UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA CIVIL

TESIS
IC2004
C37

PROGRAMA PARA EL DISEÑO DE PUENTES TUBO EN ARCO (DAPT 1.0)

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: VEINTE (20) PUNTOS

JURADO EXAMINADOR

Firma:
Nombre: JUAN A. RAMIREZ

Firma:
Nombre: MARIO MENDIVAL

Firma:
Nombre: EMILIO SOLORZANO

REALIZADO POR: Castillo Gutiérrez, José Ramón
Díaz Millán, Karla Patricia

PROFESOR GUIA: Ing. Juan Andrés Ramírez

FECHA: Caracas, Junio de 2004

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA CIVIL

PROGRAMA PARA EL DISEÑO DE PUENTES TUBO EN ARCO (DAPT 1.0)

Trabajo especial de Grado presentado como requisito parcial para optar al titulo de Ingeniero Civil

REALIZADO POR Castillo Gutiérrez, José Ramón
Díaz Millán, Karla Patricia

PROFESOR GUIA Ing. Juan Andrés Ramírez

FECHA Caracas, Junio de 2004

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA CIVIL

PROGRAMA PARA EL DISEÑO DE PUENTES TUBO EN ARCO (DAPT 1.0)

Trabajo especial de Grado presentado como requisito parcial para optar al titulo de Ingeniero Civil

Autores: Castillo Gutiérrez, José Ramón

Díaz Millán, Karla Patricia

Tutor: Ing. Juan Andrés Ramírez

Caracas, Junio de 2004

DEDICATORIA

A mis padres por ser mi fuente de energía y el pilar donde se apoyan y crecen todos mis sueños.

A mi hermana y mi novio, por compartir conmigo cada día de mi vida y apoyarme en todo momento.

A todos los miembros de mi familia por ser cada uno a su manera parte importante de lo que soy y de lo que me he convertido, ustedes son personas maravillosas que llenan de felicidad mi vida.

Karla Patricia Díaz Millán

A mis padres y a mi hermano, por todo el cariño y el apoyo que me han dado a lo largo de toda mi vida.

A mi novia y seres queridos por estar a mi lado en cada momento y en cada paso del camino.

José Ramón Castillo Gutiérrez

AGRADECIMIENTOS

A la Universidad Católica Andrés Bello por ser nuestra casa durante esta travesía que nos llevo a formarnos como profesionales.

Al Ingeniero Juan Andrés Ramírez por brindarnos su asesoría durante la realización de este trabajo especial de grado y apoyarnos durante el largo proceso de su realización.

Al Ingeniero Carlos Griffin por compartir con nosotros en el transcurso de nuestros estudios y ofrecernos la oportunidad de realizar este trabajo especial de grado.

Al Ingeniero Luciano Carestia por brindarnos su excelente y desinteresada ayuda en la realización de este trabajo especial de grado.

A nuestros padres, por ser los seres que con su dedicación, apoyo y compresión, hicieron posible que culmináramos esta etapa de nuestras vidas. Gracias.

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
Trabajo Especial de Grado

PROGRAMA PARA EL DISEÑO DE PUENTES TUBO EN ARCO (DAPT 1.0)

Autores: José Ramón Castillo
Karla Patricia Díaz
Tutor: Ing. Juan Andrés Ramírez
Fecha: Mayo, 2004

RESUMEN

En el presente trabajo especial de grado se desarrolló una aplicación informática que permite el diseño y/o la verificación de estructuras de puentes tubo en arco de acero al carbono con juntas soldadas, para ello se tomaron en cuenta las siguientes consideraciones: presión interna y colapso por vacío, ambos aspectos de origen hidráulico, seguidamente un análisis estructural que considera el puente tubo en arco como una viga empotrada en ambos extremos donde se estudia un caso de carga originado por el peso de la tubería y del fluido que transporta, fuerzas ocasionadas por las acciones del viento y una carga viva uniformemente distribuida que representa eventuales trabajos de reparación sobre la estructura, para así obtener las fuerzas y los desplazamientos a los que estará sometida la estructura. A través de estas rutinas de cálculo se selecciona el espesor de tubería que mejor se adapta a las necesidades diseño y que garantice la integridad de la aducción para posteriormente generar los planos de ingeniería básica utilizando como herramienta el programa AutoCAD.

Descriptores: Tubería, puente tubo en arco.

ÍNDICE GENERAL

Página

Portada 01
Dedicatoria 02
Agradecimientos 03
Resumen 04
Índice General 05
Índice de tablas 08
Índice de Figuras 09
Símbolos y notación utilizada..... 11
Objetivos 13
Introducción 14
Alcances 15
Antecedentes 16

Capitulo I.

1. Enfoques de la aplicación

1.1 Diseño del Puente Tubo..... 20
1.1.1 Generalidades 20
1.1.2 Información básica requerida..... 20
1.1.3 Geometría 24
1.1.4 Prediseño de fundaciones directas 30
1.2 Verificación 33
1.2.1 Generalidades 33
1.2.2 Alcances 33

Capitulo II.

2. Cálculos asociados a la Hidráulica

2.1 Presión Interna 35

2.2 Colapso por Vacío 38

Capitulo III.

3. Cálculos Estructurales

3.1 Introducción 41

3.2 Análisis Estructural del puente tubo en arco 42

3.3 Análisis Matricial por el Método de los desplazamientos..... 43

3.4 Esfuerzos 49

3.4.1 Revisión de la sección tubular sometida a esfuerzos combinados mediante la aplicación del "Criterio tentativo para aplicaciones estructurales y conductos de acero"American Iron and Steel Institute (AISI)..... 50

3.4.1.1 Relación de Esbeltez Efectiva..... 50

3.4.1.2 Compresión Axial 50

3.4.1.3 Flexión 51

3.4.1.4 Corte por Flexión 52

3.4.1.5 Esfuerzo de Corte por Torsión..... 52

3.4.1.6 Combinaciones de Carga..... 52

3.4.1.6.1 Presión Interna 52

3.4.1.6.2 Presión Externa 53

3.4.1.7 Tensiones Combinadas..... 53

Capitulo IV.

4. Desarrollo de la aplicación

4.1 El Programa 55

4.2 Requerimientos del Sistema..... 56

4.2.1 Hardware..... 56

4.2.2 Software..... 56

4.3 Flujograma..... 57

4.4 Estructura del programa..... 59

4.5 Entrada de Datos.....59

4.6 Resultados.....73

4.6.1 Numéricos 74

4.6.2 Gráficos 76

Conclusiones 79

Recomendaciones..... 80

Bibliografía 81

Anexos..... 82

ÍNDICE DE TABLAS

Página

Tabla N° 1. Pesos Específicos característicos de líquidos corrientes a la presión Atmosférica..... 21

Tabla N° 2. Especificaciones de algunos aceros comerciales..... 23

Tabla N° 3. Transformaciones del sistema internacional al sistema inglés 62

Tabla N° 4. Capacidad de soporte para diversos tipos de suelos..... 72

ÍNDICE DE FIGURAS

		Página
Figura N° 1	Puente Tubo en arco San Agustín (Mayo 2004)	17
Figura N° 2	Puente Tubo en arco Plaza Venezuela (Mayo 2004)	17
Figura N° 3	Puente Tubo en arco Macaracuay (Mayo 2004)	18
Figura N° 4	Puente Tubo en arco Puente Hierro (Mayo 2004)	18
Figura N° 5	Peso específico del agua a presión atmosférica	22
Figura N° 6	Aprox. geométrica de la sección transversal del puente tubo	24
Figura N° 7	Geometría real de la sección transversal del puente tubo	28
Figura N° 8	Detalle de tramo de tubería	29
Figura N° 9	Bloque de fundación	32
Figura N° 10	Diagrama de cuerpo libre del corte de una tubería	35
Figura N° 11	Ventana introductoria del programa	59
Figura N° 12	Ventana de selección del tipo de operación	60
Figura N° 13	Ventana de Entrada de Datos	61
Figura N° 14	Ventana de selección del tipo de fluido	63
Figura N° 15	Ventana con detalle de Temperatura y cargas aplicadas	63
Figura N° 16	Ventana de datos se sección transversal topográfica	64
Figura N° 17	Ventana de Gráfico en Excel	66
Figura N° 18	Ventana de selección del tipo de acero y factores de seguridad	67
Figura N° 19	Ventana de selección de otro tipo de Acero y asignación del esfuerzo de fluencia	68
Figura N° 20	Ventana de selección del tipo de cálculo y del Diámetro	69
Figura N° 21	Ventana de detalle de selección del diámetro	70
Figura N° 22	Ventana de selección de diámetro y número de tramos	71
Figura N° 23	Prediseño del bloque de anclaje	72
Figura N° 24	Ventana Resumen de Datos	73

Figura N° 25 Ventana de cuadro de Resultados..... 74
Figura N° 26 Ventana de Fuerzas y Momentos 75
Figura N° 27 Ventana de los desplazamientos 75
Figura Nº 28 Plano de AutoCAD del puente tubo en arco..... 76
Figura N° 29 Plano de AutoCAD,detalle de tramo de tubería 77
Figura N° 30 Plano de AutoCAD, detalle de bloque de fundación directa..... 77
Figura N° 31 Ventana final del programa 78

SÍMBOLOS Y NOTACIÓN UTILIZADA

ϑ	Angulo entre las rectas que conforman cada tramo
α	Ángulo formado por las rectas desde el punto de inicio hasta el punto final.
m_1	Pendiente de la recta formada por el punto inicial y el centro
m_2	Pendiente de la recta formada por el punto final y el centro
R	Radio del circulo
Lt	Longitud del Tramo de Tubería
P	Presión Interna
d.	Diámetro interno de la tubería
D	Diámetro externo de la tubería
k_1	Factor de seguridad para el esfuerzo de fluencia
k_2	Factor de Seguridad para el esfuerzo de rotura
k_{11}	Factor de Seguridad para el esfuerzo de fluencia y presión normal
k_{12}	Factor de Seguridad para el esfuerzo de fluencia y sobrepresión
k_{pn}	Factor de seguridad para la presión Normal de Trabajo
k_{sp}	Factor de seguridad para la Sobrepresión.
e_{pn}	Espesor de tubería para la presión Normal de Trabajo
e_s	Espesor de tubería para la Sobrepresión.
σ_{pn}	Esfuerzo de fluencia para la presión normal
σ_s	Esfuerzo de fluencia para la sobrepresión
P_{crit}	Presión de Colapso o crítica
E	Módulo de elasticidad del material
e	Espesor del conducto

D	Diámetro exterior de la tubería
μ	Relación de Poisson
σ_f	Punto efluente del acero
A	Área de la sección
J	Momento polar de Inercia
I	Inercia de la sección

OBJETIVO

El objetivo general de la elaboración de este trabajo especial de grado es el de construir una herramienta informática de cálculo, diseño y dibujo en el programa AutoCAD, a nivel de ingeniería básica, de proyectos de puentes tubo en arco para la conducción de flujo a presión. La herramienta esta en capacidad de calcular y diseñar todos los parámetros relativos al proyecto de puentes tubo en arco, dentro de los cuales se incluyen la configuración geométrica del puente, las solicitaciones generadas por las cargas, determinación del espesor de la tubería, predimensionado de las fundaciones cuando se trate de fundaciones directas y generar planos de ingeniería básica en el programa AutoCAD, particularmente un perfil longitudinal del puente tubo y el detalle de la pieza mecánica típica que compone el arco.

INTRODUCCIÓN

La colocación de tuberías a través de secciones transversales de terreno tales como cauces de cursos de agua y depresiones en general, posee una gama de soluciones que implica estructuras que son capaces de satisfacer las exigencias hidráulicas y las solicitaciones de carga que se presentan en cada caso, una de estas soluciones es el puente tubo en arco; esta clase de estructura posee unas características geométricas que pudieran colocarlas en posición ventajosa al compararlas con otros tipos de puentes tubo; no requiere de estructuras adicionales que deban ser calculadas, ni de sistemas de soporte, ni representan un montaje mecánico que pueda considerarse de extrema complicación.

El uso de la tubería en arco pudiese asociarse a casos particulares que se presentan cuando se debe atravesar una sección topográfica con una luz libre de considerable longitud, tal es el caso de los cursos de agua de importante magnitud con escasa disponibilidad de espacio en los estribos, situación apreciable en la ciudad de Caracas, donde las estructuras en arco se encuentran sobre las márgenes del Río Guaire, en zonas contiguas a la Autopista Francisco Fajardo u otras arterias viales. Podemos inferir que su construcción se debió a la dificultad que resulta de aplicar una de las soluciones más comunes a la hora de cruzar una luz libre como pudiese ser puentes tubo soportados por cables tensados, bien sean colgantes o autoportantes. Este recurso no es aplicable cuando no se cuenta con un espacio suficiente para la construcción de los apoyos extremos ya que ocuparíamos parte del espacio que esta destinado a la circulación de los vehículos por estas arterias viales.

En vista de la aplicación que en algún momento tuvieron los puentes tubo en arco y que la información técnica para el calculo de este tipo de estructuras se

encuentra muy dispersa entre la bibliografía comúnmente disponible, se ha procedido a generar esta herramienta que trata de reunir de la manera más práctica posible las consideraciones y procedimientos de cálculo necesarios para la elaboración del proyecto de ingeniería de un puente tubo en arco.

Tal y como esta expresado en la parte anterior, uno de los objetivos de la herramienta informática es el de crear un plano del perfil longitudinal de la estructura, incluyendo la sección topográfica, los bloques de fundación directa (según sea el caso) y el dimensionado detallado de la pieza mecánica típica del puente; en este sentido se ha planteado que el nivel de presentación de este plano sea de ingeniería básica y contenga las dimensiones mínimas necesarias para la fabricación de la pieza mecánica típica que conforma el arco del puente tubo y deberá ser considerado como base para la creación del o los planos de ingeniería de detalle del proyecto.

Alcances

El Presente trabajo especial de grado abarca los siguientes aspectos:

- Diseño de puentes tubo en arco de acero al carbono con juntas soldadas.
- Diseño Geométrico de la estructura.
- Diseño Estructural del puente tubo basado en parámetros hidráulicos.
- Diseño Estructural de la sección transversal.
- Diseño Estructural del elemento trabajando a flexo-compresión.
- Pre-diseño de fundaciones directas (según aplique el caso).
- Diseño Mecánico del puente tubo.
- Dibujo de plano de Ingeniería Básica del puente tubo haciendo uso del programa AutoCAD

Antecedentes

De acuerdo con lo expuesto al inicio de la introducción y considerando que se trata del diseño de estructuras muy particulares, de las cuales no es común tener acceso a bibliografía especializada con aplicación práctica ni a herramientas informáticas para el diseño de las mismas; se estableció la ejecución de un trabajo especial de grado que contemple la creación de una aplicación informática para el diseño de los puentes tubo en arco, abarcando desde la recopilación y análisis de información existente, hasta la generación de un plano de ingeniería básica del diseño del puente tubo.

Es importante destacar que para la elaboración del presente trabajo se realizó una labor de búsqueda y recopilación de información existente relativa a los puentes tubo en arco ya construidos, especialmente del área metropolitana de Caracas; este proceso arrojó como resultado la carencia de proyectos y memorias de cálculo que faciliten la obtención de las consideraciones tomadas en cuenta en el pasado a la hora de diseñar estas estructuras, a pesar de esto se logro tener acceso a los planos de construcción de algunos de los puentes tubo en arco localizados en la ciudad de Caracas, pero esta información es muy básica y de ninguna manera su contenido es relevante al momento de emprender el diseño del puente tubo.

Como complemento de este texto se anexan fotografías actuales de puentes tubo en arco, existentes en el área metropolitana de Caracas.

Figura N° 1. Puente tubo en arco San Agustín (Mayo 2004)

El puente tubo N° 1 mostrado en la figura N° 1 se encuentra localizado en las márgenes del río Guaire en la Autopista Francisco Fajardo a la altura de la zona de San Agustín con aproximadamente 30 metros de luz libre y 15 metros de altura

El siguiente puente tubo, figura N°2, se encuentra ubicado en la zona de Plaza Venezuela con una luz libre de 40 metros y una altura de 4 metros aproximadamente.

Figura N°2. Puente tubo en arco Plaza Venezuela (Mayo 2004)

En la figura N° 3 encontramos el puente tubo en arco que puede apreciarse en la Avenida San Francisco de Macaracuay, con una luz libre de 46 metros y una altura de 8 metros aproximadamente.

Figura N° 3. Puente tubo en arco Macaracuay (Mayo 2004)

Por ultimo encontramos el puente tubo que se encuentra ubicado en la Autopista Francisco Fajardo a la altura de Puente Hierro con una luz libre de 50 metros y una altura de 8 metros aproximadamente.

Figura N° 4. Puente tubo en arco Puente Hierro (Mayo 2004)

CAPÍTULO I

I. Enfoque de la aplicación

El programa elaborado cumple con las funciones de diseñar y/o verificar puentes tubo en arco. La aplicación informática se desarrollo de manera tal que el puente tubo cumpla con los requerimientos necesarios para su buen desempeño. Inicialmente el usuario deberá introducir como datos el diámetro, la presión máxima de trabajo, el caudal o gasto y las cargas a la que estará sometida la tubería, esto se debe a que se asume que la tubería en arco es una estructura adicional en una aducción y por lo tanto el diámetro de la tubería, la presión de trabajo y el caudal son el resultado de condiciones hidráulicas ya estudiadas. El programa calculará el espesor de la tubería que resista tanto la presión interna, como la presión que pudiese ocasionar el colapso por vacío en la tubería y cumplir con todas las condiciones de esfuerzos que se generan para el caso cargas establecido en el diseño del puente tubo.

El usuario del programa obtendrá los valores de las fuerzas en los diferentes ejes principales, *X*, *Y* y *Z*, para cada uno de los nodos de la estructura (Ver Capítulo III) y en el caso de que este lo desee el programa generará el plano del perfil longitudinal del puente tubo así como también el detalle mecánico de la pieza típica que compone el puente, utilizando como herramienta el programa AutoCAD. Adicionalmente podrá obtener el predimensionado del bloque de fundación directa del puente tubo (según sea el caso).

1.1. Diseño del puente Tubo

1.1.1 Generalidades

Este enfoque se refiere a la obtención de un espesor mínimo que satisfaga las solicitaciones generadas tanto por la presión interna del fluido (considerando la presión normal de trabajo y una eventual sobrepresión) como por la presión externa; dicho espesor será el punto de partida para el análisis estructural del puente tubo, bajo las condiciones geométricas particulares y las cargas adicionales a las que está sometida la estructura, tales como carga viva y el efecto del viento. Una vez concluido el análisis se obtiene el espesor, las fuerzas actuantes y los desplazamientos en los nodos de la estructura.

En caso del diseño el usuario introducirá los datos relativos a la sección transversal topográfica, las coordenadas de inicio del puente tubo y la altura del mismo. Con estos datos el programa calculará la luz libre del puente tubo, seguidamente podrá visualizar el gráfico que representa la topografía introducida y hacer los posibles cambios que considere necesarios.

Por el contrario, si se está realizando una verificación, el usuario directamente introducirá los valores de la luz libre y la altura del puente tubo, así como también el espesor que posee la tubería.

1.1.2 Información básica requerida

Para proceder a diseñar el puente tubo el usuario deberá seleccionar el tipo de fluido que será transportado por la tubería de manera que se identifique en la base de datos cual es el peso específico promedio del mismo y para el caso del agua, el peso

específico de la misma en función de la temperatura. Dichos fluidos se encuentran resumidos en la tabla N° 1. con sus respectivos pesos específicos.

Tipo de Fluido	**Peso especifico** kg/m^3	**Tipo de Fluido**	**Peso especifico** kg/m^3
ALCOHOL ETILICO	789,7	GLICERINA	262,3
BENCENO	879,5	KEROSEN	800,95
GASOLINA	674	ACEITE	868,55

Tabla N°1. Pesos específicos característicos de líquidos corrientes a la presión atmosférica

Es importante destacar que aún cuando el usuario tiene la libertad de escoger el tipo de fluido, la tubería en arco no aplica a todos los casos, se toma como ejemplo el transporte de gasolina donde debe llevarse a cabo un mantenimiento en las tuberías mediante un dispositivo que se introduce y se desplaza a lo largo de la misma, los puentes en arco poseen una geometría que impide que su recorrido se realice sin interrupciones en tramos con secciones quebradas y con ciertos ángulos, en tal sentido se considera que este tipo de información es relevante a la hora de diseñar y queda a criterio del usuario la aplicación de este tipo de tubería en casos donde no sea recomendable.

En el caso particular del agua encontramos una variación del peso específico con respecto a la temperatura que se rige según la curva del siguiente gráfico:

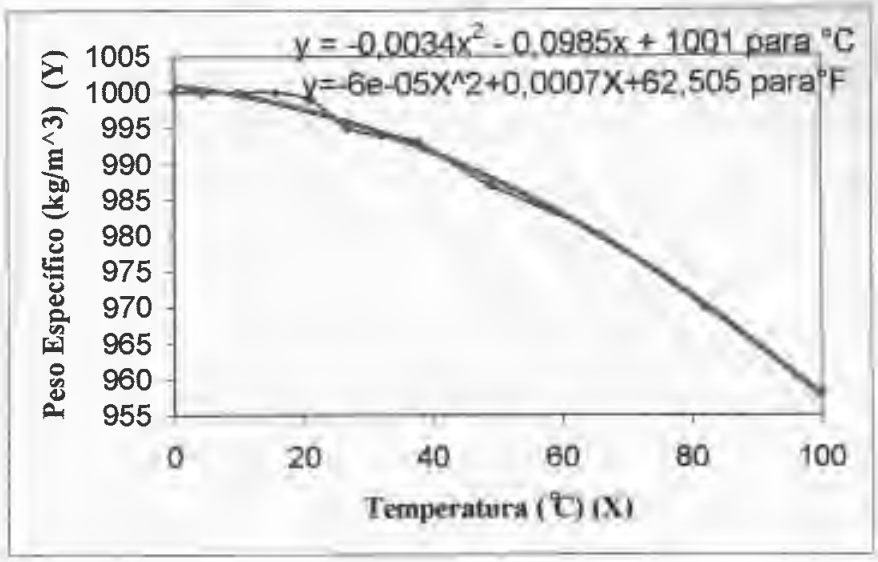


Figura N° 5. Peso especifico del agua a presión atmosférica en función de la temperatura

Tomando esto en cuenta, se puede calcular la ecuación que mejor describe la relación entre los dos aspectos (Ver figura N° 5) y de esta manera facilitar la selección del peso específico para cualquier valor de la temperatura.

Una vez seleccionado el fluido y la temperatura a la cual se encuentra, se introducen los valores de presión de trabajo, sobrepresión en el caso que exista, carga viva y carga de viento.

El siguiente paso es la introducción de los puntos en las coordenadas X e Y que conforman la sección transversal topográfica, de este gráfico se podrá obtener la luz libre del puente tubo. El desarrollo del aspecto geométrico que permite este cálculo se analiza en el punto 1.1.3 de este capítulo.

A continuación se selecciona el tipo de acero para la tubería que se pretende utilizar. Existen actualmente una gran variedad de tipos de aceros que se utilizan para

tuberías en los sistemas de aducción, estos poseen propiedades físicas y un comportamiento estructural diferente dependiendo de las posibles condiciones de carga a las que pueden estar sometidos. De esta gran variedad de tipos de acero se destacan principalmente en las aducciones las tuberías de acero al carbono, por lo tanto la búsqueda se limitara a las tuberías de acero al carbono con diámetros y espesores comerciales existentes en el país y cuya fabricación se rige por las normas del American Society for Testing and Materials (ASTM) y el American Petroleum Institute (API) y cuyas especificaciones se muestran en la siguiente tabla:

NORMA		**PUNTO DE FLUENCIA MÍNIMO**		**ESFUERZO MÁXIMO**	
		PSI	**kg/cm²**	**PSI**	**kg/cm²**
API5L	A	30000	2110	48000	3370
	B	35000	2460	60000	4220
	X42	42000	2950	60000	4220
	X52	52000	3660	66000	4640
	X60	60000	4220	75000	5270
ASTM A42		42000	2950	60000	4220
ASTM A36		36000	2530	58000	4080
ASTM 53	A	30000	2110	48000	3370
	B	35000	2460	60000	4220

Tabla Nº 2. Especificaciones de algunos Aceros comerciales

Una vez seleccionado el tipo de acero el usuario podrá ajustar el valor de los factores de seguridad que se utilizarán para el cálculo y que se explican detenidamente en el capítulo II y el capítulo III.

Si desea obtener el predimensionado del bloque de fundación directa deberá introducir el valor de la capacidad de soporte del suelo en los estribos del puente tubo.

1.1.3 Geometría

El usuario deberá introducir progresivamente los datos de la sección topográfica y cada punto de la topografía estará designado por un valor de "X" e "Y" positivos que serán introducidos manualmente. Es importante que el usuario introduzca la totalidad de la sección transversal ya que el programa generará el arco tomando como consideración principal que el punto designado como inicial y el final estén a la misma altura, así como también que la estructura debe atravesar toda la sección topográfica que introduce el usuario.

La estructura tendrá una geometría circular donde el arco no corresponde necesariamente a una semicircunferencia; los cálculos se asumen para cualquier sección de arco con la única condición de que el punto de salida este a la misma cota que el punto de llegada, por lo tanto la cuerda corresponderá a la luz libre que debe vencer la tubería tal como se muestra en la siguiente figura:

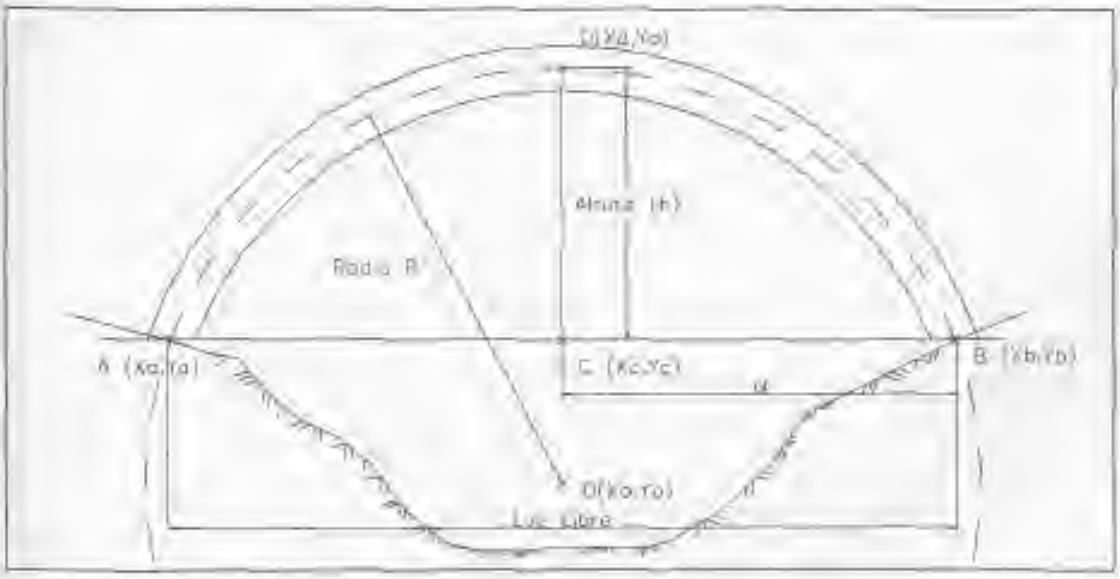


Figura Nº 6. Aproximación geométrica de la sección transversal del puente tubo.

A continuación se presentan los cálculos y consideraciones geométricas que permiten generar el arco teniendo como información las coordenadas de los puntos de la figura N° 6:

$$A = (X_a, Y_a) \qquad X_b = X_a + Luz$$
$$B = (X_a + Luz, Y_a) \qquad Y_a = Y_b = Y_c$$
$$D = (\frac{Luz}{2} + X_a, Y_a + h) \qquad X_c = \frac{X_b + X_a}{2} \Rightarrow 2X_c = X_b + X_a$$

Mediatriz

Punto medio AB = C $\qquad C = (X_c, Y_c) = (\frac{Luz}{2} + X_a, Y_a)$

Recta $X = X_c = \frac{Luz}{2} + X_a$

Intersección

$$Y - Y_a - \frac{h}{2} = -\frac{Luz}{2} \times \left(\frac{X_a}{h} + \frac{Luz}{2h} \right) + \frac{Luz^2}{8h} + \frac{Luz \times X_a}{2h}$$

$$Y - Y_a - \frac{h}{2} = \frac{Luz \times X_a}{2h} - \frac{Luz^2}{4h} + \frac{Luz^2}{8h} - \frac{Luz \times X_a}{2h}$$

$$Y - Y_a - \frac{h}{2} = \frac{-2Luz^2 + Luz^2}{8h}$$

$$Y - Y_a - \frac{h}{2} = -\frac{Luz^2}{8h}$$

$$\boxed{\begin{aligned} Y &= Y_a + \frac{h}{2} - \frac{Luz^2}{8h} \\ X &= X_c = \frac{Luz}{2} + X_a \end{aligned}}$$

Centro O

$$O\left(\frac{Luz}{2} + X_a, Y_a + \frac{h}{2} - \frac{Luz^2}{8h}\right)$$

Radio

$$(\overline{AO})^2 = \left(\frac{h^2}{4} + \frac{Luz^2}{8}\right) + \left(\frac{Luz^2}{8h}\right)^2$$

Ecuación de la Circunferencia

$$(X - X_o)^2 + (Y - Y_o)^2 = \frac{h^2}{4} + \frac{Luz^2}{8} + \frac{Luz^4}{64 \times h^2}$$

En función de: $X_a \rightarrow Dato$

$Y_a \rightarrow Dato$

$Luz \rightarrow Dato$

$h \rightarrow Dato$

Sólo para valores de $Y \geq Y_a$

$X \geq X_a$

$h > 0$

Al considerar el número de tramos rectos que conformarán la estructura en arco se tomaron en cuenta varios aspectos, inicialmente un aspecto mecánico, la soldadura.

El objetivo de la soldadura es el de unir el metal dándole a esta unión propiedades físicas y químicas adecuadas al trabajo que van a desempañar la pieza. Las tuberías de acero que conducen un fluido a presión se fabrican en aceros dulces, con poca cantidad de carbono y que generalmente son fácilmente soldables, esto debido a que los aceros de alta resistencia implican un incremento en la dificultad del proceso de soldadura, procedimiento que intrínsicamente se relaciona con la habilidad del operador.

Cuando se lleva a cabo el proceso de soldadura, en este caso usualmente en espiral o longitudinalmente, el metal es calentado localmente por los fenómenos de transferencia de calor, generando una distribución no uniforme que producen variaciones estructurales y metalúrgicas, estudios anteriores[1] han demostrado que el calor adyacente a la soldadura no afecta significativamente el metal base, y su área de afectación esta en el orden de las micras, por esto la distancia entre las juntas soldadas no es un factor determinante en la geometría del arco.

Una vez estudiado este aspecto, el problema de la geometría del arco pasa a ser de índole constructivo. Para optimizar el montaje del arco se estableció que las longitudes de cada tramo serían las mismas, esto se logra al dividir n veces el ángulo y de esta manera se evita el hecho de cortar y biselar tubos de diferentes longitudes, por otra parte a nivel visual se hace mucho más estética la estructura.

---

[1] Fernández de Sousa, Freddy y Pérez Caraballo, Víctor. (1999) **Estimación de los Esfuerzos Residuales en uniones soldadas a través del método de los elementos finitos**. Trabajo especial de Grado, Universidad Central de Venezuela, Caracas.

