UNTIL(SALVA);
END;

(****PROGRAMA PRINCIPAL DE *****)
(*EDITOR DEL ARCHIVO_PROPIEDADES;****)

BEGIN
TEXTBACKGROUND(3);
CLRSCR;
CARGAR_DISCO_TECLA;
MODIFICAR;
GUARDAR_PROP;
END;

(*****)
(ICREA O REEDITA *****)
(ARCHIVOS DE PROPIEDADES****)
(*****)

2 3 PROCEDURE
EDITOR_DEL_ARCHIVO_PROPIEDADES;
VAR
PROP:TEXT;
P:PATHSTR;
EXISTE:BOOLEAN;
NUMERAT:INTEGER;
JAJ:ARRAY [1..40] OF INTEGER;
CU,FRICT,WTOTAL,FRIO,VARA,CONFO,AK,EN:ARRAY
[1..40] OF REAL;

24 PROCEDURE VERIFICAR_PROP(var
existe:boolean);
VAR
D: DirStr;
N: NameStr;
E: ExtStr;
BIENPROP:BOOLEAN;
RESUL_PROP:INTEGER;
BEGIN
BIENPROP:=FALSE;
RESTAURAR;
ENTRAR_ARCHIVOS;
TEXTCOLOR(11);
GOTOXY(2,2);
DELLINE;
WRITE('NOMBRE DEL ARCHIVO (PROP.DAT):

);
READLN(P);
FSPLIT(P, D, N, E);
IF N="" THEN N:='PROP';
IF E="" THEN E:='.DAT';
P:=D+N+E;
ASSIGN(PROP,P);
RESUL_PROP:=0;
EXISTE:=FALSE;
(BI-1)
RESET (PROP);
RESUL_PROP := IORESULT;
RESUL_PROP := IORESULT;
(BI+1)
WINDOW(22,9,72,15);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(20,8,70,14);
TEXTBACKGROUND(4);
CLRSCR;
TEXTCOLOR(10+128);
GOTOXY(1,4);
IF RESUL_PROP=0
THEN
BEGIN
WRITELN('...||REEDITANDO EL ARCHIVO
|| : 'P);
EXISTE:=TRUE;
END
ELSE
BEGIN
WRITELN('...CREANDO EL ARCHIVO :
'P);
EXISTE:=FALSE;
REWRITE (PROP);
END;
CLOSE(PROP);
DELAY(1500);
END;

(***PARA INTRODUCIR LAS PROPIEDADES*)
(*DESDE EL TECLADO*****)
(***CUENTA CON DETECTOR DE ERROR****)

25 PROCEDURE LEER_PROP(VAR I:INTEGER);
BEGIN
RESTAURAR;
WINDOW(22,4,62,24);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(20,3,60,23);
TEXTBACKGROUND(3);
CLRSCR;

```

```

TEXTCOLOR(10);
WRITE('DATOS PARA EL MATERIAL : ',0);
TEXTCOLOR(4);
($I-1)
REPEAT
  GOTOXY(7,3);
  DELLINE;
  WRITE('FRICCION VARIABLE?1-SI,0-NO
:); READLN(JAJ[I]);
  UNTIL (IRESULT=0);
  REPEAT
    GOTOXY(7,5);
    DELLINE;
    WRITE('COHESION : '); READLN(CU[I]);
    UNTIL (IRESULT=0);
    REPEAT
      GOTOXY(7,7);
      DELLINE;
      WRITE('FRICCION : '); READLN(FRICT[I]);
      UNTIL (IRESULT=0);
    REPEAT
      GOTOXY(7,9);
      DELLINE;
      WRITE('W TOTAL
:); READLN(WTOTAL[I]);
      UNTIL (IRESULT=0);
      REPEAT
        GOTOXY(7,11);
        DELLINE;
        WRITE('FRIO : '); READLN(FRIO[I]);
        UNTIL (IRESULT=0);
        REPEAT
          GOTOXY(7,13);
          DELLINE;
          WRITE('VARA : '); READLN(VARA[I]);
          UNTIL (IRESULT=0);
          REPEAT
            GOTOXY(7,15);
            DELLINE;
            WRITE('CONFO : '); READLN(CONFO[I]);
            UNTIL (IRESULT=0);
            REPEAT
              GOTOXY(7,17);
              DELLINE;
              WRITE('AK (S.M.) : '); READLN(AK[I]);
              UNTIL (IRESULT=0);
            REPEAT
              GOTOXY(7,19);
              DELLINE;
              WRITE('EN (S.M.) : '); READLN(EN[I]);
              UNTIL (IRESULT=0);
            ($I+1)
          END;

```

(**GRABA EN DISCO LAS PROPIEDADES**)

26 PROCEDURE GUARDAR_PROP;

```

VAR
  III:INTEGER;
BEGIN
  REWRITE(PROP);
  WRITELN(PROP,NUMESTRAT);
  FOR III.=1 TO NUMESTRAT DO
    WRITELN(PROP,III,"JAJ[III]","CU[III]:3,2',
:FRICT[III]:3,2.' ;
      WTOTAL[III]:3,2.' 'FRIO[III]:3,2.'
:VARA[III]:3,2.' ;
      CONFO[III]:3,2.' 'AK[III]:3,2.'
:EN[III]:3,2);
  CLOSE(PROP);
END;

```

(**LEE DE DISCO LAS PROPIEDADES**)

27 PROCEDURE CARGAR_PROP;

```

VAR
  I:INTEGER;
BEGIN
  RESET(PROP);
  READLN(PROP,NUMESTRAT);
  FOR I.=1 TO NUMESTRAT DO
    READLN(PROP,I,JAJ[I],CU[I],FRICT[I],WTOTAL[I],FR
IO[I],VARA[I],
      CONFO[I],AK[I],EN[I]);
    CLOSE(PROP);
  END;

```

(**DESPLIEGA EN PANTALLA LAS LINEAS ACTIVAS**)

28 PROCEDURE DESPLEGAR_PROP;

```

VAR I:INTEGER;
BEGIN
  WINDOW(1,250,24);
  TEXTBACKGROUND(7);
  CLSCR;
  TEXTCOLOR(4);
  GOTOXY(20,2);
  WRITELN('EL NUMERO DE ESTRATOS
ACTIVOS ES : NUMESTRAT);
  WRITELN('J JAJ CU FRICT WTOTAL
FRIO VARA CONFO AK EN ');
  TEXTCOLOR(1);
  WRITELN;
  FOR I.=1 TO NUMESTRAT DO

```

```

WRITELN(I,'.JAJ[I]','.CU[I]:3.2','.FRICT[I]:3.2','.
WTOTAL[I]:3.2','.FRIO[I]:3.2','.VARA[I]:3.2','.
CONFO[I]:3.2','.AK[I]:3.2','.EN[I]:3.2','.');
END;

```

```

(****PROCEDIMIENTO PARA *****)
(EDITAR EL ARCHIVO DE PROPIEDADES**)

```

29 PROCEDURE CARGAR_DISCO_TECLA;

```

VAR
I:INTEGER;
BEGIN
VERIFICAR_PROP(EXISTE);
RESET(PROP);
IF NOT EXISTE
THEN
BEGIN
TEXTCOLOR(15);
GOTOXY(4,3);
WRITELN('INICIANDO ARCHIVO: '.P...');
(I-1)
REPEAT
GOTOXY(4,4);
DELLINE;
WRITE('ESTRATOS CONSIDERADOS : ');
READLN(NUMESTRAT);
UNTIL(IORESULT=0);
(I+1)
FOR I:=1 TO NUMESTRAT DO
LEER_PROP(I);
END
ELSE
BEGIN
CARGAR_PROP;
END;
DESPLEGAR_PROP;
END;

```

```

(****PROCEDIMIENTO PARA****)
(**MODIFICAR LAS PROPIEDADES*)

```

30 PROCEDURE MODIFICAR;

```

VAR CAR:CHAR;
SALYA:BOOLEAN;
I13:INTEGER;

```

31 PROCEDURE CAMBIAR;

```

VAR LUG:INTEGER;
BEGIN

```

```

(I-1)
REPEAT
WRITE(' NUMERO DE ESTRATO A MODIFICAR
);
READLN(LUG);
UNTIL (IORESULT=0);
(I+1)
LEER_PROP(LUG);
END;

```

32 PROCEDURE SUPRIMIR;

```

VAR
LIN,IL:INTEGER;
BEGIN
(I-1)
REPEAT
WRITE(' QUE LINEA DESEA BORRAR ');
READLN(LIN);
UNTIL (IORESULT=0);
(I+1)
FOR IL:=LIN TO NUMESTRAT DO
BEGIN
JAJ[IL]:=JAJ[IL+1];
CU[IL]:=CU[IL+1];
FRICT[IL]:=FRICT[IL+1];
WTOTAL[IL]:=WTOTAL[IL+1];
FRIO[IL]:=FRIO[IL+1];
VARA[IL]:=VARA[IL+1];
CONFO[IL]:=CONFO[IL+1];
AK[IL]:=AK[IL+1];
EN[IL]:=EN[IL+1];
END;
NUMESTRAT:=NUMESTRAT-I
END;

```

33 PROCEDURE AMPLLIAR;

```

VAR
SO:IAJ:ARRAY [1..40] OF INTEGER;
SOCU,SOFRICT,SOWTOTAL,SOFRIO,SOVARA,SOCOEN
FO,SOAK,SOEN:ARRAY [1..40] OF REAL;
LIN,I12:INTEGER;
BEGIN
(I-1)
REPEAT
WRITE(' QUE LINEA YA A INSERTAR ');
READLN(LIN);
UNTIL (IORESULT=0);
(I+1)
FOR I12:=LIN TO NUMESTRAT DO
BEGIN
SOJAJ[I12+1]:=JAJ[I12];
SOCU[I12+1]:=CU[I12];

```

```

SOFRIC[112+1]:=FRIC[112];
SOWTOTAL[112+1]:=WTOTAL[112];
SOFRIQ[112+1]:=FRIO[112];
SOVARA[112+1]:=VARA[112];
SOCONFO[112+1]:=CONFO[112];
SOAK[112+1]:=AK[112];
SOEN[112+1]:=EN[112];
END;
112:=LIN;
LEER_PROP(112);
NUMESTRAT:=NUMESTRAT+1;
FOR 112:=LIN+1 TO NUMESTRAT DO
  BEGIN
    JAJ[112] :=SOJAJ[112];
    CU[112] :=SOCU[112];
    FRIC[112] :=SOFRIC[112];
    WTOTAL[112]:=SOWTOTAL[112];
    FRIQ[112] :=SOFRIQ[112];
    VARA[112] :=SOVARA[112];
    CONFO[112] :=SOCONFO[112];
    AK[112] :=SOAK[112];
    EN[112] :=SOEN[112];
  END;
END;
END;
****PROGRAMA PRINCIPAL DEL****
!º PROCEDIMIENTO MODIFICAR**
BEGIN
  SALYA -F,11,SE;
  REPEAT
    WINDOW(1,1,80,1);
    TEXTBACKGROUND(1);
    CLRSKR;
    TEXTCOLOR(12);
    WRITE(' 1) MODIFICAR LINEA 2) BORRAR
  LINEA 3) INSERTAR LINEA');
    WINDOW(1,25,80,25);
    TEXTBACKGROUND(3);
    CLRSKR;
    TEXTCOLOR(4);
    WRITE('Modificando Archivo: ',P,' ');
    GOTOXY(50,1);
    WRITE('CU:ALQUER TECLA PARA TERNINAR');
    CAR:=READKEY;
    WRITELN;
    CASE CAR OF
      '1': CAMBIAR;
      '2': SUPRIMIR;
      '3': AMPLIAR;
    ELSE SALYA:=TRUE;
  END;
  DESPLEGAR_PROP;
  UNTIL(SALY.I);

```

```

END;
****PROGRAMA PRINCIPAL DE****
EDITOR_DEL_ARCHIVO_PROPIEDADES;****
BEGIN
  TEXTCOLOR(14);
  CARGAR_DISCO_TECLA;
  MODIFICAR;
  GUARDAR_PROP;
end;

```

```

*****
["CALCULA Y EDITA LOS ARCHIVO"]
DE DATOS DEL PROGRAMA ANSTA**
*****
3 4 PROCEDURE
CALCULAR_Y_EDITAR_LOS_ARCHIVOS_DE_ANSTA:

```

```

VAR
  S10,CENTXS10,CENTYS10:ARRAY[1..40,1..40]OF REAL;
  LNSTR,LNDOY;CONSTR,CONDOY:INTEGER;
  ACELPROM,CENTROX,CENTROY,OBLIGADOX,OBLIG
  ADOY:REAL;
  ****PARA ESTRATOS Y DOVELAS****
  SURFY,GRIETA,LINX:ARRAY[1..40]OF REAL;
  LINY:ARRAY[1..40,1..40]OF REAL;
  P:PATHSTR;
  RES:ARCHIVO.TEXT;
  JAJ:ARRAY[1..40]OF INTEGER;
  CU,FRIC,WTOTAL,FRIQ,VARA,CONFO,AK,EN:ARRAY
  [1..40]OF REAL;
  SALIRFIN:BOOLEAN;
  TERMINAR:CHAR;

```

```

35 PROCEDURE VERIFICAR_EXISTENCIA(VAR
EXISI:BOOLEAN);
BEGIN
  EXISI:=FALSE;
  $I-1;
  RESET(ARCHIVO);
  IF IORESULT=0 THEN BEGIN
    EXISI:=TRUE;CLOSE(ARCHIVO); END;
  $I+1;
  END;

```

```

*****FUNCIONES*****
["DISTANCIA MEDIA ENTRE DOS PUNTOS"]
FUNCTION MITAD(COR1,COR2:REAL):REAL;
BEGIN

```

```

MITAD:=-COR1+0.5*(COR2-COR1);
END;

F U N C T I O N
DISTANCIA(CORX1,CORY1,CORX2,CORY2:REAL):REAL;
BEGIN
DISTANCIA:=SQRT(SQR(CORX1-CORX2)+SQR(CORY1-
CORY2));
END;
(***)PENDIENTE ENTRE DOS PUNTOS***)
F U N C T I O N
PENDIENTE(CORX1,CORY1,CORX2,CORY2:REAL):RE
AL;
BEGIN
IF (CORX1<>CORX2)AND(CORY1<>CORY2) THEN:
PENDIENTE:=(CORY2-CORY1)/(CORX2-CORX1);
IF (CORX1=CORX2)AND(CORY1<>CORY2) THEN
PENDIENTE:=30000;
IF (CORX1<>CORX2)AND(CORY1=CORY2) THEN
PENDIENTE:=0;
END;
(***)Obtiene "C" para y=mx+c***)
FUNCTION COEF1(PEN,CORX,CORY:REAL):REAL;
BEGIN
COEF1:=-CORY-PEN*CORY;
END;
(***)encuentra el Punto de Intersección X entre dos
rectas***)
F U N C T I O N
INTERX(PEN1,COEF1,PEN2,COEF2:REAL):REAL;
BEGIN
INTERX:=(COEF2-COEF1)/(PEN1-PEN2);
END;
(***)encuentra el P1 de Intersección Y entre dos rectas***)
FUNCTION INTERY(PEN,CORX,COEF:REAL):REAL;
BEGIN
INTERY:=PEN*CORX+COEF;
END;
F U N C T I O N
INTERLINEAL(DIST1,DISINC,CELRI,CELRI2:REAL):RE
AL;
BEGIN
INTERLINEAL:=-CELRI+DISINC*(CELRI2-CELRI)/DIST1;
END;

