

# **TRABAJO ESPECIAL DE GRADO**

## **INFLUENCIA DE LAS VIGAS ACARTELADAS EN PÓRTICOS PARA GALPONES**

Presentado ante la ilustre  
Universidad Central de Venezuela

Por los Bachilleres:

Ziegler D., Ronald E.

Espinosa, Guillermo W.

Para optar por el Título de

Ingeniero Civil

Caracas, junio de 2005

# **TRABAJO ESPECIAL DE GRADO**

## **INFLUENCIA DE LAS VIGAS ACARTELADAS EN PÓRTICOS PARA GALPONES**

TUTOR ACADÉMICO: Prof. José M., Velásquez

Presentado ante la ilustre  
Universidad Central de Venezuela

Por los Bachilleres:

Ziegler D., Ronald E.

Espinosa, Guillermo W.

Para optar por el Título de  
Ingeniero Civil

Caracas, junio de 2005

## ACTA

El día 23 de Junio de 2005 se reunió el Jurado formado por los profesores: José Manuel Velásquez, María Eugenia Korody y Juan José Tejon.

Con el fin de examinar el Trabajo Especial de Grado titulado: "Influencia de las vigas acarteladas en pórticos para galpones".

Presentado ante la Ilustre Universidad Central de Venezuela para optar al título de Ingeniero Civil.

Una vez oída la defensa oral que los bachilleres hicieron de su Trabajo Especial, este Jurado decidió las siguientes calificaciones:

NOMBRE	CALIFICACIÓN	
	Número	Letras
Ronald E, Ziegler D		
Guillermo W, Espinosa		

RECOMENDACIONES (Si las hubiera):

---

---

FIRMAS DEL JURADO

---

---

Caracas, 23 de junio de 2005

## DEDICATORIA

Dedico este trabajo:

A mi abuela y mi madre Elda y Alba por el amor infinito que me han dado, y la sabiduría de sus palabras que me ha enseñado, que nunca es tarde para empezar.

A mi esposa e hija Giselle y Elizabeth por su apoyo incondicional y por lo cálido de su amor. Junto a ustedes cada amanecer es maravilloso.

Guillermo Wan Espinosa.

Dedico este trabajo:

A mi madre, padre y hermanos por ser siempre un apoyo en mis momentos difíciles.

A mi tío Andrés a quien siempre llevo en mis recuerdos, que en paz descanse.

Ronald E. Ziegler D.

## **AGRADECIMIENTOS**

Guillermo:

A la vida que es la eterna escuela, donde el único fin es vivir plenamente, eso si aprendiendo de nuestros errores.

A mis amigos Yoel, Reinaldo, Edel y Karla quienes con su amistad me ayudaron a salir adelante, gracias por su incondicionalidad. A mi familia, en especial a Diana, Carlos, Cristina y Martha, sin su apoyo y su presión este trabajo especial de grado se hubiese finalizado.

A las licenciadas Maritza, Yalitza y Milagro por brindarme su orientación y permitirme laborar como becario-pasante de O.B.E. durante mi estadía en esta universidad.

A mis compañeros de esta casa de estudios, Joao, Humberto, Álvaro (el negro), Gianfranco, Denise, Richard, Orlando, Analis, Fermín, Juan, Luis (el chino), Sorelys, Gabriela, José Alfredo, Andrés, Genaro, Rafael, Pancho y Ronald. A todos gracias por ayudar de una u otra forma en la culminación de etapa.

Al señor Felipe, por confiar en mí. Y todas las personas que sin su valiosa colaboración no hubiese sido posible culminar este trabajo especial de grado.

Ronald:

A mis amigos del alma Rodolfo Linares, Andrés Jiménez, Genaro Rodríguez, Rafael Dan y Amaury Palacios.

Muy especialmente a Anet Teixeira, Gabriela Acuña, Irina Bello y Sorelys Cabrera, quienes siempre han estado en el momento que más he necesitado, representando el apoyo para seguir adelante y completar con éxito lo que me he propuesto.

A nuestro tutor Ing. José Manuel Velásquez porque gracias a él tuvimos la orientación necesaria para llevar a cabo los objetivos de este trabajo especial de grado.

ZIEGLER D. RONALD E.  
ESPINOSA GUILLERMO W.

**“INFLUENCIA DE LAS VIGAS ACARTELADAS EN PÓRTICOS  
PARA GALPONES.”**

**Tutor Académico: Prof. José M. Velásquez, Trabajo Especial de Grado. Caracas,  
U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Civil. 2005**

**RESUMEN.**

Palabras Claves: vigas acarteladas, pórtico para galpón, perfiles de acero, programa.

El objetivo principal de este trabajo especial de grado, consiste en analizar estructuralmente la influencia de las vigas acarteladas, en pórticos para galpones, bajo el efecto de diversas combinaciones de cargas. Para así determinar las reacciones en los extremos de los miembros, las rotaciones de junta y los desplazamientos laterales. Los pórticos considerados como problema, son los de perfiles laminados de acero estructural. En particular los pórticos que no contemplan armaduras entre las luces. Realizar los cálculos a mano, sería extremadamente laborioso, implicando pérdida de tiempo considerable, cada vez que se requiera cambiar alguna de las variables como la altura, la luz, el tipo de carga, el tipo de perfil, combinaciones de carga. Para evitar esta situación se ha desarrollado un programa en el lenguaje de programación de Visual Basic 6.0. Este programa llamado ***Acartbeam V1.3*** permite determinar deformaciones en los elementos, así como también los momentos de empotramiento de una viga acartelada sometida a cargas puntuales o uniformes, de tal forma que le ayude a tomar decisiones al momento de realizar el diseño estructural del pórtico con o sin acartelamientos.