A continuación se encuentran las operaciones y cálculos que permiten establecer las longitudes de cada tramo de tubería a partir de las secciones que la conforman:

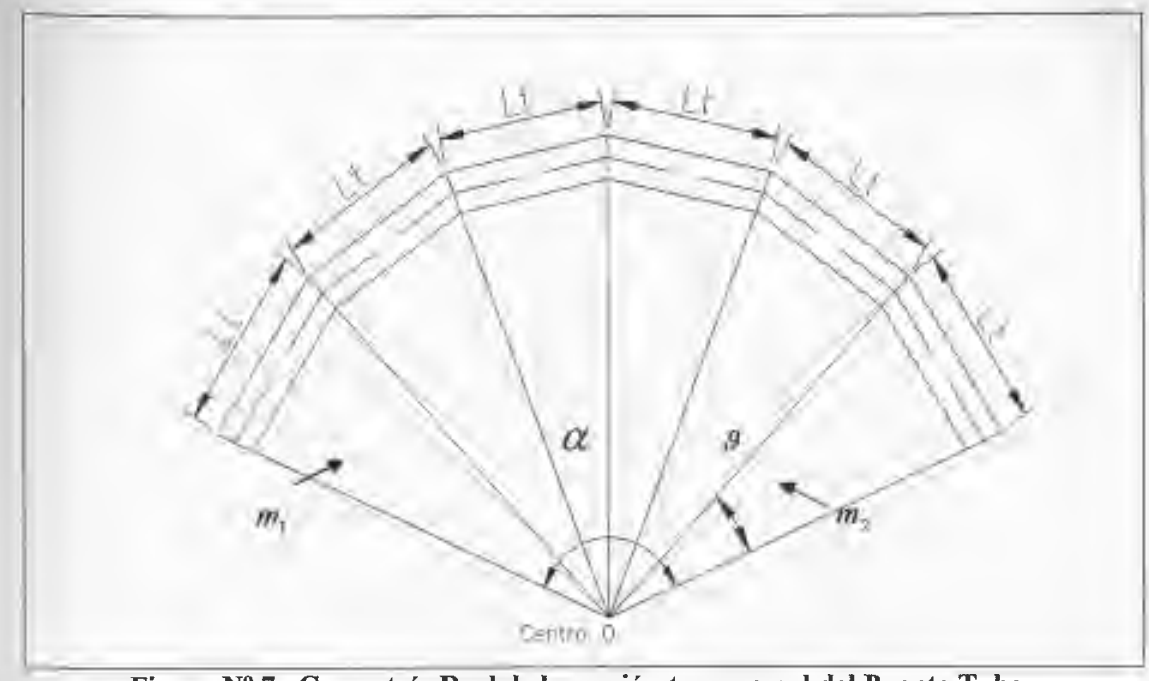


Figura Nº 7. Geometría Real de la sección transversal del Puente Tubo

Utilizando como herramienta el teorema del coseno, podemos relacionar los lados del triangulo que se forma al aislar un tramo de tubería con su respectivo centro, tal como se muestra en la figura anexa y que permite desarrollar lo siguiente:

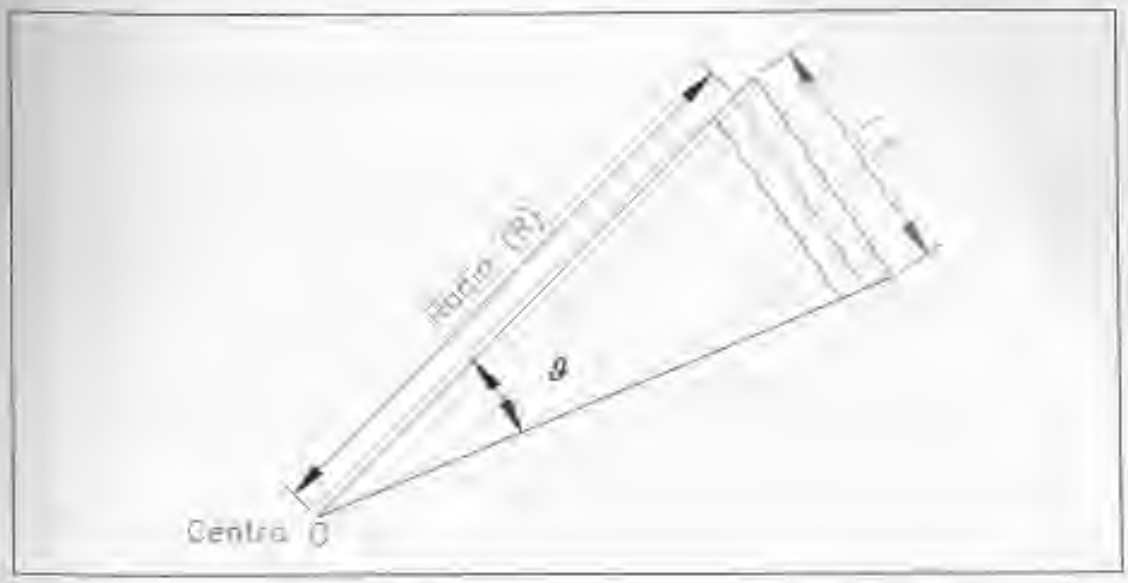


Figura N° 8. Detalle de tramo de tubería

$$Lt^2 = r^2 + r^2 - 2r \times r \times \cos\theta$$

$$\theta = \frac{\alpha}{n}, \alpha = \arctan g\left[\frac{m_2 - m_1}{1 + m_2 m_1}\right]$$

Sustituyendo,

$$Lt = \sqrt{2r^2 - 2r^2 \cos\left(\frac{\arctan g\left[\frac{m_2 - m_1}{1 + m_2 m_1}\right]}{n}\right)}$$

Donde,

$$m_1 = \frac{Yb - Yo}{Xb - Xo}, m_2 = \frac{Ya - Yo}{Xa - Xo}$$

Adicionalmente, para Diámetros menores o iguales a 16 pulgadas la longitud mínima del tramo será 0,50 metros y para mayores de 16 pulgadas la longitud mínima será el valor del diámetro, ambas consideraciones son de índole constructiva.

1.1.4 Prediseño de fundaciones directas

Las consideraciones tomadas en cuenta a la hora de prediseñar las fundaciones directas, vienen dadas por el equilibrio de las fuerzas actuantes en los nodos que se encuentran ubicados en los extremos de la tubería, tales como:

Las fuerzas resultantes del análisis estructural y que se presentan como una matriz de componentes locales Fx, Fy y Fz. En dicha matriz se seleccionan los valores correspondientes a los nodos extremos y se descomponen las fuerzas en los ejes X, Y y Z principales para luego seleccionar la de mayor valor que se denominara F_1

También encontramos la fuerza debida a la velocidad del agua, originada por la cantidad de movimiento que viene expresada por:

$$F_q = \frac{\gamma}{g} Q \times V \qquad (I\text{-}1)^2$$

Donde, γ = Peso específico del agua

g = Aceleración de la gravedad

Q = Gasto

v = Velocidad del agua en la sección considerada

Esta fuerza debida a la cantidad de movimiento es paralela al eje de la tubería por lo que igualmente es necesario su descomposición, la cual se lleva a cabo únicamente en el plano que contiene al puente tubo en arco y que esta definido por los ejes X e Y.

[2] Arocha Ravelo, Simón. "**Abastecimientos de Agua**" Caracas: Editorial Innovación Tecnológica. Pág 241.

De las fuerzas antes mencionadas se selecciona la de mayor valor para igualarla ha la ecuación que garantiza el equilibrio del bloque de fundación:

$$F = \phi \times W + h \times D \times P \qquad (I\text{-}2)^3$$

Donde, F= Fuerza actuante

ϕ= Coeficiente de fricción entre el concreto y el suelo

W= Peso del anclaje

h = Altura del bloque

D= Ancho del bloque en el plano posterior de contacto

P= Capacidad de soporte del suelo

En el diseño del bloque de fundación se asume una forma cúbica para simplificar los cálculos, por lo tanto D será igual a h y la expresión anterior se transforma en la siguiente:

$$F = \phi \times W + h^2 \times P$$

Al sustituir el valor del peso del bloque de concreto por: $PesoUnitario(\frac{kg}{m^3}) \cdot h^3(m^3)$ obtendremos:

$$F = \phi \cdot P_p \cdot h^3 + h^2 \cdot P$$

Esta ecuación de tercer grado se resolverá mediante una serie de interacciones con valores de h, hasta encontrar el valor que cumple con la ecuación y que permitirá diseñar el bloque de anclaje.

[3] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**". Fondo Polar-UCAB, Caracas 1995, Pág.9.3

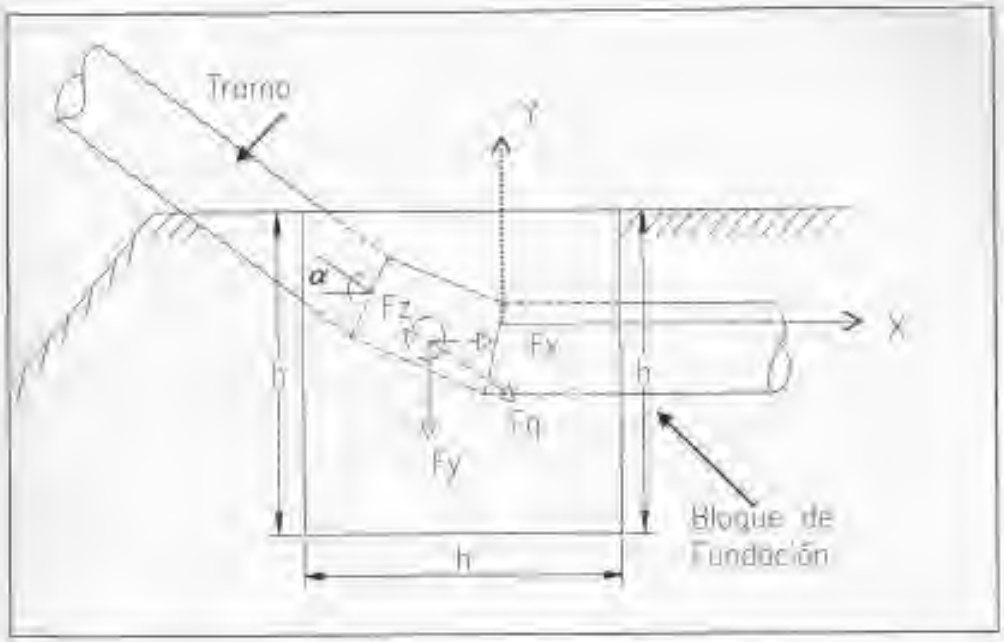


Figura Nº 9. Bloque de Fundación

1.2 Verificación

1.2.1 Generalidades

El enfoque de la verificación consiste en realizar todos los cálculos hidráulicos y estructurales haciendo uso de un espesor inicial introducido por el usuario con su respectivo diámetro para asegurarse que no sea inferior al valor que calcula el programa cerciorándose así de que cumpla con lo previsto en las consideraciones estudiadas. Esto implica que puede darse el caso en el cual el espesor introducido por el usuario sea superior al necesario pero que de igual forma verifica.

Adicionalmente al diámetro y el espesor, el usuario debe introducir la luz libre y la altura del puente tubo, no así la sección transversal topográfica ya que en el caso de la verificación, estos datos no son necesarios debido a que a partir de estos se obtiene la luz libre, dato que el usuario previamente conoce. También es necesario el valor de la carga viva, la carga transversal, la presión de trabajo y la sobrepresión.

1.2.2 Alcances

El uso del programa "DAPT 1.0" en modo de verificación se limita a informarle al usuario si el puente tubo en arco es un diseño seguro, y por lo tanto cumple con las consideraciones establecidas en el programa para el diseño. Durante este procedimiento, no se generan planos de AutoCAD, ni predimensionado de las fundaciones directas.

CAPÍTULO II

II. Cálculos asociados a la Hidráulica

Para el análisis estructural del puente tubo se debe partir con un valor inicial del espesor, este valor será el menor valor encontrado que satisfaga los requerimientos de las solicitaciones generadas en la tubería como resultado del efecto de la presión interna del fluido y la presión externa que pudiera generar colapso por vacío en la tubería.

Si bien es cierto que para el diseño de tuberías de acero se deben tomar en consideración las acciones dinámicas originadas por una serie de factores, tales como:

- Presión Interna
- Presión de colapso
- Peso de la tubería y del fluido
- Fuerzas derivadas de los gradientes térmicos
- Solicitaciones relacionadas con las condiciones particulares del apoyo
- Solicitaciones por carga viva
- La acción de cargas transversales

En el presente capítulo sólo se hará mención a las consideraciones relacionadas con el funcionamiento hidráulico del puente tubo, como lo son la presión interna del fluido y la presión que ocasiona un colapso en la tubería, en el capitulo III se analizan las restantes consideraciones del diseño estructural.

2.1 Presión Interna

En el diseño de tuberías fabricadas con acero se puede formular una simple relación entre las variables, indicadas en la figura N°10:

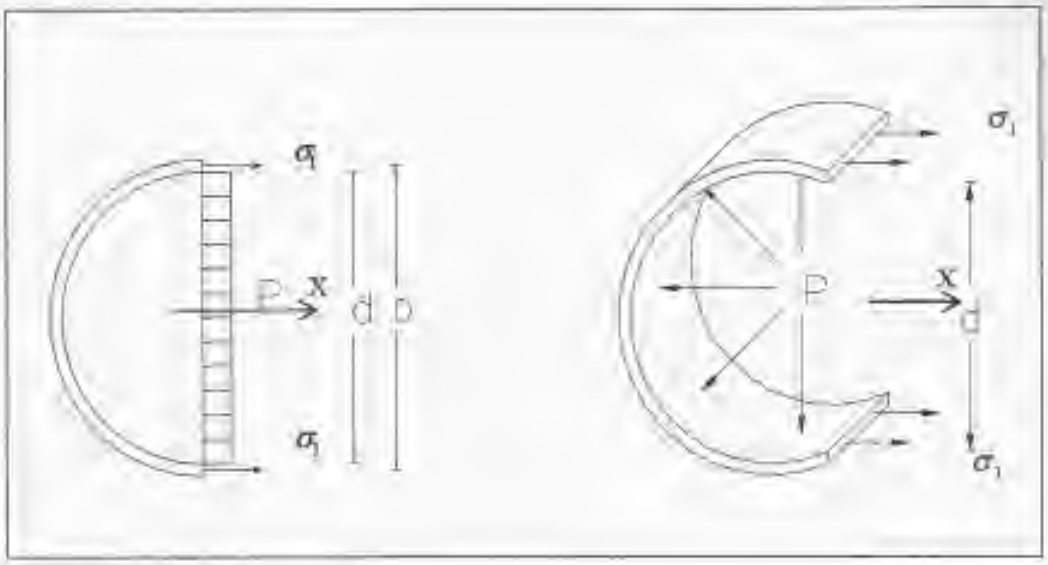


Figura N° 10. Diagrama del cuerpo Libre del Corte de una tubería

Al aislar la tubería y pasar planos perpendiculares al eje del cilindro obtenemos el diagrama del cuerpo libre del elemento para una sección de largo unitario, donde encontramos los esfuerzos normales, que al multiplicarlos por sus respectivas áreas mantienen el elemento en equilibrio con la presión interna actuante.

Considerando el equilibrio según X y admitiendo la distribución uniforme del esfuerzo tangencial "σ", se pueden calcular las siguientes fuerzas horizontales: la presión interna p multiplicada por el área proyectada dL genera la fuerza que actúa sobre el elemento cilíndrico y la fuerza desarrollada por los esfuerzos en el aro serán los esfuerzos σ multiplicados por el área correspondiente $(D-d)L$, se tiene entonces:

$$P \times dL - \sigma(D-d)L - \sigma(D-d)L = 0$$
$$P \times dL - 2\sigma(D-d)L = 0$$
$$-2\sigma(D-d)L = -P \times dL$$

Sustituyendo *(D-d)* por el espesor e y simplificando $L=1$, tendremos:

$$e = \frac{Pd}{2\sigma} \rightarrow (II-1)^{4}$$

Esta ecuación es aplicable siempre que e/d sea aproximadamente igual o menor que 0,02. Este valor es una aproximación que garantiza que existe una relación de esbeltez tal que el cilindro es de pared delgada y por lo tanto el esfuerzo puede asumirse constante en el aro, de lo contrario tendríamos que trabajar con un elemento de pared gruesa. Por otra parte, para compensar ciertas diferencias observadas entre los esfuerzos medidos y los calculados por la ecuación *II-1* se prefiere utilizar la denominada ecuación de Barlow:

$$\boxed{e = \frac{PD}{2\sigma_t} \rightarrow (II-2)}^{5}$$

En la cual P será la presión de trabajo y D igual al diámetro externo de la tubería, el esfuerzo de trabajo σ_t para tuberías de acero se define como el menor de los siguientes valores:

$$\sigma_t = k_1 \sigma_{f\,6}$$
$$\sigma_t = k_2 \sigma_r$$

[4, 5] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**". Fondo Polar-UCAB, Caracas 1995, pág 8.5

Donde, σ_f punto cedente o de fluencia del acero y σ_r esfuerzo de rotura.

Los valores de k_1 y k_2 vienen relacionados con ciertas consideraciones de institutos como el American Water Works Association (AWWA), el cual recomienda que en el caso de solicitaciones derivadas de la operación normal de la línea de flujo k_1=0,50 y para las solicitaciones ocasionales tales como las generadas por el golpe de ariete k_1=0,75. En Venezuela se adopta el menor valor de σ_t calculado utilizando k_1=0,60 y k_2=0,40, en solicitaciones derivadas del régimen permanente.

Para efectos del presente trabajo especial de grado se considera únicamente la fluencia del acero, esto permite tener un rango de seguridad a la hora de diseñar ya que evitamos llevar la resistencia del acero hasta su esfuerzo máximo. Sin embargo se considerarán dos valores de k, que serán k_{11} o k_{pn}, donde el subíndice se refiere a la presión normal de trabajo el cual tiene un valor pre-establecido de 0,60 pero que el usuario puede modificar y un k_{12} o k_{sp}, donde el subíndice se relaciona con la sobrepresión y tendrá un valor establecido de 0,75 que de igual forma puede ser alterado por el usuario según su propio criterio. Finalmente la obtención del espesor vendrá dada por el valor que satisfaga las dos expresiones:

$$e_{pn} = \frac{P_n D}{2\sigma_{pn}} \quad \text{Y} \quad e_{sp} = \frac{P_p D}{2\sigma_{sp}}$$

Donde: $\sigma_{pn} = k_{pn}\sigma_f \quad , \sigma_{sp} = k_{sp}\sigma_f$

[6] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**". Fondo Polar-UCAB, Caracas 1995, pág 8.6

e_{pn} = Espesor calculado para la presión normal de trabajo.

P_n = Presión normal de trabajo.

e_{sp} = Espesor calculado para sobrepresiones eventuales.

P_n = Sobrepresión eventual.

2.2 Colapso por vacío

Las tuberías estarán sujetas a una presión neta que actúa radialmente hacia el eje de la misma cuando la presión atmosférica es superior a la presión interna a la que esta sometida la tubería. Cuando esto ocurre los esfuerzos a los que esta sometido el conducto pueden ocasionar un colapso por inestabilidad elástica.

Para calcular la presión a la que está asociada el colapso en tuberías circulares y expuestas al ambiente atmosférico se utilizan las siguientes formulas:

$$P = \frac{2E}{1-\mu^2}\left(\frac{e}{D-e}\right)^3 \quad \text{para } \frac{e}{D} < \sqrt{\frac{(1-\mu^2)\sigma_f}{E}} \qquad \textit{(II-3)}^7$$

$$P = \frac{2\sigma_f\left(\frac{D}{e}-1\right)}{\sqrt{\left(\frac{D}{e}\right)^4 - \left(\frac{D}{e}\right)^3 + 5\left(\frac{D}{e}\right)^2 - 8\frac{D}{e} + 4}} \quad \text{para } \frac{e}{D} > \sqrt{\frac{(1-\mu^2)\sigma_f}{E}} \qquad \textit{(II-4)}^8$$

Donde:

P: Presión de Colapso

[7, 8] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**" Fondo Polar-UCAB, Caracas 1995, pág 8.8

E: Módulo de elasticidad del material

e : Espesor del conducto

D: Diámetro exterior de la tubería

μ : Relación de Poisson

σ_f : Punto efluente del acero

Por otra parte, gracias a los experimentos realizados por Stewart, se propone la ecuación (3) para el cálculo de la presión crítica de colapso en tuberías de acero expuestas. Dicha expresión toma en cuenta el efecto de eventuales ovalizaciones sobre la resistencia al colapso de la tubería y la tolerancia de la variación del espesor en los procesos de laminación del acero, actualmente esta expresión esta recomendada por la AWWA, siendo la más útil para el cálculo.

$$P_{cr} = 3.252.000\left(\frac{e}{D-e}\right)^3 \qquad \textit{(II-5)}^8$$

De las ecuaciones *(II-3)* y *(II-5)* se obtienen las relaciones *(II-6)* y *(II-7)* entre el espesor y el diámetro requeridas por tuberías de acero expuestas, sólo sujetas a la acción exterior de la presión atmosférica, para hacerlas resistentes al colapso. Se ha supuesto para el acero μ = 0,30 y Modulo de elasticidad del acero = 2,1x10^6 Kgf/cm^2; la aproximación de D-e=D y Pat=1Kgf/cm2:

$$\frac{e}{D} = 0{,}006 \qquad \textit{(II-6)} \text{ y } \frac{e}{D} = 0{,}0066 \approx 0{,}007 \quad \textit{(II-7)}$$

De acuerdo a lo anterior y a lo expresado en el aparte 2.1, la relación espesor/diámetro de la tubería debe ser tal que satisfaga la siguiente expresión:

[8] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**". Fondo Polar-UCAB, Caracas 1995, pág 8.9

$$0{,}007 < \frac{e}{D} \leq 0{,}02$$

CAPÍTULO III

3.1 Introducción

El puente tubo en arco se emplea como sistema estructural debido a que permite reducir significativamente los momentos de flexión que de otro modo existirían si se empleara por ejemplo, un alineamiento rectilíneo de la misma luz para atravesar el vano libre sin apoyos. De este modo, se disminuyen considerablemente los momentos a costa de aumentar los valores de carga axial. Sin embargo, es más económico desde el punto de vista de la sección de tubería, soportar cargas por medio de esfuerzos axiales uniformes que por esfuerzos de flexión, aunque en el caso de las cargas axiales se requiere considerar el pandeo elástico.

El modelo para el cálculo estructural del puente tubo en arco, puede asimilarse a una viga curva empotrada en ambos extremos y sometida a un estado tridimensional de esfuerzos. El puente tubo en arco estará sometido esencialmente a las acciones provenientes del peso propio de la tubería y del fluido en su interior, así como acciones eventuales provenientes de cargas variables que se encuentran contenidas tanto en el plano de la estructura como en el plano transversal a la misma. Las fuerzas debidas al peso propio y al fluido son cargas gravitacionales distribuidas en la longitud del arco, y están contenidas en el plano del mismo, mientras que por otro lado, resultan de gran importancia para el diseño las acciones de viento o sismo que actúan en dirección transversal al plano que contiene el puente tubo. Para efectos de este trabajo especial de grado, la introducción de cargas transversales representarán las originadas por el efecto del viento, sin embargo, cuando se le asignen valores grandes a estas cargas, las mismas podrán modelarse en el programa como cargas de sismos. Adicionalmente es importante destacar que para efectos de este trabajo, sólo

se analizará el caso de carga con la tubería llena por el fluido y bajo las condiciones ambientales propias de Venezuela, por tal razón no se consideran los esfuerzos generados sobre la tubería vacía a causa de la deformación de origen térmico debidos a cambios de temperatura.

3.2 Análisis Estructural del puente tubo en arco

El problema en estudio, es un sistema hiperestático del cual se requiere conocer los esfuerzos máximos para el diseño. Las estructuras convencionales tales como armaduras, vigas, pórticos rígidos, etc., pueden definirse mediante un ensamblaje de elementos estructurales unidos entre si en un numero finito de puntos discontinuos llamados nodos, y cargadas solamente en estos puntos. De forma similar, podemos dividir la longitud total del arco en segmentos rectilíneos de menor longitud, de tal modo, que si la longitud de los elementos es suficientemente pequeña, el modelo se aproximará con bastante precisión a la realidad, conformándose un polígono inscrito en la directriz curva del arco, con elementos de directriz recta conectados a través de los nodos que configuran el modelo. Debido a la naturaleza tridimensional del problema, se generarán seis componentes en las reacciones de los apoyos empotrados: tres componentes de fuerzas y momentos en las direcciones del sistema global de coordenadas "X", "Y", "Z". De manera análoga, cada uno de los elementos que configuran el modelo poseerá las mismas 6 componentes de fuerza en cada nodo.

Si bien en el análisis del problema podemos considerar que el arco está contenido en un mismo plano, el problema es tridimensional debido a la posibilidad de considerar fuerzas normales a dicho plano. Las condiciones que se deben cumplir para el equilibrio son $\sum F_{x,y,x} = 0$ y $\sum M_{x,y,z} = 0$, Si el problema fuese plano (como por ejemplo el plano XY), se podrían realizar las siguientes simplificaciones:

$\sum F_x = 0, \sum F_y = 0$ y $\sum M_z = 0$ y utilizar la simetría de la estructura para resolver la hiperestaticidad del sistema que viene dada por las seis reacciones desconocidas y las tres ecuaciones de equilibrio disponibles. Debido al efecto tridimensional que originan las cargas transversales no es posible analizar el problema utilizando las deducciones anteriores y se recurre entonces a la aplicación de un análisis matricial para resolver la hiperestaticidad existente.

3.3 Análisis Matricial por el Método de los desplazamientos

La compatibilidad aplicada en el método de los desplazamientos consiste en que la geometría de la deformación debe ser tal que los elementos de la estructura se mantengan unidos en los puntos nodales; esto es, que las deformaciones "d_i" de sus elementos se relacionan compatiblemente con los desplazamientos nodales "D_i" Sea "a_{ij}" el valor de la deformación del elemento "d_i" producida por un desplazamiento nodal unitario "D_i". El valor total de la deformación de cada elemento, producida por todos los desplazamientos nodales puede escribirse en la forma:

$$\left.\begin{aligned} d_1 &= a_{11}D_1 + a_{12}D_2 + \cdots a_{1j}D_j + \cdots a_{1n}D_n \\ d_2 &= a_{21}D_1 + a_{22}D_2 + \cdots a_{2j}D_j + \cdots a_{2n}D_n \\ d_j &= a_{j1}D_1 + a_{j2}D_2 + \cdots a_{jj}D_j + \cdots a_{jn}D_n \end{aligned}\right\} \cdots (III-1)$$

o en notación matricial es:

$$\begin{bmatrix} d_1 \\ d_2 \\ d_n \end{bmatrix} = \begin{bmatrix} a_{11} + a_{12} + \cdots a_{1n} \\ a_{21} + a_{22} + \cdots a_{2n} \\ a_{m1} + a_{m2} + \cdots a_{mn} \end{bmatrix} \times \begin{bmatrix} D_1 \\ D_2 \\ D_m \end{bmatrix} \cdots (III-2)$$

en forma matricial $[d]=[A]\times[D]$, donde:

$[A]$ = Matriz de transformación de desplazamientos, que relaciona las deformaciones internas de los elementos con los desplazamientos nodales.

"d_i" son las deformaciones internas de los nodos de los miembros de una estructura expresadas según el sistema local de coordenadas del elemento, y "D_i" los desplazamientos externos nodales en coordenadas globales.

La matriz $[A]$ es generalmente rectangular. Es una simple transformación geométrica de coordenadas que nos expresa la compatibilidad de los desplazamientos de un sistema.

El coeficiente de rigidez constituye una relación entre la fuerza y el desplazamiento unitario. Aplicando el principio de superposición, se puede expresar la componente de la fuerza en cualquier punto de un sistema, en función de un conjunto de desplazamientos impuestos.

Si tenemos que "q_i" son las acciones en extremos de miembros (acciones internas):

$$\left.\begin{aligned} q_1 &= k_{11}d_1 + k_{12}d_2 + \cdots k_{1j}d_j + \cdots k_{1n}d_n \\ q_2 &= k_{21}d_1 + k_{22}d_2 + \cdots k_{2j}d_j + \cdots k_{2n}d_n \\ q_j &= k_{j1}d_1 + k_{j2}d_2 + \cdots k_{jj}d_j + \cdots k_{jn}d_n \end{aligned}\right\}\cdots(III-3)$$

que en forma matricial se puede escribir de la siguiente forma:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_n \end{bmatrix} = \begin{bmatrix} k_{11} + k_{12} + \cdots k_{1n} \\ k_{21} + k_{22} + \cdots k_{2n} \\ k_{m1} + k_{m2} + \cdots k_{mn} \end{bmatrix} \times \begin{bmatrix} d_1 \\ d_2 \\ d_m \end{bmatrix} \cdots (III-4)$$

o simplemente:

$$[q] = [k] \times [d] \qquad (III\text{-}5)$$

donde k_{ij} es un coeficiente de rigidez, el cual se define como la fuerza que aparece en la coordenada "i" debido a un desplazamiento unitario en la coordenada "j", cuando todos los demás desplazamientos están impedidos, y $[k]$ es la matriz de rigidez del elemento.

Si denominamos "Q_i" las acciones externas en los nodos, tenemos:

$$[Q] = [K] \times [D] \qquad (III\text{-}6)$$

en donde:

"Q_i": son las fuerzas externas aplicadas en los nodos.

"D_i" · desplazamientos nodales externos.

$[K]$: matriz de rigidez de la estructura.

Para expresar las fuerzas nodales en función de los correspondientes desplazamientos nodales recurrimos al método del trabajo virtual.

La ley del trabajo virtual establece que si un cuerpo deformable está en equilibrio y permanece en equilibrio cuando el cuerpo sufre un pequeño desplazamiento virtual, el trabajo virtual exterior realizado por el sistema de fuerzas externas actuando sobre el cuerpo es igual al trabajo de deformación virtual interno realizado por las fuerzas internas. La palabra virtual indica que la acción que produce el desplazamiento es independiente del sistema de fuerzas original y se llama trabajo virtual al realizado por el sistema de fuerzas original durante el desplazamiento virtual. El principio del trabajo virtual se aplica principalmente en problemas estáticos de cuerpos deformables.

Una vez establecido este concepto podemos aplicar el mismo a las fuerzas y a los desplazamientos nodales.

$$[Q^T]\times[D]=[q^T]\times[d]\Leftrightarrow[D^T]\times[Q]=[d^T]\times[q]$$
$$[d]=[A]\times[D]\rightarrow[d^T]=[(A\times D)^T]=[D^T]\times[A^T]$$
$$Luego:\rightarrow[Q]=[A^T]\times[q]$$

De las siguientes ecuaciones podemos demostrar la relación entre la matriz de rigidez del miembro y la matriz de rigidez de la estructura.

$$[Q]=[A^T]\times[q]\cdots(1)$$
$$[Q]=[K]\times[D]\cdots(2)$$
$$[q]=[k]\times[d]\cdots(3)$$

Sustituyendo (1) en (2) tenemos:

$$[A^T]\times[q]=[K]\times[D]\Rightarrow[A^T]\times[k]\times[d]=[K]\times[D]$$
$$pero.\rightarrow[d]=[A]\times[D]$$
$$Obteniendose:\rightarrow[K]=[A]^T\times[k]\times[A]$$

$$[K]=[A^T]\times[k]\times[A]$$

Siendo $[K]$ la llamada matriz global de rigidez, o también matriz de rigidez de la estructura, que relaciona directamente las fuerzas nodales con los desplazamientos nodales.

Para aplicar el método de las rigideces o de los desplazamientos en la solución de una estructura hiperestática se necesitan determinar primero las componentes independientes de los desplazamientos (lineales y angulares) que se desconocen y que constituyen las incógnitas del problema. Las fuerzas internas de la estructura se pueden expresar en función de estos desplazamientos.

Por cada componente de desplazamiento desconocida, se establece una ecuación de equilibrio en función de las fuerzas externas conocidas y de las fuerzas internas no conocidas, las cuales están expresadas en términos de los desplazamientos. Se forma un sistema de ecuaciones cuyo número es igual al número de componentes de desplazamiento desconocidos.

La solución del sistema de ecuaciones permite conocer los valores de los desplazamientos, con los cuales se pueden calcular las fuerzas internas.

Un resumen de los pasos a aplicar para la resolución del problema de ingeniería estructural, podría ser el siguiente:

- ✓ Suponemos desplazamientos incógnitas en cada uno de los nodos que conforman el sistema. En nuestro caso existirán 3 componentes de desplazamiento en las direcciones "x", "y" "z" globales, con lo que tenemos tres grados de libertad (desplazamientos incógnitas) por cada nodo.

- ✓ No se supondrán desplazamientos en los puntos con coordenadas de desplazamiento impedidos (apoyos con coordenadas restringidas). Así, un extremo empotrado no puede tener ningún desplazamiento. Esta hipótesis es necesaria, puesto que de otro modo las determinantes de la matriz de rigidez serían nulos, y el inverso de la matriz sería indefinido, lo que corresponde a un sistema inestable.

- ✓ En cada una de las coordenadas de desplazamiento supuestas habrá cargas actuantes las cuales serán nulas o tendrán valores conocidos. Habrá n cargas, correspondientes con los n desplazamientos desconocidos.

- ✓ Se calculan los coeficientes de rigidez k_{ij} de la ecuación $III-4$. Esto puede programarse y calcularse. Estos serán los elementos de una matriz simétrica $n \times n$

- ✓ Resolvemos el sistema de ecuaciones $[D]=[K]^{-1} \times [Q]$ determinando los n desplazamientos supuestos en el primer paso.

- ✓ Obtenemos las deformaciones internas del miembro a partir de los desplazamientos calculados: $[d]=[A]\times[D]$

- ✓ Se calculan las fuerzas internas de miembros a partir de la matriz de rigidez del elemento $[q]=[k]\times[d]$

Para un mayor entendimiento y comprensión de las rutinas usadas en el cálculo estructural, remitirse al anexo 1

3.4 Esfuerzos

Una vez culminado el análisis matricial se obtiene como resultado la siguiente matriz:

$$\begin{pmatrix} Nodo. & F_x & F_y & F_z & M_x & M_y & M_z \\ i & . & . & . & & & . \\ j & . & . & & & & . \\ i & . & . & . & . & . & . \\ & & & & & & \end{pmatrix}$$

Con estos valores se procede al cálculo de esfuerzos de la siguiente forma:

- ✓ Esfuerzo por compresión: $\frac{F_x}{A}$
- ✓ Esfuerzo por corte: $\frac{\sqrt{F_y^{\,2} + F_z^{\,2}}}{A}$
- ✓ Esfuerzo Torsor: $\frac{M_x}{J_o} \times R$
- ✓ Esfuerzo Flexionante: $\frac{\sqrt{M_y^{\,2} + M_z^{\,2}}}{I} \times R$

Se procede entonces a un proceso de revisión de la sección tubular, el cual verificará el cumplimiento de la estructura bajo las condiciones actuantes.

Donde:

A= Área de la sección

J= Momento polar de Inercia

I= Inercia de la sección

R= Radio

3.4.1 Revisión de la sección tubular sometida a esfuerzos combinados mediante la aplicación del "Criterio Tentativo para aplicaciones estructurales y conductos de acero" American Iron and Steel Institute (AISI) 1974 [9]

3.4.1.1 *Relación de Esbeltez Efectiva*

En la determinación de la relación de esbeltez $\frac{K \cdot L}{r}$ de un miembro a compresión cargado axialmente, la longitud se tomará como una longitud efectiva de KL y r corresponde al radio de giro de la sección transversal. A continuación se presentan disposiciones para el valor apropiado del factor de longitud efectiva K.

La relación de esbeltez para secciones tubulares en compresión no excederá de 200.

3.4.1.2 *Compresión Axial*:

Para secciones tubulares con $\frac{D}{t} < \frac{232000}{F_y}$

Para $C_c < \frac{KL}{r}$: $F_a = \frac{\left[1 - \frac{(KL/r)^2}{2C_c^{\,2}}\right] \cdot F_{ey}}{FS}$ $\qquad FS = \frac{5}{3} + \frac{3 \cdot (KL/r)}{8C_c} - \frac{(KL/r)^3}{8C_c^{\,3}}$

[9] C.V.G Siderurgica del Orinoco, C.A. (SIDOR), Manual de Proyectos. Estructuras de Acero Tomo 1 (1982) (2da ed.).Puerto Ordaz,Pág. I-429

Donde:

$$C_c = \sqrt{\frac{2 \cdot \pi^2 \cdot E}{F_y}} \qquad r = \sqrt{\frac{I}{A}}$$

E: módulo de Young.