F U N C T I O N
INTERPOLACION(CAX,CAY,CELA,CBX,CBY,CCLB,CDX
,CDY,CELD,CCX,CCY:REAL):REAL;
V A R
DIS1,DISDELTA,PENAB,PENDC,COFAB,COFDC,CMX,
CMY,CELM:REAL;
VAR DISA,DISB,DISD:REAL;
BEGIN
PENAB:=-PENDIENTE(CBX,CBY,CAX,CAY);

```

```

COFAB:=-COEF1(PENAB,CBX,CBY);
PENDC:=-PENDIENTE(CCX,CCY,CDX,CDY);
COFDC:=-COEF1(PENDC,CCX,CCY);
CMX:=-INTERX(PENAB,COFAB,PENDC,COFDC);
CMY:=-INTERX(PENAB,CMX,COFAB);
DIS1:=-DISTANCIA(CAX,CAY,CBX,CBY);
DISDELTA:=-DISTANCIA(CAX,CAY,CMX,CMY);
CELM:=-INTERLINEAL(DIS1,DISDELTA,CELA,CELB);
DISI:=-DISTANCIA(CDX,CDY,CMX,CMY);
DISDELTA:=-DISTANCIA(CDX,CDY,CCX,CCY);
DISDELTA:=-INTERLINEAL(DISI,DISDELTA,CELD,CE
LA);
INTERPOLACION:=-DISDELTA;
END;

```

(****INICIO DEL PROCEDIMIENTO PARA ACACULAR****)
 (****LA MATRIZ DE COEFICIENTES SISMICOS****)

36 PROCEDURE COEFICIENTES_SISMICOS;

```

VAR
(***)PARA MALLA DE ELEMENTOS FINITOS***)
NUMNODOS:INTEGER;
NUMIDEN:ARRAY[1..700] OF INTEGER;
NODX,NODY,NODACEL:ARRAY [1..700] OF REAL;
CENTX,CENTY:REAL;
VAR CONDO2,CONST2:INTEGER;

```

37 PROCEDURE CARGADATOS;

```

VAR
ACELDAT:TEXT;
CAD.STRING;
Ij:INTEGER;
SAL,EXISTE:BOOLEAN;
BEGIN
REPEAT
SAL:=TRUE;
REPEAT
ENTRAR_ARCHIVOS;
REPEAT
GOTOXY(2,2);
DELLINE;
WRITE(ARCHIVO DE ACCELERACIONES: );
READLN(CAD);
UNTIL(CAD<=>');
P:=CAD;
ASSIGN(ARCHIVO,P); (***)CARGA EL
VECTOR DE ACCELERACIONES)
VERIFICAR_EXISTENCIA(EXISTE);
IF NOT EXISTE THEN NOEXISTE;

```


ANEXO V Módulo de Interfase

```

UNTIL(EXISTE);
ASSIGN(ACEL,P);
RESET(ACEL);
{$I-1}
READ(ACEL,NUMNODOS);
WRITELN(RES,'NODOS : ',NUMNODOS);
WRITELN(RES);
WRITELN(RES,'NUMERO X Y
ACELERACION ');
FOR I:=1 TO NUMNODOS DO
BEGIN
READLN(ACEL,NUMIDEN[I],NODX[I],NODY[I],NODA
CEL[I]);
WRITELN(RES,NUMIDEN[I]:3,'
',NODX[I]:3.3,' ',NODY[I]:3.3,' ',NODACEL[I]:3.3);
END;
IF IORESULT<>0 THEN BEGIN
FORMATOINCO,SAL:=FALSE;END;
{$I+1}
UNTIL(SAL);
CLOSE(ACEL);
[***CARGA EL VECTOR DE DOVELAS
Y LA MATRIZ DE ESTRATOS]
REPEAT
SAL:=TRUE;
REPEAT
ENTRAR_ARCHIVOS;
REPEAT
GOTOXY(2,2);
DEL.LINE;
WRITE('ARCHIVO DE INTERSECCIONES :
);
READLN(CAD);
UNTIL(CAD<>'');
P:=CAD;
ASSIGN(ARCHIVO,P); [***CARGA EL
VECTOR DE ACELERACIONES]
VERIFICAR_EXISTENCIA(EXISTE);
IF NOT EXISTE THEN NOEXISTE;
UNTIL(EXISTE);
ASSIGN(DAT,P);
RESET(DAT);
{$I-1}
READ(DAT,LNDOV,LNSTR);
WRITELN(RES,'LINEAS DOVELA y ESTRATOS
: ',LNDOV,' ',LNSTR);
WRITELN(RES,'CORDENADA X y CORDENADAS
Y :);
FOR I:=1 TO LNDOV DO
BEGIN
READ(DAT,LINX[I],SURFY[I],GRIETA[I]);
FOR J:=1 TO LNSTR DO
READ(DAT,LINY[LJ]);

```

```

END;
FOR I:=1 TO LNDOV DO
BEGIN
WRITE(RES,LINX[I]:2.2,' ');
FOR J:=1 TO LNSTR DO
WRITE(RES,LINY[LJ]:2.2,' ');
WRITELN(RES);
END;
IF IORESULT<>0 THEN BEGIN
FORMATOINCO,SAL:=FALSE;END;
{$I+1}
UNTIL(SAL);
CLOSE(DAT);
END;

```

[***CALCULO DEL CENTROIDE DEL ELEMENTO***]

```

38 PROCEDURE
CALCULOCENTRO(CONDO2,CONST2:INTEGER);
VAR
XI,AY,BY,X2,CY,DY,CX1,CY1,CX2,CY2:REAL;

```

[***centroide del TRIANGULO 1***]

```

39 PROCEDURE
CENTROIDE1(X1,AY,X2,BY,DY:REAL);
VAR

```

```

M1Y,M2Y,MX,PENM1D,PENM2B,COEFM1,COEFM2:R
EAL;

```

```

BEGIN
M1Y:=MITAD(AY,BY);
M2Y:=MITAD(AY,DY);
MX:=MITAD(X1,X2);
PENM1D:=PENDIENTE(MX,M1Y,X2,DY);
PENM2B:=PENDIENTE(MX,M2Y,X2,BY);
COEFM1:=COEFI(PENM1D,X2,DY);
COEFM2:=COEFI(PENM2B,X2,BY);

```

```

CX1:=INTERX(PENM1D,COEFM1,PENM2B,COEFM2);
CY1:=INTERY(PENM1D,CX1,COEFM1);
END;

```

[***CENTROIDE DEL del TRIANGULO 2***]

```

40 PROCEDURE
CENTROIDE2(X1,AY,CY,X2,DY:REAL);
VAR

```

```

M2Y,M3Y,MX,PENAM3,PENCM2,COEFM2,COEFM3:R
EAL;
BEGIN
M3Y:=MITAD(CY,DY);
M2Y:=MITAD(AY,DY);

```

```

MX := MITAD(X1,X2);
PENAM3 := PENDIENTE(X1,AY,MX,M3Y);
PENCM2 := PENDIENTE(X1,CY,MX,M2Y);
COEFM2 := COEFI(PENCM2,X1,CY);
COEFM3 := COEFI(PENAM3,X1,AY);

CX2 := INTERY(PENAM3,COEFM3,PENCM2,COEFM2);
CY2 := INTERY(PENAM3,CX2,COEFM3);
END;
[***CENTROIDE DE UNA RECTA***]
41 PROCEDURE RECTA;
BEGIN
  CENTX := MITAD(X1,X2);
  CENTY := MITAD(AY,BY);
END;
[***CENTROIDE DEL TRIANGULO SUPERIOR***]
42 PROCEDURE TRISUPERA;
BEGIN
  CENTROIDEI(X1,AY,X2,BY,DY);
  WRITELN(RES, 'Cx = ' ,CX1:4:3, ' Cy = ' ,CY1:4:3);
  CENTX := CX1;
  CENTY := CY1;
END;
[***CENTROIDE DEL TRIANGULO INFERIOR***]
43 PROCEDURE TRIINFER;
BEGIN
  CENTROIDEI2(X1,AY,CY,X2,DY);
  CENTX := CX2;
  CENTY := CY2;
  WRITELN(RES, 'Cx = ' ,CX2:4:3, ' Cy = ' ,CY2:4:3);
END;
[***CENTROIDE DEL ELEMETO COMPUESTO***]
44 PROCEDURE COMPLETO;
VAR AREA1,AREA2:REAL;
BEGIN
  CENTROIDEI(X1,AY,X2,BY,DY);
  CENTROIDEI2(X1,AY,CY,X2,DY);
  AREA1 := 0.5*(X2-X1)*(BY-DY);
  AREA2 := 0.5*(X2-X1)*(AY-CY);
  WRITELN(RES, 'Cx = ' ,CX1:4:3, ' Cy = ' ,CY1:4:3);
  WRITELN(RES, 'Cx = ' ,CX2:4:3, ' Cy = ' ,CY2:4:3);
  CENTX := (CX1*AREA1 + CX2*AREA2)/(AREA1 + AREA2);
  CENTY := (CY1*AREA1 + CY2*AREA2)/(AREA1 + AREA2);
END;
[****PROGRAMA PRICIPAL DEL PROCEDIMIENTO

```

```

CALCULOCENTRO****]
BEGIN
  X1 := LINX(CONDO2);
  AY := LINY(CONDO2,CONST2);
  CY := LINY(CONDO2,CONST2+1);
  X2 := LINX(CONDO2+1);
  BY := LINY(CONDO2+1,CONST2);
  DY := LINY(CONDO2+1,CONST2+1);
  IF (AY=CJ)AND(BY=DJ) THEN RECTA;
  IF (AY<CJ)AND(BY>DJ) THEN TRISUPER;
  IF (AY>CJ)AND(BY<DJ) THEN TRIINFER;
  IF (AY<>CJ)AND(BY<>DJ) THEN COMPLETO;
  WRITELN(RES, 'CENTROIDEI',CONDO2,',',CONST2,',',CONST2,')
  Cx = ' ,CENTX:4:3, ' Cy = ' ,CENTY:4:3);
  CENTXS10(CONDO2,CONST2) := CENTX;
  CENTYS10(CONDO2,CONST2) := CENTY;
END;

[*****CALCULA LOS TRE NUDOS MAS
CERCANOS*****]

```

```

4 5 PROCEDURE
NODOCERCA(CONDO2,CONST2:INTEGER);
VAR
  LIM,SOPNOD:INTEGER;
  NODFPOS:ARRAY [1..4]OF INTEGER;
  VALR:ARRAY [1..4]OF REAL;
  VALRADIO,SOPRAD,INTERPOL,DISY,DISX:REAL;
  NO1,NO2,NO3:INTEGER;

```

```

46 PROCEDURE CUADRO1;
BEGIN
  IF VALR[1]>VALRADIO THEN
    BEGIN
      VALR[1] := VALRADIO;
      NODFPOS[1] := NUMIDEN[1];
    END;
  END;

```

```

47 PROCEDURE CUADRO2;
BEGIN
  IF VALR[2]>VALRADIO THEN
    BEGIN
      VALR[2] := VALRADIO;
      NODFPOS[2] := NUMIDEN[1];
    END;
  END;

```

```

48 PROCEDURE CUADRO3;
BEGIN

```

```

IF VALR[3]>VALRADIO THEN
  BEGIN;
  VALR[3]:=VALRADIO;
  NODPOS[3]:=NUMIDEN[1];
  END;
END;

49 PROCEDURE CUADRO4;
  BEGIN
  IF VALR[4]>VALRADIO THEN
    BEGIN;
    VALR[4]:=VALRADIO;
    NODPOS[4]:=NUMIDEN[1];
    END;
  END;

[***PROGRAMA PRINCIPAL DEL PROCEDIMIENTO
NUDOCERCANO****]
  BEGIN
  FOR I:=1 TO 4 DO
    BEGIN
    NODPOS[I]:=0;VALR[I]:=2000
    END;
  FOR I:=1 TO NUMNODOS DO
    BEGIN
    VALRADIO:=DISTANCIA(CENTX,CENTY,NODX[I],NODY[I]);
    DISX:=NODX[I]-CENTX;
    DISY:=NODY[I]-CENTY;
    IF (DISX>=0)AND(DISY>=0) THEN CUADRO1;
    IF (DISX<0)AND(DISY<=0) THEN CUADRO2;
    IF (DISX<=0)AND(DISY<=0) THEN CUADRO3;
    IF (DISX<0)AND(DISY<0) THEN CUADRO4;
    END;
  FOR I:=1 TO 4 DO
    BEGIN
    FOR IM:=(I+1) TO 4 DO
      BEGIN
      IF VALR[IM]<VALR[I] THEN
        BEGIN
        SOPRAD:=VALR[I];SOPNOD:=NODPOS[I];
        VALR[I]:=VALR[IM];NODPOS[I]:=NODPOS[IM];
        VALR[IM]:=SOPRAD;NODPOS[IM]:=SOPNOD;
        END;
      END;
    WRITE(RES,'NUDOS MAS CERCANOS : ');
    FOR I:=1 TO 4 DO
      BEGIN
        WRITE(RES,NODPOS[I],');
      END;
      WRITELN(RES);
    FOR I:=1 TO NUMNODOS DO
      BEGIN
      IF NODPOS[1]=NUMIDEN[1] THEN NO1:=1;
      IF NODPOS[2]=NUMIDEN[1] THEN NO2:=1;
      IF NODPOS[3]=NUMIDEN[1] THEN NO3:=1;
      END;
      WRITELN(RES,'DATOS DE LOS NUNDOS
      SELECCIONADOS: ');
      WRITELN(RES,'NUMERO X Y
      ACELERACION');
      WRITELN(RES,NUMIDEN[NO1],',',NODX[NO1]:3.3,
      ',NODY[NO1]:3.3,',NODACEL[NO1]:3.3);
      WRITELN(RES,NUMIDEN[NO2],',',NODX[NO2]:3.3,
      ',NODY[NO2]:3.3,',NODACEL[NO2]:3.3);
      WRITELN(RES,NUMIDEN[NO3],',',NODX[NO3]:3.3,
      ',NODY[NO3]:3.3,',NODACEL[NO3]:3.3);
      INTERPOL:=INTERPOLACION(NODX[NO1],NODY[NO1],
      NODX[NO2],NODY[NO2],NODACEL[NO2],
      NODX[NO3],NODY[NO3],NODACEL[NO3],
      CENTX,CENTY);
      WRITELN(RES,'INTERPOLACION =',INTERPOL:3.3);
      SIO[CONDO,CONST2]=INTERPOL;
      WRITELN(RES);
      END;
    [*****PROCEDIMIENTO PARA IMPRIMIR LAS
    ACELERACIONES INTERPOLADAS****]
    50 PROCEDURE IMPRICOEF;
      VAR CONDOI,CONST1:INTEGER;
      BEGIN
      FOR CONDOI:=1 TO (LNDOV-1) DO
        BEGIN
        FOR CONST1:=1 TO (LNSTR-1) DO
          WRITE(RES,SIO[CONDOI,CONST1]:3.3,');
          WRITELN(RES);
        END;
        WRITELN(RES);
      FOR CONDOI:=1 TO (LNDOV-1) DO
        BEGIN
        FOR CONST1:=1 TO (LNSTR-1) DO
          WRITE(RES,CENTXSIO[CONDOI,CONST1]:3.3,');
          WRITELN(RES);
        END;
        WRITELN(RES);
      FOR CONDOI:=1 TO (LNDOV-1) DO

```

ANEXO V Módulo de Interfase