D: diámetro.

t: Espesor de la tubería.

r: Radio de Giro.

K: Coeficiente de Esbeltez.

A: Área.

I: Inercia de la sección.

F_y: Esfuerzo de fluencia.

F_{ey}: Esfuerzo de fluencia efectivo (ver sección 3.5.6.1).

Para $\frac{KL}{r} < C_c < 200$: $F_a = \frac{12 \cdot \pi^2 \cdot E}{23 \cdot (KL/r)^2}$

Para secciones tubulares con $\frac{232000}{F_y} < \frac{D}{t} < \frac{914000}{F_y}$, F_a no excederá:

$$F_a = \frac{46500}{D/t} + 0{,}40 \cdot F_{ey}$$

F_a calculado de acuerdo con sección 3.5.2.1.

3.4.1.3 *Flexión:*

Secciones con $\frac{D}{t} < \frac{232000}{F_y}$, $F_b = 0{,}72 F_{ey}$

Secciones con $\frac{232000}{F_y} < \frac{D}{t} < \frac{914000}{F_y}$, $F_b = \frac{46500}{D/t} + 0{,}40 \cdot F_y$

3.4.1.4 *Corte por Flexión*

Secciones con $\frac{D}{t} < \frac{851000}{F_y}$, $F_v = 0{,}40 \cdot F_{ey}$

Esfuerzo por Corte $\tau = 2 \cdot \frac{V}{A}$

3.4.1.5 *Esfuerzo de Corte por Torsión:*

Para secciones con: $\frac{D}{t} < \frac{1{,}77\left(\frac{E}{F_y}\right)^{\frac{4}{3}}}{\left(\frac{L}{D}\right)^{\frac{5}{2}}}$ ó $\frac{D}{t} < \left(\frac{E}{F_y}\right)^{\frac{2}{3}}$, $F_v = 0{,}40 \cdot F_{ey}$

Donde "L" representa la longitud total del arco sometido a torsión

Para secciones con $\frac{D}{t}$ mayores a los valores limites:

$$F_v = \frac{0{,}40 \cdot E}{\left(\frac{D}{t}\right)^{\frac{3}{2}}}$$

3.4.1.6 *Combinaciones de Carga*

3.4.1.6.1) ***Presión interna***. Para secciones tubulares que estén inevitablemente sometidas a presión p interna significativa, el diseño por carga axial, flexión o torsión se basará en la tensión de fluencia efectiva apropiada listada a continuación:

Por cargas que originan tensiones normales:

$$F_{ey} = \left(1 - \frac{pD}{2tF_y}\right) \cdot F_y$$

Para cargas que originan tensiones de corte:

$$F_{ey} = \sqrt{1 - \left(\frac{pD}{2tF_y}\right)^2} \cdot F_y$$

3.4.1.6.2) ***Presión Externa.*** Para secciones tubulares redondas que estén inevitablemente sometidas a presión p externa, la tensión admisible de compresión axial, flexión o torsión será reducida por un factor de:

$$1 - \frac{p}{P_{crit}}$$

Donde, $$P_{crit} = \frac{1{,}32E}{\left(\frac{D}{t}\right)^3}$$

3.4.1.6.7 *Tensiones Combinadas*

En las expresiones anteriores se ha reemplazado el valor de F_y por el valor de fluencia efectiva F_{ey} para tomar en cuenta la existencia de presión interna o externa. La presión externa pudiera también asimilarse a una succión o presión interna negativa.

a) $F_{ey_1} = \left(1 - \frac{PD}{2tF_y}\right) \cdot F_y$ Para el cálculo de F_a y F_b

b) $F_{ey_2} = \sqrt{1 - \frac{PD}{2tF_y}^2} \cdot F_y$ Para el cálculo de F_v

Los miembros sometidos simultáneamente a tensiones de **carga axial, flexión y torsión** se dimensionarán para cumplir con los siguientes requerimientos:

$$\left\{ \begin{array}{l} \dfrac{f_a}{F_a} + \dfrac{C_m \cdot f_b}{\left(1 + \dfrac{f_a}{F_e'}\right) \cdot F_b} + \left(\dfrac{f_v}{F_v}\right)^2 \leq 1.0 \\ \\ \dfrac{f_a}{0{,}6 \cdot F_{ey}} + \dfrac{f_b}{F_b} + \left(\dfrac{f_v}{F_v}\right)^2 \leq 1.0 \end{array} \right\}$$

Donde,

$C_m = 0{,}85$ (para miembros cuyos extremos se encuentren empotrados)

$$F_{é} = \frac{12 \cdot \pi^2 \cdot E}{23 \cdot (KL/r)^2}$$

f_a=Esfuerzo por compresión

f_v= Esfuerzo por flexión + Esfuerzo por torsión

f_b= Esfuerzo cortante

Al cumplirse las relaciones anteriores se garantiza la integridad estructural del puente tubo en arco y se concluye el proceso de verificación del mismo.

CAPÍTULO IV

4 Desarrollo de la aplicación

4.1. El Programa

El programa "DAPT 1.0", se desarrolló en Lenguaje Visual Basic 6.0, en conjunto con las aplicaciones Microsoft Excel y AutoCAD 2004, ya que gracias a estas, se logra crear una interfaz gráfica bastante amigable y útil para el usuario. A diferencia de otros programas desarrollados para el cálculo de estructuras en la rama de Ingeniería, "DAPT 1.0" permite al usuario el manejo gráfico de los datos introducidos y de esta manera permite la verificación visual del diseño, obteniéndose resultados de alta calidad en el diseño de puentes tubo en arco.

El programa se basa en todas las consideraciones expuestas en los capítulos anteriores y tiene como objeto el diseño de puentes tubo en arco de una manera rápida y segura en cualquier tipo de situación y de topografía del cauce.

La aplicación cuenta con una serie de módulos de procedimiento o sub-rutinas, que son llamadas cada vez que estas sean necesarias. Así mismo, posee una base de datos desarrollada en Microsoft Excel, que contiene más de 100 registros de tuberías comerciales, de distintos fabricantes y que son usadas actualmente con mucha frecuencia en el país.

Estos registros son de mucha utilidad a la hora de realizar el diseño, ya que evitan que el usuario se vea obligado a consultar catálogos de tuberías durante el proceso de diseño. Sin embargo, el usuario puede agregar información adicional a la de la base de

datos y se le da la posibilidad de realizar una verificación de su diseño introduciendo los datos correspondientes para dicho procedimiento.

4.2. Requerimientos del Sistema. Configuración

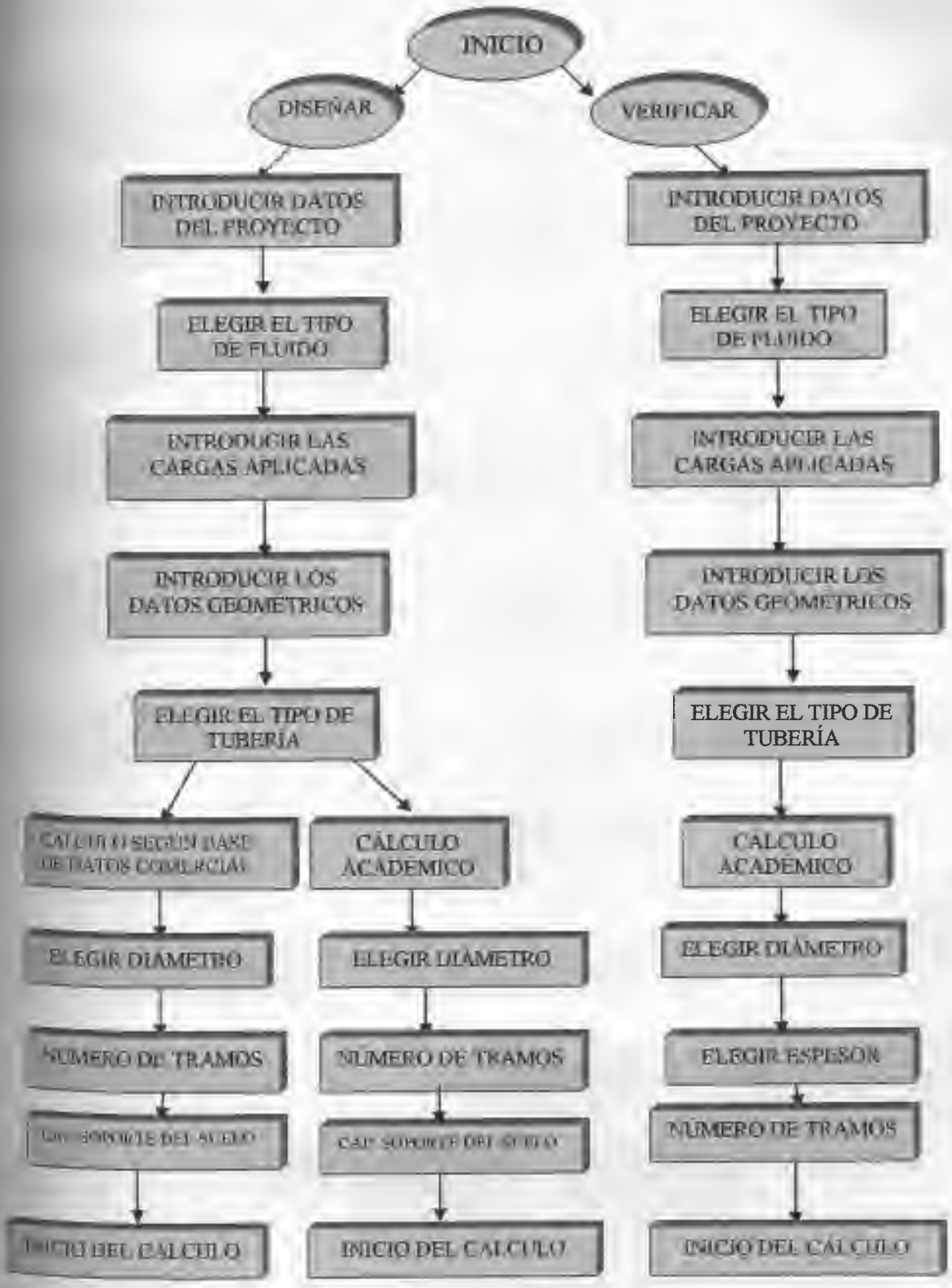
4.2.1 Hardware

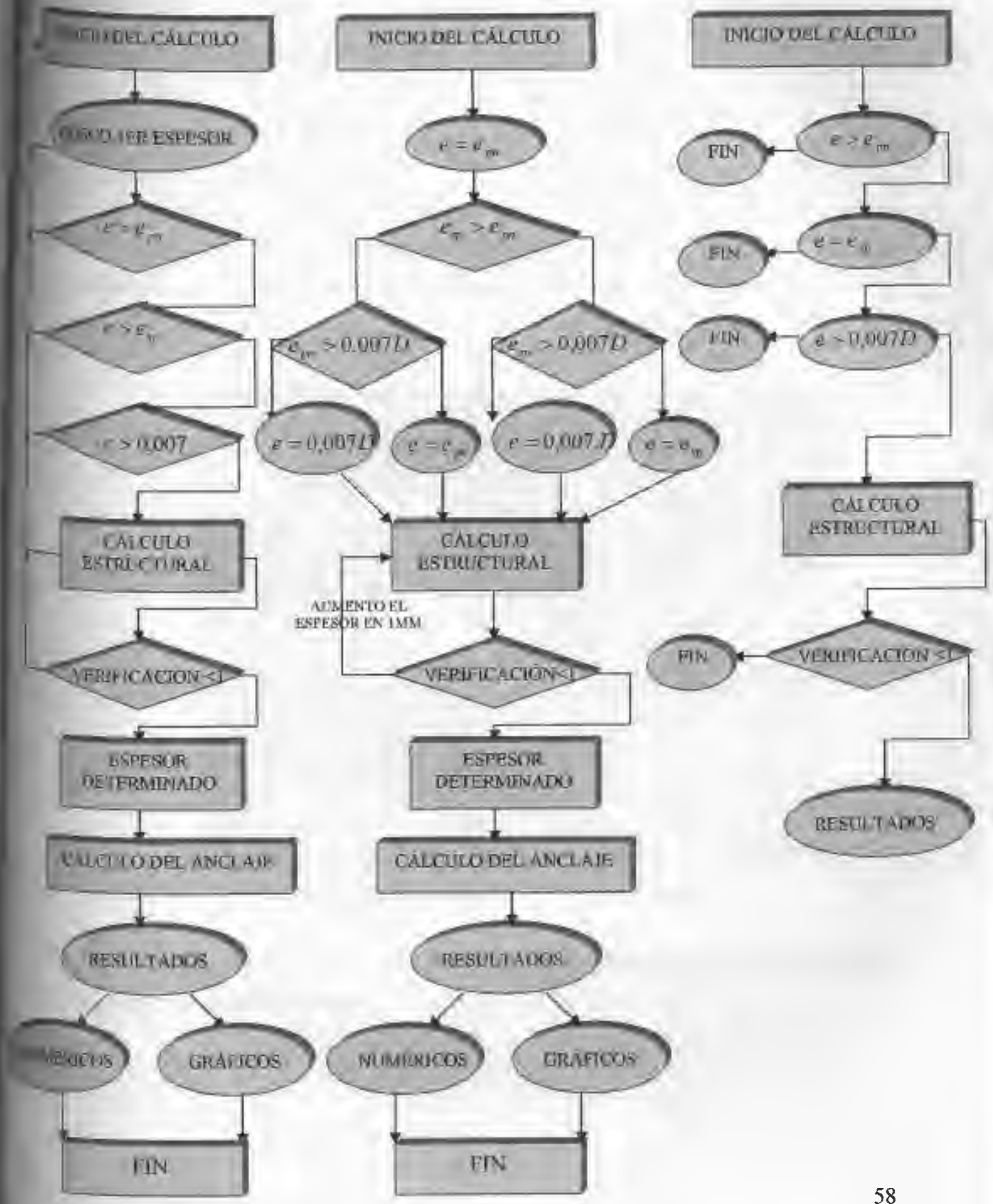
COMPONENTE	MÍNIMO	RECOMENDADO
PROCESADOR	INTEL 486DX 66MHZ	PENTIUM III 500MHZ O SUPERIOR.
MEMORIA	16MB	64MB O SUPERIOR.
ESPACIO DISPONIBLE	35MB	45MB O SUPERIOR
MONITOR	VGA (640X480, 256 COLOR)	SVGA (800X600 HIGH COLOR 16BITS)
CD –ROM	2X O SUPERIOR	12X O SUPERIOR
TARJETA DE AUDIO	NINGUNA	16BITS O SUPERIOR
MOUSE	DISPOSITIVO O APUNTADOR	MOUSE COMPATIBLE

4.2.2.Software

TIPO	VERSIÓN MÍNIMA	VERSIÓN RECOMENDADA
SISTEMA OPERATIVO	WINDOWS 95	WINDOWS 98 O SUPERIOR
APLICACIÓN	EXCEL 97,AUTOCAD 2004	EXCEL 2000, AUTOCAD 2004

4.3 *Flujograma*




INICIO DEL CÁLCULO
INICIO DEL CÁLCULO
INICIO DEL CÁLCULO
ESPESOR
$e = e_m$
$e > e_m$
FIN
$e > e_m$
$e = e_0$
FIN
$e > e_0$
$e > 0,007D$
FIN
$e > 0,007D$
$e_m > 0,007D$
$e_m > 0,007D$
$e > 0,007$
$e = 0,007D$
$e = e_m$
$e = 0,007D$
$e = e_m$
CÁLCULO ESTRUCTURAL
CÁLCULO ESTRUCTURAL
CÁLCULO ESTRUCTURAL
AUMENTO EL ESPESOR EN 1MM
VERIFICACIÓN<1
VERIFICACIÓN<1
FIN
VERIFICACIÓN<1
ESPESOR DETERMINADO
ESPESOR DETERMINADO
RESULTADOS
CÁLCULO DEL ANCLAJE
CÁLCULO DEL ANCLAJE
RESULTADOS
RESULTADOS
NUMÉRICOS
GRÁFICOS
NUMÉRICOS
GRÁFICOS
FIN
FIN

4.4. Estructura del Programa

4.4.1. *Entrada de Datos*

El Programa para el diseño de puentes tubos en arco "DAPT 1.0", esta elaborado de tal manera que el usuario pueda introducir los datos de una forma rápida, lógica y ordenada. A continuación se presentan las diferentes ventanas que guían al usuario durante el proceso de diseño, inicialmente encontramos la ventana introductoria:

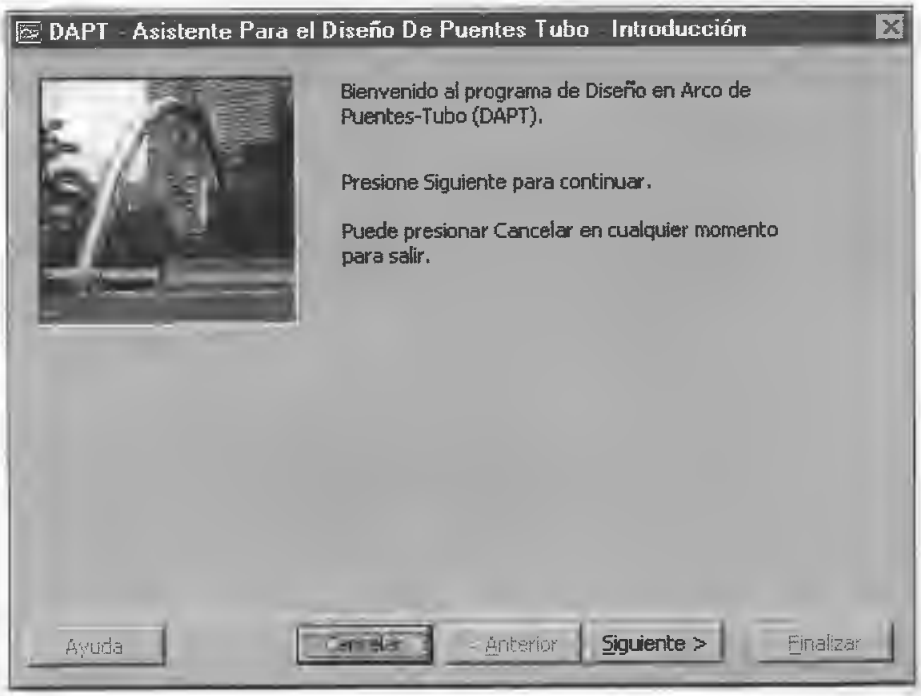


Figura N°13. Ventana introductoria del programa.

En esta ventana se le da la bienvenida al usuario al programa y las instrucciones para continuar o salir del diseño.

Posteriormente se selecciona el tipo de Operación que se desea realizar, tal como se muestra en la siguiente figura N° 12:

Figura N° 12. Ventana de selección del tipo de operación

La selección del tipo de operación es uno de los datos más importantes requeridos por el programa ya que de ella depende el tipo de cálculo que se va a realizar. La opción diseñar, crea un puente tubo en arco basado en la información que introduce el usuario y en las consideraciones estructurales e hidráulicas, para luego crear el plano de ingeniería básica en AutoCAD, mientras que la opción verificar, se asegura que el diámetro y el espesor que se utilizó en la construcción de un puente tubo en arco cumpla con los requerimientos necesarios sin generar algún tipo de plano en AutoCAD.

La siguiente ventana se utiliza para recopilar la información que describe el proyecto que se pretende realizar (Ver figura N° 13).

Figura N° 13. Ventana de entrada de datos

Se solicitan los siguientes datos del proyecto: el nombre, la localidad, la fecha y el sistema de unidades con el que se pretende trabajar. La selección del sistema de unidades es de mucha importancia ya que a partir de ese momento todos los datos que se introduzcan deben estar en las unidades específicas, de lo contrario se estaría incurriendo en un error grave en el diseño que estaría bajo la responsabilidad del usuario.

Es importante destacar que a pesar de elegir el sistema de unidades en esta ventana el programa solicita todos los datos en las unidades correspondientes indicando en un recuadro la elección realizada. A continuación se presenta la tabla que permite realizar las conversiones de un sistema de unidades a otro en algunos de los parámetros más importantes para el diseño de la estructura.

	DE:	MULTIPLICAR POR:	PARA OBTENER:
PESO ESPECIFICO	kg/m^3	0,0624	$libra/ft^3$
TEMPERATURA	°C	$\frac{9}{5}°C + 32$	°F
CAUDAL	m^3/seg	35,3147	ft^3/seg
DIAMETRO	mm	0,0394	in
ESFUERZO	kg/cm^2	14,2233	lib/in^2
COORDENADAS	m	3,2808	Ft
MASA	kg	2,2046	Lb
CARGA VIVA	kg/m	0,6720	lb/ft
PRESION MAXIMA	kg/cm^2	14,2233	Lib/in^2

Tabla N° 3. Transformaciones entre sistemas de unidades

Seguidamente se procede a la selección de datos importantes a la hora del diseño o verificación como lo son: el tipo de fluido y las cargas a las que se encuentra expuesto el puente tubo en arco (Ver figura N° 14).

El tipo de fluido que será transportado por el puente tubo en arco se presenta con una selección a partir de una lista establecida por el programa y que además cuenta con una base de datos que respalda cada opción con su respectivo peso especifico, en el caso particular del agua, se presenta la variación del peso especifico en función de la temperatura, es por esto que adicionalmente se debe especificar la temperatura a la cual se encuentra el fluido (ver figura N° 15), de no encontrarse el liquido que se desea transportar se puede utilizar la opción "*otro*" para un tipo de fluido no definido, siempre y cuando se introduzca el peso especifico correspondiente

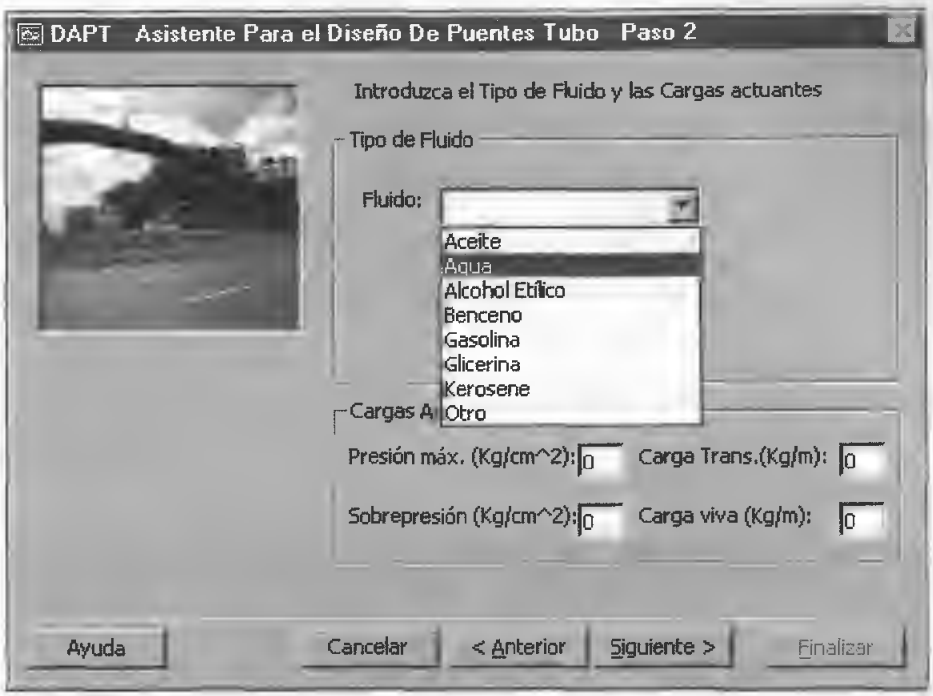


Figura N° 14 Ventana de selección del tipo fluido

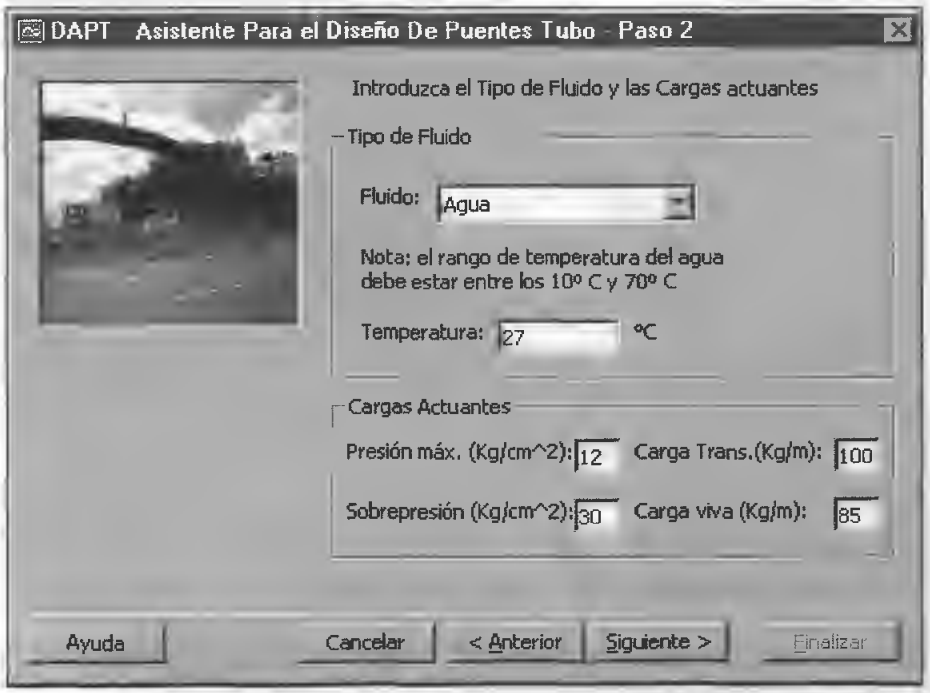


Figura N° 15. Ventana de detalle de la Temperatura y cargas aplicadas

También son solicitados en esta pantalla otros datos básicos en el diseño como lo son: la carga viva, la carga tramsversal, la presión de trabajo y en caso de que el usuario lo desee puede introducirse el valor de una sobrepresión que pueda ocurrir eventualmente en la tubería. El usuario debe introducir el valor en los campos adecuados, de lo contrario se tomaran como cero (0) y se continuará con el cálculo.

El siguiente paso corresponde a la sección transversal topográfica, tal como se muestra en la figura N° 16.

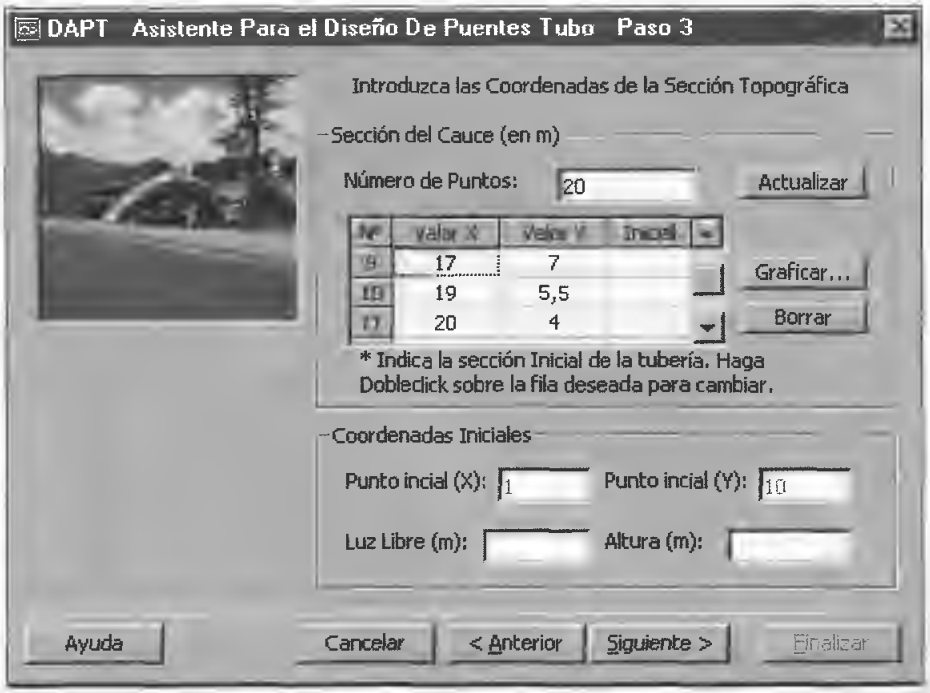


Figura N° 16. Ventana de datos de sección transversal topográfica

Uno de los atractivos del programa radica en su capacidad grafica, el usuario puede introducir las coordenadas de la sección topográfica en estudio (Ver figura N°16) y realizar una pre-visualización del diseño, permitiendo así ajustar ciertos parámetros antes de realizar cualquier tipo de cálculo, esto le da un carácter flexible al diseño y lo hace ameno para el usuario.

Es importante señalar que si el usuario esta realizando la operación de verificar esta pantalla tendrá inhabilitada la tabla de secciones topográficas y únicamente podrá introducir las coordenadas del punto de inicio, los valores de la altura y la luz libre.

La pantalla donde se introducen los valores de las coordenadas de los puntos que conforman la sección transversal topográfica, esta constituida por una tabla que tendrá establecido 3 secciones en las filas y 3 en las columnas, cuyos títulos serán X, Y e inicio. Las columnas de X e Y se encontrarán en blanco y la de inicio estará acompañada de un asterisco en la primera fila, este indica cual es el punto de coordenadas X, Y donde se inicia el arco y podrá ocupar cualquier casilla que el usuario desee, adicionalmente identifica el punto final ya que este tendrá igual valor de Y que el punto de inicio, en el caso de encontrar una situación donde existen varios valores de Y que concuerden con el Y de inicio, el programa adoptará como el correcto, el último valor de Y encontrado.

El número de puntos que definen la sección principal puede incrementarse hasta un máximo de 40, dependiendo de la información topográfica que posea el usuario y una vez establecido esto, se debe actualizar la tabla de manera que se incrementen las casillas hasta cumplir con el valor introducido.

Al oprimir el botón de graficar se llama al programa Excel, durante este procedimiento los valores que fueron introducidos por el usuario se cargan en filas y columnas y se crea un gráfico que representa la sección del cauce (ver figura N° 17). En este gráfico se identifica la luz libre que vencerá el puente tubo en arco con una línea color rojo y la topografía en color verde.

Luego de crearse el gráfico se puede regresar a la ventana anterior donde ahora se indica cual será la luz libre, la altura del arco y las coordenadas del punto inicial. La luz libre se calcula basándose en las consideraciones geométricas estudiadas en el Capitulo I y

la altura se establecerá por defecto como un quinto de la luz libre. Si el usuario lo desea podrá modificar este valor de la altura, llevándolo a un máximo de ½ de la luz libre.

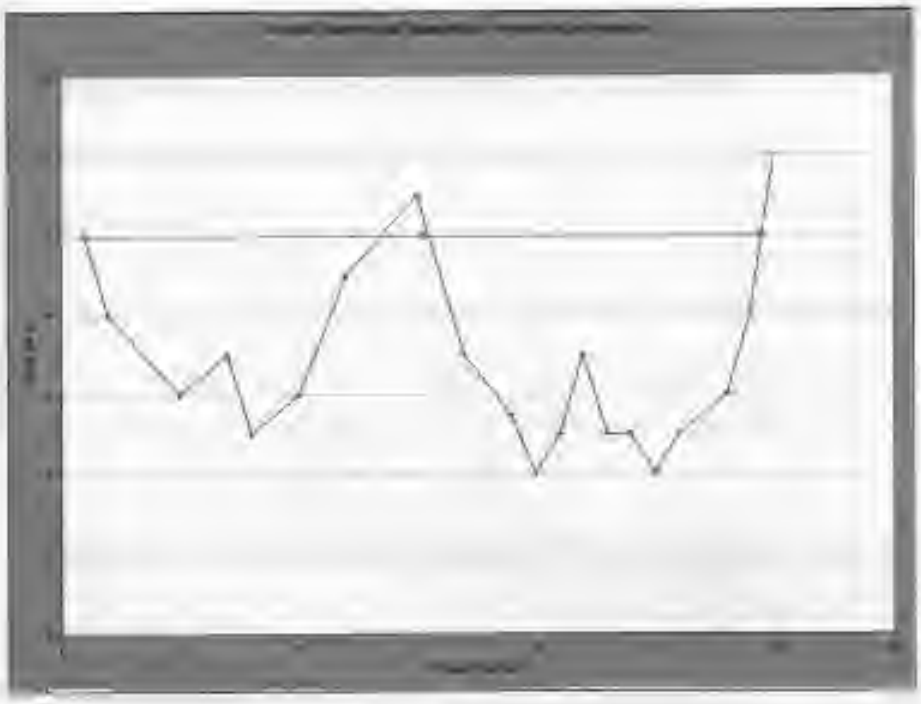


Figura N° 17. Ventana de Gráfico en Excel

De existir alguna duda para completar los datos se puede recurrir al menú de ayuda donde encontrará la información de todos los valores que solicita el programa y adicionalmente pensando en la comodidad del usuario, a cada solicitud se le asignó un rango de valores entre los cuales podría estar el valor de diseño, de esta manera se evitan la asignación de valores errados que reflejaran resultados incoherentes. Para acceder al menú de ayuda sólo debe presionar el botón en la parte inferior izquierda de cada ventana y se abrirá un documento de sólo lectura en la aplicación Word donde obtendrá los detalles para cada una de las pantallas

Los datos referentes al tipo de acero con el cual se piensa realizar el diseño, se solicita en una pantalla de selección simple (Ver figura N° 18), donde se encuentran los

diferentes aceros de las normas API y ASTM, al seleccionar cualquiera de ellos el programa directamente cargará una base de datos que contiene diámetros y espesores para el respectivo tipo de acero (Ver Anexo 2).

Por otra parte se encuentran en esta pantalla los valores de k_{11} y k_{12}, los cuales son los factores de seguridad mencionados en el capítulo II, que garantizan que la resistencia del acero no llegará hasta su esfuerzo máximo. Estos se encuentran con valores pre-establecidos con opción a ser modificados. El k de pandeo se utilizará para obtener una longitud efectiva y por consiguiente la relación de esbeltez efectiva, tendrá un valor pre-establecido de 1 y de igual forma podrá modificarse. Si el usuario introdujo un valor de sobrepresión se habilitara la casilla de k_{sp} de lo contrario esta casilla estará inhabilitada.

Figura N°18. Ventana de selección del tipo de Acero y factores de seguridad

Si el usuario decide colocar un tipo de acero que no forma parte del catalogo comercial dispuesto en la base de datos del programa, deberá introducir el valor del

esfuerzo de fluencia (Ver figura N° 19) y además el cálculo asociado al espesor será únicamente el académico, es decir, el resultado de los cálculos no necesariamente corresponde a una tubería que se pueda encontrar en el mercado, pero si por el contrario el tipo de acero forma parte de la base de datos el cálculo del espesor localizará inmediatamente el valor igual o superior que este disponible.

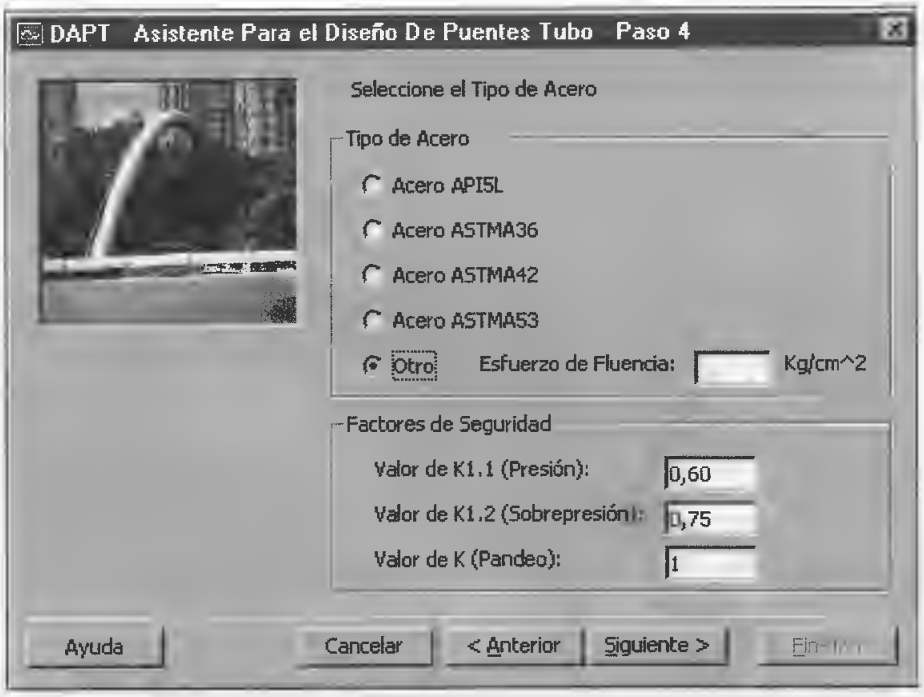


Figura N° 19. Selección de otro tipo de acero y asignación del esfuerzo de fluencia

Una vez culminada la selección del acero, el programa solicita la selección del tipo de cálculo, presentando dos opciones, el cálculo académico y el cálculo según la base de datos comercial (Ver figura N° 20).

El cálculo académico se activará automáticamente cuando el usuario haya seleccionado un tipo de acero que no se encuentra en la base de datos y su resultado generará un espesor que probablemente no se encuentra disponible en el mercado.

El cálculo según la base de datos comercial, carga la base de datos relacionada con la alternativa y muestra los diámetros disponibles en el mercado para ese tipo de acero (Ver figura N° 21), el usuario podrá entonces realizar la elección del diámetro de la tubería que desea utilizar para la construcción del puente tubo en arco.

Puede observarse igualmente una casilla de espesor, la cual se habilitara en el caso de una verificación, ya que el usuario conoce previamente cual es el espesor con el que esta trabajando y por lo tanto deberá introducirlo, mientras que en el caso del diseño este valor es justamente el que se desea encontrar.

Figura N° 20. Ventana de selección del tipo de cálculo y del diámetro

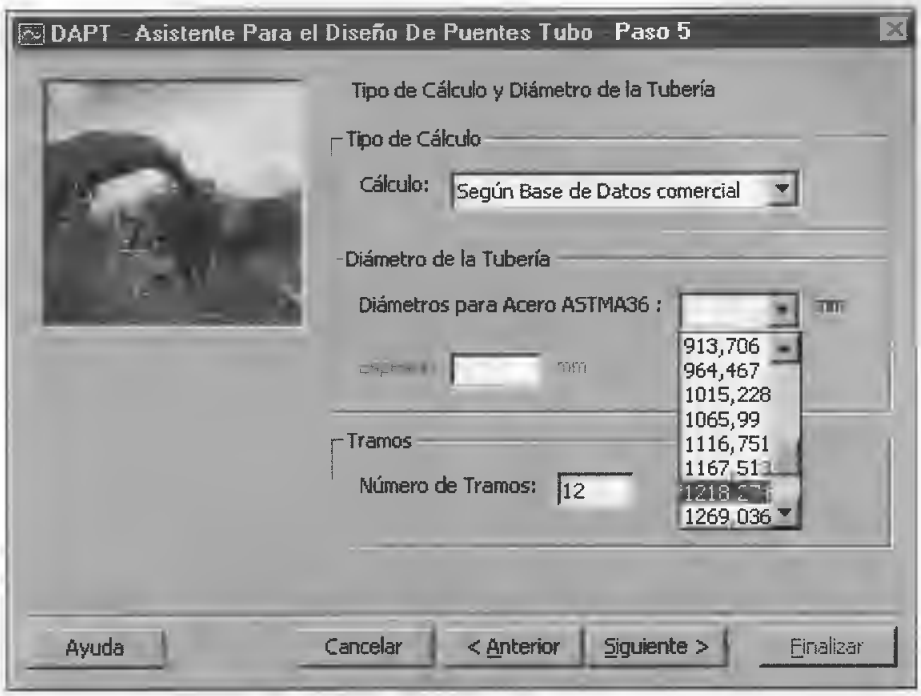


Figura N°21. Detalle de selección del Diámetro

Una vez que el usuario selecciona el diámetro con el que desea trabajar, aparecerá en la casilla correspondiente al número de tramos un valor que se modificará para cada elección de un diámetro. Esto se debe a un parámetro introducido para el cálculo del número de tramos, que establece una longitud mínima recomendable para el tramo de tubería, por lo que puede presentarse el caso donde no consiga esa longitud establecida como óptima y por lo tanto la casilla aparecerá vacía y el usuario se verá obligado a introducir el dato con el valor que desee o a elegir un diámetro que cumpla con la longitud mínima (Ver figura N°22).

Figura N° 22. Selección del diámetro y número de tramos

Como complemento del diseño del puente tubo en arco, se calculan las fuerzas actuantes en los extremos de la tubería y se utilizan las mismas en el pre-diseño del bloque de fundación directa, es importante señalar que esta opción solo se presentará si el usuario se encuentra bajo la operación diseñar y para poder ejecutarla se le solicitan los datos que se observan en la figura N° 23, tales como: caudal de trabajo, coeficiente de fricción entre el concreto y el suelo, el cual depende del tipo de terreno y varía entre 0,30 y 0,50[10], el peso unitario del concreto y la capacidad de soporte del suelo.

Para el valor de la capacidad de soporte del suelo se suministra una tabla que contiene valores máximos y mínimos de dicho soporte para distintos tipos de terreno y que se presenta a continuación:

[10] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**" Fondo Polar-UCAB, Caracas 1995, Pág 9.3

TIPO DE SUELO	CAPACIDAD DE SOPORTE DEL SUELO "P" (KG/CM²)	
	VALOR MAXIMO	**VALOR MINIMO**
SUELOS ALUVIALES SATURADOS	0,50	0,25
ARENA COMPACTA, SECA	3,00	2,00
ARENA SIN MAYOR COMPACTACIÓN, SECA	2,00	1,00
ARCILLA, SECA	3,00	2,00
ARCILLA CON CIERTA HUMEDAD	2,00	1,00
GRAVA ARENOSA	4,00	3,00

Tabla N° 4 Capacidad de soporte para diversos tipos de suelo[11]

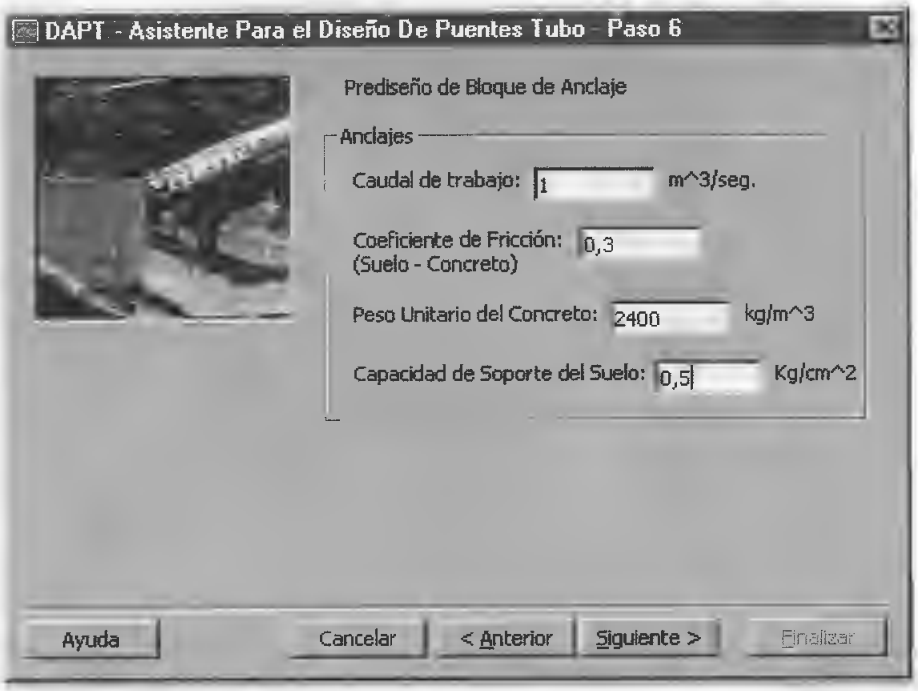


Figura N° 23 Ventana de pre-diseño del bloque de fundación directa

[11] Méndez, Manuel Vicente, "**Tuberías a presión en los sistemas de abastecimiento de agua**". Fondo Polar-UCAB, Caracas 1995, Pág 9.4

Con el prediseño del bloque de anclaje se culmina con la introducción de los datos requeridos por el programa, los cuales se presentan en la ventana de resumen de datos, tal como se muestra en la Figura N° 24.

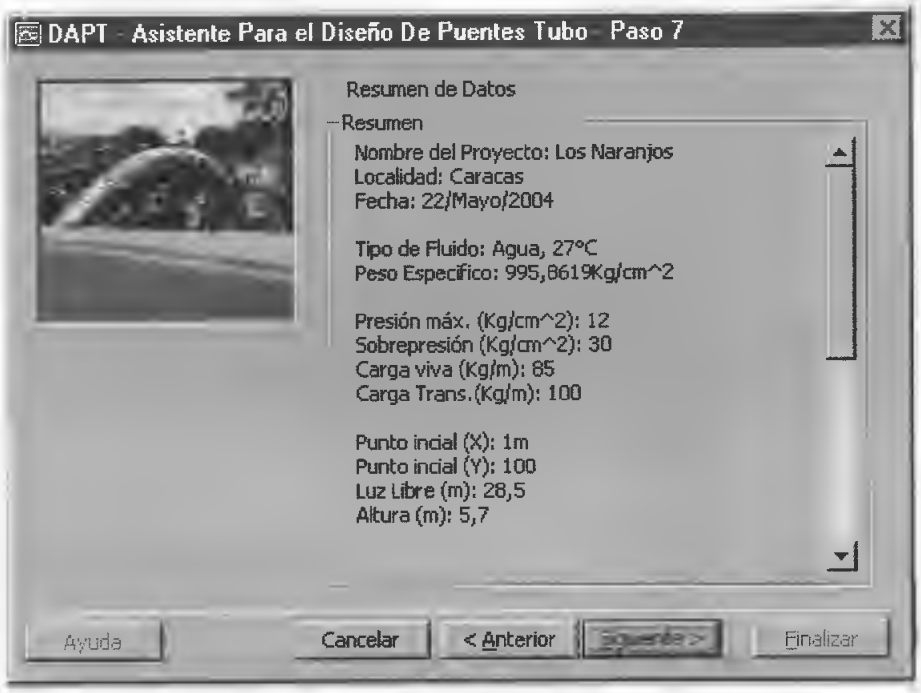


Figura N° 24 Ventana de Resumen de datos

4.1.5 *Resultados*

Una vez que se le suministran al programa todos los datos, este procede a realizar los cálculos correspondientes; pudiéndose apreciar el valor del espesor obtenido; posteriormente se presentan los resultados correspondientes numéricamente y con opción gráfica (plano en AutoCAD). Se puede acceder ha ambos resultados a partir de la siguiente ventana que se muestra en la figura N° 25.

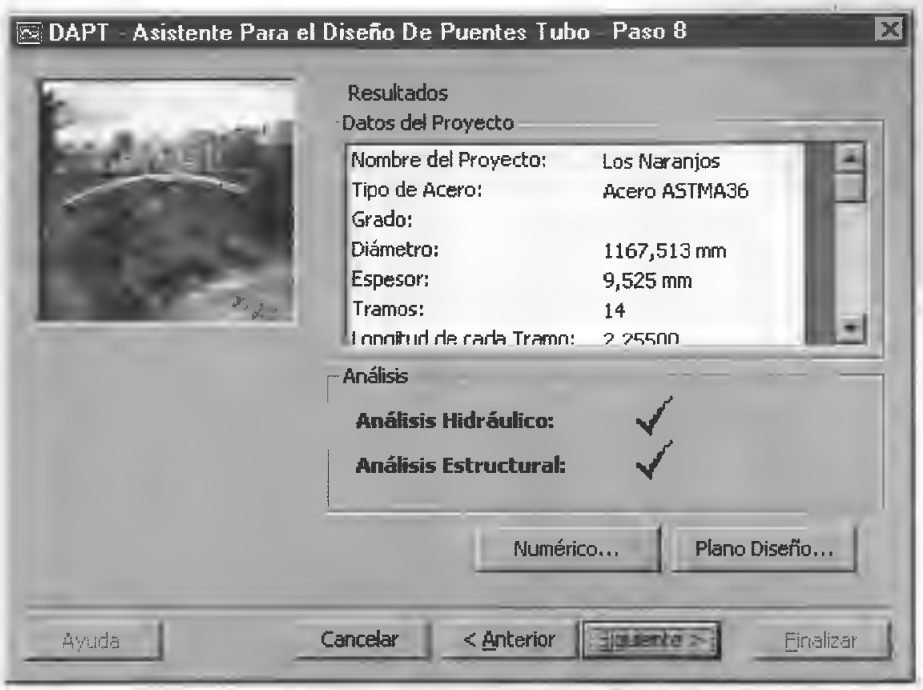
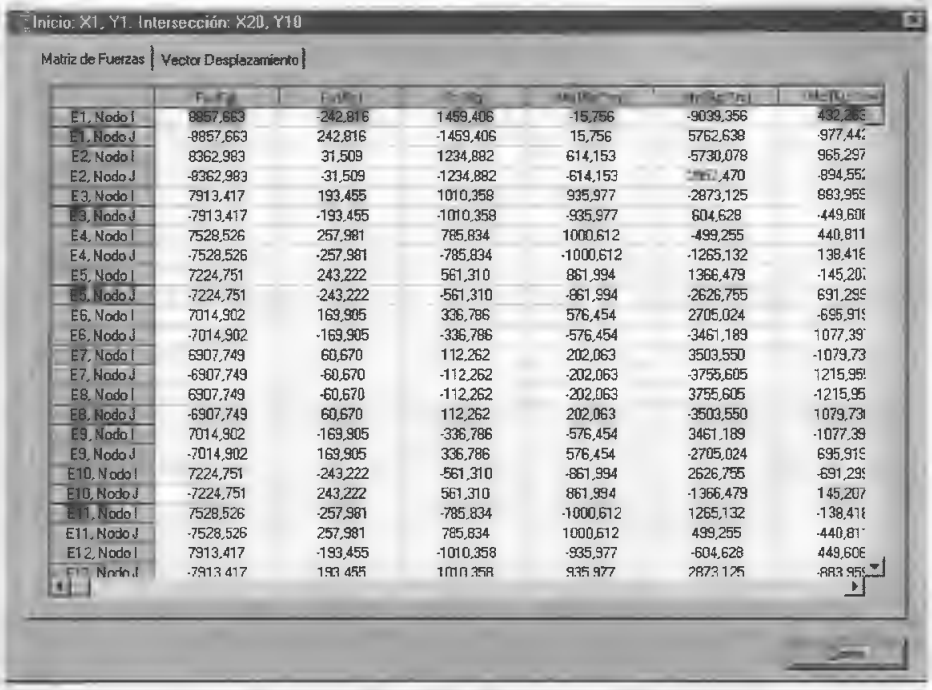


Figura N° 25 Ventana de Cuadro de Resultados

4.5.1. Resultados Numéricos

Estos resultados reflejan los valores de las Fuerzas, los momentos y los desplazamientos en cada uno de los nodos que conforman la estructura. Se presentan en una pantalla con un arreglo matricial donde las filas corresponden a la posición de los nodos y las columnas a las reacciones Fx, Fy, Fz, Mx, My y Mz (Ver figura N°26), igualmente se encuentra el arreglo matricial para los desplazamientos en cada uno de los nodos (Ver figura N° 27)



E1, Nodo I	8857,663	-242,816	1459,406	-15,756	-9039,356	432,26
E1, Nodo J	-8857,663	242,816	-1459,406	15,756	5762,639	-977,44
E2, Nodo I	8362,983	31,509	1234,882	614,153	-5730,078	965,297
E2, Nodo J	-8362,983	-31,509	-1234,882	-614,153	[illegible],470	-894,55
E3, Nodo I	7913,417	193,455	1010,358	935,977	-2873,125	883,959
E3, Nodo J	-7913,417	-193,455	-1010,358	-935,977	604,628	-449,60
E4, Nodo I	7528,526	257,981	785,834	1000,612	-499,255	440,811
E4, Nodo J	-7528,526	-257,981	-785,834	-1000,612	-1265,132	138,41
E5, Nodo I	7224,751	243,222	561,310	861,994	1366,479	-145,20
E5, Nodo J	-7224,751	-243,222	-561,310	-861,994	-2626,755	691,29
E6, Nodo I	7014,902	169,905	336,786	576,454	2705,024	-695,91
E6, Nodo J	-7014,902	-169,905	-336,786	-576,454	-3461,189	1077,39
E7, Nodo I	6907,749	60,670	112,262	202,063	3503,550	-1079,73
E7, Nodo J	-6907,749	-60,670	-112,262	-202,063	-3755,605	1215,95
E8, Nodo I	6907,749	-60,670	-112,262	-202,063	3755,605	-1215,95
E8, Nodo J	-6907,749	60,670	112,262	202,063	-3503,550	1079,73
E9, Nodo I	7014,902	-169,905	-336,786	-576,454	3461,189	-1077,39
E9, Nodo J	-7014,902	169,905	336,786	576,454	-2705,024	695,91
E10, Nodo I	7224,751	-243,222	-561,310	-861,994	2626,755	-691,29
E10, Nodo J	-7224,751	243,222	561,310	861,994	-1366,479	145,20
E11, Nodo I	7528,526	-257,981	-785,834	-1000,612	1265,132	-138,41
E11, Nodo J	-7528,526	257,981	785,834	1000,612	499,255	-440,81
E12, Nodo I	7913,417	-193,455	-1010,358	-935,977	-604,628	449,60
E12, Nodo J	-7913,417	193,455	1010,358	935,977	2873,125	-883,95

Figura N° 26 Ventanas de Fuerzas y Momentos

N° 1	,0000
N° 2	,0000
N° 3	-,0002
N° 4	-,0001
N° 5	,0001
N° 6	,0000
N° 7	,0000
N° 8	-,0001
N° 9	-,0006
N° 10	,0001
N° 11	,0002
N° 12	,0000
N° 13	,0000
N° 14	-,0002
N° 15	-,0011
N° 16	-,0002
N° 17	,0002
N° 18	,0000
N° 19	,0000
N° 20	0003
N° 21	-,0016
N° 22	-,0002
N° 23	0002
N° 24	,0000
N° 25	0000

Figura N° 27. Ventana de los desplazamientos

4.5.2. Gráficos de Diseño

Haciendo click en el botón de plano de Diseño, se crea el plano del puente tubo utilizando como herramienta el programa AutoCAD, el cual contendrá: la sección del cauce, los bloques de fundación directa (según sea el caso), el arco que conforma la estructura y el detalle del tramo de tubería, todos estos valores se encontrarán adecuadamente dimensionados obteniéndose así un plano de diseño de Ingeniería Básica de alta calidad. En el caso que el usuario este conforme con los resultados, puede proceder a imprimirlos, completándose así el diseño del puente tubo en arco.

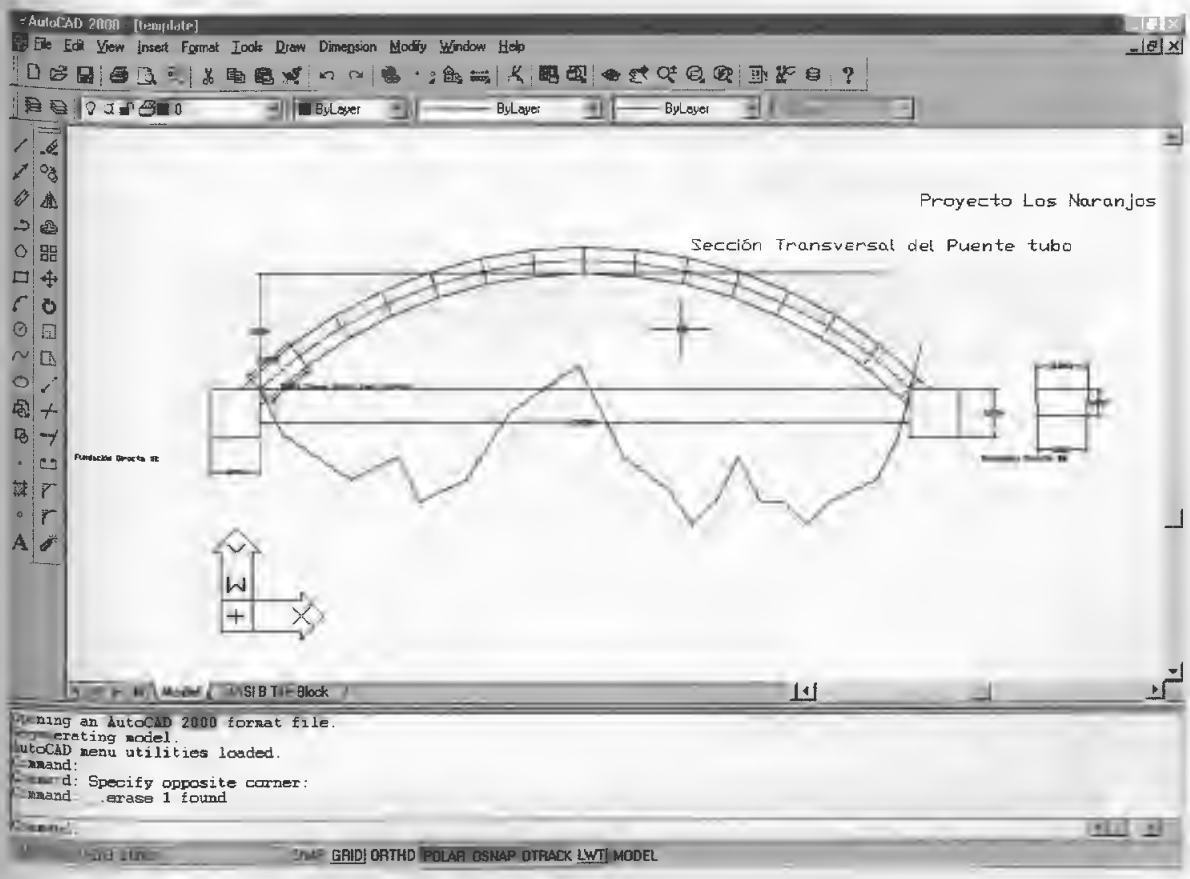


Figura N° 28. Plano de AutoCAD del puente tubo en arco

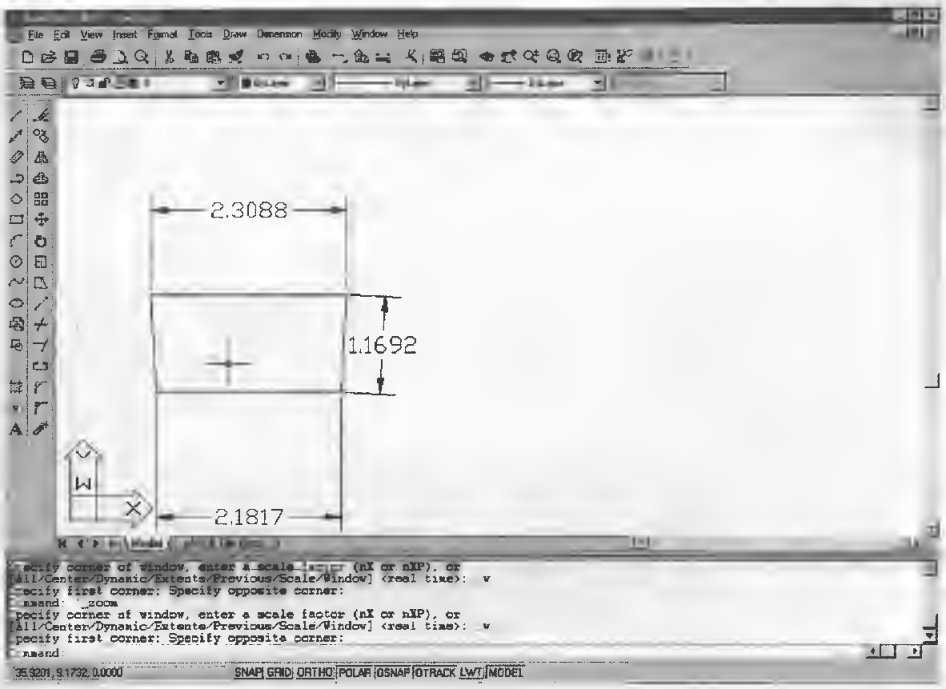


Figura N° 29. Plano de AutoCAD, detalle de tramo de tubería

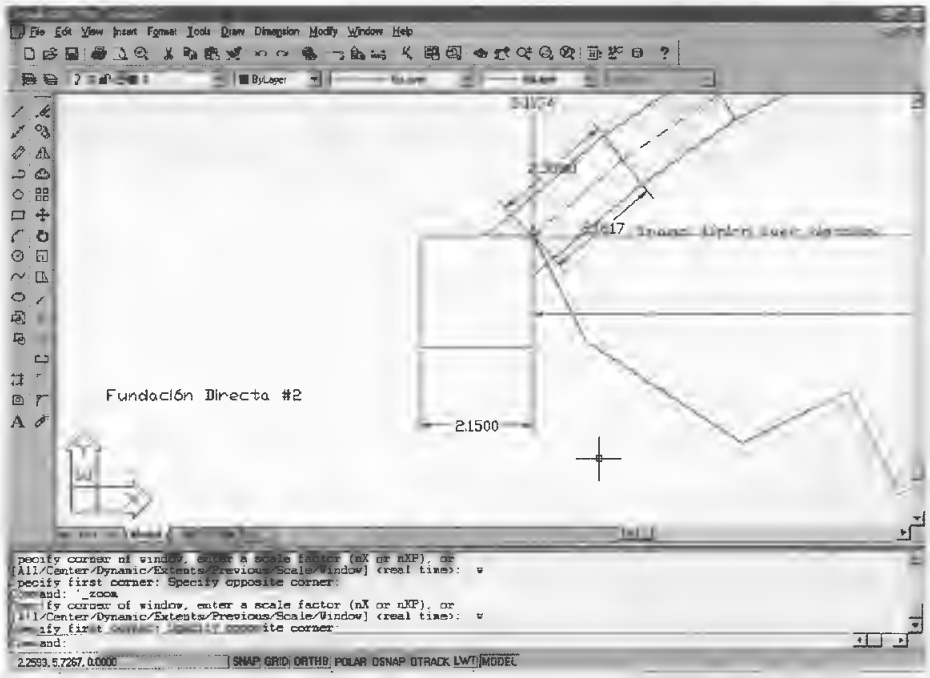


Figura N° 30. Plano de AutoCAD, detalle del bloque de fundación directa

Una vez completado los cálculos y por ende finalizado el proceso de diseño o verificación, el programa proporciona la opción *Guardar Proyecto*, que permite recopilar toda la información para almacenarla en un archivo Excel. Esta opción permite regresar a un diseño en particular en el momento que el usuario así lo considere.

Figura N° 31. Pantalla final del programa.

CONCLUSIONES

El desarrollo de la herramienta informática que permite el diseño y/o la verificación de estructuras de Puentes tubo en Arco, genera una gran ventaja en el tiempo de ejecución de los cálculos establecidos en las condiciones hidráulicas y estructurales que garantizan el buen funcionamiento del puente tubo. Como resultado de su aplicación se optimiza el proceso de diseño y de obtención de planos de dibujo en el programa AutoCAD.

Al establecer los diferentes aspectos que intervienen en el proceso de diseño y/o verificación de puentes tubo en arco, comprendimos que este trabajo abarca una cantidad de ramas de la Ingeniería que incluye estudios de Suelos, Hidráulica, Estructuras, Mecánica, por lo que fue necesaria asesoría en cada una de estas especialidades.

Podemos confiar en la veracidad de los resultados obtenidos en la matriz de fuerzas y momentos, ya que se realizó la comparación de estos valores contra el programa STAAD/pro, resultando una aproximación razonable entre ambas matrices. De igual forma se realizaron ejemplos en cada una de las rutinas que conforman el programa para garantizar que una vez integradas como un todo el funcionamiento sea correcto. La diferencia entre los resultados obtenidos en el *DAPT 1.0* y el STAAD/Pro están en el orden del 3%.

RECOMENDACIONES

El usuario del programa "DAPT 1.0"deberá estar familiarizado con los datos solicitados para no cometer errores a la hora de asignar valores a cada una de las variables que conforman el programa. Si por el contrario, esto no ocurre, no se podrá garantizar la veracidad de los resultados que se generen.

La creación de una versión avanzada que contemple el aspecto económico en el diseño de los puentes tubo en arco, estableciendo las cantidades de obras que implican la construcción del puente tubo y la estimación de los costos para su elaboración. Adicionalmente crear actualizaciones de la base de datos comercial para mantener una gama de opciones de diámetros y espesores acorde con el mercado nacional.

BIBLIOGRAFÍA

American Water Works Association. **Manual M-11 Steel Pipe a guide for Design and installation.** (2da ed.) U.S.A (1987)

Arocha Ravelo, Simón. (1997). **Abastecimientos de Agua**. (3ª ed.). Caracas: Editorial Innovación Tecnológica.

Borg, Sydney y Gennaro, Joseph. (1963). **Análisis Estructural Avanzado.** México: Compañía Editorial Continental.

C.V.G Siderurgica del Orinoco, C.A. (SIDOR), Manual de Proyectos. Estructuras de Acero Tomo 1 (1982) (2da ed.).Puerto Ordaz.

Fernández de Sousa, Freddy y Pérez Caraballo, Víctor. (1999) **Estimación de los Esfuerzos Residuales en uniones soldadas a través del método de los elementos finitos**. Trabajo especial de Grado, Universidad Central de Venezuela, Caracas.

Hunter, Rouse. (1951). **Hidráulica Mecánica Elemental de los Fluidos**. Editorial Dossat

Lehman, Charles. (1996). **Geometría Analítica**. México: Editorial Limosa.

Luthe, Rodolfo. (1971). **Análisis Estructural**. México: Representaciones y Servicios de Ingeniería México

Méndez, Manuel Vicente. (1995) **Tuberías a presión en los sistemas de abastecimiento de agua**. Caracas: Universidad Católica Andrés Bello. Fondo Polar.

Popov, Egor. (1995). **Introducción a la Mecánica de Sólidos**. México: Editorial Limusa.

ANEXOS

ANEXO 1: CÁLCULO ESTRUCTURAL

Las operaciones que se ejecutan en el programa una vez que se inicia el cálculo estructural, vienen establecidas en una serie de rutinas que se describirán a continuación, dando una breve reseña de cuales son los datos de entrada, cuales son los comandos ejecutados y cuales son los datos de salida que se crean.

Datos de Entrada

- ✓ **DIM X, DIM Y ,DIM Z (1 TO NNODOS)**

Coordenadas X, Y y Z de cada uno de los nodos que conforman la estructura, este dato se presenta como un vector de dimensiones iguales al número de nodos que existan en la estructura.

- ✓ **DIM GRLIB (1 TO NNODOS,1 TO 6)**

Matriz de grados de libertad de la estructura. Este arreglo está conformado por ceros (0) y unos (1), colocando 1 en aquellos puntos donde las coordenadas de los nodos estén restringidas y 0 cuando estén libres.

- ✓ **DIM INCDA**

La matriz de incidencia de miembros, la cual contiene los nodos ***i-j*** en los extremos de cada elemento. Para el caso de los modelos estructurales analizados en este trabajo el valor de ***i*** se corresponde con el número del elemento y el valor de ***j*** con el número del elemento mas uno.

- ✓ **DIM PROP(1 TO NELEMEN, 1 TO 4)**

La matriz de propiedades de cada elemento, la cual contendrá los valores del Área (*A*) , Momento de Inercia alrededor del eje *z* (**Iz**), Momento de Inercia alrededor del eje *y* (**Iy**) , Momento polar de inercia ó constante de Torsión (**Jo**).