```

BEGIN
FOR CONST1:=1 TO (LNSTR-1) DO
WRITE(RES,CENTYS10[CONDO1,CONST1]:3:3,',');
Writeln(RES);
END;
Writeln(RES);
END;

[****PROGRAMA PRINCIPAL DE
COEFICIENTES_SISMICOS****]

BEGIN
CARGADATOS;
TRABAJANDO;
FOR CONDO2:=1 TO (LNDOV-1) DO
FOR CONST2:=1 TO (LNSTR-1) DO
BEGIN
CALCULOCENTRO(CONDO2,CONST2);
NODOCERCA(CONDO2,CONST2);
END;
END;
IMPRICOEF;
END;
[*****FIN DEL PROCEDIMIENTO
COEFICIENTES_SISMICOS****]

[*****PROCEDIMIENTO PARA CALCULAR EL
COEFICIENTE PROMEDIO****]

51 PROCEDURE COEF_PROMEDIO;
VAR
RADMAXIMO,RADCENTRO:REAL;
SUMACOF,SUMANDOS:REAL;
CONDO3,CONST3:INTEGER;

52 PROCEDURE LEERCIRCULO;
BEGIN
RESTAURAR;
WINDOW(12,9,72,21);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(10,8,70,20);
TEXTBACKGROUND(1);
CLRSCR;
TEXTCOLOR(14);
($I-1)
REPEAT
GOTOXY(3,2);
DELLINE;
WRITE('CORDENADA X DEL CENTRO DEL
CIRCULO : ');
READ(CENTROX);
UNTIL(IORESULT=0);
REPEAT
GOTOXY(3,4);
DELLINE;
WRITE('CORDENADA Y DEL CENTRO DEL
CIRCULO : ');
READ(CENTROY);
UNTIL(IORESULT=0);
REPEAT
GOTOXY(3,6);
DELLINE;
WRITE('CORDENADA X DEL PUNTO OBLIGADO
: ');
READ(OBLIGADOX);
UNTIL(IORESULT=0);
REPEAT
GOTOXY(3,8);
DELLINE;
WRITE('CORDENADA Y DEL PUNTO OBLIGADO
: ');
READ(OBLIGADOY);
UNTIL(IORESULT=0);
($I+1)
END;

BEGIN
LEERCIRCULO;
SUMACOF:=0;
SUMANDOS:=0;

RADMAXIMO:=DISTANCIA(CENTROX,CENTROY,OBLI
GADOX,OBLIGADOY);
Writeln(RES,'EL RADIO MAXIMO ES
:',RADMAXIMO:3:3);
FOR CONDO3:=1 TO (LNDOV-1) DO
FOR CONST3:=1 TO (LNSTR-1) DO
BEGIN
RADCENTRO:=DISTANCIA(CENTROX,CENTROY,
CENTXS10[CONDO3,CONST3],CENTYS10[CONDO3,C
ONST3]);
IF RADCENTRO<=RADMAXIMO THEN
BEGIN
SUMACOF:=SUMACOF+S10[CONDO3,CONST3];
SUMANDOS:=SUMANDOS+1;
WRITE(RES,'SE SUMO EL COEFICIENTE
(',CONDO3,CONST3,')');
Writeln(RES,' SU VALOR ES : ');
S10[CONDO3,CONST3]:3:3);
END;
END;
Writeln(RES,' LA SUMA FINAL ES :
',SUMACOF:3:3);
Writeln(RES,' EL NUMERO DE SUMANDOS ES :
',SUMANDOS:2:0);
ACELPROM:=(SUMACOF/SUMANDOS);

```

```

WRITELN(RES,' EL PROMEDIO ES :
'ACELPROM:3:3);
END;

```

```

(*****FIN DEL PROCEDIMIENTO
COEFI_PROMEDIO*****)

```

```

(***PROCEDIMIENTO PARA DECLARAR
PROPIEDADES*****)

```

53 PROCEDURE PROPIEDADES;

```

VAR
PROP:TEXT;
EXISTE,SAL,SALI:BOOLEAN;
NUMPROP:INTEGER;
CAD:STRING;
BEGIN
REPEAT
SAL:=TRUE;
REPEAT
SALI:=TRUE;
REPEAT
ENTRAR_ARCHIVOS;
REPEAT
GOTOXY(2,2);
DELLINE;
WRITE('ARCHIVO DE PROPIEDADES : ');
READLN(CAD);
UNTIL(CAD<>' ');
P:=CAD;
ASSIGN(ARCHIVO,P);
IF ERIFICAR_EXISTENCIA(EXISTE);
IF NOT EXISTE THEN NOEXISTE;
UNTIL(EXISTE);
ASSIGN(PROP,P);
RESET(PROP);
($I-1)
READLN(PROP,NUMPROP);
IF (IORESULT <> 0) OR (NUMPROP <>
LNSTR-1) THEN
BEGIN
FORMATOINCO;
SALI:=FALSE;
END;
($I+1)
UNTIL(SALI);
WRITELN(RES,'EL NUMERO DE ESTRATOS ES
: ',NUMPROP);
WRITELN(RES);
WRITELN(RES,'X SURFY GRIETAY COTAS DE
Y... ');
($I-1)
FOR I:=1 TO NUMPROP DO

```

```

READLN(PROP,I,JA[I],CU[I],FRICT[I],WTOTAL[I],

```

```

FRIO[I],VARAI[I],CONFO[I],AK[I],EN[I]);
IF IORESULT<>0 THEN BEGIN
FORMATOINCO:SAL:=FALSE;END;
($I+1)
UNTIL(SALI);
CLOSE(PROP);
END;

```

```

(***FIN DEL PROCEDIMIENTO PARA DECLARAR
PROPIEDADES*****)

```

```

(*****EDICION DEL ARCHIVO PARA EL PROGRAMA
ANSTA******)

```

54 PROCEDURE EDITAR_ANSTA;

```

VAR
EDITAR:TEXT;
P1:PATHSTR;
SAL:BOOLEAN;
CAD:STRING;

```

55 PROCEDURE MATRIZ_COEFI;

```

VAR CONDO4,CONST4:INTEGER;
BEGIN
WRITELN(EDITAR,ACELPROM:1:3,'10,0.05,1');
WRITELN(EDITAR,0,0,'LNDOV. ','LNSTR',0,0,'ACEL
PROM:1:3,'0.64.2');
FOR CONDO4:=1 TO (LNDOV-1) DO
BEGIN
FOR CONST4:=1 TO (LNSTR-1) DO
WRITE(EDITAR,SIO(CONDO4,CONST4):1:3,',');
WRITELN(EDITAR);
FOR CONST4:=1 TO (LNSTR-1) DO
WRITE(EDITAR,0,');
WRITELN(EDITAR);
END;
END;

```

56 PROCEDURE CIRC_FALLA;

```

BEGIN
WRITELN(EDITAR,CENTROX:4:3,';'-1*CENTROY:4:3,'
,0');
WRITELN(EDITAR,OBLIGADOX:4:3,';'-1*OBLIGADYO
:4:3);
END;

```

57 PROCEDURE INTER_DOV_ESTR;

```

VAR LJ:INTEGER;
BEGIN
  FOR I:=1 TO LNDOV DO
  BEGIN
    WRITE(EDITAR,LINX[I]:3:3,'.-'-'*SURFY[I]:3:3.'
    '-I*GRIETA[I]:3:3.' );
    FOR J:=1 TO LNSTR DO
      WRITE(EDITAR,-1*LINY[I,J]:3:3,');
    WRITELN(EDITAR);
  END;
END;

```

58 PROCEDURE GRAVAR_PROPIEDADES;

```

VAR
  I:INTEGER;
BEGIN
  FOR I:=1 TO LNSTR-I DO
    WRITELN(EDITAR,I,'.JAJ[I].'.CU[I]:3:3.'
    'FRIC[I]:3:3.' '.WTOTAL[I]:3:3.' 'FRIO[I]:3:3.'
    'VARA[I]:3:3.' '.CONF0[I]:3:3.' 'AK[I]:3:3.'
    'EN[I]:3:3);
  END;

```

```

[*****PROGRAMA PRINCIPAL DE EDITOR DE
DATOS PARA ANSTA*****]

```

```

BEGIN
  READLN;
  REPEAT
    SAL:=TRUE;
    ENTRAR_ARCHIVOS;
  REPEAT
    GOTOXY(5,1);
    WRITE('archivo para alimentar ANTA.EXE');
    GOTOXY(2,2);
    DELLINE;
    WRITE('ARCHIVO PARA GUARDAR RESULTADOS
: ');
    READLN(CAD);
    UNTIL(CAD<>' ');
    P1:=CAD;
    ASSIGN(EDITAR,P1);
    [SI-I]
    REWRITE(EDITAR);
    IF IORRESULT<>0 THEN
      BEGIN

```

```

NOCAMINO;
SAL:=FALSE;
END;
UNTIL(SAL);
[SI+I]
MATRIZ_COEFI;
CIRC_FALLA;
INTER_DOV_ESTR;
GRAVAR_PROPIEDADES;
CLOSE(EDITAR);
END;

```

```

[*****PROCEDIMIENTO PRINCIPAL DEL
PROGRAMA *****]

```

```

BEGIN
  ASSIGN(RES,'RESPALDO.INT');
  REWRITE(RES);
  COEFICIENTES_SISMICOS;
  WRITE(W7);
  PROPIEDADES;
  REPEAT
    SALIRFIN:=TRUE;
    COEFI_PROMEDIO;
    EDITAR_ANSTA;
    TEXTCOLOR(15);
    CLRSCR;
    GOTOXY(2,2);
    WRITE('DESEA PREPAR OTRO ARCHIVO DE DATOS
PARA ANSTA.EXE (N), S ');
    WRITE(W7);DELAY(30);WRITE(W7);
    TERMINAR:=READKEY;
    IF (TERMINAR='S') OR (TERMINAR='3') THEN
      SALIRFIN:=FALSE;
    UNTIL(SALIRFIN);
    CLOSE(RES);
  END;

```

```

[*****]
[*****]
[** PROCEDIMIENTO PARA EDITAR ARCHIVO
CONTROLADOR DEL PROGRAMA ANTA.EXE**]
[*****]

```

59 PROCEDURE EDITAR_CONTROL_ANSTA_EXE;

```

VAR
  P:PATHSTR;

```

PF.TEXT:
CIERTO:BOOLEAN;
CONT.CHAR:

60 PROCEDURE AVISO1;

VAR

S:CHAR;

BEGIN

WINDOW(7,22,77,24);

TEXTBACKGROUND(0);

CLRSCL;

WINDOW(5,21,75,23);

TEXTBACKGROUND(1);

CLRSCL;

TEXTCOLOR(12+128);

WRITELN(' ! ARCHIVO NO EXISTE o

NOMBRE INCORRECTO !');

TEXTCOLOR(11);

WRITE('EL ARCHIVO PUEDE SER UTILIZADO EN LA
CORRIDA SI ANTES DE EJECURAR ANSTA');WRITE('SALE AL MS-DOS Y LO COPIA A ESTE
DICTECTORIO: (N), S ');

S:=READKEY;

IF S='S' THEN CIERTO:=TRUE;

WINDOW(1,21,80,25);

TEXTBACKGROUND(7);

CLRSCL;

END;

61 PROCEDURE AVISO2;

VAR

S:CHAR;

BEGIN

WINDOW(7,22,77,24);

TEXTBACKGROUND(0);

CLRSCL;

WINDOW(5,21,75,23);

TEXTBACKGROUND(3);

CLRSCL;

TEXTCOLOR(10+128);

WRITELN(' ! ARCHIVO DE RESULTADOS YA

EXISTE !');

TEXTCOLOR(9);

WRITELN('SI DESEA REUTILIZAR EL ARCHIVO PARA

GUARDAR RESULTADOS');

WRITE('OPRIMA "S", DE LO CONTRARIO

CUALQUIER TECLA ');

S:=READKEY;

IF (S='S')OR(S='s') THEN CIERTO:=FALSE;

WINDOW(1,21,80,25);

TEXTBACKGROUND(7);

CLRSCL;

END;

62 PROCEDURE BORRARARCHIVO;

BEGIN

ASSIGN(PF,P);

REWRITE(PF);

CLOSE(PF);

ERASE(PF);

END;

63 PROCEDURE EXISTE(VAR CIERTO:BOOLEAN);

BEGIN

ASSIGN(PF,P);

(\$I-1)

RESET(PF);

IF IORESULT<> 0 THEN CIERTO:=FALSE ELSE BEGIN

CIERTO:=TRUE;CLOSE(PF);END;

(\$I+1)

END;

64 PROCEDURE CONTROL;

VAR

L1,I,NUMERO,FRUNC,FRFIN,TAM:INTEGER;

F:TEXT;

NOMBRE1,NOMBRE2:STRING[12];

BEGIN

RESTAURAR;

WINDOW(17,6,67,15);

TEXTBACKGROUND(0);

CLRSCL;

WINDOW(15,5,65,14);

TEXTBACKGROUND(3);

CLRSCL;

TEXTCOLOR(4);

ASSIGN(F,'CONTROL.DAT');

REWRITE(F);

ARCHIVO CONTROLADOR DE

ANSTA.EXE');

(\$I-1)

REPEAT

GOTOXY(1,3);

CLREOL;

WRITE(' CUANTOS ARCHIVOS DE DATOS YA A

UTILIZAR : ');READ(NUMERO);

UNTIL(IORESULT=0);

GOTOXY(1,5);

WRITELN(' DATOS PARA VARIAR EL ANGULO DE

FRICCION');

REPEAT

GOTOXY(1,7);

CLREOL;

ANEXO V Módulo de Interfase

```
A:= 1/(4.65*A)-1/(1.34*A*A)+1/(1.16*A*A*A);
A:= H*A;
DELTA:= 2*A*(H/(BS+B));
END;
```

66 PROCEDURE LEER;

```
VAR
ARC:TEXT;
P:PATHSTR;
ESTA:BOOLEAN;
CAD1:STRING[18];
CAD2:STRING[1];
CAD3:STRING[15];
CAD4:STRING[2];
CAD5:STRING[30];
CODIGO:INTEGER;
I:INTEGER;

BEGIN
REPEAT
REPEAT
REPEAT
REPEAT
RESTAURAR;
ENTRAR_ARCHIVOS;
CLRSCR;
TEXTCOLOR(11);
GOTOXY(2,2);
DELLINE;
WRITE('ARCHIVO DE RESULTADOS DE
ANSTA.EXE : ');
READLN(P);
UNTIL(P<>' ');
ASSIGN(ARC,P);
{31-1}
RESET(ARC);
IF IORESULT<>0
THEN
BEGIN
ESTA:=FALSE;
AVISAR;
NOEXISTE;
END
ELSE
ESTA:=TRUE;
UNTIL(ESTA);
READLN(ARC,CAD1,CAD2,CAD3,CAD4);
WRITELN(RESP,CAD1,"",CAD2,"",CAD3,"",CAD4);
READLN(ARC,CAD5);
WRITELN(RESP,VALOR_MINIMO_DE_FRICCION:
'CAD5);
```

```
VAL(CAD2,INICIAL,CODIGO);
```

```
IF IORESULT<>0
```

```
THEN
```

```
BEGIN
```

```
ESTA:=FALSE;
```

```
AVISAR;
```

```
FORMATOINCO;
```

```
END
```

```
ELSE
```

```
ESTA:=TRUE;
```

```
UNTIL(ESTA);
```

```
VAL(CAD4,FINAL,CODIGO);
```

```
IF IORESULT<>0
```

```
THEN
```

```
BEGIN
```

```
ESTA:=FALSE;
```

```
AVISAR;
```

```
FORMATOINCO;
```

```
END
```

```
ELSE
```

```
ESTA:=TRUE;
```

```
UNTIL(ESTA);
```

```
WRITELN(RESP,INICIAL,'',FINAL);
```

```
CAMPO:=FINAL+1-INICIAL;
```

```
FOR I:=1 TO CAMPO DO
```

```
BEGIN
```

```
READLN(ARC,FC[I],EEP[I],TTP[I],ESFNP[I],ESNTP[I]);
```

```
WRITELN(RESP,INICIAL+I-1,'',FC[I]:5:3,'
```

```
'EEP[I]:5:3,'',TTP[I]:5:3,'',ESFNP[I]:5:3.'
```

```
'ESNTP[I]:5:3);
```

```
END;
```

```
CLOSE(ARC);
```

```
{31+1}
```

```
END;
```

67 PROCEDURE LEER_DATOS_PANTALLA;

```
BEGIN
```

```
{31-1}
```

```
WINDOW(22,9,62,15);
```

```
TEXTBACKGROUND(0);
```

```
CLRSCR;
```

```
WINDOW(20,8,60,14);
```

```
TEXTBACKGROUND(5);
```

```
CLRSCR;
```

```
TEXTCOLOR(14);
```

```
REPEAT
```

```
GOTOXY(4,2);
```

```
DELLINE;
```

```
WRITE('ALTURA DE LA CUÑA : ');
```

```
READLN(ALTURA);
```

```
UNTIL(IORESULT=0);
```

```
REPEAT
```

```
GOTOXY(4,4);
```

```

DELLENE;
WRITE('BASE SUPERIOR :');
READLN(BASESUPER);
UNTIL(IORRESULT=0);
REPEAT
  GOTOXY(4,6);
  DELLENE;
  WRITE('BASE INFERIOR :');
  READLN(BASEINFER);
  UNTIL(IORRESULT=0);
  ($I+1)
  WRITELN(RESPI,' ALTURA BASE SUP BASE INF');
  WRITELN(RESPI,ALTA:4:3,' ',BASESUPER:4:3,' ',
  'BASEINFER:4:3);
END;

```

68 PROCEDURE CORRECFR;

```

VAR I:INTEGER;
BEGIN
  FOR I:=1 TO CAMPO DO
  BEGIN
    FR[I]:=FREAL(FC[I]);
    TTP[I]:=ABS(TTP[I]);
  END;
END;

```

69 PROCEDURE DATBASE;

```

VAR I:INTEGER;
BEGIN
  FOR I:=1 TO CAMPO DO
  IF FR[I]>1.00001 THEN
  BEGIN
    LUGAR:=I;
    EXIT
  END;
END;

```

70 PROCEDURE COREEP;

```

AR
I:INTEGER;
BEGIN
  FOR I:=LUGAR TO CAMPO DO
  EEP[I]:=EEP(FC[TTP[LUGAR],EEP[LUGAR],TTP[I]]);
  FOR I:=1 TO CAMPO DO
  WRITELN(RESPI.INICIAL-1+I,' ',FC[I]:5:3,' ',
  'EEP[I]:3:3,' ',TTP[I]:5:3,' ',ESFNP[I]:3:3,' ',
  'ESNTP[I]:3:3);
  WRITELN(RESPI,'ANGULO MINIMO PARA CORRECCION
  DE EEP : ',INICIAL+LUGAR-1);
END;

```

71 PROCEDURE CALPBL;

```

VAR I:INTEGER;
BEGIN
  FOR I:=1 TO CAMPO DO
  BEGIN
    PBL[I]:=-DELTA(ALTA,BASESUPER,BASEINFER,FR
    [I],EEP[I],TTP[I]);
    WRITELN(RESPI,'CALCULANDO LA PBL ');
    WRITELN(RESPI,'PARA ',INICIAL+I-1,' ');
  END;
END;

```

72 PROCEDURE GUARDAR;

```

VAR R:TEXT;
P:PATHSTR;
ESTA:BOOLEAN;
I:INTEGER;
BEGIN
  REPEAT
    RESTAURAR;
    ENTRAR_ARCHIVOS;
    TEXTBACKGROUND(5);
    CLRSCR;
    TEXTCOLOR(11);
    GOTOXY(2,2);
    DELLENE;
    WRITE('ARCHIVO PARA GUARDAR RESULTADOS
    ');
    READLN(P);
    UNTIL(P<>'');
    ASSIGN(R,P);
    ($I-1)
    REWRITE(R);
    IF IORRESULT<>0
    THEN
      BEGIN
        ESTA:=FALSE;
        AVISAR;
        MOCAMINO;
        END
      ELSE
        ESTA:=TRUE;
    UNTIL(ESTA);
    ($I+1)
  FOR I:=1 TO CAMPO DO
  IF PBL[I]>20 THEN PBL[I]:=0;
  WRITELN(R,'FRIC PBL FR FC EEP

```

```

TTP');
FOR I:=1 TO CAMPO DO
WRITELN(R,INICIAL+I-1,'      ',PBL[I]:2:5,'
'FR[I]:1:3,'      ',FC[I]:1:3,'      ',EEP[I]:6:1,'
'TTP[I]:6:1);
CLOSE(R);
END;

BEGIN
RESTAURAR;
{B+I}
ASSIGN(RESF,'RESPALDO.PBL');
REPEAT
  REWRITE(RESF);
  LEER;
  LEER_DATOS_PANTALLA;
  CORR:CFR;
  DATOBASE;
  COREE:;
  CALPBL;
  GUARDAR;
  AVISAR;
  TEXTCOLOR(15);
  GOTOXY(4,4);
  WRITE('OTRO CALCULO : (N).S');
  SALIR:=READKEY;
  IF (SALIR='S')OR(SALIR='x') THEN
TERMINAR:=FALSE ELSE TERMINAR:=TRUE;
UNTIL(TERMINAR);
CLOSE(RESF);
{B+I}
END;

[*****]
[***** PROGRAMA QUE ACTIVA *****]
[**LOS PROCEDIMIENTOS GENERALES **]
[*****]

73 PROCEDURE CARGAR;
BEGIN
X[1]:=5:Y[1]:=-5;
PAL[1]:=' EDITAR ARCHIVO DE INTERSECCIONES ';
X[2]:=5:Y[2]:=-7;
PAL[2]:=' EDITAR ARCHIVO DE PROPIEDADES ';
X[3]:=5:Y[3]:=-9;
PAL[3]:=' EDITAR ARCHIVOS DE DATOS PARA
ANSTA.EXE ';
X[4]:=5:Y[4]:=-11;
PAL[4]:=' EDITAR ARCHIVO CONTROLADOR DE
ANSTA.EXE ';

```

```

X[5]:=5:Y[5]:=-13;
PAL[5]:=' CALCULAR PERDIDA DE BORDO LIBRE ';
X[6]:=5:Y[6]:=-15;
PAL[6]:=' SALIR ';
END;

74 PROCEDURE DESPLEGAR;
BEGIN
FOR I:=1 TO 6 DO
  BEGIN
  GOTOXY(X[I],Y[I]);WRITE(PAL[I])
  END;
END;

75 PROCEDURE ACTIVAR(I:INTEGER);
BEGIN
  TextBackground(3);
  TEXTCOLOR(11+128);
  GOTOXY(X[I],Y[I]);WRITE(PAL[I])
  END;

76 PROCEDURE DESACTIVAR(VAR I:INTEGER);
BEGIN
  TextBackground(1);
  TEXTCOLOR(12);
  GOTOXY(X[I],Y[I]);WRITE(PAL[I])
  END;

77 PROCEDURE ADELANTE(VAR I:INTEGER);
BEGIN
  DESACTIVAR(I);
  IF I=6 THEN I:=1 ELSE I:=I+1;
  ACTIVAR(I);
  END;

78 PROCEDURE REGRESA(VAR I:INTEGER);
BEGIN
  DESACTIVAR(I);
  IF I=1 THEN I:=6 ELSE I:=I-1;
  ACTIVAR(I);
  END;

79 PROCEDURE INICIAR;
BEGIN;
RESTAURAR;
TEXTBACKGROUND(7);
CLRSCR;
WINDOW(17,6,67,23);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(15,5,65,22);
TEXTBACKGROUND(1);

```

```

CLSCLR;
TEXTCOLOR(10);
WRITE('OPCIONES PARA INTERFAS CON
ANSTA.EXE');
TEXTCOLOR(12);
DESPLEGAR;
I:=1;
ACTIVAR(I);
END;

```

```

80 PROCEDURE ACTIVIDAD(VAR I:INTEGER);
BEGIN
CASE I OF
1:BEGIN
EDITOR_DEL_ARCHIVO_INTERSECCIONES;
INICIAR.END;
2:BEGIN
EDITOR_DEL_ARCHIVO_PROPIEDADES;
INICIAR.END;
3 : B E G I N
CALCULAR_Y_EDITAR_LOS_ARCHIVOS_DE_ANSTA;
INICIAR.END;
4:BEGIN EDITAR_CONTROL_ANSTA_EXE;
INICIAR.END;
5:BEGIN PERDIDA_DE_BORDO_LIBRE;
INICIAR.END;
END;
END;

```

81 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: REGRESA(I);
#13: IF I=6 THEN SALIR:=TRUE ELSE
ACTIVIDAD(I);
END;
UNTIL(SALIR);
END;

```

```

{***PROGRAMA PRINCIPAL DEL****}
{** MODULO INTERFASE*****}
BEGIN
CARGAR;

```

```

INICIAR;
MENU;
RESTAURAR;
TEXTCOLOR(1);
GOTOXY(5,4);
WRITELN(' FIN DEL PROGRAMA INTERFAS
CON ANSTA.EXE I ...');
GOTOXY(15,6);
WRITELN('...CREADO EN EL INSTITUTO DE
INGENIERIA UNAM 1993 ');
GOTOXY(15,10);
END.

```

ANEXO VI

MODULO EXPLICATIVO

DESCRIPCION DE LOS PROCEDIMIENTO UTILIZADOS DENTRO DEL MODULO¹

En orden numérico:

- | | |
|-------------|---|
| 1 RESTAURAR | Limpia la pantalla. |
| 2 CUADROS1 | Activa ventana para desplegar información. |
| 3 VENTANA1 | Activa ventana en la parte inferior de la pantalla. |
| 4 VENTANA2 | Activa ventana en la parte superior de la pantalla. |
| 5 MENSAJE2 | Despliega letrero de información para el usuario. |
| 6 GUAR | Procedimiento general para activar la búsqueda de información, contiene a los procedimientos del 6 al 14. |
| 7 LECTURA | Procedimiento general que permite iluminar una palabra a la vez de acuerdo con la opción utilizada por el usuario, contiene los procedimientos del 8 al 14. |
| 8 ACTIVAR | Declaración del vector de palabras parpadeantes, y de las posiciones en las que se desea se desplieguen, procedimiento interno de 8. |
| 9 INICIO | Inicializa la búsqueda de información de manera automática, procedimiento interno de 8. |
| 10 LEER | Despliega la información seleccionada por el usuario (cuando oprime enter sobre una palabra parpadeante, el programa despliega la información relacionada con dicha palabra), procedimiento interno de 8. |
| 11 ESCRIBIR | Despliega las palabras parpadeantes relacionadas con la información que actualmente se encuentra en pantalla, procedimiento interno de 8. |

¹ LOS NOMBRES QUE IDENTIFICAN A LOS PROCEDIMIENTOS UTILIZADOS, EN LO GENERAL SON PALABRAS COMPLETAS O COMPUESTAS (TODAS EN ESPAÑOL) LO QUE HACE RELATIVAMENTE FACIL DIFERENCIAR LOS PROCEDIMIENTOS DE LOS COMANDOS PROPIOS DE TURBO PASCAL.

- 12 ANTES** Procedimiento para iluminar la palabra parpadeante anterior, procedimiento interno de 8.
- 13 SIGUIENTE** Procedimiento para activar la palabra parpadeante siguiente, procedimiento interno de 8.
- 14 SELECCIONAR** Procedimiento que permite al usuario seleccionar la palabra, parpadeante de su interés procedimiento interno de 8.
- 15 MENSAJEI** Procedimiento que despliega información relacionada con las instrucciones para utilizar esta opción del módulo.
- 16 MANUAL** Procedimiento general que permite visualizar los manuales de los programas en Fortran PTLUSH Y ANSTA, contiene a los procedimientos de: 17 al 22.
- 17 MENSAJEI** Despliega información sobre las instrucciones para utilizar esta parte del módulo, procedimiento interno de 16.
- 18 MENSAJEJ** Informa al usuario que no existen más páginas en el manual activo en pantalla, procedimiento interno de 16.
- 19 LEER** Despliega página por página el manual de usuario activo en memoria, procedimiento interno de 16.
- 20 ANTES** Permite desplegar la página anterior a la que se encuentra en pantalla, procedimiento interno de 16.
- 21 SIGUIENTE** Permite desplegar la siguiente página del texto, procedimiento interno de 16.
- 22 SELECCIONAR** Permite al usuario decidir que página desea desplegar en pantalla, procedimiento interno de 16.
- 23 IMAGEN** Procedimiento general para desplegar la explicación gráfica del análisis dinámico, contiene los procedimientos del 24 al 44.
- 24 DETENER** Procedimiento que permite detener la ejecución del programa el tiempo suficiente para que el usuario lea la información desplegada, procedimiento interno de 16.
- 25 LEERCOTAS** Declaración de las vértices de la malla de elementos finitos, procedimiento interno de 16.
- 26 LEERLINEAS** Declaración de las líneas que se van a dibujar durante toda la explicación gráfica, procedimiento interno de 16.
- 26 DETECTAR** Procedimiento para determinar que tipo de monitor se va a utilizar durante la explicación gráfica, procedimiento interno de 16.
- 28 DIBUJARI** Procedimiento para desplazar la sección media de la presa, procedimiento interno de 16.
- 29 DIBUJAR** Procedimiento para dibujar líneas, procedimiento interno de 16.