- ✓ **DIM COMPL (1 TO NNODOS, 1 TO 6)**

Por ultimo será necesaria el arreglo **COMPL** que contiene todas las fuerzas aplicadas en los nodos de la estructura en coordenadas globales, que resultan del cálculo del caso complementario como:

$$\mathbf{Q}^c = \mathbf{Q} - \mathbf{Q}^p$$

Donde: $\mathbf{Q}^p$ es el vector de cargas resultante del caso primario (desplazamientos nodales impedidos); **Q** representa las fuerzas externas aplicadas en los nodos según las coordenadas globales de desplazamientos.

EJECUCIÓN DEL PROGRAMA

- ✓ **Determinación del vector de conectividad y de los grados de libertad**

Crea dos arreglos en forma de matrices los cuales van a ser utilizados para ubicar las posiciones de las coordenadas de desplazamientos nodales dentro del a matriz de rigidez. También se requiere para el cálculo de la fuerzas en extremos de miembros en la rutina **FUERZAS.**

Sub VCONECT (GRLIB(),INCDA(),NODCOR(),ML(),NNODOS,NELEM,NGRL)

Donde:

- ✓ GRLIB (NNODOS, 6): Matriz que indica si la coordenada está libre o restringida (1: restringida; 0: libre).

- ✓ INCDA (2*NELEM): Vector que contiene las incidencias de miembros (número de los nodos en extremos de miembros).
- ✓ NODCOR (NNODOS, 6): Arreglo que contiene la posición de las coordenadas de desplazamiento nodales (según los ejes globales de la estructura) dentro de la matriz de rigidez.
- ✓ ML (NELEM, 12): Vector de Conectividad. Arreglo que contiene la posición de las coordenadas de desplazamiento (según los ejes globales) del elemento dentro de la matriz de rigidez.
- ✓ NNODOS: Número de nodos
- ✓ NELEM: Número de elementos
- ✓ NGRL: Número de grados de libertad

Parámetros de Entrada: GRLIB (), INCDA (), NNODOS, NELEM
Parámetros de Salida: NODCOR (), ML (), NGRL

- ✓ **Determinación del máximo semi-ancho de banda**

El máximo ancho de banda corresponde al número de elementos distintos de cero contenidos en una fila de la matriz de rigidez, contados a partir de la diagonal. Corresponderá a la mayor posición a partir de la diagonal de la matriz más allá de la cual todos los elementos de la fila serán iguales a cero. Conociendo el semi-ancho de banda, y aprovechando la condición de simetría de la matriz de rigidez podremos construir una matriz rectangular reducida con un número de filas igual a la matriz original (y que corresponde al número de grados de libertad) y un número de columnas igual al semi-ancho de banda. Esto nos permitirá agilizar los cálculos y ahorrar espacio de memoria, al no incluir en ellos valores que previamente se conocen y que serán iguales a cero.

Sub MAXKBU (ML (), KBU, NELEM)

Donde:

ML (NELEM, 12): Vector de Conectividad, resultado obtenido en la rutina VCONECT

KBU: Semi- ancho de banda – 1 (el semi-ancho de banda se define como *KBU* + 1

NELEM: Número de elementos

Parámetros de Entrada: NELEM, ML ()

Parámetros de Salida: KBU

- ✓ **Determinación del vector de cargas Q**

Este arreglo genera el vector de cargas **Q** a partir del arreglo **COMPL ()** y conociendo las posiciones de las coordenadas de desplazamiento nodales dentro de la matriz de rigidez obtenidas del arreglo **NODCOR ().** Se utilizará en la resolución del sistema de ecuaciones y en el cálculo de fuerzas en los extremos de miembros.

Sub VECQ (NODCOR (), COMPL (), Q (), NNODOS)

Donde:

NODCOR (NNODOS, 6) Matriz que contiene la posición de las coordenadas nodales en la matriz de rigidez. Se obtiene de la rutina VCONECT

COMPL (NNODOS, 6) Matriz que contiene el caso complementario "Q^c"

Q (NGRL) Vector de cargas generalizadas

NNODOS Número de nodos

Parámetros de Entrada: NODCOR (), COMPL (); NNODOS

Parámetros de Salida: Q ()

- ✓ **Determinación de la matriz de rigidez del elemento en coordenadas globales**

Genera la matriz de rigidez del elemento.

Sub MATKE (I, INCDA (), X (), Y (), Z (), KE (), E, G, A, IY, IZ, Jo)

Donde:
INCDA (2*NELEM) Vector que contiene incidencias de miembros. Producto de la subrutina VCONECT.
KE (12,12) Matriz de rigidez del elemento
X (NNODOS) Vector de coordenadas X
Y (NNODOS) Vector de coordenadas Y
Z (NNODOS) Vector de coordenadas Z
E Modulo de elasticidad
G módulo de corte
A área
Ix, Inercia en X
Iy, Inercia en Y
Jo, Constante de torsión

Parámetros de Entrada: INCDA (), X (), Y (), Z (), E, G, A, IY, IZ, Jo

Parámetros de Salida: KE ()

Es importante dar una explicación más amplia en relación a dos de los cálculos desarrollados en esta rutina. El primero de ellos es el relacionado con el cálculo de la matriz de transformación "T" el cual se dedujo de la siguiente forma:

T (12, 12) Matriz de transformación "T"

TT (12, 12) Matriz de transformación "T" Traspuesta

✓ **Transformación de Coordenadas**

A través de los valores de las coordenadas ***x, y*** se pueden obtener los cosenos directores de los vectores unitarios ***x , y , z*** que determinan la orientación del sistema de ejes locales con respecto al sistema de ejes globales:

$$\left.\begin{array}{l} i(x_1, y_1) \rightarrow \Delta x = x_2 - x_1 \\ j(x_2, y_2) \rightarrow \Delta y = y_2 - y_1 \end{array}\right\} \rightarrow L = \sqrt{\Delta x^2 + \Delta y^2}$$

$$Cx = Cos\vartheta = \frac{\Delta x}{L} ; Sy = Sen\vartheta = \frac{\Delta y}{L}$$

$$P = \begin{bmatrix} XLX & XLY & XLZ \\ YLX & YLY & YLZ \\ ZLX & ZLY & ZLZ \end{bmatrix} \qquad P^T = \begin{bmatrix} XLX & YLX & ZLX \\ XLY & YLY & ZLY \\ XLZ & YLZ & ZLZ \end{bmatrix}$$

$$Donde : \begin{cases} XLX = Cx & YLX = Sy & ZLX = 0 \\ XLY = -Sy & YLY = Cx & ZLY = 0 \\ XLZ = 0 & YLZ = 0 & ZLZ = 1 \end{cases}$$

La matriz **[P]** contiene los cosenos directores de los vectores unitarios ***x , y , z***. La matriz **[T]** se define como la matriz de transformación de coordenadas:

$$T = \begin{bmatrix} P & & & \\ & P & & \\ & & P & \\ & & & P \end{bmatrix} \qquad T^T = \begin{bmatrix} P^T & & & \\ & P^T & & \\ & & P^T & \\ & & & P^T \end{bmatrix}$$

La matriz de transformación así deducida es válida únicamente para el caso elementos contenidos en el plano ***x-y*** (coordenada ***z*** constante).

La matriz de rigidez del elemento **[Ke]*** en coordenadas globales se obtiene a partir de la matriz **[Ke]** del elemento y la matriz de transformación [T]:

$$\mathbf{[Ke]^* = [T^T] \times [Ke] \times [T]}$$

Matriz de Rigidez del elemento:

$$Ke=\begin{bmatrix}
\frac{EA}{L} & & & & & & \frac{-EA}{L} & & & & & \\
 & \frac{12EI}{L^3} & & & \frac{6EI}{L^2} & & & \frac{-12EI}{L^3} & & & & \frac{6EI}{L^2} \\
 & & \frac{12EI}{L^3} & & \frac{-6EI}{L^2} & & & & \frac{-12EI}{L^3} & & \frac{-6EI}{L^2} & \\
 & & & \frac{GJ}{L} & & & & & & \frac{-GJ}{L} & & \\
 & & \frac{-6EI}{L^2} & & \frac{4EI}{L} & & & & \frac{6EI}{L^2} & & \frac{2EI}{L} & \\
 & \frac{6EI}{L^2} & & & & \frac{4EI}{L} & & \frac{-6EI}{L^2} & & & & \frac{2EI}{L} \\
\frac{-EA}{L} & & & & & & \frac{EA}{L} & & & & & \\
 & \frac{-12EI}{L^3} & & & \frac{-6EI}{L^2} & & & \frac{12EI}{L^3} & & & & \frac{-6EI}{L^2} \\
 & & \frac{-12EI}{L^3} & & \frac{6EI}{L^2} & & & & \frac{12EI}{L^3} & & \frac{6EI}{L^2} & \\
 & & & \frac{-GJ}{L} & & & & & & \frac{GJ}{L} & & \\
 & & \frac{-6EI}{L^2} & & \frac{2EI}{L} & & & & \frac{6EI}{L^2} & & \frac{4EI}{L} & \\
 & \frac{6EI}{L^2} & & & & \frac{2EI}{L} & & \frac{-6EI}{L^2} & & & & \frac{4EI}{L}
\end{bmatrix}$$

En donde:

$$SA = \frac{E*A}{L}; TA = \frac{G*J}{L}$$

$$GY = \frac{2*E*IY}{L}; GZ = \frac{2*E*IZ}{L}$$

$$BY = \frac{3*GY}{L}; BZ = \frac{3*GZ}{L}$$

$$AY = \frac{2*BY}{L}, AZ = \frac{2*BZ}{L}$$

- ✓ **Ensamblaje de la matriz de rigidez de la estructura con almacenamiento de semi-ancho de banda constante**

Conociendo el semi-ancho de banda (**KBU**) se puede ensamblará la matriz de rigidez dentro de un arreglo unidimensional en el cual las posiciones ***i, j*** se transforma en la posición: **KBU * (*i* – 1) + *j***, es decir que:

K (*i, j*) = K (KBU * (*i* – 1) + *j*)

Sub MATKABC (I, KE (), K (), ML (), KBU)

Donde:

K (NGRL*(KBU+1)): Matriz de rigidez de la estructura.

KE (12, 12): Matriz de rigidez del elemento. Producto de la subrutina MATKE.

ML (NELEM, 12): Vector de Conectividad, resultado obtenido en la rutina VCONECT.

KBU: Producto de la subrutina MAXKBU

Parámetros de Entrada: KE (), ML (), KBU

Parámetros de Salida: K ()

✓ **Resolución del sistema de ecuaciones**

Para el cálculo de los desplazamientos se resuelve el sistema de ecuaciones lineales, lo cual es equivalente a calcular la matriz inversa [K^{-1}] de la matriz de rigidez [K]. Para ello emplearemos el Método de Eliminación de Gauss por fila. Este método se puede dividir en tres procesos separados:

- Triangularización de la matriz de rigidez simétrica [K] (*Sub MATGAUSSFTR*).
- Modificación del término independiente, que en este caso lo conforma el vector de cargas [Q] (*Sub MATGAUSSFMI*).
- Solución del sistema de ecuaciones por retro-sustitución (*Sub MATGAUSSFS*).

✓ **Triangularización con almacenamiento por semi-ancho de banda constante**

El proceso de triangularización consistirá transformar en ceros todos los elementos de la matriz [K] situados por debajo de la diagonal (triángulo inferior), mediante combinaciones lineales entre filas. La descripción del proceso se ilustra en la figura inferior:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdot & \cdot & \cdot & K_{1n} \\ K_{21} & K_{22} & K_{23} & \cdot & \cdot & \cdot & K_{2n} \\ K_{31} & K_{32} & K_{33} & \cdot & \cdot & & K_{3n} \\ \cdot & \cdot & & & & & \cdot \\ \cdot & \cdot & & & & & \cdot \\ \cdot & \cdot & & & & & \cdot \\ K_{n1} & K_{n2} & K_{n3} & & & & K_{nn} \end{bmatrix} \quad \begin{matrix} L=1 \rightarrow n \quad 1 \quad (Fila \quad Pivote) \\ i=L+1 \rightarrow n \end{matrix}$$

"n" es el número de grados de libertad NGR.

$$L = 1 \rightarrow n-1 \;;\; i = L+1 \rightarrow n \;;\; j = L+1 \rightarrow n$$

$$K_{ij} = K_{ij} - \frac{K_{iL}}{K_{LL}} \times K_{Lj} \rightarrow (A-1)$$

El elemento pivote estará ubicado en la diagonal. A cada una de las filas se le restará la fila pivote previamente dividida por el elemento pivote y multiplicada por el elemento de la fila situado en la misma columna del elemento pivote. Mediante este procedimiento todos los elementos en la columna por debajo del elemento pivote se harán iguales a cero. El procedimiento se repite haciendo variar la fila pivote desde la primera fila hasta la penúltima fila., obteniendo una matriz triangular.

Cuando se emplea una matriz de rigidez almacenada por semi-ancho de banda constante en un vector unidimensional, en la expresión (A-1) el elemento $K_{i,L}$ es reemplazado por su simétrico o transpuesto $K_{L,i}$ y toma la forma siguiente:

$$K_{(II+j)} = K_{(II+j)} - \frac{K_{(LL+i)}}{K_{(LL+L)}} \times K_{(LL+j)}$$

$$Siendo: \begin{cases} II = KBU \times (i-1) \\ LL = KBU \times (L-1) \end{cases}$$

Sub MATGAUSSFTR (K (), NGRL, KBU)

Donde:

K (NGRL*(KBU+1)) Matriz de rigidez de la estructura. Producto de la subrutina MATKABC

NGRL Número de grados de libertad. Producto de la subrutina VCONECT

KBU Máximo ancho de banda. Producto de la subrutina MAXKBU

Parámetros de Entrada: K (), NGRL; KBU

Parámetros de Salida: K ()

- ✓ **Modificación del Término independiente con almacenamiento con semi-ancho de banda constante**

El proceso de modificación del vector de cargas [Q] (correspondiente al termino independiente del sistema de ecuaciones) se realiza en total correspondencia con el proceso de triangularización descrito anteriormente:

$$L = 1 \rightarrow n-1 \;\; ; \;\; j = L+1 \rightarrow n$$
$$Q(j) = Q(j) - Q(L) \times \frac{K(j,L)}{K(L,L)}$$

Pero como:

$$\frac{K(j,L)}{K(L,L)} = K(L,j)$$

Tenemos:

$$Q(j) = Q(j) - Q(L) \times K(L,j)$$

Siendo: "L" la fila pivote que sería de 1 a n-1

Para el caso de semiancho de banda constante:

$$Q(j) = Q(j) - Q(L) \times K(II + j)$$

La razón para realizar este proceso separado, es la de permitir analizar diferentes casos de cargas que dan lugar a diferentes vectores de cargas [Q] , sin

tener que triangularizar la matriz de cargas [K] en cada caso (ya la matriz [K] es independiente de las cargas). Esto es posible por ser simétrica la matriz [K].

Sub MATGAUSSFMI (K (), Q (); NGRL; KBU)

Donde:

K (NGRL*(KBU+1)) Matriz de rigidez de la estructura. Producto de la subrutina MATGAUSSFTR
NGRL Número de grados de libertad. Producto de la subrutina VCONECT
KBU Máximo ancho de banda. Producto de la subrutina MAXKBU
Q (NGRL) Vector de cargas generalizadas. Producto de la subrutina VECQ

Parámetros de Entrada: K (), NGRL; KBU, Q ()
Parámetros de Salida: K (); Q ()

- ✓ **Solución del sistema de Ecuaciones por Retro-sustitución**

Los desplazamientos nodales se pueden obtener sustituyendo los valores de las incógnitas ya conocidas en las ecuaciones, procediendo en sentido inverso desde la ecuación "n" hasta la primera ecuación:

$$Q(n) = \frac{Q(n)}{K(n,n)}$$

$$Q(n-1) = \frac{Q(n-1)}{K(n-1,n-1)} - K(n-1,n) \times Q(n)$$

$$Q(n-2) = \frac{Q(n-2)}{K(n-2,n-2)} - K(n-2,n-1) \times Q(n-1) - K(n-2,n) \times Q(n)$$

De tal modo que:

$$Para \quad i = n-1 \quad \rightarrow \quad 1 \quad ; \quad j = i+1 \quad \rightarrow \quad n$$
$$Q(i) = Q(i) - Q(j) \times K(i,j)$$

Los desplazamientos incógnitas son almacenados en el mismo arreglo del vector cargas **Q ()**

Sub MATGAUSSFS (K (), Q (); NGRL; KBU)

Donde:

K (NGRL*(KBU+1)) Matriz de rigidez de la estructura. Producto de la subrutina MATGAUSSFMI
NGRL Número de grados de libertad. Producto de la subrutina VCONECT
KBU Máximo ancho de banda. Producto de la subrutina MAXKBU
Q (NGRL) Vector de cargas generalizadas. Producto de la subrutina MATGAUSSFMI

Parámetros de Entrada: K (), NGRL; KBU, Q ()
Parámetros de Salida: K (); Q ()

- ✓ **Cálculo de fuerzas en extremos de miembros en coordenadas locales**

Una vez resuelto el sistema de ecuaciones esta rutina permite calcular las fuerzas en los extremos de cada elemento
Sub FZAS (I, INCDA (), X (), Y (), Z (), ML (), Q (), FZAS (), E, G, A; IY, IZ, Jo)

Donde:

INCDA (2*NELEM) Vector que contiene incidencias de miembros. Producto de la subrutina VCONECT.

Q (NGRL) Vector de cargas generalizadas. Producto de la subrutina MATGAUSSFS

ML (NELEM, 12) Vector de Conectividad, resultado obtenido en la rutina VCONECT

FZAS (NELEM, 12) Matriz que contiene fuerzas en extremos de miembro

X (NNODOS) Vector de coordenadas X

Y (NNODOS) Vector de coordenadas Y

Z (NNODOS) Vector de coordenadas Z

E Modulo de elasticidad

G módulo de corte

A área

Iz Inercia en Z

Iy Inercia en Y

Jo Constante de torsión

Parámetros de Entrada: INCDA (), X (), Y (), Z (), Q (), ML (), E, G, A, IY, IZ, Jo

Parámetros de Salida: FZAS ()

Similarmente a la rutina MATKE se construye la matriz de rigidez de la estructura y se construye la matriz de transformación.

ANEXO 2: BASE DE DATOS COMERCIAL[12]

Espesores de tubería para Aceros **API5L A, ASTM A53 A y B**

Diametro (pulgadas)	Espesores de Tubería (pulgadas)																			
	0,237	0,256	0,28	0,312	0,322	0,344	0,365	0,375	0,406	0,432	0,438	0,469	0,5	0,531	0,562	0,593	0,688	0,75	0,864	0,875
4	■							■										■		
5		■								■									■	
6			■										■							■
8					■								■							
10							■						■							
12								■	■				■							
14								■			■		■							
16								■					■							
18								■					■		■					
20								■					■			■				
22								■					■							
24								■					■				■			
26								■					■							
28								■					■							
30								■					■							
32								■					■							
34								■					■							
36								■					■							
38								■					■							
40								■					■							
42								■					■							

[12] Catalogo TUBOACERO aprobado ISO 9002. Fondo Norma, Catalogo IMOSA, Tuboacero fabricación C.A.

Espesores de tubería para Aceros **API5L X60**

Diametro (pulgadas)	Espesores de Tubería (pulgadas)																	
	0,258	0,28	0,312	0,322	0,344	0,365	0,375	0,406	0,432	0,436	0,469	0,5	0,531	0,562	0,593	0,625	0,688	0,875
20			■		■		■			■	■							
22			■		■		■			■	■							
24			■		■		■			■	■	■						
26			■		■		■			■	■	■						
28			■		■		■			■	■	■						
30			■		■		■			■	■	■	■	■				
32			■		■		■			■	■	■	■	■				
34			■		■		■			■	■	■	■	■				
36			■		■		■			■	■	■	■	■		■	■	

Espesores de Tuberías para Aceros **API5L X52**

Diametro Tubería (pulgadas)	Espesores de Tubería (pulgadas)																	
	0,237	0,258	0,28	0,312	0,322	0,344	0,365	0,375	0,406	0,432	0,438	0,469	0,5	0,531	0,562	0,593	0,625	0,875
20																		
22																		
24																		
26																		
28																		
30																		
32																		
34																		
36																		
38																		
40																		
42																		
44																		
46																		
48																		
50																		

Espesores de Tuberías para Aceros **API5L B y X42, ASTM A36 y A42**

Diametro de Tubería (pulgadas)	Espesores de Tubería (pulgadas)																		
	0,237	0,258	0,28	0,312	0,322	0,344	0,365	0,375	0,406	0,432	0,438	0,469	0,5	0,531	0,562	0,593	0,625	0,688	0,875
20																			
22																			
24																			
26																			
28																			
30																			
32																			
34																			
36																			
38																			
40																			
42																			
44																			
46																			
48																			
50																			
52																			
54																			
56																			
58																			
60																			