ANEXO VI Módulo Explicativo

- 30 CONTORNO2** *Dibuja el contorno de la sección media aguas abajo, procedimiento interno de 16.*
- 31 MALLA2** *Dibuja la malla de elementos finitos correspondiente a la sección media aguas abajo, procedimiento interno de 16.*
- 32 MATERIALES2** *Dibuja los estratos contenidos en la sección media de la presa aguas abajo, procedimiento interno de 16.*
- 33 SUBMATERIALES2** *Dibuja los subestratos propuestos para la sección media, procedimiento interno de 16.*
- 34 CONTORNO** *Dibuja el contorno de la malla de elementos finitos utilizada para toda la presa, procedimiento interno de 16.*
- 35 CUERPO** *Dibuja el cuerpo de la presa, procedimiento interno de 16.*
- 36 MATERIALES** *Dibuja los límites que dividen el cambio de los materiales a lo largo de toda la presa, procedimiento interno de 16.*
- 37 ELEMFRENTE** *Dibuja los elementos finitos de la sección transversal de la cortina, procedimiento interno de 16.*
- 38 ELEMREBANADA** *Dibuja los elementos finitos utilizados para discretizar la presa sobre el eje Z, procedimiento interno de 16.*
- 39 ELEMSECCION** *Dibuja los diferentes elementos utilizados para dividir en secciones transversales a la presa, procedimiento interno de 16.*
- 40 CONTORNOMOY** *Dibuja el contorno de la presa para lograr animar el desplazamiento, procedimiento interno de 16.*
- 41 DOVELAS** *Procedimiento para dibujar las dovelas utilizadas durante el análisis, contiene al procedimiento #1, procedimiento interno de 16.*
- 42 ASIGNAR** *Procedimiento para declarar las posiciones donde se desea dibujar las dovelas, procedimiento interno de 16.*
- 43 ARCO1** *Procedimiento para dibujar el primer círculo de falla, procedimiento interno de 16.*
- 44 ARCOS** *Procedimiento para dibujar los círculos de falla complementarios para el análisis de pérdida de bordo libre, procedimiento interno de 16.*
- 45 SELECCION** *Procedimiento que permite al usuario seleccionar el tipo de explicación requerida.*
- 46 LEVANTA** *Procedimiento para activar las diferentes opciones que contiene el módulo.*

En orden alfabético:

- 8 ACTIVAR** *Declaración del vector de palabras parpadeantes, y de las posiciones en las que se desea se desplieguen, procedimiento interno de 8.*
- 12 ANTES** *Procedimiento para iluminar la palabra parpadeante anterior, procedimiento interno de 8.*
- 20 ANTES** *Permite desplegar la página anterior a la que se encuentra en pantalla, procedimiento interno de 16*
- 43 ARCOI** *Procedimiento para dibujar el primer círculo de falla, procedimiento interno de 16.*
- 44 ARCOS** *Procedimiento para dibujar los círculos de falla complementarios para el análisis de pérdida de bordo libre, procedimiento interno de 16.*
- 42 ASIGNAR** *Procedimiento para declarar las posiciones donde se desea dibujar las dovelas, procedimiento interno de 16.*
- 34 CONTORNO** *Dibuja el contorno de la malla de elementos finitos utilizada para toda la presa, procedimiento interno de 16.*
- 30 CONTORNO2** *Dibuja el contorno de la sección media aguas abajo, procedimiento interno de 16.*
- 40 CONTORNOMOV** *Dibuja el contorno de la presa para lograr animar el desplazamiento, procedimiento interno de 16.*
- 2 CUADROS1** *Activa ventana para desplegar información.*
- 35 CUERPO** *Dibuja el cuerpo de la presa, procedimiento interno de 16.*
- 26 DETECTAR** *Procedimiento para determinar que tipo de monitor se va a utilizar durante la explicación gráfica, procedimiento interno de 16.*
- 24 DETENER** *Procedimiento que permite detener la ejecución del programa el tiempo suficiente para que el usuario lea la información desplegada, procedimiento interno de 16.*
- 29 DIBUJAR** *Procedimiento para dibujar líneas, procedimiento interno de 16.*
- 28 DIBUJARI** *Procedimiento para desplazar la sección media de la presa, procedimiento interno de 16.*
- 41 DOVELAS** *Procedimiento para dibujar las dovelas utilizadas durante el análisis, contiene al procedimiento 41, procedimiento interno de 16.*
- 38 ELEMREBANADA** *Dibuja los elementos finitos utilizados para discretizar la presa sobre el eje Z, procedimiento interno de 16.*
- 39 ELEMSECCION** *Dibuja los diferentes elementos utilizados para dividir en secciones transversales a la presa, procedimiento interno de 16.*

- 11** **ESCRIBIR** *Despliega las palabras parpadeantes relacionadas con la información que actualmente se encuentra en pantalla, procedimiento interno de 8.*
- 6** **GUIAR** *Procedimiento general para activar la búsqueda de información, contiene a los procedimientos del 6 al 14.*
- 23** **IMAGEN** *Procedimiento general para desplegar la explicación gráfica del análisis dinámico, contiene los procedimientos del 24 al 44.*
- 9** **INICIO** *Inicializa la búsqueda de información de manera automática, procedimiento interno de 8.*
- 7** **LECTURA** *Procedimiento general que permite iluminar una palabra a la vez de acuerdo con la opción utilizada por el usuario, contiene los procedimientos del 8 al 14.*
- 19** **LEER** *Despliega página por página el manual de usuario activo en memoria, procedimiento interno de 16.*
- 10** **LEER** *Despliega la información seleccionada por el usuario (cuando oprime enter sobre una palabra parpadeante, el programa despliega la información relacionada con dicha palabra), procedimiento interno de 8.*
- 25** **LEERCOTAS** *Declaración de los vértices de la malla de elementos finitos, procedimiento interno de 16.*
- 26** **LEERlíneas** *Declaración de las líneas que se van a dibujar durante toda la explicación gráfica, procedimiento interno de 16.*
- 46** **LEVANTA** *Procedimiento para activar las diferentes opciones que contiene el módulo.*
- 31** **MALLA2** *Dibuja la malla de elementos finitos correspondiente a la sección media aguas abajo, procedimiento interno de 16.*
- 16** **MANUAL** *Procedimiento general que permite visualizar los manuales de los programas en Fortran PTLUSH Y ANSTA, contiene a los procedimientos del 17 al 22.*
- 36** **MATERIALES** *Dibuja los límites que dividen el cambio de los materiales a lo largo de toda la presa, procedimiento interno de 16.*
- 37** **ELEMFRENTE** *Dibuja los elementos finitos de la sección transversal de la cortina, procedimiento interno de 16.*
- 32** **MATERIALES2** *Dibuja los estratos contenidos en la sección media de la presa aguas abajo, procedimiento interno de 16.*
- 15** **MENSAJE1** *Procedimiento que despliega información relacionada con las instrucciones para utilizar este opción del módulo.*

ANEXO VI Módulo Explicativo

17 MENSAJE1	<i>Despliega información sobre las instrucciones para utilizar esta parte del módulo, procedimiento interno de 16.</i>
5 MENSAJE2	<i>Despliega letrero de información para el usuario.</i>
18 MENSAJE3	<i>Informa al usuario que no existen más páginas en el manual activo en pantalla, procedimiento interno de 16.</i>
1 RESTAURAR	<i>Limpia la pantalla.</i>
45 SELECCION	<i>Procedimiento que permite al usuario seleccionar el tipo de explicación requerida.</i>
22 SELECCIONAR	<i>Permite al usuario decidir que página desea desplegar en pantalla, procedimiento interno de 16.</i>
14 SELECCIONAR	<i>Procedimiento que permite al usuario seleccionar la palabra, parpadeante de su interés procedimiento interno de 8.</i>
21 SIGUIENTE	<i>Permite desplegar la siguiente página del texto, procedimiento interno de 16.</i>
13 SIGUIENTE	<i>Procedimiento para activar la palabra parpadeante siguiente, procedimiento interno de 8.</i>
33 SUBMATERIALES2	<i>Dibuja los substratos propuestos para la sección media, procedimiento interno de 16.</i>
3 VENTANA1	<i>Activa ventana en la parte inferior de la pantalla.</i>
4 VENTANA2	<i>Activa ventana en la parte superior de la pantalla.</i>

Listado del Programa:

```
USES CRT,GRAPH;
[DECLARACION DE LAS VARIABLES VALIDAS EN
TODO EL CUERPO DEL PROGRAMA]
```

```
VAR TERMINAR:BOOLEAN;
S:STRING;
F:TEXT;
ESPRESFIN:INTEGER;
FUERA,REGRESAR:BOOLEAN;
```

1 PROCEDURE RESTAURAR;

```
BEGIN
WINDOW(1,1,80,25);
TEXTBACKGROUND(5);
CLRSCR;
END;
```

2 PROCEDURE CUADROS1;

```
VAR A:INTEGER;
BEGIN
TEXTBACKGROUND(0);
CLRSCR;
TEXTCOLOR(12);
FOR A:=1 TO 79 DO
BEGIN
GOTOXY(A,1); WRITE(' ');
GOTOXY(A,24);WRITE(' ');
IF A<25 THEN
BEGIN
GOTOXY(1,A);WRITE(' ');
GOTOXY(80,A);WRITE(' ');
END;
END;
GOTOXY(1,1); WRITE(' ');
GOTOXY(1,24);WRITE(' ');
GOTOXY(80,1);WRITE(' ');
GOTOXY(80,24);WRITE(' ');
TEXTCOLOR(11);
GOTOXY(10,24);
WRITE(' INSTITUTO DE INGENIERIA ');
GOTOXY(60,24);
TEXTCOLOR(13+128);
WRITE(' U N A M ');
END;
```

3 PROCEDURE VENTANA1;

```
BEGIN
WINDOW(2,3,79,23);
TEXTBACKGROUND(1);
CLRSCR;
END;
```

4 PROCEDURE VENTANA2;

```
BEGIN
WINDOW(2,2,79,2);
TEXTBACKGROUND(3);
CLRSCR;
END;
```

5 PROCEDURE MENSAJE1;

```
BEGIN
TEXTCOLOR(10);GOTOXY(25,1);WRITE('CONFIRME
SALIDA ');
TEXTCOLOR(15+128);GOTOXY(45,1);
WRITE("(N) S");
GOTOXY(47,1);
END;
```

[CONTIENE LOS PROCEDIMIENTOS DEL 7 AL 14]

5 PROCEDURE GUIAR;**6 PROCEDURE LECTURA;**

```
[CONTIENE LOS PROCEDIMIENTOS DEL 8 AL 14]
[DECLARACION DE VARIABLES DE EN LOS
PROCEDIMIENTOS DEL 8 AL 14]
```

```
VAR
CH:STRING;
VECTOR,PRIN,FIN,I:INTEGER;
PAL :ARRAY[1..190] OF STRING[15];
COL :ARRAY[1..190] OF INTEGER;
TAM :ARRAY[1..190] OF INTEGER;
REN :ARRAY[1..190] OF INTEGER;
PPAL :ARRAY[1..190] OF INTEGER;
FPAL :ARRAY[1..190] OF INTEGER;
PLEC :ARRAY[1..190] OF INTEGER;
FLEC :ARRAY[1..190] OF INTEGER;
```

8 PROCEDURE ACTIVAR;

```
VAR ARC:TEXT;
PALABRA:STRING[15];
K:INTEGER;
BEGIN
Assign(ARC,'I.PAS');
RESET(ARC);
READLN(ARC,VECTOR);
FOR K:=1 TO VECTOR DO
BEGIN
READLN(ARC,PAL[K],TAM[K],REN[K],
COL[K],PPAL[K],FPAL[K],PLEC[K],FLEC[K]);
END;
CLOSE(ARC);
```

```
FOR K:=1 TO VECTOR DO
PAL[K]:=-COPY(PAL[K],I,TAM[K]);
END;
```

9 PROCEDURE INICIO;

```
BEGIN
PRIN:=-1;
FIN:=3;
ESPRI:=1;
ESFIN:=20;
END;
```

10 PROCEDURE LEER;

```
VAR
N:INTEGER;
CH:STRING;
BEGIN
CLASC;
TEXTCOLOR(14);
Assign(F,'AYUDA.PAS');
Reset(F);
FOR N:=1 TO 1000 DO
BEGIN
READLN(F,CH);
IF N>=ESPRI THEN
IF N<=ESFIN THEN WRITELN(CH)
ELSE
BEGIN
CLOSE(F);
EXIT;
END;
END;
END;
```

11 PROCEDURE ESCRIBIR;

```
VAR I:INTEGER;
BEGIN
ACTIVAR;
TEXTCOLOR(10);
FOR I:=PRIN TO FIN DO
BEGIN
GOTOXY(COL[I],REN[I]);WRITE(PAL[I]);
END;

TEXTCOLOR(12+144);GOTOXY(COL[PRIN],REN[PRIN
]);WRITE(PAL[PRIN]);
END;
```

12 PROCEDURE ANTES;

```
BEGIN
```

```
TEXTCOLOR(10);GOTOXY(COL[I],REN[I]);WRITELN(
PAL[I]);
IF I=PRIN THEN I:=FIN
ELSE I:=I-1;
```

```
TEXTCOLOR(12+144);GOTOXY(COL[I],REN[I]);WRIT
E(PAL[I]);
END;
```

13 PROCEDURE SIGUIENTE;

```
BEGIN
TEXTCOLOR(10);GOTOXY(COL[I],REN[I]);WRITE(PAL
[I]);
IF I=FIN THEN I:=PRIN
ELSE I:=I+1;

TEXTCOLOR(12+144);GOTOXY(COL[I],REN[I]);WRIT
E(PAL[I]);
END;
```

**14 PROCEDURE SELECCIONAR(VAR
REGRESAR:BOOLEAN);**

```
CONST
NULO =#0;
ENTER =#13;
ALARMA =#7;
ARRIBA =#72;
ABAJO =#80;
DERECHA =#77;
IZQUIERDA =#75;
CASA =#119;
SELECCION:SET OF CHAR=[NULO,ENTER,#27];
MOVER:SET OF CHAR
=[ARRIBA,IZQUIERDA,DERECHA,ABAJO,CASA];
VAR
TECLA:CHAR;
BEGIN
REGRESAR:=FALSE;
REPEAT
TECLA:=UPCASE(READKEY);
IF NOT (TECLA IN SELECCION) THEN WRITE
(ALARMA);
UNTIL (TECLA IN SELECCION);
CASE TECLA OF
NULO:
BEGIN
TECLA:=READKEY;
IF TECLA IN MOVER THEN
CASE TECLA OF
ARRIBA,IZQUIERDA:ANTES;
ABAJO,DERECHA:SIGUIENTE;
CASA:
```

```

BEGIN
PRIN:=1;
FIN:=3;
ESPRI:=0;
ESFIN:=20;
CLRSCR;
LEER;
ESCRIBIR;
I:=PRIN;
END;
END
ELSE WRITE(ALARMA)
END;

```

ENTER:

```

BEGIN
PRIN:=PPAL[I];
FIN:=FPAL[I];
ESPRI:=PLEC[I];
ESFIN:=FLEC[I];
CLRSCR;
LEER;
ESCRIBIR;
I:=PRIN;
END;

```

#27: REGRESAR:=TRUE

end;

```

END;
BEGIN
CLRSCR;
INICIO;
LEER;
ESCRIBIR;
I:=PRIN;
REPEAT
SELECCIONAR(REGRESAR);
UNTIL REGRESAR;
END;

```

15 PROCEDURE MENSAJE1;

```

BEGIN
TEXTCOLOR(0+144);WRITE('UTILICE:');
TEXTCOLOR(15);GOTOXY(10,1);
WRITE('mover: *24,*26,*25,*27. Seguir:
ENTER Principio: Ctrl+Home Salir: Esc');
END;