ANEXO 3: CÓDIGO DEL PROGRAMA

```
Sub Resultados (FZAS (), Porcentaje, Verificacion, pres, w, Fy, Lverificacion,
NELEM, E, diam, espesor)
Dim Prueba1 As Single, Prueba2 As Single, Prueba3 As Single, Prueba4 As Single
If Vars.SistemaDec Then diam = diam * 100 Else diam = diam * 12
If Vars.SistemaDec Then espesor = espesor * 100 Else espesor = espesor * 12
If Vars.SistemaDec Then E = E / 10000 Else E = E / 144
Vars.Verificacion = True
Vars.Porcentaje = 0
pres = pres
w = w
Fy = Fy
Lverificacion = Lverificacion
If Vars.SistemaDec Then Lverificacion = Lverificacion * 100 Else Lverificacion =
Lverificacion * 12
If Vars.SistemaDec = False Then
  diam = diam * 2.54
  espesor = espesor * 2.54
  E = E / 14.2233
  pres = pres / 14.2233
  Fy = Fy / 14.2233
  Lverificacion = Lverificacion * 2.54
End If

ro = diam / 2
ri = (diam - 2 * espesor) / 2
A = 3.1416 * (ro ^ 2 - ri ^ 2)                'Calculos en centimetros
IY = (3.1416 / 4) * (ro ^ 4 - ri ^ 4)
Jo = (3.1416 / 2) * (ro ^ 4 - ri ^ 4)

r = Sqr(IY / A)                               'Radio de Giro en centimetros

Factor = (pres * diam) / (2 * Fy * espesor)
Fey1 = Fy * (1 - Factor)
Fey2 = Fy * Sqr(1 - (Factor) ^ 2)             'Factores para el calculo
PP = diam / espesor
TT = (w * Lverificacion) / (r)
Fee = 12 * 3.1416 ^ 2 * E / (23 * TT ^ 2)

diam = diam / 100
```

```
espesor = espesor / 100
ro = diam / 2
ri = (diam - 2 * espesor) / 2
A = 3.1416 * (ro ^ 2 - ri ^ 2)
IY = (3.1416 / 4) * (ro ^ 4 - ri ^ 4)
Jo = (3.1416 / 2) * (ro ^ 4 - ri ^ 4)

  For I = 1 To NELEM
    Axial1 = FZAS(I, 1) / A / 10000
    Cortante1 = (Sqr(FZAS(I, 2) ^ 2 + FZAS(I, 3) ^ 2) / (0.5 * A)) / 10000
    Torsor1 = (FZAS(I, 4) * ro / Jo) / 10000
    Flexion1 = ((Sqr(FZAS(I, 5) ^ 2 + FZAS(I, 6) ^ 2) * ro) / (IY)) / 10000
    Axial2 = -(FZAS(I, 7) / A) / 10000
    Cortante2 = (Sqr(FZAS(I, 8) ^ 2 + FZAS(I, 9) ^ 2) / (0.5 * A)) / 10000
    Torsor2 = (FZAS(I, 10) * ro / Jo) / 10000
    Flexion2 = ((Sqr(FZAS(I, 11) ^ 2 + FZAS(I, 12) ^ 2) * ro) / (IY)) / 10000

      If PP < (232000 / Fy) Then
        Fb = 0.72 * Fey1
        Cc = Sqr(2 * 3.1416 ^ 2 * E / Fy)
        If Cc < TT Then
          Fs = (5 / 3) + ((3 * TT) / (8 * Cc)) - (((TT) ^ 3) / (8 * ((Cc) ^ 3)))
          Fa = ((1 - (TT ^ 2 / (2 * Cc ^ 2))) * Fey1) / (Fs)
        ElseIf TT < Cc And Cc < 200 Then
          Fa = (12 * ((3.1416) ^ 2) * E) / (23 * ((TT) ^ 2))
        End If
      ElseIf (232000 / Fy) < PP And PP < (914000 / Fy) Then
        Fb = (46500 / PP) + 0.4 * Fey1



        Cc = Sqr((2 * ((3.1416) ^ 2) * E) / Fy)
        If Cc < TT Then
          Fs = (5 / 3) + ((3 * TT) / (8 * Cc)) - (((TT) ^ 3) / (8 * ((Cc) ^ 3)))
          Fa = ((1 - ((TT ^ 2) / (2 * (Cc ^ 2)))) * Fey1) / (Fs)
        ElseIf TT < Cc And Cc < 200 Then
          Fa = (12 * ((3.1416) ^ 2) * E) / (23 * ((TT) ^ 2))
        End If
        Fa1 = (46500 / PP) + 0.4 * Fey1
        If Fa > Fa1 Then Fa = Fa1 Else Fa = Fa
```

```
        'Fa = (46500 / PP) + 0.4 * Fey1
      End If

      If PP < (851000 / Fy) Then
        FfV = 0.4 * Fey2
      End If

      If PP < ((1.77 * ((E / Fy) ^ (4 / 5))) / ((Lverificacion / diam) ^ (5 / 2))) Or PP
< ((E / Fy) ^ (2 / 3)) Then
        FfV = 0.4 * Fey2
      ElseIf PP > ((1.77 * ((E / Fy) ^ (4 / 5))) / ((Lverificacion / diam) ^ (5 / 2))) Or
PP > ((E / Fy) ^ (2 / 3)) Then
        FfV = (0.4 * E) / ((PP) ^ (3 / 2))
      End If

      Prueba1 = (Axial1 / Fa) + ((0.85 * Flexion1) / ((1 - (Axial1 / Fee)) * Fb)) +
(((Cortante1 + Torsor1) / (FfV)) ^ 2)
      Prueba2 = (Axial1 / (0.6 * Fey1)) + (Flexion1 / Fb) + (((Cortante1 + Torsor1)
/ (FfV)) ^ 2)
      Prueba3 = (Axial2 / Fa) + ((0.85 * Flexion2) / ((1 - (Axial2 / Fee)) * Fb)) +
(((Cortante2 + Torsor2) / (FfV)) ^ 2)
      Prueba4 = (Axial2 / (0.6 * Fey1)) + (Flexion2 / Fb) + (((Cortante2 + Torsor2)
/ (FfV)) ^ 2)

      If Prueba1 > Prueba2 And Prueba1 > Prueba3 And Prueba1 > Prueba4 Then
        Vars.Porcentaje = Prueba1
      ElseIf Prueba2 > Prueba1 And Prueba2 > Prueba3 And Prueba2 > Prueba4
Then
        Vars.Porcentaje = Prueba2
      ElseIf Prueba3 > Prueba1 And Prueba3 > Prueba2 And Prueba3 > Prueba4
Then
        Vars.Porcentaje = Prueba3
      Else
        Vars.Porcentaje = Prueba4
      End If
      If Vars.Porcentaje > 1 Then
        Vars.Verificacion = False
        Exit For
      End If

    Next I
    Vars.Fza = FZAS(NELEM, 7)
```

```
    Vars.Fza2 = FZAS(NELEM, 8)
    Vars.Fza3 = FZAS(NELEM, 9)

End Sub
Sub FUERZAS(N, INCDA(), X(), Y(), Z(), ML(), D(), FZAS(), E, G, A, IY, IZ, Jo)
Dim KE(1 To 12, 1 To 12)
Dim T(1 To 12, 1 To 12)
Dim DL(1 To 12)
Ni = INCDA(2 * N - 1)
Nj = INCDA(2 * N)
disX = X(Nj) - X(Ni)
disY = Y(Nj) - Y(Ni)
disZ = Z(Nj) - Z(Ni)
L = Sqr(disX ^ 2 + disY ^ 2 + disZ ^ 2)
disX = disX / L
disY = disY / L
disZ = disZ / L
ZLX = 0
ZLY = 0
ZLZ = 1
YLX = -disY
YLY = disX
YLZ = 0
For I = 0 To 3
    K = 3 * I
    T(K + 1, K + 1) = disX
    T(K + 1, K + 2) = disY
    T(K + 1, K + 3) = disZ
    T(K + 2, K + 1) = YLX
    T(K + 2, K + 2) = YLY
    T(K + 2, K + 3) = YLZ
    T(K + 3, K + 1) = ZLX
    T(K + 3, K + 2) = ZLY
    T(K + 3, K + 3) = ZLZ
Next I
SA = (E * A) / L
TA = (G * Jo) / L
GY = (2 * E * IY) / L
GZ = (2 * E * IZ) / L
By = (3 * GY) / L
BZ = (3 * GZ) / L
AY = (2 * By) / L
AZ = (2 * BZ) / L
```

```
KE(1, 1) = SA
KE(2, 2) = AZ
KE(2, 6) = BZ
KE(3, 3) = AY
KE(3, 5) = -By
KE(4, 4) = TA
KE(5, 5) = 2 * GY
KE(6, 6) = 2 * GZ
KE(6, 2) = KE(2, 6)
KE(5, 3) = KE(3, 5)

KE(1, 7) = -SA
KE(2, 8) = -AZ
KE(2, 12) = BZ
KE(3, 9) = -AY
KE(3, 11) = -By
KE(4, 10) = -TA
KE(5, 9) = By
KE(5, 11) = GY
KE(6, 8) = -BZ
KE(6, 12) = GZ

KE(7, 1) = -SA
KE(8, 2) = -AZ
KE(8, 6) = -BZ
KE(9, 3) = -AY
KE(9, 5) = By
KE(10, 4) = -TA
KE(11, 3) = -By
KE(11, 5) = GY
KE(12, 2) = BZ
KE(12, 6) = GZ

KE(7, 7) = SA
KE(8, 8) = AZ
KE(8, 12) = -BZ
KE(9, 9) = AY
KE(9, 11) = By
KE(10, 10) = TA
KE(11, 11) = 2 * GY
KE(12, 12) = 2 * GZ
KE(12, 8) = KE(8, 12)
```

```
KE(11, 9) = KE(9, 11)

For I = 1 To 12
   For j = 1 To 12
      K = ML(N, j)
      DL(I) = DL(I) + T(I, j) * D(K)
   Next j
Next I
For I = 1 To 12
   For j = 1 To 12
      FZAS(N, I) = FZAS(N, I) + KE(I, j) * DL(j)
   Next j
Next I
End Sub
Sub MATGAUSSFMI(K(), Q(), NGRL, KBU)
For I = 1 To NGRL - 1
   LK = I + KBU
   If LK > NGRL Then
      LK = NGRL
   End If
   II = KBU * (I - 1)
   For j = I + 1 To LK
      Q(j) = Q(j) - Q(I) * K(II + j)
   Next j
Next I
End Sub
Sub MATGAUSSFS(K(), Q(), NGRL, KBU)
NG = KBU * (NGRL - 1)
Q(NGRL) = Q(NGRL) / K(NG + NGRL)
For I = NGRL - 1 To 1 Step -1
   II = KBU * (I - 1)
   Q(I) = Q(I) / K(II + I)
   LK = I + KBU
   If LK > NGRL Then
      LK = NGRL
   End If
   For j = I + 1 To LK
      Q(I) = Q(I) - Q(j) * K(II + j)
   Next j
Next I
End Sub
Sub MATGAUSSFTR(K(), NGRL, KBU)
For L = 1 To (NGRL - 1)
```

```
    LK = L + KBU
    If LK > NGRL Then
        LK = NGRL
    End If
    LL = KBU * (L - 1)
    For I = L + 1 To LK
        KLI = K(LL + I) / K(LL + L)
        II = KBU * (I - 1)
        For j = I To LK
            K(II + j) = K(II + j) - KLI * K(LL + j)
        Next j
        K(LL + I) = KLI
    Next I
Next L
End Sub
Sub MATKABC(I, KE(), K(), ML(), KBU)
j = 1
Do While j <= 12
    MLI = ML(I, j)
    If MLI <> 0 Then
        p = 1
        Do While p <= 12
            MLJ = ML(I, p)
            If MLI <= MLJ Then
                II = KBU * (MLI - 1)
                K(II + MLJ) = K(II + MLJ) + KE(j, p)
            End If
            p = p + 1
        Loop
    End If
    j = j + 1
Loop
End Sub
Sub MATKE(N, INCDA(), X(), Y(), Z(), KE(), E, G, A, IY, IZ, Jo)
For I = 1 To 12
    For j = 1 To 12
        KE(I, j) = 0
    Next j
Next I
Dim KG(1 To 12, 1 To 12)
Dim T(1 To 12, 1 To 12)
Dim TT(1 To 12, 1 To 12)
Ni = INCDA(2 * N - 1)
```

```
Nj = INCDA(2 * N)
disX = X(Nj) - X(Ni)
disY = Y(Nj) - Y(Ni)
disZ = Z(Nj) - Z(Ni)
L = Sqr((disX ^ 2) + (disY ^ 2) + (disZ ^ 2))
disX = disX / L
disY = disY / L
disZ = disZ / L
ZLX = 0
ZLY = 0
ZLZ = 1
YLX = -disY
YLY = disX
YLZ = 0
For I = 0 To 3
   K = 3 * I
   T(K + 1, K + 1) = disX
   T(K + 1, K + 2) = disY
   T(K + 1, K + 3) = disZ
   T(K + 2, K + 1) = YLX
   T(K + 2, K + 2) = YLY
   T(K + 2, K + 3) = YLZ
   T(K + 3, K + 1) = ZLX
   T(K + 3, K + 2) = ZLY
   T(K + 3, K + 3) = ZLZ
Next I
For I = 0 To 3
   K = 3 * I
   TT(K + 1, K + 1) = disX
   TT(K + 1, K + 2) = YLX
   TT(K + 1, K + 3) = ZLX
   TT(K + 2, K + 1) = disY
   TT(K + 2, K + 2) = YLY
   TT(K + 2, K + 3) = ZLY
   TT(K + 3, K + 1) = disZ
   TT(K + 3, K + 2) = YLZ
   TT(K + 3, K + 3) = ZLZ
Next I
SA = (E * A) / L
TA = (G * Jo) / L
GY = (2 * E * IY) / L
GZ = (2 * E * IZ) / L
By = (3 * GY) / L
```

```
BZ = (3 * GZ) / L
AY = (2 * By) / L
AZ = (2 * BZ) / L

KE(1, 1) = SA
KE(2, 2) = AZ
KE(2, 6) = BZ
KE(3, 3) = AY
KE(3, 5) = -By
KE(4, 4) = TA
KE(5, 5) = 2 * GY
KE(6, 6) = 2 * GZ
KE(6, 2) = KE(2, 6)
KE(5, 3) = KE(3, 5)

KE(1, 7) = -SA
KE(2, 8) = -AZ
KE(2, 12) = BZ
KE(3, 9) = -AY
KE(3, 11) = -By
KE(4, 10) = -TA
KE(5, 9) = By
KE(5, 11) = GY
KE(6, 8) = -BZ
KE(6, 12) = GZ

KE(7, 1) = -SA
KE(8, 2) = -AZ
KE(8, 6) = -BZ
KE(9, 3) = -AY
KE(9, 5) = By
KE(10, 4) = -TA
KE(11, 3) = -By
KE(11, 5) = GY
KE(12, 2) = BZ
KE(12, 6) = GZ

KE(7, 7) = SA
KE(8, 8) = AZ
KE(8, 12) = -BZ
KE(9, 9) = AY
KE(9, 11) = By
KE(10, 10) = TA
```

```
KE(11, 11) = 2 * GY
KE(12, 12) = 2 * GZ
KE(12, 8) = KE(8, 12)
KE(11, 9) = KE(9, 11)

For I = 1 To 12
   For j = 1 To 12
      For K = 1 To 12
         KG(I, j) = KG(I, j) + TT(I, K) * KE(K, j)
      Next K
   Next j
Next I
For I = 1 To 12
   For j = 1 To 12
      KE(I, j) = 0
      For K = 1 To 12
         KE(I, j) = KE(I, j) + KG(I, K) * T(K, j)
      Next K
   Next j
Next I
End Sub
Sub VECQ(NODCOR(), COMPL(), Q(), NNODOS)
For I = 1 To NNODOS
   For j = 1 To 6
      K = NODCOR(I, j)
      If K <> 0 Then
         Q(K) = COMPL(I, j)
      End If
   Next j
Next I
End Sub
Sub MAXKBU(ML(), KBU, NELEM)
KBU = 0
For I = 1 To NELEM
   MLMIN = 99999
   MLMAX = 0
   For j = 1 To 12
      If ML(I, j) > 0 Then
         If ML(I, j) > MLMAX Then
            MLMAX = ML(I, j)
         End If
         If ML(I, j) < MLMIN Then
            MLMIN = ML(I, j)
```

```
        End If
      End If
    Next j
    DIFF = MLMAX - MLMIN
    If DIFF > KBU Then
      KBU = DIFF
    End If
  Next I
End Sub
Sub VCONECT(GRLIB(), INCDA(), NODCOR(), ML(), NNODOS, NELEM,
NGRL)
K = 0
For I = 1 To NNODOS
  For j = 1 To 6
    If GRLIB(I, j) = 0 Then
      K = K + 1
      NODCOR(I, j) = K
    End If
  Next j
Next I
NGRL = K
For I = 1 To NELEM
  Ni = INCDA(2 * I - 1)
  Nj = INCDA(2 * I)
  For j = 1 To 6
    ML(I, j) = NODCOR(Ni, j)
    ML(I, j + 6) = NODCOR(Nj, j)
  Next j
Next I
End Sub

Public Sub Luciano()
Dim valx As Double, valx2 As Double, Valx3 As Double, Valx4 As Double
frmWizard.pmsg True, "Calculando...", 70, "Análisis Estructural"
With frmWizard
NNODOS = Val(.txtP5Tramos.Text) + 1
NELEM = Val(.txtP5Tramos.Text)
If Vars.SistemaDec Then diam = CDbl(.txtP5Diametro.Text) / 1000 Else diam =
CDbl(.txtP5Diametro.Text) / 12
If Vars.SistemaDec Then espesor = Vars.espesor / 100 Else espesor = Vars.espesor /
12
diamInterno = diam - (2 * espesor)
ro = diam / 2
```

```
ri = diamInterno / 2
If Vars.SistemaDec Then E = 21000000000# Else E = 4301139016#
G = 0.41 * E
A = 3.1416 * (ro ^ 2 - ri ^ 2)
IY = (3.1416 / 4) * (ro ^ 4 - ri ^ 4)
IZ = IY
Jo = (3.1416 / 2) * (ro ^ 4 - ri ^ 4)

pres = CDbl(frmWizard.txtP2Presion.Text)
Fy = Vars.Sigma
w = CDbl(frmWizard.txtP4Pandeo.Text)

ReDim X(1 To NNODOS)
ReDim Y(1 To NNODOS)
ReDim Z(1 To NNODOS)
ReDim Xa(1 To NNODOS)
ReDim Ya(1 To NNODOS)
ReDim GRLIB(1 To NNODOS, 1 To 6)
ReDim COMPL(1 To NNODOS, 1 To 6)
ReDim NODCOR(1 To NNODOS, 1 To 6)
ReDim INCDA(1 To 2 * NELEM)
ReDim PROP(1 To NELEM, 1 To 4)
ReDim ML(1 To NELEM, 1 To 12)
Dim KE(1 To 12, 1 To 12)
ReDim FZAS(1 To NELEM, 1 To 12)

QQ = Vars.QQ
If QQ < 0 Then QQ = 180 + QQ
r = Vars.r

Lverificacion = (QQ * r * 3.1416) / 180          "Este es el valor de la longuitud del
tramo, en m

X(1) = CDbl(.grdP3Secciones.TextMatrix(Vars.Inicial, 1))
Y(1) = CDbl(.grdP3Secciones.TextMatrix(Vars.Inicial, 2))
Xa(1) = X(1)
Ya(1) = Y(1)
Z(1) = 0
Alfa = QQ / NELEM
Beta = (180 - QQ) / 2

For I = 1 To NELEM
   AnguloAlfa = Alfa * I
```

```
    Lp = Sqr((2 * (r ^ 2)) * (1 - Cos((3.1416 / 180) * (AnguloAlfa))))
    GKL = (180 - AnguloAlfa) / 2
      If I = 1 Then
        angulodecorte = 90 - GKL
        Vars.Lt = Lp
      End If
    AnguloCalculo = GKL - Beta
    deltax = Lp * Cos((3.1416 / 180) * (AnguloCalculo))
    deltay = Lp * Sin((3.1416 / 180) * (AnguloCalculo))
    X(I + 1) = X(1) + deltax
    Y(I + 1) = Y(1) + deltay
    Xa(I + 1) = X(I + 1)
    Ya(I + 1) = Y(I + 1)
    Z(I + 1) = 0
    INCDA(2 * I - 1) = I
    INCDA(2 * I) = I + 1
Next I

For I = 1 To NNODOS
    For j = 1 To 6
      GRLIB(I, j) = 0
    Next j
Next I
For I = 1 To 6 'NNODOS
    GRLIB(1, I) = 1
    GRLIB(NNODOS, I) = 1
Next I
For I = 1 To NNODOS
    For j = 1 To 6
      COMPL(I, j) = 0
    Next j
Next I

If Vars.SistemaDec Then tmp = 7760 Else tmp = 484.2244
valx = -CDbl(frmWizard.txtP2cargaviva.Text)
valx2 = -CDbl(frmWizard.txtP2cargaviento.Text)
Valx3 = tmp * A
Valx4 = Vars.PesoFluido * A
For I = 1 To NELEM
    Ni = INCDA(2 * I - 1)
    Nj = INCDA(2 * I)
    dx = X(Nj) - X(Ni)
    dy = Y(Nj) - Y(Ni)
```

```
    L = Sqr(dx ^ 2 + dy ^ 2)
    PP1 = -Valx3 * L / dx
    PF1 = -Valx4 * L / dx
    CX = dx / L
    CY = dy / L
    COMPL(Ni, 2) = COMPL(Ni, 2) + (PP1 + PF1 + valx) * dx / 2
    COMPL(Ni, 6) = COMPL(Ni, 6) + (PP1 + PF1 + valx) * dx ^ 2 / 12
    COMPL(Nj, 2) = COMPL(Nj, 2) + (PP1 + PF1 + valx) * dx / 2
    COMPL(Nj, 6) = COMPL(Nj, 6) - (PP1 + PF1 + valx) * dx ^ 2 / 12

    COMPL(Ni, 3) = COMPL(Ni, 3) + valx2 * L / 2
    COMPL(Ni, 5) = COMPL(Ni, 5) - (valx2 * L ^ 2 / 12) * CX
    COMPL(Ni, 4) = COMPL(Ni, 4) + (valx2 * L ^ 2 / 12) * CY
    COMPL(Nj, 3) = COMPL(Nj, 3) + valx2 * L / 2
    COMPL(Nj, 5) = COMPL(Nj, 5) + (valx2 * L ^ 2 / 12) * CX
    COMPL(Nj, 4) = COMPL(Nj, 4) - (valx2 * L ^ 2 / 12) * CY
Next I
'valx = (CDbl(frmWizard.txtP2cargaviva.Text) * Vars.Lt * Val(.txtP5Tramos.Text)) /
(Val(.txtP5Tramos.Text) - 2)
'valx2 = CDbl(frmWizard.txtP2cargaviento.Text) * Val(.txtP5Tramos.Text) * Vars.Lt
/ (Val(.txtP5Tramos.Text) - 2)
'For I = LBound(COMPL) + 1 To UBound(COMPL) - 1
  'COMPL(I, 2) = -valx
  'COMPL(I, 3) = -valx2
'Next I

For I = 1 To NELEM
      PROP(I, 1) = A
      PROP(I, 2) = IY
      PROP(I, 3) = IZ
      PROP(I, 4) = Jo
Next I

Call VCONECT(GRLIB(), INCDA(), NODCOR(), ML(), NNODOS, NELEM,
NGRL)
Call MAXKBU(ML(), KBU, NELEM)
ReDim K(1 To NGRL * (KBU + 1))
ReDim Q(0 To NGRL)
Call VECQ(NODCOR(), COMPL(), Q(), NNODOS)
For I = 1 To NELEM
    A = PROP(I, 1)
    IY = PROP(I, 2)
    IZ = PROP(I, 3)
```

```
    Jo = PROP(I, 4)
    Call MATKE(I, INCDA(), X(), Y(), Z(), KE(), E, G, A, IY, IZ, Jo)
    Call MATKABC(I, KE(), K(), ML(), KBU)
Next I
Call MATGAUSSFTR(K(), NGRL, KBU)
Call MATGAUSSFMI(K(), Q(), NGRL, KBU)
Call MATGAUSSFS(K(), Q(), NGRL, KBU)
For I = 1 To NELEM
    A = PROP(I, 1)
    IY = PROP(I, 2)
    IZ = PROP(I, 3)
    Jo = PROP(I, 4)
    Call FUERZAS(I, INCDA(), X(), Y(), Z(), ML(), Q(), FZAS(), E, G, A, IY, IZ, Jo)
Next I

Call Resultados(FZAS(), Porcentaje, Verificacion, pres, w, Fy, Lverificacion,
NELEM, E, diam, espesor)
With frmGraph.grdNum
  frmWizard.pmsg True, "Llenando Tablas..."
  .Clear
  .Rows = (UBound(FZAS) * 2) + 1
  .Cols = 7
  .WordWrap = True
  .RowHeight(0) = 500
  If Vars.SistemaDec Then tmp = "Kg" Else tmp = "Lb"
  .TextMatrix(0, 1) = "Fuerza Axial (" & tmp & ")"
  .TextMatrix(0, 2) = "Fuerza Cortante Fy (" & tmp & ")"
  .TextMatrix(0, 3) = "Fuerza Cortante Fz (" & tmp & ")"
  If Vars.SistemaDec Then tmp = "Kg * m" Else tmp = "Lb * Ft"
  .TextMatrix(0, 4) = "Momento Torsor (" & tmp & ")"
  .TextMatrix(0, 5) = "Momento Flector My (" & tmp & ")"
  .TextMatrix(0, 6) = "Momento Flector Mz (" & tmp & ")"

  For I = 1 To UBound(FZAS) * 2
    cn = cn + 1
    For j = 1 To 6
      .TextMatrix(I, j) = Format(FZAS(cn, j), "#0.000")
      .TextMatrix(I + 1, j) = Format(FZAS(cn, j + 6), "#0.000")
      .TextMatrix(I, 0) = "E" & cn & ", Nodo I"
      .TextMatrix(I + 1, 0) = "E" & cn & ", Nodo J"
    Next j
    I = I + 1
  Next I
```

```
  For I = 1 To .Cols - 1
    .ColAlignment(I) = 4
    .ColWidth(I) = 1500
  Next I
  .ColAlignment(0) = 4
  .ColWidth(0) = 990
End With
With frmGraph.grdNum2
  .Clear
  .Cols = 2
  .Rows = UBound(Q) + 1
  .ColAlignment(0) = 4
  .ColAlignment(1) = 4
  .ColWidth(0) = 1500
  If Vars.SistemaDec Then tmp = " (m)" Else tmp = " (Ft)"
  cn = 0
  For I = 1 To UBound(Q)
    cn = cn + 1
    If cn <= 3 Then
      tmpx = tmp
    ElseIf cn > 3 Then
      tmpx = " (rad)"
    End If
    .TextMatrix(I, 1) = Format(Q(I), "#0.0000")
    .TextMatrix(I, 0) = "N° " & I & tmpx
    If cn = 6 Then cn = 0
  Next I

End With
Vars.AnguloCorte = angulodecorte
End With
End Sub

Private Sub Form_Activate()
If loading Then
  pb1.value = 10
  Me.Refresh
  frmWizard.Show
  loading = False
End If
End Sub

Private Sub Form_Load()
```

```
loading = True
End Sub

Const NUM_STEPS = 12

Const RES_ERROR_MSG = 30000

'VALOR DE BASE PARA EL ARCHIVO DE AYUDA DE ESTE ASISTENTE:
Const HELP_BASE = 1000
Const HELP_FILE = "MIASISTENTE.HLP"

Const BTN_HELP = 0
Const BTN_CANCEL = 1
Const BTN_BACK = 2
Const BTN_NEXT = 3
Const BTN_FINISH = 4

Const STEP_INTRO = 0
Const STEP_1 = 1
Const STEP_2 = 2
Const STEP_3 = 3
Const STEP_4 = 4
Const STEP_5 = 5
Const STEP_6 = 6
Const STEP_7 = 7
Const STEP_8 = 8
Const STEP_9 = 9
Const STEP_FINISH = 10

Const DIR_NONE = 0
Const DIR_BACK = 1
Const DIR_NEXT = 2

Const FRM_TITLE = "DAPT - Asistente Para el Diseño De Puentes Tubo"
Const INTRO_KEY = "Introducción"
Const SHOW_INTRO = "ShowIntro"
Const TOPIC_TEXT = "<TOPIC_TEXT>"

'variables a nivel de módulo
Dim mnCurStep       As Integer
Dim mbHelpStarted   As Boolean
Public VBInst       As VBIDE.VBE
Dim mbFinishOK      As Boolean
```

```
Dim nRow As Integer, nCol As Integer

Private Sub cmbP4Grado_Click()
Select Case cmbP4Grado.Text
Case "A"
  If Vars.SistemaDec Then Vars.Sigma = 30000 / 14.2233 Else Vars.Sigma = 30000
Case "B"
  If Vars.SistemaDec Then Vars.Sigma = 35000 / 14.2233 Else Vars.Sigma = 35000
Case "X42"
  If Vars.SistemaDec Then Vars.Sigma = 42000 / 14.2233 Else Vars.Sigma = 42000
Case "X52"
  If Vars.SistemaDec Then Vars.Sigma = 52000 / 14.2233 Else Vars.Sigma = 52000
Case "X60"
  If Vars.SistemaDec Then Vars.Sigma = 60000 / 14.2233 Else Vars.Sigma = 60000
End Select
End Sub

Private Sub cmbP5Calculo_Click()
Dim tmp As String
If cmbP5Calculo.ListIndex = 0 Then
  txtP5Diametro.Visible = True
  cmbP5Diametro.Visible = False
  lblP5Diametro.Caption = "Ingrese el Diámetro de la tubería:"
Else
  txtP5Diametro.Visible = False
  cmbP5Diametro.Visible = True
  If cmbP4Grado.Visible = True Then tmp = cmbP4Grado.Text
  lblP5Diametro.Caption = "Diámetros para " & 
optP4Acero(optP4Acero(0).Tag).Caption & " " & tmp & ":"
End If
End Sub

Private Sub chkShowIntro_Click()
   If chkShowIntro.value Then
      SaveSetting APP_CATEGORY, WIZARD_NAME, INTRO_KEY,
SHOW_INTRO
   Else
      SaveSetting APP_CATEGORY, WIZARD_NAME, INTRO_KEY,
vbNullString
   End If
End Sub

Private Sub cmbP5Diametro_Click()
```

```
txtP5Diametro.Text = cmbP5Diametro.Text
End Sub

Private Sub cmdAutoCAD_Click()
Dim ln As AcadLine, start(0 To 2) As Double, endx(0 To 2) As Double, Yo As
Double, Xo  As Double, dimx2 As AcadDimAngular
Dim dimx As AcadDimAligned, entry As AcadLineType, lntemp As AcadLine, lnx
As AcadLine, miny As Double, maxy As Double
Dim found As Boolean, endx2(0 To 2) As Double, p As Double, Q As Double, minx
As Double, maxx As Double, tmp As String
Dim XaT As Double, YaT As Double, anc(0 To 3, 0 To 3), linep1(0 To 2) As
Double, linep2(0 To 2) As Double, angvert(0 To 2) As Double
pmsg True, "Creando Plano ACAD...", 25, "Inicializando Objeto ACAD..."
Set acadx = CreateObject("AutoCAD.Application")
pmsg True, "Creando Plano ACAD...", 40, "Abriendo Plantilla..."
acadx.Documents.Open (App.Path & "\template.dwg")
pmsg True, "Creando Plano ACAD...", 75, "Diseñando Sección del Cauce..."
Xo = (CDbl(txtP3Luz.Text) / 2) + CDbl(grdP3Secciones.TextMatrix(Vars.Inicial, 1))
Yo = CDbl(grdP3Secciones.TextMatrix(Vars.Inicial, 2)) + (CDbl(txtP3Altura.Text) /
2) - ((CDbl(txtP3Luz.Text) ^ 2) / (8 * CDbl(txtP3Altura.Text)))
rr = Sqr((Vars.Altura ^ 2 / 4) + (CDbl(txtP3Luz.Text) ^ 2 / 8) +
((CDbl(Me.txtP3Luz.Text) ^ 4) / (64 * Vars.Altura ^ 2)))
If Vars.SistemaDec Then
  r1 = rr - (((Vars.Diametro / 1000) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2)
  r2 = rr + (((Vars.Diametro / 1000) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2)
Else
  r1 = rr - (((Vars.Diametro / 12) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2)
  r2 = rr + (((Vars.Diametro / 12) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2)
End If
With acadx.ActiveDocument

  'Dimensionado de la luz libre
  start(0) = Xa(1)
  start(1) = Ya(1)
  start(2) = 0
  endx(0) = Xa(UBound(Xa))
  endx(1) = Ya(UBound(Ya))
  endx(2) = 0
  endx2(0) = (endx(0) + start(0)) / 2
  endx2(1) = Ya(1) - 2.5
  endx2(2) = 0
  Set dimx = .ModelSpace.AddDimAligned(start, endx, endx2)
  dimx.LinetypeScale = 0.3
```

```
If Vars.SistemaDec Then tmp = "m" Else tmp = "Ft"

'Dimensionado de la altura
start(0) = (Xa(UBound(Xa)) + Xa(1)) / 2
start(1) = Ya(1)
start(2) = 0
endx(0) = (Xa(UBound(Xa)) + Xa(1)) / 2
If InStr(1, Str$(txtP5Tramos.Text / 2), ".") > 0 Or InStr(1, Str$(txtP5Tramos.Text /
2), ",") > 0 Then
  If Vars.SistemaDec Then endx(1) = Ya(1) + Vars.Altura - (Vars.Diametro / 1000) /
2 Else endx(1) = Ya(1) + Vars.Altura - (Vars.Diametro / 12) / 2
Else
  If Vars.SistemaDec Then endx(1) = Ya(1) + Vars.Altura - (((Vars.Diametro /
1000) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2) Else endx(1) = Ya(1) +
Vars.Altura - (((Vars.Diametro / 12) / Cos((Vars.AnguloCorte * 3.1416) / 180)) / 2)
End If
endx(2) = 0
endx2(0) = (Xa(UBound(Xa)) + Xa(1)) / 2
endx2(1) = ((endx(1) + start(1)) / 2)
endx2(2) = 0
Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
dimx.LinetypeScale = 0.3

'If Vars.SistemaDec Then
' tmp = "m"
' dimx.TextOverride = Format(Vars.Altura + (Vars.Diametro / 1000) / 2, "0.##") &
" " & tmp
'Else
' tmp = "Ft"
' dimx.TextOverride = Format(Vars.Altura + (Vars.Diametro / 12) / 2, "0.##") & " "
& tmp
'End If
'dimx.Update


'Nombre del Proyecto
start(0) = ((Xa(UBound(Xa)) + Xa(1)) / 2)
start(1) = endx(1) + 6
Set tx = .ModelSpace.AddText("Proyecto " & Vars.NombreProy, start, 0.7)

start(0) = ((Xa(UBound(Xa)) + Xa(1)) / 2) - ((Xa(UBound(Xa)) - Xa(1)) / 4)
start(1) = endx(1) + 3.3
```

```
Set tx = .ModelSpace.AddText("Sección Transversal del Puente tubo", start, 0.5)

start(0) = Xo - 5
start(1) = Ya(1) - 2
Set tx = .ModelSpace.AddText("Sección Transversal del Cauce", start, 0.25)

start(0) = (grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) - 1
start(1) = endx(1) - 1
Set tx = .ModelSpace.AddText("Detalle del Tramo", start, 0.5)

'Dibujo de la seccion del cauce
For I = 1 To frmWizard.grdP3Secciones.Rows - 2
  start(0) = frmWizard.grdP3Secciones.TextMatrix(I, 1)
  start(1) = frmWizard.grdP3Secciones.TextMatrix(I, 2)
  start(2) = 0
  endx(0) = frmWizard.grdP3Secciones.TextMatrix(I + 1, 1)
  endx(1) = frmWizard.grdP3Secciones.TextMatrix(I + 1, 2)
  endx(2) = 0
  Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
  found = False
  For Each entry In acadx.ActiveDocument.Linetypes
    If StrComp(entry.name, "TRAZOS", 1) = 0 Then
       found = True
       Exit For
    End If
  Next
  If Not (found) Then acadx.ActiveDocument.Linetypes.Load "TRAZOS",
"acad.lin"
  ln.Linetype = "TRAZOS"
  ln.LinetypeScale = 0.3
  ln.Update
Next I

'Dibujo del eje de la tuberia
pmsg True, "Creando Plano ACAD...", 95, "Diseñando Tubería..."
For I = 1 To UBound(Xa) - 1
  start(0) = Xa(I)
  start(1) = Ya(I)
  start(2) = 0
  endx(0) = Xa(I + 1)
  endx(1) = Ya(I + 1)
  endx(2) = 0
  Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
```

```
  found = False
  For Each entry In acadx.ActiveDocument.Linetypes
    If StrComp(entry.name, "DASHDOT", 1) = 0 Then
       found = True
       Exit For
    End If
  Next
  If Not (found) Then acadx.ActiveDocument.Linetypes.Load "DASHDOT",
"acad.lin"
  ln.Linetype = "DASHDOT"
  ln.LinetypeScale = 0.4
  ln.Update
 Next I

 'Dibujo de la Tuberia
 For I = 1 To UBound(Xa)
  If (Format(Xa(I), "0.##") - Format(Xo, "0.##")) <> 0 Then
    m = (Ya(I) - Yo) / (Xa(I) - Xo)
    If m < 0 Then
      start(0) = -Sqr((r2 ^ 2) / (1 + (m ^ 2))) + Xo
      start(1) = Yo + (m * (start(0) - Xo))
      start(2) = 0
      endx(0) = -Sqr((r1 ^ 2) / (1 + (m ^ 2))) + Xo
      endx(1) = Yo + (m * (endx(0) - Xo))
      endx(2) = 0
    Else
      start(0) = Sqr((r2 ^ 2) / (1 + m ^ 2)) + Xo
      start(1) = Yo + (m * (start(0) - Xo))
      start(2) = 0
      endx(0) = Sqr((r1 ^ 2) / (1 + m ^ 2)) + Xo
      endx(1) = Yo + (m * (endx(0) - Xo))
      endx(2) = 0
    End If
  Else
    endx(0) = Xa(1) + CDbl(txtP3Luz.Text) / 2
    endx(1) = Yo + r1
    endx(2) = 0
    start(0) = Xa(1) + CDbl(txtP3Luz.Text) / 2
    start(1) = Yo + r2
    start(2) = 0
  End If
  Select Case I
  Case 1
```

```
    anc(0, 0) = start(0)
    anc(0, 1) = start(1)
    anc(0, 2) = endx(0)
    anc(0, 3) = endx(1)
  Case 2
    anc(1, 0) = start(0)
    anc(1, 1) = start(1)
    anc(1, 2) = endx(0)
    anc(1, 3) = endx(1)
  Case UBound(Xa) - 1
    anc(2, 0) = start(0)
    anc(2, 1) = start(1)
    anc(2, 2) = endx(0)
    anc(2, 3) = endx(1)
  Case UBound(Xa)
    anc(3, 0) = start(0)
    anc(3, 1) = start(1)
    anc(3, 2) = endx(0)
    anc(3, 3) = endx(1)
  End Select
  Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
  If I > 1 Then
    start2 = lntemp.StartPoint
    end2 = ln.StartPoint
    Set lnx = acadx.ActiveDocument.ModelSpace.AddLine(start2, end2)
    start2 = lntemp.EndPoint
    end2 = ln.EndPoint
    Set lnx = acadx.ActiveDocument.ModelSpace.AddLine(start2, end2)
  End If
  Set lntemp = ln
Next I

'Dibujo del bloque de anclaje este

'1 a 2
ancl = CDbl(Mid$(Me.grdResumen.TextMatrix(9, 1), 1,
Len(grdResumen.TextMatrix(9, 1)) - 2))
start(0) = Xa(UBound(Xa))
start(1) = Ya(UBound(Ya))
start(2) = 0
endx(0) = Xa(UBound(Xa)) + ancl
endx(1) = Ya(UBound(Ya))
endx(2) = 0
```

```
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'2 a 3
start(0) = Xa(UBound(Xa)) + ancl
start(1) = Ya(UBound(Ya))
start(2) = 0
endx(0) = Xa(UBound(Xa)) + ancl
endx(1) = Ya(UBound(Ya)) - ancl
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'Dimension de la recta 2-3 anclaje este
endx2(0) = Xa(UBound(Xa)) + ancl + 0.5
endx2(1) = ((endx(1) + start(1)) / 2)
endx2(2) = 0
Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
dimx.LinetypeScale = 0.3
If Vars.SistemaDec Then tmp = "m" Else tmp = "Ft"
'dimx.TextOverride = Format(Vars.Lt + ((Vars.Diametro / 1000) * Tan((3.1416 /
180) * Vars.AnguloCorte)), "0.##") & " " & tmp
dimx.Update

start(0) = endx(0)
start(1) = endx(1) - 1
Set tx = .ModelSpace.AddText("Fundación Directa #2", start, 0.2)

'3 a 4
start(0) = Xa(UBound(Xa)) + ancl
start(1) = Ya(UBound(Ya)) - ancl
start(2) = 0
endx(0) = Xa(UBound(Xa))
endx(1) = Ya(UBound(Ya)) - ancl
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'4 a 5
start(0) = Xa(UBound(Xa))
start(1) = Ya(UBound(Ya)) - ancl
start(2) = 0
endx(0) = Xa(UBound(Xa))
endx(1) = Ya(UBound(Ya))
endx(2) = 0
```

```
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'Dibujo del bloque de anclaje oeste

'1 a 2
start(0) = Xa(1)
start(1) = Ya(1)
start(2) = 0
endx(0) = Xa(1) - ancl
endx(1) = Ya(1)
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'2 a 3
start(0) = Xa(1) - ancl
start(1) = Ya(1)
start(2) = 0
endx(0) = Xa(1) - ancl
endx(1) = Ya(1) - ancl
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

start(0) = endx(0) - 4
start(1) = endx(1) - 1
Set tx = .ModelSpace.AddText("Fundación Directa #1", start, 0.2)

'3' a 4'
start(0) = Xa(1) - ancl
start(1) = Ya(1) - ancl
start(2) = 0
endx(0) = Xa(1)
endx(1) = Ya(1) - ancl
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
endx2(0) = ((endx(0) + start(0)) / 2)
endx2(1) = Ya(1) - ancl - 0.5
endx2(2) = 0
Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
dimx.LinetypeScale = 0.3
If Vars.SistemaDec Then tmp = "m" Else tmp = "Ft"
'dimx.TextOverride = Format(Vars.Lt + ((Vars.Diametro / 1000) * Tan((3.1416 /
180) * Vars.AnguloCorte)), "0.##") & " " & tmp
```

```
dimx.Update

'4' a 5'
start(0) = Xa(1)
start(1) = Ya(1) - ancl
start(2) = 0
endx(0) = Xa(1)
endx(1) = Ya(1)
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)


'Lineas para anclar la tuberia anclaje derecho
start(0) = anc(3, 0)
start(1) = anc(3, 1)
start(2) = 0
endx(0) = (((anc(2, 0) - anc(3, 0)) / (anc(2, 1) - anc(3, 1))) * (Ya(UBound(Ya)) -
anc(3, 1))) + anc(3, 0)
endx(1) = Ya(UBound(Ya))
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
start(0) = anc(3, 2)
start(1) = anc(3, 3)
start(2) = 0
endx(0) = Xa(UBound(Xa))
endx(1) = (((anc(2, 3) - anc(3, 3)) / (anc(2, 2) - anc(3, 2))) * (Xa(UBound(Xa)) -
anc(3, 2))) + anc(3, 3)
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

'Lineas para anclar la tuberia anclaje izquierdo
start(0) = anc(0, 0)
start(1) = anc(0, 1)
start(2) = 0
endx(0) = (((anc(1, 0) - anc(0, 0)) / (anc(1, 1) - anc(0, 1))) * (Ya(1) - anc(0, 1))) +
anc(0, 0)
endx(1) = Ya(1)
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
start(0) = anc(0, 2)
start(1) = anc(0, 3)
start(2) = 0
endx(0) = Xa(1)
```

```
 endx(1) = (((anc(1, 3) - anc(0, 3)) / (anc(1, 2) - anc(0, 2))) * (Xa(1) - anc(0, 2))) +
anc(0, 3)
 endx(2) = 0
 Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)



 'Usar para dimensionar el tramo pqra detalle
 start(0) = anc(0, 0)
 start(1) = anc(0, 1)
 start(2) = 0
 endx(0) = anc(1, 0)
 endx(1) = anc(1, 1)
 endx(2) = 0
 endx2(0) = ((endx(0) + start(0)) / 2) + 1.5
 endx2(1) = ((endx(1) + start(1)) / 2) + 1.5
 endx2(2) = 0
 Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
 dimx.LinetypeScale = 0.001
 dimx.DimensionLineColor = acBlue
 If Vars.SistemaDec Then tmp = "m" Else tmp = "Ft"
 'dimx.TextOverride = Format(Vars.Lt + ((Vars.Diametro / 1000) * Tan((3.1416 /
180) * Vars.AnguloCorte)), "0.##") & " " & tmp
 dimx.Update

 start(0) = anc(0, 2)
 start(1) = anc(0, 3)
 start(2) = 0
 endx(0) = anc(1, 2)
 endx(1) = anc(1, 3)
 endx(2) = 0
 endx2(0) = ((endx(0) + start(0)) / 2) - 1.5
 endx2(1) = ((endx(1) + start(1)) / 2) - 1.5
 endx2(2) = 0
 Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
 dimx.LinetypeScale = 0.3
 dimx.DimensionLineColor = acBlue
 If Vars.SistemaDec Then tmp = "m" Else tmp = "Ft"
 'dimx.TextOverride = Format(Vars.Lt + ((Vars.Diametro / 1000) * Tan((3.1416 /
180) * Vars.AnguloCorte)), "0.##") & " " & tmp
```

```
dimx.Update



'Dibujo da la seccion

start(0) = Xa(1) + 2
start(1) = Ya(1)
Set tx = .ModelSpace.AddText("Tramo típico (ver detalle)", start, 0.2)

start(0) = grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5
start(1) = Ya(UBound(Ya))
start(2) = 0
If Vars.SistemaDec Then endx(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Vars.Lt +
(Tan(3.1416 * Vars.AnguloCorte / 180) * (Vars.Diametro / 1000))) Else endx(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Vars.Lt +
(Tan(3.1416 * Vars.AnguloCorte / 180) * (Vars.Diametro / 12)))
endx(1) = Ya(UBound(Ya))
endx(2) = 0
Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
endx2(0) = ((endx(0) + start(0)) / 2)
endx2(1) = Ya(UBound(Ya)) + 1
endx2(2) = 0
Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
dimx.LinetypeScale = 0.3
dimx.Update

angvert(0) = endx(0)
angvert(1) = endx(1)
angvert(2) = 0
linep1(0) = ((endx(0) - start(0)) / 3) + start(0)
linep1(1) = start(1)
linep1(2) = 0
If Vars.SistemaDec Then start(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Vars.Lt +
Tan(3.1416 * Vars.AnguloCorte / 180) * (Vars.Diametro / 1000)) Else start(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Vars.Lt +
Tan(3.1416 * Vars.AnguloCorte / 180) * (Vars.Diametro / 12))
start(1) = Ya(UBound(Ya))
```

```
 start(2) = 0
 endx(0) = (grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + Vars.Lt
 If Vars.SistemaDec Then endx(1) = Ya(UBound(Ya)) - (Vars.Diametro / 1000) Else
endx(1) = Ya(UBound(Ya)) - (Vars.Diametro / 12)
 endx(2) = 0
 Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
 endx2(0) = ((endx(0) + start(0)) / 2) + 1
 endx2(1) = ((endx(1) + start(1)) / 2) + 1
 endx2(2) = 0
 dimx.LinetypeScale = 0.3
 dimx.Update
 Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)

 linep2(0) = ((start(0) - endx(0)) / 3) + endx(0)
 linep2(1) = ((start(1) - endx(1)) / 3) + endx(1)
 linep2(2) = 0
 endx2(0) = (linep1(0) + linep2(0)) / 2
 endx2(1) = (linep1(1) + linep2(1)) / 2
 endx2(2) = 0
 Set dimx2 = acadx.ActiveDocument.ModelSpace.AddDimAngular(angvert, linep2,
linep1, endx2)

 start(0) = (grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + Vars.Lt
 If Vars.SistemaDec Then start(1) = Ya(UBound(Ya)) - (Vars.Diametro / 1000) Else
start(1) = Ya(UBound(Ya)) - (Vars.Diametro / 12)
 start(2) = 0
 If Vars.SistemaDec Then endx(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Tan(3.1416 *
Vars.AnguloCorte / 180) * (Vars.Diametro / 1000)) Else endx(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Tan(3.1416 *
Vars.AnguloCorte / 180) * (Vars.Diametro / 12))
 If Vars.SistemaDec Then endx(1) = Ya(UBound(Ya)) - (Vars.Diametro / 1000) Else
endx(1) = Ya(UBound(Ya)) - (Vars.Diametro / 12)
 endx(2) = 0
 Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)
 endx2(0) = ((endx(0) + start(0)) / 2)
 If Vars.SistemaDec Then endx2(1) = Ya(UBound(Ya)) - (Vars.Diametro / 1000) -
1.5 Else endx2(1) = Ya(UBound(Ya)) - (Vars.Diametro / 12) - 1.5
 endx2(2) = 0
 Set dimx = acadx.ActiveDocument.ModelSpace.AddDimAligned(start, endx,
endx2)
 dimx.LinetypeScale = 0.3
```

```
 dimx.Update

 If Vars.SistemaDec Then start(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Tan(3.1416 *
Vars.AnguloCorte / 180) * (Vars.Diametro / 1000)) Else start(0) =
(grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5) + (Tan(3.1416 *
Vars.AnguloCorte / 180) * (Vars.Diametro / 12))
 If Vars.SistemaDec Then start(1) = Ya(UBound(Ya)) - (Vars.Diametro / 1000) Else
start(1) = Ya(UBound(Ya)) - (Vars.Diametro / 12)
 start(2) = 0
 endx(0) = grdP3Secciones.TextMatrix(grdP3Secciones.Rows - 1, 1) + 5
 endx(1) = Ya(UBound(Ya))
 endx(2) = 0
 Set ln = acadx.ActiveDocument.ModelSpace.AddLine(start, endx)

 acadx.ActiveDocument.SaveAs (App.Path & "\" & Vars.NombreProy)


End With
acadx.Visible = True
acadx.ZoomAll
acadx.WindowState = acMax
pmsg False
End Sub

Private Sub cmdNav_Click(Index As Integer)
   Dim nAltStep As Integer
   Dim lHelpTopic As Long
   Dim rc As Long

   Select Case Index
      Case BTN_HELP
         'If wd = Nothing Then
         'If Not mbHelpStarted Then
           On Error Resume Next
           pmsg True, "Leyendo...", 75
           dd = wd.Visible
           If Err = 91 Or Err = 462 Then
             On Error GoTo 0
             pmsg True, "Activando...", 25
             Set wd = CreateObject("Word.Application")
             If Err <> 0 Then Set wd = CreateObject("Word.Application")
           End If
```

```
        If Not CheckOpen("ayuda_0" & mnCurStep - 1 & ".doc", "wd") Then
wd.Documents.Open App.Path & "\ayuda_0" & mnCurStep - 1, , True
        On Error GoTo 0
        'mbHelpStarted = True
        pmsg False
      'End If
      'End If

      wd.ActiveDocument.Activate
      wd.Visible = True
      wd.Activate
      wd.WindowState = wdWindowStateMaximize

      'wd.Documents ("ayuda").

      'lHelpTopic = HELP_BASE + 10 * (1 + mnCurStep)
      'rc = WinHelp(Me.hwnd, HELP_FILE, HELP_CONTEXT, lHelpTopic)

    Case BTN_CANCEL
      pmsg True, "Cerrando...", 35, "Cerrando vinculos dinámicos..."
      xs.Quit
      wd.Quit
      acadx.Quit
      Unload Me
      End

    Case BTN_BACK
      'colocar aquí casos especiales para saltar
      'a pasos alternativos
      'If mnCurStep = 7 And optModo(1).Value Then
      '  nAltStep = 1
      'Else
      nAltStep = mnCurStep - 1
      If nAltStep = 7 And Vars.Modo = 1 Then nAltStep = 6
      'End If
      SetStep nAltStep, DIR_BACK

    Case BTN_NEXT
      'colocar aquí casos especiales para saltar
      'a pasos alternativos
      'If mnCurStep = 1 And optModo(1).Value Then
      '  nAltStep = 7
      'Else
```

```
        nAltStep = mnCurStep + 1
        'End If
        If nAltStep = 7 And Vars.Modo = 1 Then nAltStep = 8
        SetStep nAltStep, DIR_NEXT

      Case BTN_FINISH
        'el código de creación de asistentes va aquí

        Unload Me

        If  GetSetting(APP_CATEGORY,  WIZARD_NAME,  CONFIRM_KEY,
vbNullString) = vbNullString Then
          frmConfirm.Show vbModal
        End If

    End Select
End Sub

Private Sub cmdNumerico_Click()
grdP3Secciones.Tag = False
frmGraph.Show
pmsg False
End Sub

Private Sub cmdOpen_Click()
With cmDialog
  .Flags = cdlOFNFileMustExist Or cdlofnhisereadonly Or cdlOFNLongNames Or
cdlOFNNoReadOnlyReturn
  .Filter = "Archivos DAPT/Excel (*.xls)|*.xls|Todos los archivos (*.*)|*.*"
  .FilterIndex = 1
  .InitDir = App.Path
  On Error Resume Next
  .ShowOpen
  If Err = 32755 Then Exit Sub
End With
pmsg True, "Abriendo...", 25, "Abriendo Archivo..."
optModo(0).value = True
xs.Workbooks.Open (cmDialog.FileName)
With xs.Workbooks(cmDialog.FileTitle).Sheets("Proyecto")
  pmsg True, "Verificando...", 0
  If .Range("F1") <> "DAPT 1.0" Then
    Beep
```

```
  MsgBox "El Archivo " & fpath & " no es un archivo DAPT válido.", vbCritical +
vbOKOnly, "Error"
  pmsg False
  Exit Sub
 End If
 pmsg True, "Abriendo...", 35, "Recopilando Data"
 txtP1Nombre.Text = .Range("B1")
 txtP1loc.Text = .Range("B2")
 txtP1Fecha.Text = .Range("B3")
 If CBool(.Range("B4")) Then Me.optSistInternacional = True: Vars.SistemaDec =
True Else Me.optSistIngles.value = True: Vars.SistemaDec = False
 cmbP2Fluido.ListIndex = Val(.Range("B5"))
 Me.txtP2Temp.Text = .Range("B6")
 Me.txtP2Otro.Text = .Range("B7")
 Me.txtP2Presion.Text = .Range("B8")
 Me.txtP2Sobrepresion.Text = .Range("B9")
 Me.txtP2cargaviento.Text = .Range("B10")
 Me.txtP2cargaviva.Text = .Range("B11")
 pmsg True, "Abriendo...", 65, "Recopilando Data..."
 Me.txtP3Luz.Text = .Range("B12")
 Me.txtP3Altura.Text = .Range("B13")
 Vars.Altura = txtP3Altura.Text
 optP4Acero(.Range("B14")).value = True
 cmbP4Grado.ListIndex = .Range("B15")
 Me.txtP4K1.Text = .Range("B17")
 Me.txtP4K2.Text = .Range("B18")
 Me.txtP4Pandeo.Text = .Range("B19")
 cmbP5Calculo.ListIndex = .Range("B20")
 Vars.book = .Range("B28")
 cmbP5Diametro.Clear
 For Each cell In
xs.Workbooks("espesores.dat").Worksheets(Vars.book).Range("A1:A30")
  If cell.value = "" Then Exit For
  If Vars.SistemaDec Then cmbP5Diametro.AddItem Format((cell.value / 0.0394),
"###.###") Else cmbP5Diametro.AddItem cell.value
 Next cell
 If cmbP5Calculo.ListIndex = 0 Then txtP5Diametro.Text = .Range("B21") Else
cmbP5Diametro.ListIndex = .Range("B21")
 Me.txtP5Espesor.Text = .Range("B22")
 txtP5Tramos.Text = .Range("B23")
 txtP6Caudal.Text = .Range("B24")
 txtP6Friccion.Text = .Range("B25")
 txtP6Peso.Text = .Range("B26")
```

```
  txtP6Soporte.Text = .Range("B27")
  txtP3Secciones.Text = .Range("C2")
  txtP3Altura.Enabled = True
  Call cmdP3Secciones_Click
  pmsg True, "Abriendo...", 85, "Recopilando Data..."
  For I = 1 To grdP3Secciones.Rows - 1
    grdP3Secciones.TextMatrix(I, 1) = .Range("D" & I)
    grdP3Secciones.TextMatrix(I, 2) = .Range("E" & I)
  Next I
  Me.txtP3InitX.Text = grdP3Secciones.TextMatrix(1, 1)
  Me.txtP3InitY.Text = grdP3Secciones.TextMatrix(1, 2)
End With
Call cmdNav_Click(BTN_NEXT)
pmsg False
End Sub

Public Sub cmdP3Borrar_Click()
InitGrid
txtP3Secciones.Text = 3
Call cmdP3Secciones_Click
EnumGrid
txtP3Luz.Text = ""
txtP3Altura.Text = ""
txtP3InitX = ""
txtP3InitY = ""
End Sub

Private Sub cmdP3Graph_Click()
Dim I As Integer
For I = 1 To grdP3Secciones.Rows - 1
  If grdP3Secciones.TextMatrix(I, 1) = "" Then grdP3Secciones.TextMatrix(I, 1) = 0
  If grdP3Secciones.TextMatrix(I, 2) = "" Then grdP3Secciones.TextMatrix(I, 2) = 0
Next I
grdP3Secciones.Tag = True
frmGraph.Show
End Sub

Public Sub cmdP3Secciones_Click()
  Dim rw As Integer
  If Not IsNumeric(Val(txtP3Secciones.Text)) Or (Val(txtP3Secciones.Text) < 3 Or
Val(txtP3Secciones.Text) > 40) Then
    Beep
```

```
    MsgBox "Debe introducir un valor numérico válido entre 3 y 40.", vbCritical +
vbOKOnly, "Error"
    txtP3Secciones.Text = grdP3Secciones.Rows - 1
    Exit Sub
  End If
  grdP3Secciones.Rows = Val(txtP3Secciones.Text) + 1
  If Vars.Inicial >= grdP3Secciones.Rows Then
    grdP3Secciones.TextMatrix(1, 3) = "*"
    Vars.Inicial = 1
  End If
  EnumGrid
  cmdP3Secciones.Default = False
  If mnCurStep <> 1 Then
    txtP3Luz.Text = ""
    txtP3Altura.Text = ""
    Vars.Altura = 0
  End If
End Sub

Private Sub cmdp5calc_Click()
Dim diam As Double, luz As Double, Altura As Double, r As Double, m1 As Double,
m2 As Double, QQ As Double
Dim Lt As Double, NELEM As Byte, maxdiam As Double, diam1 As Double, diam2
As Double, tmp As Double
Vars.Diametro = CDbl(txtP5Diametro.Text)
If optModo(0).Index = 0 Then
  If Vars.SistemaDec Then
    diam = Vars.Diametro / 1000
    maxdiam = 0.4572
    diam1 = 0.5
    diam2 = 1
  Else
    diam = Vars.Diametro / 12
    maxdiam = 1.5
    diam1 = 1.6404
    diam2 = 3.2808
  End If
  luz = CDbl(txtP3Luz.Text) '13.8
  Altura = CDbl(txtP3Altura.Text)  '2.2
  r = Sqr(((Altura ^ 2) / 4) + ((luz ^ 2) / 8) + ((luz ^ 4) / (64 * (Altura ^ 2))))
  m1 = (r - Altura) / (luz / 2)
  m2 = (r - Altura) / (-1 * (luz / 2))
  QQ = (180 / 3.1416) * (Atn((m2 - m1) / (1 + m1 * m2)))
```

```
  Vars.QQ = QQ
  Vars.r = r

  For NELEM = 2 To 25
     Lt = Sqr((2 * (r ^ 2)) * (1 - Cos((3.1416 / 180) * (QQ / NELEM))))
     If diam > maxdiam Then
       If Lt > diam And Lt < 2 * diam Then
         txtP5Tramos.Tag = NELEM
         'Vars.Lt = Lt
         Exit Sub
       End If
     Else
       If Lt > diam1 And Lt < diam2 Then
         txtP5Tramos.Tag = NELEM
         'Vars.Lt = Lt
         Exit Sub
       End If
     End If
  Next NELEM
  If Vars.SistemaDec Then
    diam1 = 1.5
    maxdiam = 1.45
  Else
    diam1 = 4.92
    maxdiam = 4.76
  End If
  If diam > maxdiam Then
    tmp = 1 - (diam1 ^ 2 / (2 * r ^ 2))
    NELEM = Int(QQ / (Atn(-tmp / Sqr(-tmp * tmp + 1)) * 180 / 3.1416) + 2 * Atn(1)
* (180 / 3.1416))
  Else
    tmp = 1 - (maxdiam ^ 2 / (2 * r ^ 2))
    NELEM = Int(QQ / (Atn(-tmp / Sqr(-tmp * tmp + 1)) * 180 / 3.1416) + 2 * Atn(1)
* (180 / 3.1416))
  End If
  If NELEM > 50 Then NELEM = 50
  txtP5Tramos.Tag = NELEM
End If
End Sub

Private Sub Command1_Click()
pmsg True, "Guardando...", 25, "Creando Archivo..."
xs.Workbooks.Open (App.Path & "\save.dat")
```

```
With xs.Workbooks("save.dat").Worksheets("Proyecto")
  pmsg True, "Guardando...", 35, "Recopilando Data"
  .Unprotect ("daptxdaptx")
  .Range("B1") = Vars.NombreProy
  .Range("B2") = Vars.LocalidadProy
  .Range("B3") = Vars.FechaProy
  .Range("B4") = Vars.SistemaDec
  .Range("B5") = Vars.TipoFluido
  .Range("B6") = Vars.Temperatura
  .Range("B7") = Vars.PesoFluido
  .Range("B8") = CDbl(Me.txtP2Presion.Text)
  .Range("B9") = CDbl(Me.txtP2Sobrepresion.Text)
  .Range("B10") = CDbl(Me.txtP2cargaviento.Text)
  .Range("B11") = CDbl(Me.txtP2cargaviva.Text)
  pmsg True, "Guardando...", 65, "Recopilando Data..."
  .Range("B12") = CDbl(Me.txtP3Luz.Text)
  .Range("B13") = CDbl(Me.txtP3Altura.Text)
  .Range("B14") = optP4Acero(0).Tag
  .Range("B15") = cmbP4Grado.ListIndex
  .Range("B16") = Vars.Sigma
  .Range("B17") = CDbl(Me.txtP4K1.Text)
  .Range("B18") = CDbl(Me.txtP4K2.Text)
  .Range("B19") = CDbl(Me.txtP4Pandeo.Text)
  .Range("B20") = cmbP5Calculo.ListIndex
  If cmbP5Calculo.ListIndex = 0 Then .Range("B21") = CDbl(txtP5Diametro.Text)
Else .Range("B21") = cmbP5Diametro.ListIndex
  .Range("B22") = Vars.espesor
  .Range("B23") = txtP5Tramos.Text
  .Range("B24") = CDbl(txtP6Caudal.Text)
  .Range("B25") = CDbl(txtP6Friccion.Text)
  .Range("B26") = CDbl(txtP6Peso.Text)
  .Range("B27") = CDbl(txtP6Soporte.Text)
  .Range("B28") = Vars.book
  .Range("C2") = Val(txtP3Secciones.Text)
  pmsg True, "Guardando...", 85, "Recopilando Data..."
  For I = 1 To grdP3Secciones.Rows - 1
    .Range("D" & I) = grdP3Secciones.TextMatrix(I, 1)
    .Range("E" & I) = grdP3Secciones.TextMatrix(I, 2)
  Next I
  .Range("F1") = "DAPT 1.0"
End With
With xs.Workbooks("save.dat").Worksheets("Resultados Numéricos")
For I = 0 To 6
```

```
  For j = 1 To frmGraph.grdNum.Rows - 1
    .Range(Chr$(65 + I) & j + 2) = frmGraph.grdNum.TextMatrix(j, I)
  Next j
Next I
For I = 0 To 1
  For j = 1 To frmGraph.grdNum2.Rows - 1
    .Range(Chr$(73 + I) & j + 2) = frmGraph.grdNum2.TextMatrix(j, I)
  Next j
Next I
End With
If CheckOpen(Vars.NombreProy & ".xls", "xs") = True Then
  xs.Workbooks(Vars.NombreProy).Save
Else
  xs.ActiveWorkbook.SaveAs (App.Path & "\" & Vars.NombreProy & ".xls")
End If
xs.ActiveWorkbook.Worksheets("Proyecto").Protect "daptxdaptx", True, True, True,
True
pmsg False
MsgBox "Archivo guardado exitosamente en " & App.Path & "\" &
Vars.NombreProy & ".xls"
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
   If KeyCode = vbKeyF1 Then
      cmdNav_Click BTN_HELP
   End If
End Sub

Private Sub Form_Load()
   Dim I As Integer
   'inicializar todas las variables
   pmsg True, "Inicializando...", 15
   mbFinishOK = False

   For I = 0 To NUM_STEPS - 1
     fraStep(I).Left = -10000
   Next

   pmsg True, "Inicializando...", 45, "Creando vínculos dinámicos..."
   Set xs = CreateObject("Excel.Application")
   xs.Workbooks.Open (App.Path & "\espesores.dat")
   'pmsg True, "AutoCAD..."
   'Set acadx = CreateObject("AutoCAD.application")
```

```
  'Cargar toda la información de cadenas para el formulario
  'LoadResStrings Me

  'Determinar el primer paso:

  pmsg True, "Inicializando...", 65, "Inicializando..."
  Vars.Separator = Asc(xs.International(xlDecimalSeparator))

  If GetSetting(APP_CATEGORY, WIZARD_NAME, INTRO_KEY, vbNullString)
= SHOW_INTRO Then
     chkShowIntro.value = vbChecked
     SetStep 1, DIR_NEXT
  Else
     SetStep 0, DIR_NONE
  End If
  With cmbP2Fluido
    .AddItem "Agua"
    .AddItem "Alcohol Etílico"
    .AddItem "Benceno"
    .AddItem "Gasolina"
    .AddItem "Kerosene"
    .AddItem "Glicerina"
    .AddItem "Aceite"
    .AddItem "Otro"
  End With
  grdResumen.ColAlignment(1) = 2
  grdResumen.ColWidth(0) = 2000
  grdResumen.ColWidth(1) = 1650
  cmbP5Calculo.AddItem "Cálculo Académico (Ideal)"
  cmbP5Calculo.AddItem "Según Base de Datos comercial"
  Call cmdP3Borrar_Click
  pmsg False
End Sub

Private Sub SetStep(nStep As Integer, nDirection As Integer)
 Dim I As Double, PesoAgua As Double, xt As Integer, tmp As Double, diam As
Double
 Dim book As Byte, rng As Variant, cell As Object, jm As String, msg As String
 Dim E1 As Double, E2 As Double, E3 As Double, found As Boolean, cval As
Double, Lt As Double
```

```
 Dim F1 As Double, Fangle As Double, Fz As Double, Fz2 As Double, FzFinal As
Double
  If Not CheckOpen("espesores.dat", "xs") Then xs.Workbooks.Open (App.Path &
"\espesores.dat")
  jm = Chr$(13) & Chr$(10)
  Select Case nStep
    Case STEP_INTRO

    Case STEP_1 'Datos generales...

    Case STEP_2
     If optModo(0).value = 0 And optModo(1).value = 0 Then
      Beep
      MsgBox "Debe seleccionar un modo de trabajo para continuar.", vbCritical +
vbOKOnly, "Error"
      Exit Sub
     End If
    Case STEP_3
     '*************** Verificacion Paso 1
     If nDirection = DIR_BACK Then GoTo continue
     If txtP1Nombre.Text = "" Or txtP1Fecha.Text = "" Or txtP1loc.Text = "" Then
      MsgBox "No ha introducido los datos completos", vbCritical, "Error"
      Exit Sub
     ElseIf Not IsDate(txtP1Fecha.Text) Then
      MsgBox "Debe introducir una fecha válida.", vbCritical + vbOKOnly, "Error"
      Exit Sub
     Else
      Vars.NombreProy = txtP1Nombre.Text
      Vars.LocalidadProy = txtP1loc.Text
      Vars.FechaProy = txtP1Fecha.Text
     End If
     txtP2Temp.Text = ""
     If optSistIngles Then
      Vars.SistemaDec = False
      lblP2Temp2.Caption = "°F"
      lblP2Otro2.Caption = "lb/ft^3"
      lblP2Presion.Caption = "Presión máx. (PSI):"
      lblP2Sobrepresion.Caption = "Sobrepresión (PSI):"
      lblP2cargaviento.Caption = "Carga Trans. (Lb/Ft):"
      lblP2Cargaviva.Caption = "Carga viva (Lb/Ft):"
      lblP3Luz.Caption = "Luz Libre (Ft):"
      lblP3Altura.Caption = "Altura (Ft):"
      Frame5.Caption = "Sección del Cauce (en Ft)"
```

```
        lblP4Sigma2.Caption = "PSI"
        lblP5Diametro2.Caption = "in"
        lblP2Temp3.Caption = "Nota: el rango de temperatura del agua debe estar
entre los 50° F y 160° F"
        lblP6Caudal2.Caption = "ft^3/seg."
        lblP6Soporte2.Caption = "PSI"
        lblP6Peso2.Caption = "lb/ft^3"
        lblP5Espesor2.Caption = "in"
      Else
        Vars.SistemaDec = True
        lblP2Temp2.Caption = "°C"
        lblP2Otro2.Caption = "kg/m^3"
        lblP2Presion.Caption = "Presión máx. (Kg/cm^2):"
        lblP2Sobrepresion.Caption = "Sobrepresión (Kg/cm^2):"
        lblP2cargaviento.Caption = "Carga Trans. (Kg/m):"
        lblP2Cargaviva.Caption = "Carga viva (Kg/m):"
        lblP3Luz.Caption = "Luz Libre (m):"
        lblP3Altura.Caption = "Altura (m):"
        Frame5.Caption = "Sección del Cauce (en m)"
        lblP4Sigma2.Caption = "Kg/cm^2"
        lblP5Diametro2.Caption = "mm"
        lblP2Temp3.Caption = "Nota: el rango de temperatura del agua debe estar
entre los 10° C y 70° C"
        lblP6Caudal2.Caption = "m^3/seg."
        lblP6Soporte2.Caption = "Kg/cm^2"
        lblP6Peso2.Caption = "kg/m^3"
        lblP5Espesor2.Caption = "mm"
      End If

    Case STEP_4
    '************ Verificacion Paso 3
    Select Case cmbP2Fluido.Text
    Case "Agua"
      If Vars.SistemaDec Then
        If Not IsNumeric(Val(txtP2Temp.Text)) Or (Val(txtP2Temp.Text) > 70 Or
Val(txtP2Temp.Text) < 10) Then
          Beep
          MsgBox "Introduzca un valor numérico válido entre 10°C y 70°C para la
temperatura del agua.", vbCritical + vbOKOnly, "Error"
          Exit Sub
        Else
          Vars.Temperatura = Val(txtP2Temp.Text)
        End If
```

```
      Else
        If Not IsNumeric(Val(txtP2Temp.Text)) Or (Val(txtP2Temp.Text) < 50 Or
Val(txtP2Temp.Text) > 160) Then
          Beep
          MsgBox "Introduzca un valor numérico válido entre 50ºF y 160ºF para la
temperatura del agua.", vbCritical + vbOKOnly, "Error"
          Exit Sub
        Else
          Vars.Temperatura = Val(txtP2Temp.Text)
        End If
      End If
      xt = Vars.Temperatura
      If Vars.SistemaDec Then
        Vars.PesoFluido = (-0.0034 * (xt ^ 2)) - (0.0985 * xt) + 1001
      Else
        Vars.PesoFluido = (-0.00006 * (xt ^ 2)) + (0.0007 * xt) + 62.505
      End If
    Case "Aceite"
      If Vars.SistemaDec Then
        Vars.PesoFluido = 860
      Else
        Vars.PesoFluido = PesoEspecifico(860)
      End If
    Case "Alcohol Etílico"
      If Vars.SistemaDec Then
        Vars.PesoFluido = 790
      Else
        Vars.PesoFluido = PesoEspecifico(790)
      End If
    Case "Gasolina"
      If Vars.SistemaDec Then
        Vars.PesoFluido = 674
      Else
        Vars.PesoFluido = PesoEspecifico(674)
      End If
    Case "Kerosene"
      If Vars.SistemaDec Then
        Vars.PesoFluido = 800
      Else
        Vars.PesoFluido = PesoEspecifico(800)
      End If
    Case "Benceno"
      If Vars.SistemaDec Then
```

```
      Vars.PesoFluido = 879.5
    Else
      Vars.PesoFluido = PesoEspecifico(879.5)
    End If
  Case "Glicerina"
    If Vars.SistemaDec Then
      Vars.PesoFluido = 262.3
    Else
      Vars.PesoFluido = PesoEspecifico(262.3)
    End If
  Case "Otro"
    If Not IsNumeric(txtP2Otro.Text) Or Val(txtP2Otro.Text) <= 0 Then
      Beep
      MsgBox "Introduzca un valor numérico positivo para el Peso Específico del
fluido.", vbCritical + vbOKOnly, "Error"
      Exit Sub
    End If
    Vars.PesoFluido = txtP2Otro.Text
  Case Else
    Beep
    MsgBox "Debe seleccionar un tipo de fluido para continuar.", vbCritical +
vbOKOnly, "Error"
    Exit Sub
  End Select
  If Not IsNumeric(Vars.PesoFluido) Then
    Beep
    MsgBox "Peso Específico del Fluido inválido. Por favor seleccione una opción
o ingrese un valor numérico válido.", vbCritical + vbOKOnly, "Error"
    Exit Sub
  End If
  tmp = Val(txtP2Presion.Text)
  If Vars.SistemaDec Then
    If tmp < 0 Or tmp > 70 Then
      Beep
      MsgBox "Presión inválida. El rango de presión establecido está entre 0 y 70
Kg/cm^2.", vbCritical + vbOKOnly, "Error"
      Exit Sub
    End If
    tmp = Val(txtP2Sobrepresion.Text)
    If tmp < 0 Or tmp > 120 Then
      Beep
      MsgBox "Sobrepresión inválida. El rango de sobrepresión establecido está
entre 0 y 120 Kg/cm^2.", vbCritical + vbOKOnly, "Error"
```

```
        Exit Sub
      End If
      If tmp > 0 Then txtP4K2.Enabled = True Else txtP4K2.Enabled = False:
txtP4K2.Text = "0"
      tmp = Val(txtP2cargaviva.Text)
      If tmp > 500 Or tmp < 0 Then
        Beep
        MsgBox "Carga Viva inválida. El valor debe estar entre 0 y 500 Kg/m",
vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      tmp = Val(txtP2cargaviento.Text)
      If tmp > 336 Or tmp < 0 Then
        Beep
        MsgBox "Carga Transversal inválida. El valor debe estar entre 0 y 300
Kg/m", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
    Else
      If tmp < 0 Or tmp > 1000 Then
        Beep
        MsgBox "Presión inválida. El rango de presión establecido está entre 0 y 1000
Lb/in^2.", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      tmp = Val(txtP2Sobrepresion.Text)
      If tmp < 0 Or tmp > 1707 Then
        Beep
        MsgBox "Sobrepresión inválida. El rango de sobrepresión establecido está
entre 0 y 1707 Lb/in^2.", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      tmp = Val(txtP2cargaviva.Text)
      If tmp > 336 Or tmp < 0 Then
        Beep
        MsgBox "Carga Viva inválida. El valor debe estar entre 0 y 336 Lb/Ft",
vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      tmp = Val(txtP2cargaviento.Text)
      If tmp > 202 Or tmp < 0 Then
        Beep
```

```
        MsgBox "Carga Transversal inválida. El valor debe estar entre 0 y 202
Lb/Ft", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
    End If
    If txtP2cargaviento.Text = "" Then txtP2cargaviento.Text = 0
    If txtP2cargaviva.Text = "" Then txtP2cargaviva.Text = 0
    If txtP2Presion.Text = "" Then txtP2Presion.Text = 0
    If txtP2Sobrepresion.Text = "" Then txtP2Sobrepresion.Text = 0
    If txtP2Sobrepresion.Text = 0 Then
      txtP4K2.Enabled = False
      txtP4K2.Text = "0" & Chr$(Vars.Separator) & "75"
    Else
      txtP4K2.Enabled = True
    End If
    If txtP2Presion.Text = 0 Then
      txtP4K1.Enabled = False
      txtP4K1.Text = "0" & Chr$(Vars.Separator) & "60"
    Else
      txtP4K1.Enabled = True
    End If

    Case STEP_5
    '******************* Verificacion Paso 3

      If Vars.Modo = 0 Then
        If txtP3Luz.Text = "" Then MsgBox "Debe introducir los datos topográficos y
hacer click en el botón 'Graficar' para continuar.", vbCritical + vbOKOnly, "Error":
Exit Sub
      Else
        If Val(txtP3Luz.Text) = 0 Or Val(txtP3Altura.Text) = 0 Then MsgBox "Los
valores de Luz y de Altura deben ser distintos a cero.", vbCritical + vbOKOnly,
"Error": Exit Sub
      End If

    Case STEP_6
    '************ Verificacion Paso 4

    If ((optP4Acero(0).Tag = 0 Or optP4Acero(0).Tag = 3) And
cmbP4Grado.ListIndex = -1) Then
      Beep
      MsgBox "Por favor seleccione un Tipo de Acero para continuar.", vbCritical +
vbOKOnly, "Error"
```

```
    Exit Sub
  End If
  If optP4Acero(4).value And Val(txtP4Sigma.Text) <= 0 Then
    Beep
    MsgBox "Debe escribir un valor positivo de Esfuerzo de Fluencia para el
Acero.", vbCritical + vbOKOnly, "Error"
    Exit Sub
  End If
  If txtP4K1.Text = "" Then txtP4K1.Text = 0
  tmp = CDbl(txtP4K1.Text)
  If tmp < 0.4 Or tmp > 0.6 Then
    Beep
    MsgBox "Por favor introduzca valores numéricos válidos para K1.1 entre 0,4 y
0,6", vbCritical + vbOKOnly, "Error"
    Exit Sub
  End If
  If txtP4K2.Enabled Then
    If txtP4K2.Text = "" Then txtP4K2.Text = 0
    tmp = CDbl(txtP4K2.Text)
    If tmp < 0.4 Or tmp > 0.75 Then
      Beep
      MsgBox "Por favor introduzca valores numéricos válidos para K1.2 entre 0,4
y 0,75", vbCritical + vbOKOnly, "Error"
      Exit Sub
    End If
  End If
  If txtP4Pandeo.Enabled Then
    If txtP4Pandeo.Text = "" Then txtP4Pandeo.Text = 0
    tmp = CDbl(txtP4Pandeo.Text)
    If tmp < 0.5 Or tmp > 1.5 Then
      Beep
      MsgBox "Por favor introduzca valores numéricos válidos para K Pandeo entre
0,5 y 1,5", vbCritical + vbOKOnly, "Error"
      Exit Sub
    End If
  End If
  If optP4Acero(0).Tag <> 4 And Not nDirection = DIR_BACK Then
  Select Case optP4Acero(0).Tag
  Case 0 'api5l
    Select Case cmbP4Grado.ListIndex
    Case 0 'A
      book = 1
    Case 1, 2 'B, X42
```

```
        book = 2
      Case 3 'X52
        book = 3
      Case 4 'X60
        book = 4
      End Select