```

```

BEGIN
CUADROSI;
REPEAT

```

```

VENTANA2;
MENSAJE1;
VENTANA1;
TERMINAR:=FALSE;
LECTURA;
VENTANA2;
MENSAJE2;
S:=READKEY;
VENTANA1;
IF (S='S') OR (S='s') THEN TERMINAR:=TRUE;
UNTIL TERMINAR;
END;

```

16 PROCEDURE MANUAL;
[CONTIENE LOS PROCEDIMIENTOS DEL 17 AL 22]

17 PROCEDURE MENSAJE1;

```

BEGIN
TEXTCOLOR(0+144);WRITE('UTILICE:');
TEXTCOLOR(15);GOTOXY(10,1);
WRITE('Para desplegar: Up-Page y Down-Page
Principio: Ctrl+Home Salir: Esc');
END;

```

18 PROCEDURE MENSAJE3;

```

BEGIN
GOTOXY(20,10);
TEXTCOLOR(12+144);
WRITE('¡¡ FIN DE LECTURA !!');
END;

```

19 PROCEDURE LEER;

```

VAR
N:INTEGER;
CH:STRING;
BEGIN
CLRSCR;
TEXTCOLOR(14);

```

```

Reset(F);
FOR N:=1 TO 1000 DO
BEGIN
READLN(F,CH);
IF N>=ESPRI THEN
IF N<=ESFIN THEN WRITELN(CH)
ELSE
BEGIN
CLOSE(F);
EXIT;
END;

```

```

END;
IF EOF(F) THEN
BEGIN
CLOSE(F);
REGRESAR:=TRUE;
CLRSCR;
MENSAJE3;
EXIT;
END;
END;

```

20 PROCEDURE ANTES;

```

BEGIN
IF ESPRI=1 THEN
BEGIN
ESPRI:=ESPRI;
ESFIN:=ESFIN;
END
ELSE
BEGIN
ESPRI:=ESPRI+20;
ESFIN:=ESFIN+20;
END;
END;

```

21 PROCEDURE SIGUIENTE;

```

BEGIN
ESPRI:=ESPRI+20;
ESFIN:=ESFIN+20;
END;

```

22 PROCEDURE SELECCIONAR(VAR

```
REGRESAR:BOOLEAN);
```

```

CONST
NULO =#0;
ARRIBA =#73;
ABAJO =#81;
CASA =#119;
SELECCION.SET OF CHAR=[NULO,#27];
MOVER.SET OF CHAR =[ARRIBA,ABAJO,CASA];

```

```

VAR
TECLA:CIAR;
BEGIN
REPEAT
REGRESAR:=FALSE;
REPEAT
TECLA:=UPCASE(READKEY);
IF NOT (TECLA IN SELECCION) THEN WRITE
(#7):
UNTIL (TECLA IN SELECCION);
CASE TECLA OF

```

```

NULO:
BEGIN
TECLA:=READKEY;
IF TECLA IN MOVER THEN
CASE TECLA OF
#81:
BEGIN
CLRSCR;
SIGUIENTE;
LEER;
END;

```

```

#73:
BEGIN
CLRSCR;
ANTES;
LEER;
END;
#119:
BEGIN
ESPRI:=1;
ESFIN:=20;
CLRSCR;
LEER;
END;

```

```

END
ELSE WRITE(#7)
END;

```

```

#27: REGRESAR:=TRUE
END;
UNTIL(REGRESAR);
end;

```

```

BEGIN
CUADROSI;
REPEAT
VENTANAZ;
MENSAJE1;
VENTANA1;
TERMINAR:=FALSE;
ESPRI:=1;
ESFIN:=20;
LEER;
SELECCIONAR(REGRESAR);
VENTANAZ;
MENSAJE2;
S:=READKEY;
VENTANA1;
IF (S='S') OR (S='s') THEN TERMINAR:=TRUE ELSE

```

```

TERMINAR:=FALSE;
UNTIL TERMINAR;
END;

```

23 PROCEDURE IMAGEN;
{CONTIENE LOS PROCEDIMIENTOS DEL 24 AL 44}
{VARIABLES VALIDAS EN LOS PRECEDIMIENTOS
DEL 24 AL 44}

```

var
  Gd, Gm, L, PRN, FIN, D : Integer;
  X, Y, ARRAY [1..85] OF INTEGER;
  LP, LF: ARRAY [1..89] OF INTEGER;
  CON: CHAR;

```

24 PROCEDURE DETENER;

```

BEGIN
  DELAY(1700);
END;

```

25 PROCEDURE LEERCOTAS;

```

VAR
  FFI: TEXT;
  I: INTEGER;
BEGIN
  ASSIGN(FFI, 'up|fuente\COTAS');
  RESET(FFI);
  FOR I:=1 TO 86 DO
    READLN(FFI, X[I], Y[I]);
  CLOSE(FFI)
END;

```

26 PROCEDURE LEERlineas;

```

VAR
  FFI: TEXT;
  I: INTEGER;
BEGIN
  ASSIGN(FFI, 'up|fuente\lineas');
  RESET(FFI);
  FOR I:=1 TO 89 DO
    READLN(FFI, LP[I], LF[I]);
  CLOSE(FFI)
END;

```

26 PROCEDURE DETECTAR;

```

begin
  Gd := Detect; InitGraph(Gd, Gm, '');
  if GraphResult <> grOk then Halt(1);
  setibcolor(1);
  setcolor(14);
  line(638, 1, 638, 479);
  line(1, 1, 1, 479);

```

```

line(1, 479, 650, 479);
line(1, 1, 638, 1);
end;

```

28 PROCEDURE DIBUJARI;

```

VAR
  D, I: INTEGER;
BEGIN;
  FOR D:=-1 TO 50 DO
    BEGIN;
    SETCOLOR(14);
    FOR I:=-PRN TO FIN DO

```

```

LINE(X[LP[I]]+D, Y[LP[I]]-D, X[LF[I]]: D, Y[LF[I]]-D);
  DELAY(100);
  SETCOLOR(0);
  FOR I:=-PRN TO FIN DO

```

```

LINE(X[LP[I]]+D, Y[LP[I]]-D, X[LF[I]]+D, Y[LF[I]]-D);
  END;
  SETCOLOR(14);
  FOR I:=-PRN TO FIN DO

```

```

LINE(X[LP[I]]+D, Y[LP[I]]-D, X[LF[I]]+D, Y[LF[I]]-D);
  SETFILLSTYLE(2, 10);
  FLOODFILL(300, 200, 14);
  END;

```

29 PROCEDURE DIBUJAR;

```

VAR I: INTEGER;
BEGIN;
  FOR I:=-PRN TO FIN DO
    BEGIN

```

```

LINE(X[LP[I]]+D, Y[LP[I]]-D, X[LF[I]]+D, Y[LF[I]]-D);
  END;
END;

```

30 PROCEDURE CONTORNO2;

```

BEGIN PRN:=-1;
  FIN:=6;
  DIBUJAR;
END;

```

31 PROCEDURE MALLA2;

```

BEGIN PRN:=-7;
  FIN:=-18;
  DIBUJAR;
END;

```

32 PROCEDURE MATERIALES2;

```

BEGIN

```

PR:=-19;
FIN:=-20;
DIBUJAR;
END;

33 PROCEDURE SUBMATERIALES2;

BEGIN PRN:=-21;
FIN:=-30;
DIBUJAR;
END;

34 PROCEDURE CONTORNO;

BEGIN PRN:=-31;
FIN:=-39;
DIBUJAR;
END;

35 PROCEDURE CUERPO;

BEGIN PRN:=-40;
FIN:=-48;
DIBUJAR;
END;

36 PROCEDURE MATERIALES;

BEGIN
PRN:=-49;
FIN:=-56;
DIBUJAR;
END;

37 PROCEDURE ELEMFRENTE;

BEGIN
PRN:=-69;
FIN:=-77;
DIBUJAR;
END;

38 PROCEDURE ELEMREBANADA;

BEGIN
PRN:=-57;
FIN:=-68;
DIBUJAR;
END;

39 PROCEDURE ELEMSECCION;

BEGIN
PRN:=-78;
FIN:=-89;
DIBUJAR;
END;

40 PROCEDURE CONTORNOMOV;

BEGIN
PRN:=-31;
FIN:=-39;
DIBUJAR;
END;

41 PROCEDURE DOVELAS;
(PROCEDIMIENTO QUE CONTIENE AL
PROCEDIMIENTO 43)
(VARIABLES VALIDAS EN LOS PROCEDIMIENTOS
41 Y 42)

VAR DOV:ARRAY[1..20] OF INTEGER;

42 PROCEDURE ASIGNAR;

BEGIN
DOV[1]:=-40; DOV[5]:=-240;
DOV[9]:=-220;DOV[13]:=-380;
DOV[2]:=-60; DOV[6]:=-320;
DOV[10]:=-260;DOV[14]:=-400;
DOV[3]:=-80; DOV[7]:=-200; DOV[11]:=-300;
DOV[4]:=-120; DOV[8]:=-160;DOV[12]:=-360;
END;

VAR I:INTEGER;

BEGIN;
ASIGNAR;
DETENER;
OUTTEXTXY(100,420,'— OBLIGADAS: Donde existe
cambio en la dirección del estrato');

FOR I:=1 TO 14 DO

BEGIN

LINE(DOV[I],400,DOV[I],60);

DELAY(300);

IF I=7 THEN

Begin

DELAY(1500);

SETCOLOR(10);

OUTTEXTXY(100,440,'— OPCIONALES: Para
hacer más pequeños los elementos de suelo');

END;

END;

END;

43 PROCEDURE ARCO1;

BEGIN
SETCOLOR(11);
CONTORNO2;
ARC(200,80,180,270,120);
END;

44 PROCEDURE ARCOS;

BEGIN
DETENER;DETENER;
SETCOLOR(11);
ARC(240,80,180,297,180);
SETFILLSTYLE(5,4);
FLOODFILL(70,100,11);
DETENER;DETENER;


```

SETCOLOR(11);
ARC(280,120,178,301,240);
SETFILLSTYLE(1,1,12);
FLOODFILL(300,300,11);
DETENER,DETENER;
END;

BEGIN
LEERCOTAS:
LEERlineas:
D:=0;
DETECTAR;
SETCOLOR(14);
CUERPO;
CONTORNO;
DETENER;
SETFILLSTYLE(9,7);
FLOODFILL(2,2,14);
SETCOLOR(15);
OUTTEXTXY(100,340,'UNA PRESA DE TIERRA Y
ENROCAMIENTO SE COMPONE DE:');
DETENER,DETENER;
SETCOLOR(10);
OUTTEXTXY(140,360,'----- CORAZON DE
ARCILLA');
DETENER;
SETCOLOR(13);
MATERIALES:
LINE(180,320,320,320);
LINE(330,80,350,80);
SETFILLSTYLE(1,10);
FLOODFILL(245,300,13);
DETENER,DETENER;
SETCOLOR(11);
OUTTEXTXY(140,380,'----- FILTROS');
DETENER;
SETFILLSTYLE(2,11);
FLOODFILL(290,300,13);
FLOODFILL(190,300,13);
DETENER,DETENER;
SETCOLOR(12);
OUTTEXTXY(140,400,'----- ENROCAMIENTO');
DETENER;
SETCOLOR(15);
CONTORNO;
CUERPO;
MATERIALES:
SETFILLSTYLE(2,12);
FLOODFILL(140,300,15);
FLOODFILL(360,300,15);
SETFILLSTYLE(2,12);
FLOODFILL(420,200,15);
FLOODFILL(322,85,15);
FLOODFILL(253,179,15);

SETFILLSTYLE(5,12);
FLOODFILL(430,300,15);
FLOODFILL(400,160,15);
DETENER,DETENER,DETENER,DETENER;
SETFILLSTYLE(1,1);
FLOODFILL(1,1,1);
SETCOLOR(14);
DETECTAR;
CONTORNO;
CUERPO;
SETCOLOR(12);
OUTTEXTXY(130,390,'Al ser sometida a una
EXCITACION en la base');
OUTTEXTXY(130,410,'se comporta como un cuerpo
FLEXIBLE');
DETENER,DETENER,DETENER;
SETCOLOR(10);
OUTTEXTXY(100,340,'-----> ----->
----->');
SETCOLOR(11);
DETENER,DETENER;
OUTTEXTXY(100,340,'-----> ----->
----->');
OUTTEXTXY(130,390,'Al ser sometida a una
EXCITACION en la base');
OUTTEXTXY(130,410,'se comporta como un cuerpo
FLEXIBLE');
SETCOLOR(10);
OUTTEXTXY(90,350,'SE SUPONEN VALIDAS LAS
SIGUIENTES HIPOTESIS:');
OUTTEXTXY(120,370,'- La presa puede modelarse
como una malla de elementos finitos');
DETENER,DETENER;
SETCOLOR(13);
ELEMFRENTE;
ELEMREBANADA;
ELEMSECCION;
MATERIALES:
DETENER;
SETCOLOR(9);
OUTTEXTXY(120,390,'- La excitación se considera un
fenómeno aleatorio');
DETENER,DETENER;
SETCOLOR(11);
OUTTEXTXY(90,350,'SE SUPONEN VALIDAS LAS
SIGUIENTES HIPOTESIS:');
OUTTEXTXY(120,370,'- La presa puede modelarse
como una malla de elementos finitos');
OUTTEXTXY(120,390,'- La excitación se considera un
fenómeno aleatorio');
SETCOLOR(11);
OUTTEXTXY(100,350,'Utilizando el programa
PTLUSH es posible conocer');
OUTTEXTXY(100,370,'las aceleraciones máximas en

```

los Puntos Nodales');
 DETENER;DETENER;DETENER;DETENER;
 SETCOLOR(12);
 OUTTEXTXY(325,195,'<--- Pt. Nodal');
 DETENER;DETENER;
 SETCOLOR(10);
 OUTTEXTXY(20,400,'Conocidas las aceleraciones es posible efectuar un Análisis de Estabilidad');
 OUTTEXTXY(20,420,'de tipo Bishop Modificado en las Secciones de Frontera entre ');
 OUTTEXTXY(20,440,'Elementos Finitos tridimensionales (EJEMPLO EN LA SECCION MEDIA)');
 DETENER;DETENER;DETENER;DETENER;
 SETCOLOR(12);
 OUTTEXTXY(130,125,'Sec. MEDIA ---->');
 DETENER;DETENER;
 SETCOLOR(1);
 ELEMFRONTE;
 ELEMREBANADA;
 MATERIALES;
 OUTTEXTXY(325,195,'<--- Pt. Nodal');
 CONTORNO;
 SETCOLOR(15);
 CONTORNOMOI;
 D:=0;
 DETENER;DETENER;

 DETECTAR;
 SETBKOLOR(5);
 SETCOLOR(1);
 CONTORNO2;
 SETCOLOR(14);
 OUTTEXTXY(20,400,'El Análisis de Estabilidad sólo se realiza para el Talud Aguas Abajo');
 DETENER;DETENER;
 SETCOLOR(10);
 MALLA2;
 DETENER;
 OUTTEXTXY(20,420,'La reconstrucción Gráfica de la Sec. de Análisis se realiza a partir');
 OUTTEXTXY(20,440,'de las coordenadas utilizadas en la Malla de Elementos Finitos');
 DETENER;DETENER;DETENER;
 OUTTEXTXY(210,195,'<--- Pt. Nodal');
 DETENER;DETENER;
 DETECTAR;
 SETCOLOR(13);
 CONTORNO2;
 SETCOLOR(14);
 OUTTEXTXY(190,100,'Se procede a Estratificar la Sec. a analizar');
 DETENER;DETENER;
 MATERIALES2;
 DETENER;DETENER;

SETCOLOR(10);
 OUTTEXTXY(190,120,'Es posible definir substratos');
 DETENER;DETENER;
 SUBMATERIALES2;
 DETENER;DETENER;DETENER;DETENER;
 SETCOLOR(1);
 OUTTEXTXY(190,100,'Se procede a Estratificar la Sec. a analizar');
 OUTTEXTXY(190,120,'Es posible definir substratos');
 SETCOLOR(12);
 OUTTEXTXY(190,40,'A continuación se definen las Dovelas');
 DETENER;
 DOVELAS;
 DETENER;DETENER;DETENER;
 SETCOLOR(1);
 OUTTEXTXY(100,440,'<--- OPCIONALES: Para hacer más pequeños los elementos de suelo');
 OUTTEXTXY(100,420,'<--- OBLIGADAS: Donde existe cambio en la dirección del estrato');
 OUTTEXTXY(190,40,'A continuación se definen las Dovelas');
 SETCOLOR(15);
 OUTTEXTXY(10,30,'Conocidos los Elementos de Suelo se pasa a determinar sus centroides');
 DETENER;DETENER;DETENER;
 LINE(240,280,260,280);
 LINE(240,280,240,320);
 LINE(260,320,260,280);
 LINE(260,320,240,320);
 DETENER;
 SETFILLSTYLE(1,12);
 FLOODFILL(245,300,15);
 DETENER;
 OUTTEXTXY(250,300,'<--- ELEMENTO DE SUELO');
 DETENER;DETENER;
 SETCOLOR(1);
 OUTTEXTXY(250,300,'<--- ELEMENTO DE SUELO');
 SETCOLOR(14);
 OUTTEXTXY(246,300,'* <--- CENTROIDE DEL ELEMENTO DE SUELO');
 DETENER;DETENER;
 OUTTEXTXY(10,420,'Conocidos los Centroides de los elementos se interpola la aceleración');
 OUTTEXTXY(10,430,'a la que están sujetos utilizando los valores de');
 OUTTEXTXY(10,440,'las aceleraciones en los Pts. Nodales');
 DETENER;DETENER;DETENER;DETENER;
 SETCOLOR(1);
 OUTTEXTXY(10,30,'Conocidos los Elementos de Suelo se pasa a determinar sus centroides');

```

OUTTEXTXY(10,420,'Conocidos los Centroides de los
elementos se interpola la aceleración');
OUTTEXTXY(10,430,'a la que estan sujetos utilizando
los valores de');
OUTTEXTXY(10,440,' las aceleraciones en los Pto.
Nodales');
OUTTEXTXY(246,300,'* <— CENTROIDE DEL
ELEMENTO DE SUELO');
SETCOLOR(14);
OUTTEXTXY(10,20,'PARA CONOCER LA
ESTABILIDAD DEL TALUDE SE UTILIZA EL PROGRAMA
ANSTA');
DETENER;
ARCO;
DETENER;
OUTTEXTXY(10,40,'Finalmente se determina la PBL
para la Superficie de falla propuesta');
DETENER;DETENER;DETENER;DETENER;
SETCOLOR(1);
OUTTEXTXY(10,20,'PARA CONOCER LA
ESTABILIDAD DEL TALUDE SE UTILIZA EL PROGRAMA
ANSTA');
OUTTEXTXY(10,40,'Finalmente se determina la PBL
para la Superficie de falla propuesta');
SETCOLOR(12);
OUTTEXTXY(10,20,'PARA CONOCER LA PERDIDA
DE BORDO LIBRE TOTAL');
OUTTEXTXY(10,40,'se divide el terraplen en varias
rebanadas');
DETENER;DETENER;
SETFILLSTYLE(3,10);
FLOODFILL(120,140,11);
ARCO;
CLOSEGRAPH;
END;
    
```

45 PROCEDURE SELECCION;

```

BEGIN;
TEXTBACKGROUND(5);
CLRSCR;
WINDOW(24,6,61,21);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(23,5,60,20);
TEXTBACKGROUND(1);
CLRSCR;
TEXTCOLOR(14);
WRITE(' MODULO EXPLICATIVO);
TEXTCOLOR(11);
GOTOXY(3,5); WRITE('1) EXPLICACION
METODOLOGICA');
GOTOXY(3,7);WRITE('2) MANUAL DE ANALISIS
    
```

```

DINAMICO);
GOTOXY(3,9);WRITE('3) MANUAL DE ANALISIS
ESTABILIDAD');
GOTOXY(3,11);WRITE('4) EXPLICACION ANIMADA');
GOTOXY(3,13);WRITE('5) SALIR);
TEXTCOLOR(12);
GOTOXY(19,5);WRITE('M');
GOTOXY(26,7);WRITE('D');
GOTOXY(26,9);WRITE('E');
GOTOXY(19,11);WRITE('A');
GOTOXY(7,13);WRITE('S');
GOTOXY(3,5);WRITE('1');
GOTOXY(3,7);WRITE('2');
GOTOXY(3,9);WRITE('3');
GOTOXY(3,11);WRITE('4');
GOTOXY(3,13);WRITE('5');
textcolor(10);
GOTOXY(8,15); WRITE(' Elija una Opción.....');
END;
    
```

46 PROCEDURE LEVANTA;

```

CONST
MOVER: SET OF CHAR
='M', 'D', 'E', 'A', 'S', 'm', 'd', 'e', 'a', 's', '1', '2', '3', '4', '5';
VAR
TECLA:CHAR;
BEGIN
REPEAT
FUERA:=FALSE;
REPEAT
TECLA:=READKEY;
IF NOT (TECLA IN MOVER) THEN WRITE (#7);
UNTIL (TECLA IN MOVER);
CASE TECLA OF
'M', 'm', '1':BEGIN
Asign(F,'AYUDA.ASC');
GUIAR;
RESTAURAR;
SELECCION;
END;
'D', 'd', '2':BEGIN
Asign(F,'DINA.MAN');
MANUAL;
RESTAURAR;
SELECCION;
END;
'E', 'e', '3':BEGIN
Asign(F,'DINA.MAN');
MANUAL;
RESTAURAR;
SELECCION;
END;
    
```

```
'A','a','4':BEGIN
  IMAGEN;
  TEXTMODE(3);
  RESTAURAR;
  SELECCION;
  END;
'S','s','5': FUERA:=TRUE;
  END;
UNTIL(FUERA);
END;
```

**{PROGRAMA PRINCIPAL DEL MODULO
EXPLICATIVO}**

```
BEGIN
SELECCION;
LEVANTA;
RESTAURAR;
GOTOXY(10,10);
WRITE('FIN DEL MODULO EXPLICATIVO .....');
GOTOXY(1,15);
END.
```

ANEXO VII

MODULO PRINCIPAL

DESCRIPCION DE LOS PROCEDIMIENTOS UTILIZADOS DENTRO DEL MODULO¹

En orden numérico:

- | | |
|---------------|--|
| 1 RESTAURAR | Permite limpiar la pantalla. |
| 2 DESCONOCIDA | Mensaje sonoro que alerta al usuario cuando existe algún error en la ejecución de los módulos. |
| 3 AVISAR | Activa ventana para enviar un mensaje de estado al usuario. |
| 4 CARGAR | Procedimiento para declarar todos los letreros utilizados en este módulo. |
| 5 DESPLEGAR | Procedimiento que despliega las opciones del menú principal. |
| 6 ACTIVAR | Procedimiento que activa palabras parpadeantes. |
| 7 DESACTIVAR | Procedimiento que desactiva palabras parpadeantes. |
| 8 ADELANTE | Procedimiento que activa la palabra siguiente. |
| 9 REGRESA | Procedimiento que activa la palabra anterior. |
| 10 INICIAR | Procedimiento que activa la primer palabra parpadeante automáticamente. |
| 11 EXECUTAR | Procedimiento que ejecuta el módulo relacionado con la palabra parpadeante. |
| 12 MENU | Procedimiento para relacionar las palabras parpadeantes con los módulos con que cuenta el SE. |

¹ LOS NOMBRES QUE IDENTIFICAN A LOS PROCEDIMIENTOS UTILIZADOS, EN LO GENERAL SON PALABRAS COMPLETAS O COMPUESTAS (TODAS EN ESPAÑOL) LO QUE HACE RELATIVAMENTE FACIL DIFERENCIAR LOS PROCEDIMIENTOS DE LOS COMANDOS PROPIOS DE TURBO PASCAL.

En orden alfabético:

- | | |
|----------------------|--|
| 6 ACTIVAR | <i>Procedimiento que activa palabras parpadeantes.</i> |
| 8 ADELANTE | <i>Procedimiento que activa la palabra siguiente.</i> |
| 3 AVISAR | <i>Activa ventana para enviar un mensaje de estado al usuario.</i> |
| 4 CARGAR | <i>Procedimiento para declarar todos los letreros utilizados en este módulo.</i> |
| 7 DESACTIVAR | <i>Procedimiento que desactiva palabras parpadeantes.</i> |
| 2 DESCONOCIDA | <i>Casación que alerta al usuario cuando existe algún error en la ejecución de los módulos.</i> |
| 5 DESPLEGAR | <i>Procedimiento que despliega las opciones del menú principal.</i> |
| 11 EXECUTAR | <i>Procedimiento que ejecuta el módulo relacionado con la palabra parpadeante.</i> |
| 10 INICIAR | <i>Procedimiento que activa la primer palabra parpadeante automáticamente.</i> |
| 12 MENU | <i>Procedimiento para relacionar las palabras parpadeantes con los módulos con que cuenta el SE.</i> |
| 9 REGRESA | <i>Procedimiento que activa la palabra anterior.</i> |
| 1 RESTAURAR | <i>Permite limpiar la pantalla.</i> |

Listado del Programa:

```

($M $4000,0,0 )
(declaracion de la memoria a utilizar)
USES CRT,DOS;
(VARIABLES UTILIZADAS DURANTE TODO EL PROGRAMA)
VAR
PAL: ARRAY [1..5] OF STRING(43);
X,Y:ARRAY [1..5] OF INTEGER;
VAR LDURACION:INTEGER;
MODULO:ARRAY[1..4] OF STRING(12);
ACLARAR:ARRAY [1..18] OF STRING(25);
PROGRAMA:STRING;

```

1 PROCEDURE RESTAURAR;

```

BEGIN
WINDOW(1,1,80,25);
TEXTBACKGROUND(7);
CLRSCR;
END;

```

2 PROCEDURE DESCONOCIDA;

```

VAR I:INTEGER;
begin
FOR I:= 1 TO 5 DO
BEGIN
Sound(300);
Delay(280);
Sound(500);
Delay(230);
Sound(370);
Delay(230);
Sound(500);
Delay(230);
END;
NOSOUND;
END;

```

3 PROCEDURE AVISAR;

```

VAR A:INTEGER;
BEGIN
WINDOW(22,8,62,14);
TEXTBACKGROUND(0);
CLRSCR;
WINDOW(20,7,60,13);
TEXTBACKGROUND(5);
CLRSCR;
TEXTCOLOR(10);
FOR A:=2 TO 39 DO
BEGIN
GOTOXY(A,1); WRITE('=');
GOTOXY(A,7);WRITE('=');
IF A<7 THEN
BEGIN
GOTOXY(1,A);WRITE('1');
GOTOXY(40,A);WRITE('7');
END;
END;

```

```
GOTOXY(1,1); WRITE( " ");
GOTOXY(1,7);WRITE( " ");
GOTOXY(40,1);WRITE( " ");
GOTOXY(40,7);WRITE( " ");
```

END;

4 PROCEDURE CARGAR;

BEGIN

```
X[1]:=-3;Y[1]:=-5;
X[2]:=-3;Y[2]:=-7;
X[3]:=-3;Y[3]:=-9;
X[4]:=-3;Y[4]:=-11;
X[5]:=-3;Y[5]:=-13;
PAL[1]:=' MODULO EXPLICATIVO "MODEXP" ';
PAL[2]:=' MODULO ADMINISTRATIVO "MODAD" ';
PAL[3]:=' MODULO INTERFASCON ANSTA "MODINT" ';
PAL[4]:=' PROGRAMA DE ESTABILIDAD "ANSTA" ';
PAL[5]:=' SALIR ';
MODULO[1]:=' "MODEXP" ';
MODULO[2]:=' "MODAD" ';
MODULO[3]:=' "MODINT" ';
MODULO[4]:=' "ANSTA" ';
ACLARAR[1]:=' NO EXISTIO ERROR !..';
ACLARAR[2]:=' ARCHIVO NO ENCONTRADO !..';
ACLARAR[3]:=' CAMINO NO ENCONTRADO !..';
ACLARAR[4]:=' ACCESO PROHIBIDO !..';
ACLARAR[5]:=' MANIPULACION NO VALIDA !..';
ACLARAR[6]:=' FALTA DE MEMORIA !..';
ACLARAR[7]:=' ENTORNO NO VALIDO !..';
ACLARAR[8]:=' FORMATO NO VALIDO !..';
ACLARAR[9]:=' NO EXISTEN MAS ARCHIVOS !..';
```

END;

5 PROCEDURE DESPLEGAR;

BEGIN

FOR I:=1 TO 5 DO

BEGIN

GOTOXY(X[I],Y[I]);WRITE('PAL[I]

END;

END;

6 PROCEDURE ACTIVAR(I:INTEGER);

BEGIN

TextBackground(0);

GOTOXY(X[1],Y[1]);

textcolor(12+128);

write(#16,#16);

TEXTCOLOR(10);

WRITE(PAL[I]);

textcolor(12+128);

write(#17,#17);

END;

7 PROCEDURE DESACTIVAR(VAR I:INTEGER);

BEGIN

TextBackground(1);

TEXTCOLOR(12);


```
GOTOXY(X[I],Y[I]);WRITE(' .PAL[I], ' )
END;
```

8 PROCEDURE ADELANTE(VAR I:INTEGER);

```
BEGIN
  DESACTIVAR(I);
  IF I=5 THEN I:=1 ELSE I:=I+1;
  ACTIVAR(I);
END;
```

9 PROCEDURE REGRESA(VAR I:INTEGER);

```
BEGIN
  DESACTIVAR(I);
  IF I=1 THEN I:=5 ELSE I:=I-1;
  ACTIVAR(I);
END;
```

10 PROCEDURE INICIAR;

```
BEGIN;
  RESTAURAR;
  TEXTBACKGROUND(7);
  CLRSCR;
  WINDOW(17,5,67,20);
  TEXTBACKGROUND(0);
  CLRSCR;
  WINDOW(15,4,65,19);
  TEXTBACKGROUND(1);
  CLRSCR;
  TEXTCOLOR(10);
  WRITE(' ENTRADA A LOS DIFERENTES MODULOS. . ');
  TEXTCOLOR(12);
  DESPLEGAR;
  I:=1;
  ACTIVAR(I);
END;
```