    Case 1, 2 'astma36 astma42
      book = 2
    Case 3 'astma53
      book = 1
    End Select
    If book <> Vars.book Then
      cmbP5Diametro.Clear
      For Each cell In xs.Workbooks("espesores.dat").Worksheets(book).Range("A1:A30")
        If cell.value = "" Then Exit For
        If Vars.SistemaDec Then cmbP5Diametro.AddItem Format((cell.value / 0.0394), "#### ###") Else cmbP5Diametro.AddItem cell.value
        'If Vars.SistemaDec Then cmbP5Diametro.AddItem Format((cell.value / 0.0394), "#") Else cmbP5Diametro.AddItem cell.value
      Next cell
      Vars.book = book
    End If
    End If

    Case STEP_7
      If (Not IsNumeric(txtP5Tramos.Text)) Or (Val(txtP5Tramos.Text) < 2 Or Val(txtP5Tramos.Text) > 50) Then
        Beep
        MsgBox "El número de tramos es inválido. Por Favor especifique un número entre 2 y 50 o seleccione un diámetro de tubería para calcularlo automáticamente.", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      If cmbP5Calculo.ListIndex = 1 And cmbP5Diametro.ListIndex = -1 Then
        Beep
        MsgBox "Seleccione un Diametro para continuar.", vbCritical + vbOKOnly, "Error"
        Exit Sub
      ElseIf cmbP5Calculo.ListIndex = 0 And Not IsNumeric(txtP5Diametro.Text) Then
        Beep
```

```
        MsgBox "Escriba un Diametro para continuar.", vbCritical + vbOKOnly,
"Error"
        Exit Sub
      End If
      Call cmdp5calc_Click
      If nDirection = DIR_NEXT And txtP5Tramos.Text <> txtP5Tramos.Tag And
txtP5Tramos.Tag <> "" Then
        Select Case MsgBox("El número actual de tramos es " & txtP5Tramos.Text &
" " & Chr(10) & "DAPT ha calculado que el número de tramos ideal es " &
txtP5Tramos.Tag & " " & Chr(10) & Chr(10) & "¿Desea mantener el número
actual?", vbExclamation + vbYesNo, "DAPT")
        Case vbNo
          txtP5Tramos.Text = txtP5Tramos.Tag
        Case vbCancel
          Exit Sub
        End Select
      End If


    Case STEP_8
      If Vars.Modo = 0 Then
        If CDbl(txtP6Caudal.Text) <= 0 Or CDbl(txtP6Friccion.Text) <= 0 Or
CDbl(txtP6Peso.Text) <= 0 Or CDbl(txtP6Soporte.Text) <= 0 Then
          Beep
          MsgBox "Los valores para el cálculo de los Anclajes de la Tubería no
pueden ser menores a 0.", vbCritical + vbOKOnly, "Error"
          Exit Sub
        End If
        If CDbl(txtP6Friccion.Text) < 0.3 Or CDbl(txtP6Friccion.Text) > 0.5 Then
          Beep
          MsgBox "La Fricción debe estar entre 0.3 y 0.5", vbCritical + vbOKOnly,
"Error"
          Exit Sub
        End If
      ElseIf (Vars.Modo = 1 And cmbP5Calculo.ListIndex = 0) And
(txtP5Espesor.Text = "" Or txtP5Espesor.Text = "0") Then
        Beep
        MsgBox "Debe introducir un Espesor para continuar.", vbCritical +
vbOKOnly, "Error"
        Exit Sub
      ElseIf Vars.Modo = 1 Then
        If (Not IsNumeric(txtP5Tramos.Text)) Or (Val(txtP5Tramos.Text) < 2 Or
Val(txtP5Tramos.Text) > 50) Then
```

```
        Beep
        MsgBox "El número de tramos es inválido. Por Favor especifique un número
entre 2 y 50 o seleccione un diámetro de tubería para calcularlo automáticamente.",
vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      If cmbP5Calculo.ListIndex = 0 And Not IsNumeric(txtP5Diametro.Text)
Then
        Beep
        MsgBox "Escriba un Diametro para continuar.", vbCritical + vbOKOnly,
"Error"
        Exit Sub
      End If
    End If
    msg = "Nombre del Proyecto: " & Vars.NombreProy & jm & "Localidad: " &
Vars.LocalidadProy & jm & "Fecha: " & Vars.FechaProy & jm & jm
    msg = msg & "Tipo de Fluido: " & Vars.TipoFluido
    If Vars.TipoFluido = "Agua" Then msg = msg & ", " & Vars.Temperatura & "
" & lblP2Temp2.Caption
    msg = msg & jm & "Peso Especifico: " & Vars.PesoFluido & " " &
Me.lblP2Otro2.Caption & jm & jm
    msg = msg & lblP2Presion.Caption & " " & txtP2Presion.Text & jm &
lblP2Sobrepresion.Caption & " " & txtP2Sobrepresion.Text & jm
    msg = msg & lblP2Cargaviva.Caption & " " & txtP2cargaviva.Text & jm &
lblP2cargaviento.Caption & " " & txtP2cargaviento.Text & jm & jm
    If Vars.SistemaDec Then tmpx = "m" Else tmpx = "Ft"
    msg = msg & lblP3InitX.Caption & " " & txtP3InitX.Text & " " & tmpx & jm
& lblP3InitY.Caption & " " & txtP3InitY.Text & " " & tmpx & jm
    msg = msg & lblP3Luz.Caption & " " & Format(txtP3Luz.Text, "0.0###") & "
" & tmpx & jm & lblP3Altura.Caption & " " & Format(txtP3Altura.Text, "0.0###") &
" " & tmpx & jm & jm
    msg = msg & "Tipo de Acero: " & Vars.TipoAcero & jm
    If optP4Acero(0).Tag = 0 Or optP4Acero(0).Tag = 3 Then msg = msg &
"Grado: " & Vars.GradoAcero & jm
    msg = msg & "Esfuerzo de Fluencia: " & Format(Vars.Sigma, "0.0###") & " "
& Me.lblP4Sigma2.Caption & jm
    msg = msg & lblP4K1.Caption & " " & txtP4K1.Text & jm & lblP4K2.Caption
& " " & txtP4K2.Text & jm & "Valor de K (Pandeo): " & Me.txtP4Pandeo.Text & jm
& jm
    msg = msg & "Tipo de Cálculo: " & cmbP5Calculo.Text & jm & jm &
"Diametro: "
    msg = msg & Vars.Diametro & " " & lblP5Diametro2.Caption & jm
```

```
    If txtP5Espesor.Enabled = True Then msg = msg & "Espesor: " & txtP5Espesor.Text & " " & lblP5Espesor2.Caption
    msg = msg & jm & "Tramos: " & txtP5Tramos.Text & jm & jm
    msg = msg & "Caudal de Trabajo: " & txtP6Caudal.Text & " " & lblP6Caudal2.Caption & jm & "Coeficiente de Fricción: " & txtP6Friccion.Text & jm
    msg = msg & "Peso Unitario Concreto: " & txtP6Peso.Text & " " & lblP6Peso2.Caption & jm & "Capacidad de Soporte del Suelo: " & txtP6Soporte.Text & " " & lblP6Soporte2.Caption
    txtResumen.Text = msg

  Case STEP_9
    pmsg True, "Calculando...", 25, "Análisis Hidráulico"
    If Vars.SistemaDec Then diam = Vars.Diametro / 10 Else diam = Vars.Diametro
    E1 = (diam * CDbl(txtP2Presion.Text)) / (2 * (CDbl(txtP4K1.Text) * Vars.Sigma))
    If Not CDbl(txtP4K2.Text) = 0 Then E2 = (diam * CDbl(txtP2Sobrepresion.Text)) / (2 * (CDbl(txtP4K2.Text) * Vars.Sigma)) Else E2 = 0
    E3 = 0.007 * diam
    If cmbP5Calculo.ListIndex = 1 Then
     found = False

     For Each cell In xs.Workbooks("espesores.dat").Worksheets(Vars.book).Range("B" & cmbP5Diametro.ListIndex + 1 & ":K" & cmbP5Diametro.ListIndex + 1)
      If cell.value = "" Then Exit For
      If Vars.SistemaDec Then cval = cell.value * 2.54 Else cval = cell.value
      If cval > E1 And cval > E2 And cval > E3 Then
       found = True
       Vars.espesor = cval
       If Vars.SistemaDec Then txtP5Espesor.Text = Vars.espesor * 10 Else txtP5Espesor.Text = Vars.espesor
       Exit For
      End If
     Next cell
     If Not found Then
      pmsg False
      Beep
      MsgBox "El diámetro especificado no posee un espesor comercial capaz de cumplir todos los requerimientos Hidráulicos. Por favor seleccione otro diámetro.", vbCritical + vbOKOnly, "Error"
      SetStep 7, DIR_BACK
```

```
        Exit Sub
      End If
    Else
      If E1 > E2 And E1 > E3 Then
        Vars.espesor = E1
      ElseIf E2 > E1 And E2 > E3 Then
        Vars.espesor = E2
      Else
        Vars.espesor = E3
      End If
      If Vars.Modo = 1 Then
        If Vars.SistemaDec Then
          If Vars.espesor > (CSng(txtP5Espesor.Text) / 10) Then
            Beep
            pmsg False
            MsgBox "El Espesor introducido no cumple con las especificaciones
establecidas.", vbCritical + vbOKOnly, "Error"
            Exit Sub
          Else
            Vars.espesor = (CSng(txtP5Espesor.Text) / 10)
          End If
        Else
          If Vars.espesor > (CSng(txtP5Espesor.Text)) Then
            Beep
            pmsg False
            MsgBox "El Espesor introducido no cumple con las especificaciones
establecidas.", vbCritical + vbOKOnly, "Error"
            Exit Sub
          Else
            Vars.espesor = (CSng(txtP5Espesor.Text))
          End If
        End If
      Else
        If Vars.SistemaDec Then txtP5Espesor.Text = Vars.espesor * 10 Else
txtP5Espesor.Text = Vars.espesor
      End If
    End If
    grdP3Secciones.Tag = False
    Call msgWnd.Luciano
    If Vars.SistemaDec Then g1 = 9.81 Else g1 = 32.1846
    If cmbP5Calculo.ListIndex = 1 Then
      If Vars.Verificacion Then
        imgHid.Picture = LoadPicture(App.Path & "\checkmrk.ico")
```

```
        imgEst.Picture = LoadPicture(App.Path & "\checkmrk.ico")
        If Vars.Modo = 0 Then
         F1 = (4 * Vars.PesoFluido * CDbl(txtP6Caudal.Text) ^ 2) / (3.1416 *
Vars.Diametro ^ 2 * g1)
         Fangle = (Vars.QQ * (Val(txtP5Tramos.Text - 1))) / (2 *
Val(txtP5Tramos.Text))
         Fz = -(Vars.Fza2 * Sin(Fangle) - ((Vars.Fza + F1) * Cos(Fangle)))
         Fz2 = -(Vars.Fza2 * Cos(Fangle) - ((Vars.Fza + F1) * Sin(Fangle)))
         If Fz > Fz2 And Fz > Vars.Fza3 Then
          FzFinal = Fz
         ElseIf Fz2 > Fz And Fz2 > Vars.Fza3 Then
          FzFinal = Fz2
         Else
          FzFinal = Vars.Fza3
         End If
         If Sgn(FzFinal) = -1 Then FzFinal = FzFinal * -1
         Do
          If Vars.SistemaDec Then lancl = lancl + 0.1 Else lancl = lancl + 0.3281
          If Vars.SistemaDec Then tmp = CDbl(txtP6Soporte.Text) * 1000 Else tmp
= CDbl(txtP6Soporte.Text) * 144
          If FzFinal < (CDbl(txtP6Friccion.Text) * CDbl(txtP6Peso.Text) * lancl ^
3 + lancl ^ 2 * tmp) Then Exit Do
         Loop While True
         cmdAutoCAD.Enabled = True
         cmdNumerico.Enabled = True
        Else
         cmdAutoCAD.Enabled = False
         cmdNumerico.Enabled = True
        End If
       Else
        imgHid.Picture = LoadPicture(App.Path & "\checkmrk.ico")
        imgEst.Picture = LoadPicture(App.Path & "\w95mbx01.ico")
        cmdAutoCAD.Enabled = False
        cmdNumerico.Enabled = False
       End If
      Else
       Do
        If Vars.Verificacion Then
         imgHid.Picture = LoadPicture(App.Path & "\checkmrk.ico")
         imgEst.Picture = LoadPicture(App.Path & "\checkmrk.ico")
         If Vars.Modo = 0 Then
          If Vars.SistemaDec Then tmp = Vars.Diametro / 1000 Else tmp =
Vars.Diametro / 12
```

```
            F1 = (4 * Vars.PesoFluido * CDbl(txtP6Caudal.Text) ^ 2) / (3.1416 * tmp
^ 2 * CDbl(txtP6Peso.Text))
            Fangle = (Vars.QQ * (Val(txtP5Tramos.Text - 1))) / (2 *
Val(txtP5Tramos.Text))
            Fz = -(Vars.Fza2 * Sin(Fangle) - ((Vars.Fza + F1) * Cos(Fangle)))
            Fz2 = -(Vars.Fza2 * Cos(Fangle) - ((Vars.Fza + F1) * Sin(Fangle)))
            If Fz > Fz2 And Fz > Vars.Fza3 Then
              FzFinal = Fz
            ElseIf Fz2 > Fz And Fz2 > Vars.Fza3 Then
              FzFinal = Fz2
            Else
              FzFinal = Vars.Fza3
            End If
            If Sgn(FzFinal) = -1 Then FzFinal = FzFinal * -1
            If Vars.SistemaDec Then tmp = CDbl(txtP6Soporte.Text) * 1000 Else tmp
= CDbl(txtP6Soporte.Text) * 144
            Do
              If Vars.SistemaDec Then lancl = lancl + 0.1 Else lancl = lancl + 0.3281
              If FzFinal < (CDbl(txtP6Friccion.Text) * CDbl(txtP6Peso.Text) * lancl ^
3 + lancl ^ 2 * tmp) Then Exit Do
            Loop While True
            cmdAutoCAD.Enabled = True
            cmdNumerico.Enabled = True
          Else
            cmdAutoCAD.Enabled = False
            cmdNumerico.Enabled = True
          End If
          Exit Do
        Else
          If Vars.Modo = 0 Then
            If Vars.SistemaDec Then
              Vars.espesor = Vars.espesor + 0.01
              If Vars.Diametro > (20 * Vars.espesor) Then
                txtP5Espesor.Text = Vars.espesor * 10
                Call msgWnd.Luciano
              Else
                Beep
                MsgBox "No se encontró ningún Espesor asociado a este Diámetro que
cumpla los requerimientos Estructurales.", vbCritical + vbOKOnly, "Error"
                Exit Sub
              End If
            Else
              Vars.espesor = Vars.espesor + 0.0033
```

```
            If Vars.Diametro > (2 * Vars.espesor) Then
              txtP5Espesor.Text = Vars.espesor
              Call msgWnd.Luciano
            Else
              Beep
              MsgBox "No se encontró ningún Espesor asociado a este Diámetro que
cumpla los requerimientos Estructurales.", vbCritical + vbOKOnly, "Error"
              Exit Sub
            End If
          End If
        Else
          Beep
          pmsg False
          MsgBox "El Espesor introducido no cumple con las especificaciones
establecidas.", vbCritical + vbOKOnly, "Error"
          Exit Sub
        End If
      End If
    Loop While True
  End If
  With grdResumen
    .TextMatrix(0, 0) = "Nombre del Proyecto:"
    .TextMatrix(0, 1) = Vars.NombreProy
    .TextMatrix(1, 0) = "Tipo de Acero:"
    .TextMatrix(1, 1) = Vars.TipoAcero
    .TextMatrix(2, 0) = "Grado:"
    .TextMatrix(2, 1) = Vars.GradoAcero
    .TextMatrix(3, 0) = "Diámetro:"
    .TextMatrix(3, 1) = Format(Vars.Diametro, "0.0##") & " " &
lblP5Diametro2.Caption
    .TextMatrix(4, 0) = "Espesor:"
    .TextMatrix(4, 1) = Format(txtP5Espesor.Text, "0.0##") & " " &
lblP5Espesor2.Caption
    .TextMatrix(5, 0) = "Tramos:"
    .TextMatrix(5, 1) = txtP5Tramos.Text
    .TextMatrix(6, 0) = "Longitud de cada Tramo:"
    If Vars.SistemaDec Then tmpx = "m" Else tmpx = "Ft"
    .TextMatrix(6, 1) = Format(Vars.Lt, "0.0#") & " " & tmpx
    .TextMatrix(7, 0) = "Angulo de Corte:"
    .TextMatrix(7, 1) = Format(Vars.AnguloCorte, "0.0##") & "°"
    .TextMatrix(8, 0) = "Ratio %:"
    .TextMatrix(8, 1) = Format(Vars.Porcentaje, "0.0##")
    .TextMatrix(9, 0) = "Dimensión del Anclaje:"
```

```
        .TextMatrix(9, 1) = Format(lancl, "0.0#") & " " & tmpx
      End With
      pmsg False
      mbFinishOK = False

    Case STEP_FINISH
      If cmdNumerico.Enabled = False Then
        Beep
        MsgBox "Debe aprobar el Análisis Hidráulico y el Análisis Estructural para
continuar.", vbCritical + vbOKOnly, "Error"
        Exit Sub
      End If
      mbFinishOK = True

  End Select
continue:
  'pasar al siguiente paso
  fraStep(mnCurStep).Enabled = False
  fraStep(nStep).Left = 0
  If nStep <> mnCurStep Then
    fraStep(mnCurStep).Left = -10000
  End If
  fraStep(nStep).Enabled = True

  SetCaption nStep
  SetNavBtns nStep

End Sub

Private Sub SetNavBtns(nStep As Integer)
  mnCurStep = nStep

  If mnCurStep = 0 Then
    cmdNav(BTN_BACK).Enabled = False
    cmdNav(BTN_NEXT).Enabled = True
  ElseIf mnCurStep = NUM_STEPS - 1 Then
    cmdNav(BTN_NEXT).Enabled = False
    cmdNav(BTN_BACK).Enabled = True
  Else
    cmdNav(BTN_BACK).Enabled = True
    cmdNav(BTN_NEXT).Enabled = True
  End If
  'If mnCurStep = 2 Then cmdNav(BTN_NEXT).Enabled = False
```

```
   If mbFinishOK Then
      cmdNav(BTN_FINISH).Enabled = True
      cmdNav(BTN_NEXT).Enabled = False
   Else
      cmdNav(BTN_FINISH).Enabled = False
      cmdNav(BTN_NEXT).Enabled = True
   End If
   Select Case mnCurStep
   Case 1, 2, 3, 4, 5, 6, 7
     cmdNav(BTN_HELP).Enabled = True
   Case Else
     cmdNav(BTN_HELP).Enabled = False
   End Select
End Sub

Private Sub SetCaption(nStep As Integer)
   On Error Resume Next

   Me.Caption = FRM_TITLE & " - " & LoadResString(fraStep(nStep).Tag)
   If nStep = 1 Then Me.Caption = FRM_TITLE & " - " & "Inicio"
   If nStep = 10 Then Me.Caption = FRM_TITLE & " - " & "Finalizado"
End Sub

'=================================================================
'esta subrutina muestra un mensaje de error cuando el
'usuario no ha escrito suficientes datos para continuar
'=================================================================
Sub IncompleteData(nIndex As Integer)
   On Error Resume Next
   Dim sTmp As String

   'obtener el mensaje de error de base
   sTmp = LoadResString(RES_ERROR_MSG)
   'obtener el mensaje específico
   sTmp = sTmp & vbCrLf & LoadResString(RES_ERROR_MSG + nIndex)
   Beep
   MsgBox sTmp, vbInformation
End Sub

Private Sub Form_Unload(Cancel As Integer)
   On Error Resume Next
   Dim rc As Long
   'ver si hay que guardar la configuración
```

```
  If chkSaveSettings.value = vbChecked Then

        SaveSetting APP_CATEGORY, WIZARD_NAME, "OptionName", Option
Value

  End If
  On Error Resume Next
  xs.DisplayAlerts = False
  xs.Quit
  wd.DisplayAlerts = wdAlertsNone
  xs.Quit
  acadx.Quit
  Set xs = Nothing
  Set wd = Nothing
  Set acadx = Nothing
  End
  If mbHelpStarted Then rc = WinHelp(Me.hwnd, HELP_FILE, HELP_QUIT, 0)
End Sub

Private Sub grdP3Secciones_DblClick()
 txtP3grd.Visible = False
 If grdP3Secciones.Row > 0 And grdP3Secciones.Col > 0 And grdP3Secciones.Col
< 3 Then
  With txtP3grd
   nRow = grdP3Secciones.Row
   nCol = grdP3Secciones.Col
   .Text = grdP3Secciones.TextMatrix(nRow, nCol)
   .Left = grdP3Secciones.CellLeft + grdP3Secciones.Left
   .Height = grdP3Secciones.CellHeight
   .Top = grdP3Secciones.CellTop + grdP3Secciones.Top
   .Width = grdP3Secciones.CellWidth
   .Visible = True
   .SetFocus
  End With
 ElseIf grdP3Secciones.Row > 0 And grdP3Secciones.Col = 3 Then
  With grdP3Secciones
   .TextMatrix(Vars.Inicial, 3) = ""
   nRow = .Row
   nCol = .Col
   .TextMatrix(nRow, nCol) = "*"
   Vars.Inicial = nRow
   txtP3InitX.Text = .TextMatrix(nRow, 1)
   txtP3InitY.Text = .TextMatrix(nRow, 2)
```

```
    txtP3Luz.Text = ""
    txtP3Altura.Text = ""
   End With
  End If
End Sub

Private Sub grdP3Secciones_Scroll()
If grdP3Secciones.Col <> 3 Then Call txtP3grd_KeyPress(13)
End Sub

Private Sub optModo_Click(Index As Integer)
Select Case Index
Case 0
  Frame5.Enabled = True
  Label1.Caption = "El * Indica el punto inicial de la tubería. Haga Dobleclick sobre
la fila deseada para cambiar."
  Label1.Enabled = True
  grdP3Secciones.Enabled = True
  cmdP3Graph.Enabled = True
  cmdP3Secciones.Enabled = True
  cmdP3Borrar.Enabled = True
  lblP3Secciones.Enabled = True
  txtP3Luz.Enabled = False
  txtP3Altura.Enabled = False
  txtP3InitX.Enabled = False
  txtP3InitY.Enabled = False
  txtP5Espesor.Enabled = False
  lblP5Espesor.Enabled = False
  lblP5Espesor2.Enabled = False
  cmdAutoCAD.Enabled = True
  Frame11.Caption = "Diámetro de la Tubería"
  cmbP5Calculo.Enabled = True
Case 1
  Frame5.Enabled = False
  Label1.Caption = "No se aplica en Modo Verificar. Introduzca los valores de Inicio,
Luz y Altura"
  Label1.Enabled = False
  grdP3Secciones.Enabled = False
  cmdP3Graph.Enabled = False
  cmdP3Secciones.Enabled = False
  cmdP3Borrar.Enabled = False
  lblP3Secciones.Enabled = False
  txtP3Luz.Enabled = True
```

```
  txtP3Altura.Enabled = True
  txtP3InitX.Enabled = True
  txtP3InitY.Enabled = True
  txtP5Espesor.Enabled = True
  lblP5Espesor.Enabled = True
  lblP5Espesor2.Enabled = True
  cmdAutoCAD.Enabled = False
  txtP3Luz.Text = 0
  txtP3Altura.Text = 0
  txtP3InitX.Text = 0
  txtP3InitY = 0
  cmbP5Calculo.ListIndex = 0
  cmbP5Calculo.Enabled = False
  Frame11.Caption = "Diámetro y Espesor de la Tubería"
End Select
Call cmdP3Borrar_Click
Vars.Modo = Index
End Sub

Private Sub cmbP2Fluido_Click()
  lblP2Temp.Visible = False
  lblP2Temp2.Visible = False
  txtP2Temp.Visible = False
  lblP2Otro.Visible = False
  lblP2Otro2.Visible = False
  txtP2Otro.Visible = False
  lblP2Temp3.Visible = False
  Select Case cmbP2Fluido.Text
  Case "Agua"
    lblP2Temp.Visible = True
    lblP2Temp2.Visible = True
    lblP2Temp3.Visible = True
    txtP2Temp.Visible = True
    txtP2Temp.SetFocus
  Case "Otro"
    lblP2Otro.Visible = True
    lblP2Otro2.Visible = True
    txtP2Otro.Visible = True
    txtP2Otro.SetFocus
  End Select
  Vars.TipoFluido = cmbP2Fluido.Text
End Sub
```

```
Private Sub optP4Acero_Click(Index As Integer)
  Screen.MousePointer = 13
  lblP4Grado.Visible = False
  cmbP4Grado.Visible = False
  lblP4Sigma.Visible = False
  lblP4Sigma2.Visible = False
  txtP4Sigma.Visible = False
  lblP4Grado.Top = optP4Acero(Index).Top
  optP4Acero(0).Tag = Index
  With cmbP4Grado
    .Top = optP4Acero(Index).Top
    .Clear
    If Vars.Modo = 0 Then
      cmbP5Calculo.Enabled = True
      cmbP5Calculo.ListIndex = 1
    End If
    Select Case Index
    Case 0
      .AddItem "A"
      .AddItem "B"
      .AddItem "X42"
      .AddItem "X52"
      .AddItem "X60"
    Case 1
      If Vars.SistemaDec Then Vars.Sigma = (36000 / 14.2233) Else Vars.Sigma =
36000
      Vars.GradoAcero = ""
    Case 2
      If Vars.SistemaDec Then Vars.Sigma = (42000 / 14.2233) Else Vars.Sigma =
42000
      Vars.GradoAcero = ""
    Case 3
      .AddItem "A"
      .AddItem "B"
    Case 4
      lblP4Sigma.Visible = True
      lblP4Sigma2.Visible = True
      txtP4Sigma.Visible = True
      cmbP5Calculo.ListIndex = 0
      cmbP5Calculo.Enabled = False
      Vars.GradoAcero = ""
    End Select
    If Index = 0 Or Index = 3 Then
```

```
    .ListIndex = 0
    .Visible = True
    lblP4Grado.Visible = True
   End If
  End With
  Vars.TipoAcero = optP4Acero(Index).Caption
  Vars.GradoAcero = cmbP4Grado.Text
  Call cmbP5Calculo_Click
  Screen.MousePointer = 0
End Sub

Private Sub grdP3Secciones_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
  Case 13
   Call grdP3Secciones_DblClick
   txtP3grd.SelStart = 0
   txtP3grd.SelLength = Len(txtP3grd.Text)
  Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57
   Call grdP3Secciones_DblClick
   txtP3grd.Text = Chr$(KeyAscii)
   txtP3grd.SelStart = Len(txtP3grd.Text)
  Case Else
   If grdP3Secciones.Col = 3 Then Call grdP3Secciones_DblClick
  End Select
End Sub

Private Sub txtP2cargaviento_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP2cargaviva_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP2Otro_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP2Presion_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP2Sobrepresion_KeyPress(KeyAscii As Integer)
```

```
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP2Temp_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii) Then KeyAscii = 0
End Sub

Private Sub txtP3Altura_Change()
'If IsNumeric(txtP3Altura.Text) Then Vars.Altura = CDbl(txtP3Altura.Text) Else
Vars.Altura = 0
End Sub

Private Sub txtP3Altura_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP3Altura_LostFocus()
If Vars.Modo = 0 Then
 If CDbl(txtP3Altura.Text) <= 0 Or CDbl(txtP3Altura.Text) >=
CDbl(txtP3Luz.Text) / 2 Then
  Beep
  MsgBox "La Altura debe ser mayor a 0 y menor que " & CSng(txtP3Luz.Text) / 2,
vbCritical + vbOKOnly, "Error"
  txtP3Altura.Text = txtP3Altura.Tag
  txtP3Altura.SetFocus
  Exit Sub
 End If
End If
Vars.Altura = CSng(txtP3Altura.Text)
End Sub

Private Sub txtP3grd_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
Case 13
 If CheckGrdText Then
  grdP3Secciones.TextMatrix(nRow, nCol) = txtP3grd.Text
  txtP3grd.Visible = False
  txtP3Luz.Text = ""
  txtP3Altura.Text = ""
 Else
  MsgBox "Datos invalidos. Introduzca datos numércios válidos.", vbCritical +
vbOKOnly, "Error"
 End If
```

```
Case 27
  txtP3grd.Visible = False
  txtP3grd.Text = grdP3Secciones.TextMatrix(nRow, nCol)
Case Vars.Separator, 8, 45, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57 '. 47=/
  If KeyAscii = 45 And grdP3Secciones.Col = 1 Then KeyAscii = 0
Case Else
  KeyAscii = 0
End Select
End Sub

Private Sub txtP3grd_LostFocus()
txtP3grd.Visible = False
If Not CheckGrdText Then Exit Sub
grdP3Secciones.TextMatrix(nRow, nCol) = txtP3grd.Text
If Vars.Inicial = nRow Then
  If nCol = 1 Then txtP3InitX.Text = txtP3grd.Text Else txtP3InitY.Text =
txtP3grd.Text
End If
grdP3Secciones.SetFocus
End Sub

Private Sub txtP3InitX_Change()
grdP3Secciones.TextMatrix(Vars.Inicial, 1) = txtP3InitX.Text
End Sub

Private Sub txtP3InitX_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii) Or KeyAscii = 45 Then KeyAscii = 0
End Sub

Private Sub txtP3InitY_Change()
grdP3Secciones.TextMatrix(Vars.Inicial, 2) = txtP3InitY.Text
End Sub

Private Sub txtP3InitY_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii) Then KeyAscii = 0
End Sub

Private Sub txtP3InitY_LostFocus()
If Not IsNumeric(txtP3InitY.Text) Then
  Beep
  MsgBox "Debe introducir un valor numérico válido para Y Inicial.", vbCritical +
vbOKOnly, "Error"
  txtP3InitY.SetFocus
```

```
  Exit Sub
End If
End Sub

Private Sub txtP3Luz_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii) Then KeyAscii = 0
End Sub

Private Sub txtP3Secciones_Change()
  If mnCurStep = 3 Then cmdP3Secciones.Default = True
End Sub

Public Sub InitGrid()
  Screen.MousePointer = 13
  With grdP3Secciones
    .Clear
    .ColWidth(0) = 330
    .ColWidth(1) = 850
    .ColWidth(2) = 850
    .ColWidth(3) = 650
    .TextMatrix(0, 0) = "Nº"
    .TextMatrix(0, 1) = "Valor X"
    .TextMatrix(0, 2) = "Valor Y"
    .TextMatrix(0, 3) = "Inicial"
    .ColAlignment(0) = 4
    .ColAlignment(1) = 4
    .ColAlignment(2) = 4
    .ColAlignment(3) = 4
    Vars.Inicial = 1
    .TextMatrix(1, 3) = "*"
  End With
  Screen.MousePointer = 0
End Sub

Public Sub EnumGrid()
  Dim I As Byte
  With grdP3Secciones
    For I = 1 To .Rows - 1
      .TextMatrix(I, 0) = I
    Next I
  End With
End Sub
```

```
'Private Sub txtP3grd_KeyDown(KeyCode As Integer, Shift As Integer)
'MsgBox KeyCode
'End Sub

Private Sub txtP3Secciones_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii) Then KeyAscii = 0
End Sub

Private Sub txtP3Secciones_LostFocus()
cmdP3Secciones.Default = False
End Sub

Public Function CheckGrdText() As Boolean
If Not IsNumeric(Val(txtP3grd.Text)) Then
  Beep
  MsgBox "Debe introducir un valor numérico válido para las coordenadas",
vbCritical + vbOKOnly, "Error"
  CheckGrdText = False
Else
  CheckGrdText = True
End If
End Function

Private Sub txtP4K1_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP4K2_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Public Function NumCheck(kascii As Integer, Optional UseDecimal As Boolean =
False) As Boolean
If UseDecimal = True And kascii = Vars.Separator Then NumCheck = True: Exit
Function
Select Case kascii
Case 8, 45, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57 '. 47=/ 44=,
  NumCheck = True
Case Else
  NumCheck = False
End Select
End Function
```

```
Private Sub txtP4Pandeo_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP4Sigma_Change()
If IsNumeric(txtP4Sigma.Text) Then Vars.Sigma = txtP4Sigma.Text
End Sub

Private Sub txtP4Sigma_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP5Diametro_Change()
If IsNumeric(txtP5Diametro.Text) Then
  Vars.Diametro = CDbl(txtP5Diametro.Text)
  Call cmdp5calc_Click
  txtP5Tramos.Text = txtP5Tramos.Tag
Else
  txtP5Tramos.Text = ""
End If
End Sub

Private Sub txtP5Diametro_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP5Espesor_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP5Tramos_Change()
If txtP5Tramos.Text <> "" And Val(txtP5Tramos.Text) <> 0 Then
  Vars.Lt = Sqr((2 * (Vars.r ^ 2)) * (1 - Cos((3.1416 / 180) * (Vars.QQ /
Val(txtP5Tramos.Text)))))
End If
End Sub

Private Sub txtP5Tramos_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, False) Then KeyAscii = 0
End Sub

Public Sub pmsg(showit As Boolean, Optional msg As String = "", Optional value As
Integer = 0, Optional msg2 As String = "")
```

```
With msgWnd
  If showit Then
    Screen.MousePointer = 11
    .pb1.value = value
    .lblMsg.Caption = msg
    .lblMsg2.Caption = msg2
    .Show
    .Refresh
  Else
    .pb1.value = 0
    .lblMsg.Caption = ""
    .lblMsg2.Caption = ""
    .Hide
    Screen.MousePointer = 0
  End If
End With
End Sub

Private Sub txtP6Caudal_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP6Caudal_LostFocus()
If txtP6Caudal.Text = "" Then txtP6Caudal.Text = 0
End Sub

Private Sub txtP6Friccion_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP6Friccion_LostFocus()
If txtP6Friccion.Text = "" Then txtP6Friccion.Text = 0
End Sub

Private Sub txtP6Peso_KeyPress(KeyAscii As Integer)
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP6Peso_LostFocus()
If txtP6Peso.Text = "" Then txtP6Peso.Text = 0
End Sub

Private Sub txtP6Soporte_KeyPress(KeyAscii As Integer)
```

```
If Not NumCheck(KeyAscii, True) Then KeyAscii = 0
End Sub

Private Sub txtP6Soporte_LostFocus()
If txtP6Soporte.Text = "" Then txtP6Soporte.Text = 0
End Sub

Public Function CoordX(cval As Double, minx As Double, maxy As Double, miny
As Double) As Double
CoordX = (((cval - minx) * 6.2) / (maxy - miny)) + 1
End Function

Public Function CoordY(cval As Double, minx As Double, maxy As Double, miny
As Double) As Double
CoordY = (((cval - minx) * 6.2) / (maxy - miny)) + 2.3
End Function

Public Function CoordX2(cval As Double, minx As Double, maxy As Double, miny
As Double) As Double
CoordX2 = (((cval - minx) * 4.7) / (maxy - miny)) + 10.5
End Function

Public Function CoordY2(cval As Double, minx As Double, maxy As Double, miny
As Double) As Double
CoordY2 = (((cval - minx) * 4.7) / (maxy - miny)) + 2.5
End Function
```