11 PROCEDURE EXECUTAR(VAR I:INTEGER);

```
BEGIN
  CASE I OF
    1:PROGRAMA:='MODEXP.EXE';
    2:PROGRAMA:='MODAD.EXE';
    3:PROGRAMA:='MODINT.EXE';
    4:PROGRAMA:='ANSTA.EXE';
  END;
  AVISAR;
  GOTOXY(3,3);
  TEXTCOLOR(14);
  WRITE(' ACTIVANDO EL MODULO: ');
  TEXTCOLOR(11);
  GOTOXY(25,4);
  WRITELN(MODULO[I]);
  GOTOXY(12,6);
  TEXTCOLOR(12);
  WRITE('i espere un momento!');
  TEXTCOLOR(10+144);
  WRITE(' ... ');
  DELAY(2500);
```

```

Exec("PROGRAMA.");
RESTAURAR;
AVISAR;
GOTOXY(3,3);
TEXTCOLOR(14);
WRITE("... REGRESANDO DEL MODULO. ");
TEXTCOLOR(11);
WRITELN(MODULO(I));
GOTOXY(8,5);
TEXTCOLOR(10+144);
if DotError <> 0 then
  BEGIN
    WRITE("      ii PROBLEMA !!");
    TEXTCOLOR(11);
    GOTOXY(10,6);
    WRITE(ACLARAR(I));
    DESCONOCIDA;
  END
else
  BEGIN
    WRITE("ii TERMINACION EXITOSA !!");
    DELAY(2500);
  END;
INICIAR;
end;

```

12 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: REGRESA(I);
      #13: IF I=5 THEN SALIR:=TRUE ELSE EXECUTAR(I);
    END;
  UNTIL(SALIR);
END;

```

(PROGRAMA PRINCIPAL DEL MODULO)

```

BEGIN
  CARGAR;
  INICIAR;
  MENU;
  RESTAURAR;
  TEXTCOLOR(1);
  GOTOXY(5,4);
  WRITELN("i FIN DEL PROGRAMA QUE LIGA LOS DIFERENTES MODULOS !.. ");
  GOTOXY(15,6);
  WRITELN("i...CREADO EN EL INSTITUTO DE INGENIERIA UNAM 1993 ");
  GOTOXY(15,10);
END.

```

ANEXO VIII

GLOSARIO

PALABRAS UTILIZADAS FRECUENTEMENTE

Algoritmo	Es un conjunto de ordenes (para la computadora) con las cuales se cumple un cierto objetivo.
Ambiente de Programación	Sistema de trabajo donde se cuenta con opciones adicionales de trabajo pudiendo utilizarlas sin la necesidad de salir del proceso que se esta realizando, por ejemplo un entorno de programación permite acceder al compilador y la ayuda del mismo sin la necesidad de archivar el programa fuente activo en el editor.
ASCII	Código de caracteres que reconoce la computadora.
Base de Hechos	Representa las condiciones bajo las cuales el SE resuelve un problema particular en la base del conocimiento para el cual es válido.
Base de Conocimiento	Representa el conocimiento para el cual es válido el SE.
Entorno	Sistema de trabajo donde se cuenta con opciones adicionales que pueden ser utilizadas sin la necesidad de salir del proceso que se esta realizando, por ejemplo un entorno de programación permite acceder al compilador y la ayuda del mismo sin la necesidad de archivar el programa fuente activo en el editor.
Excitación	Movimiento inducido a un cuerpo.
Experto Humano	Se dice de la persona especializada en un área o campo del conocimiento cuya opinión es reconocida por los relacionados con el tema.
Flujo del Programa	Un programa puede tener varias opciones de cálculo, cuando se ejecuta sólo realiza algunos de estos cálculos de acuerdo con los datos de entrada. Las opciones que toma el programa se le denomina flujo del programa.
Lenguaje	Medio de comunicación con la computadora para ordenarle realice algunas tareas específicas.
Memoria disponible	Se refiere a la memoria en RAM que queda libre cuando se carga un programa en la computadora.
Memoria activa	Es la memoria RAM, utilizada para correr programas.
Motor de Inferencia	Representa la lógica con que funciona el SE.
MS-DOS	Comandos básicos para comunicarse con la computadora.

<i>Nodo</i>	<i>Vértice de un elemento finito (en este caso tridimensional);</i>
<i>Programa fuente</i>	<i>Es un archivo editado por el programador y por medio de él es posible determinar objetivos y cálculos que realiza la computadora al correr dicho programa.</i>
<i>Prototipo</i>	<i>Es la construcción del SE bajo un medio controlado para observar y evaluar su comportamiento bajo condiciones parametrizables.</i>
<i>Reconfiguración</i>	<i>Proceso que permite reorganización las condiciones de trabajo del SE (borrar archivos, respaldar información, etc.).</i>
<i>Sentencia</i>	<i>En una orden básica para la computadora.</i>
<i>Shell</i>	<i>Sistema Experto Vacío, el cual puede utilizarse como soporte para desarrollar SE en diferentes áreas del conocimiento.</i>
<i>Soporte</i>	<i>Herramienta o medio de apoyo para resolver un determinado problema.</i>
<i>Velocidad de procesamiento</i>	<i>Se refiere a la velocidad de respuesta de la computadora y se mide en Hz.</i>
<i>Work-Station</i>	<i>Computadora que permite realizar más de un proceso a la vez, es decir puede dar servicio a más de un usuario.</i>

COMANDOS DE TURBO PASCAL UTILIZADOS FRECUENTEMENTE¹

<i>ABS(A)</i>	<i>Devuelve el absoluto de A, donde A Puede ser real o entera.</i>
<i>APPEND(F)</i>	<i>Abre el archivo F para anexar datos al final del mismo.</i>
<i>ARC(X,Y,ANGLANGF,R)</i>	<i>Dibuja un arco donde X y Y son el centro, ANGL y ANGF son los ángulos de recorrido y R es el radio del arco.</i>
<i>ASSIGN(F,'archivo')</i>	<i>Asignación de el nombre de un archivo a una variable de tipo TEXTO.</i>
<i>BAR(X1,Y1,X2,Y2)</i>	<i>Dibuja un rectángulo con vértice superior izquierdo X1 y Y1 y vértice inferior derecho X2 y Y2.</i>
<i>CHDIR(CAMINO)</i>	<i>Deja el directorio actual y se cambia de directorio a CAMINO.</i>
<i>CLEARDEVICE</i>	<i>Limpia la pantalla en modo gráfico y lleva el cursor al punto (0,0) arriba a la izquierda de la pantalla.</i>
<i>CLEARVIEWPORT</i>	<i>Limpia la ventana activa en modo gráfico.</i>
<i>CLOSE(F)</i>	<i>Cierra el archivo F.</i>
<i>CLOSEGRAPH</i>	<i>Cierra el modo gráfico y configura la pantalla en modo texto.</i>

¹ Para mayor información consulte - Joyanes, Luis; "Programación en Turbo Pascal (versiones 4.0, 5.0 y 5.5); Ed. McGraw-Hill; España; 1990.

CLRSCR	<i>Limpa la pantalla o la ventana activa en modo texto.</i>
COS(A)	<i>Devuelve el coseno de X, donde X es una variable real y esta en radianes.</i>
DELAY(X)	<i>Retarda la ejecución del programa X milisegundos.</i>
DELLINE	<i>Borra la línea donde se encuentra el cursor.</i>
DETECTGRAFFH	<i>Detecta el tipo de monitor con que cuenta la computadora.</i>
DISKFREE(B)	<i>Devuelve el número de bytes disponibles en la unidad B.</i>
DOSVERSION	<i>Devuelve el número de versión del sistema operativo.</i>
EOF(F)	<i>Devuelve el estado de fin de archivo, es positivo cuando encuentra el final del archivo del que se esta extrayendo información.</i>
ERASE(F)	<i>Borra el archivo F.</i>
EXÉC(A,B)	<i>Ejecuta el programa designado por A, cuando se trata de el Comand.com es posible además dirigir la ejecución a un comando en particular designado por B por ejemplo B:="DIR".</i>
EXIT	<i>Abandona inmediatamente el procedimiento activo, en el caso de utilizarse dentro de un algoritmo de tipo iterativo provoca la interrupción del mismo y pasa a ejecutar la siguiente orden.</i>
EXP(X)	<i>Devuelve el número e elevado a la potencia X.</i>
FILLELIPSEX,Y,RX,RY)	<i>Dibuja un elipse rellena con origen X,Y y radios RX, RY.</i>
FLOODFILL(X,Y,MARCO)	<i>Rellena la región que contiene al punto (X,Y) y que esta limitada por el color MARCO.</i>
GETTIME(H,M,S,C)	<i>Regresa la hora actual donde H, M, S, y C son la horas, minutos, segundos y centésimas de segundo respectivamente.</i>
GOTOXY(X,Y)	<i>Desplaza el cursor a la coordenada X,Y donde X son columnas y Y son renglones. La posición 0,0 se encuentra en la esquina superior izquierda de la pantalla.</i>
INITGRAPH(CG,CM)	<i>Asigna valores iniciales al modo gráfico, donde CG significa tipo de monitor, si no conoce el tipo de monitor que se tiene se le puede asignar el valor de cero (0), con lo que sirve para indicar a Turbo Pascal que declare el monitor, detectando el tipo de éste a través de un análisis automático; y CM significa el camino donde se encuentra almacenado el archivo de gráficos de turbo pascal.</i>
INT(A)	<i>Devuelve la parte real de la variable entera A.</i>
IORESULT	<i>Devuelve el código de error después de realizar una sentencia cuando esta se ejecuta correctamente regresa el valor de cero.</i>
KEYPRESSED	<i>Toma el valor positivo cuando se oprime cualquier tecla.</i>
LINE(X1,Y1,X2,Y2)	<i>Dibuja una línea teniendo como principio el punto X1,Y1 y como fin el punto X2,Y2.</i>

LN(A)	Regresa el logaritmo natural de A, donde A es una variable real.
MKDIR(SUBDIR)	Crea el subdirectorio especificado por SUBDIR (variable cadena).
NOSOUND	Desactiva la bocina interna de la PC.
OUTTEXT(MENSAJE)	Despliega textos el modo gráfico, donde MENSAJE que es de tipo string.
OUTTEXTXY(MENSAJE)	Despliega textos en modo gráfico, en la posición X, Y; donde MENSAJE es de tipo string.
FI	Devuelve el valor de la constante π .
PUTPIXEL(X,Y,C)	Ilumina el píxel X,Y con el color dado por la constante C.
RANDOM(INTERVALO)	Devuelve un número aleatorio dentro del intervalo dado, donde INTERVALO es una variable de cualquier tipo. Cuando no se especifica el intervalo devuelve un número aleatorio de tipo real de cero a uno.
RANDOMIZE	Regresa un valor aleatorio tomado del reloj del sistema.
READ(F,V1,V2...)	Lee valores del archivo F, donde V1, V2,... pueden ser variables de cualquier tipo, si se omite la variable F Turbo Pascal lee los valores del teclado de la PC, cuando termina la lectura deja cursor en ese lugar.
READKEY	Permite leer un carácter del teclado sin la necesidad de oprimir ENTER, la variable a la que asigna la tecla leída debe ser de tipo CHAR.
READLN(F,V1,V2...)	Lee valores del archivo F línea por línea, donde V1, V2,... pueden ser variables de cualquier tipo, si se omite la variable F Turbo Pascal lee los valores del teclado de la PC, cuando termina la lectura mueve el cursor a la siguiente línea.
RECTANGLE(X1,Y1,X2,Y2)	Dibuja un rectángulo teniendo como esquina superior izquierda a X1,Y1 y como esquina inferior derecha a X2, Y2.
RENAME(F,NUVNO)	Cambia el nombre del archivo F por NUVNO.
RESET(F)	Abre el archivo F para lectura.
REWRITE(F)	Abre el archivo F para escritura.
RMDIR(SUBDIR)	Suprime el subdirectorio SUBDIR siempre y cuando éste se encuentre vacío.
ROUND(A)	Redondea el valor del real X a el entero más cercano.
SEEK(F,POS)	Desplaza la posición de lectura hasta POS dentro del archivo F.
SETBACKCOLOR(COLOR)	Define el color de fondo para el modo gráfico, donde COLOR es la constante de color utilizada.
SETCOLOR(COLOR)	Define el color de escritura para el modo gráfico, donde COLOR es la constante de color utilizada.
SETDATE(A,M,D)	Devuelve la fecha registrada en el sistema, donde A es el año, M es el mes y D es el día.

ANEXO VIII Glosario

SETFILLSTYLE(PAT,CL)	Define el patrón de relleno y el color de éste, donde PAT es el patrón y COL es la constante de color utilizada.
SETGRAPHMODE	Restablece el modo gráfico bajo las condiciones especificadas en UNITGRAPH, en caso que el modo este activo limpia la pantalla.
SETLINESTYLE(T,P,G)	Define el tipo de línea que se va a desplegar, donde T es el tipo de línea (entera, punteada...), P es el patrón de relleno y G es el grosor de la misma.
SETTEXTJUSTIFY(H,V)	Configura el alineamiento del texto (en modo gráfico), H es el alineamiento horizontal, donde: 0 es izquierda, 1 es centrado y 2 derecha. V es el alineamiento vertical. donde: 0 es inferior, 1 es centrado y 2 superior.
SETTEXTSTYLE(T,O,A)	Configura el tipo de carácter utilizada para desplegar textos en modo gráfico, donde T es el tipo de carácter, O es la orientación (vertical u horizontal) y A es la amplificación del mismo.
SETVIEWPORT(A,B,C,D)	Declara una ventana en el modo gráfico, donde A y B son las coordenadas de la esquina superior izquierda y C y D son las de la esquina inferior izquierda.
SIN(X)	Devuelve el valor seno de X donde X es una variable real y esta en radianes.
SOUND(HZ)	Activa la bocina interna y produce un sonido igual a HZ donde HZ es una variable real y esta en ciclos por segundo, para desactivar la bocina se utiliza NOSOUND.
SQR(X)	Devuelve el cuadrado de X, donde X es de tipo real.
SQRT(X)	Devuelve la raíz de X, donde x es de tipo real positivo.
TEXTBACKGROUND(A)	Define el color de fondo para el modo texto, donde A es la constante de color seleccionada.
TEXTCOLOR(A)	define el color de escritura en el modo texto, donde A es la constante de color seleccionada.
UPCASE(CH)	Devuelve la letra mayúscula de la variable CH si esta existe.
WINDOW(X1,Y1,X2,Y2)	Define una ventana de salida en el modo texto, donde X1 y Y1 es la esquina superior izquierda y X2, Y2 es la esquina inferior derecha de la misma.
WRITE(F,V1,V2...)	Escribe valores en el archivo F, donde V1, V2,... pueden ser variables de cualquier tipo, si se omite la variable F turbo Pascal lee los valores del teclado de la PC, cuando termina de escribir deja el cursor en esa posición.
WRITELN(F,V1,V2...)	Escribe valores en el archivo F línea por línea, donde V1, V2,... pueden ser variables de cualquier tipo, si se omite la variable F turbo Pascal lee los valores del teclado de la PC, cuando termina de escribir mueve el cursor al principio de la siguiente línea.