Se puede analizar vigas acarteladas empotradas en sus extremos, de cualquier perfil comercial, esto con el objetivo de determinar el momento, con lo que el usuario puede tener idea del comportamiento estructural de la viga.

Adicionalmente como parte del anexo, se presentan tablas que permiten determinar los momentos en los extremos, para los perfiles de tipo HEA, HEB, IPE e IPN, considerando las cartelas. Para un mejor entendimiento y rendimiento del programa se creo el manual del usuario, que explica en forma sencilla el manejo del programa, ***Acartbeam V1.3***, por medio de gráficos y ejemplos de aplicación.

## ÍNDICE.

<b>CAPÍTULO</b>	<b>PÁGINA</b>
<b>I. INTRODUCCIÓN</b>	1
<b>II. PLANTEAMIENTO DEL PROBLEMA</b>	5
<b>III. OBJETIVOS</b>	8
<b>III.1 OBJETIVO GENERAL</b>	9
<b>III.2 OBJETIVOS ESPECÍFICOS</b>	9
<b>IV. ALCANCE</b>	10
<b>V. MARCO TEÓRICO</b>	12
<b>V.1 MÉTODO DE LAS ROTACIONES</b>	18
<b>V.2 DEFINICIONES</b>	23
<b>VI ¿POR QUÉ VISUAL BASIC?</b>	25
<b>VI.1 CONCEPTOS PROPIOS DE VISUAL BASIC</b>	27
<b>VI.1.1 FORMULARIOS Y CONTROLES</b>	27
<b>VI.1.2 EL FORMULARIO O FORM</b>	27
<b>VI.1.3 OBJETOS Y PROPIEDADES</b>	27
<b>VI.1.4 EVENTOS</b>	28
<b>VI.1.5 MÉTODOS</b>	28

<b>VI.2 FACILIDADES QUE OFRECE VISUAL BASIC</b>	<b>28</b>
<b>VII METODOLOGÍA</b>	<b>31</b>
<b>VII.1 FLUJOGRMA</b>	<b>34</b>
<b>VII.1.1 CODIGO FUENTE</b>	<b>35</b>
<b>VII.1.1.1 CODIGO FUENTE DEL FORM PORTICOPORT</b>	<b>35</b>
<b>VII.1.1.2 CODIGO FUENTE DEL FORM MEMPOT</b>	<b>52</b>
<b>VII.1.1.3 CODIGO FUENTE DEL FORM DB_PERFIL</b>	<b>61</b>
<b>VII.2 CREACION DEL MANUAL DEL USUARIO</b>	<b>83</b>
<b>VIII CONCLUSIONES</b>	<b>84</b>
<b>IX RECOMENDACIONES</b>	<b>87</b>
<b>X APÉNDICE A</b>	<b>89</b>
<b>XI ANEXO</b>	<b>109</b>
<b>XI.1 MOMENTOS DE EMPOTRAMIETO PARA UNA CARGA</b>	
<b>UNIFORMEMNTE DISTRIBUIDA</b>	<b>110</b>
<b>X.I.2 MEMORIA FOTOGRÁFICA</b>	<b>112</b>
<b>XI.3 MANUAL DEL USUARIO</b>	<b>116</b>
<b>XII REFERENCIAS BIBLIOGRÁFICAS</b>	<b>117</b>

## I. INTRODUCCIÓN

---

El ingeniero civil es básicamente un analista de problemas físicos concretos. Por medio de análisis puede diseñar soluciones prácticas. Sin embargo, en la mayoría de los casos, los procesos que permiten encontrar soluciones son muy repetitivos, iterativos y extensos, como es el caso de cálculos estructurales. Los resultados suelen ser conservadores si estos cálculos se hacen manualmente. Como consecuencia se hace necesario el uso de programas de computadoras que permiten optimizar el proceso de cálculo y arrojar resultados más precisos.

Es cierto que la tecnología de la informática cada día da pasos agigantados en lo referente a la creación de programa, aun así el ingeniero se ve en la necesidad de crear programas para dar solución a problemas muy particulares que escapan de los grandes paquetes de programa comerciales. Para tal fin se requiere que el profesional maneje algún lenguaje de programación. Como por ejemplo, Turbo Pascal, C++, Delphi, Fortran, Builder, Visual, Visual Basic, entre otros.

Este trabajo especial de grado busca como objetivo analizar estructuralmente la influencia de las vigas acarteladas en pórticos para galpones, sometidas a diversas combinaciones de carga, para obtener parámetros típicos dentro del comportamiento con relación a las reacciones, cortes, momentos flectores y desplazamientos laterales.

Por lo tanto se creó el programa, en el lenguaje de programación M. S. Visual Basic 6.0. Dado que es un lenguaje de programación dirigido a objetos, ofrece una gran cantidad de herramientas de programación, que para ser usadas no requieren de una amplia experiencia en el área de la programación. Es un programa que permite evitar

escribir numerosas líneas de código, para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente se agregan objetos prefabricados en su lugar dentro de la pantalla. Lo que se traduce en programas, más amigables, con semejanzas a los propios paquetes que tiene Windows y permite una interfaz con el paquete de Microsoft Office.

La toma de decisiones es una de las características de la vida, pero en ingeniería civil, estas decisiones normalmente tienen que ver con resolver un problema, partiendo de un análisis primario. Entre una de las variables que provoca límites, se encuentra el factor económico, sin dejar de tomar en cuenta los factores de seguridad que se involucran en la posible solución. Esto depende del tipo de obra y las normas que le rigen en el diseño.

El rendimiento en una obra, está relacionada con el tiempo de ejecución y el ahorro de materiales, sin que esto signifique desmejorar la calidad. Es en este punto donde entran las secciones acarteladas como respuesta a lo antes expuesto.

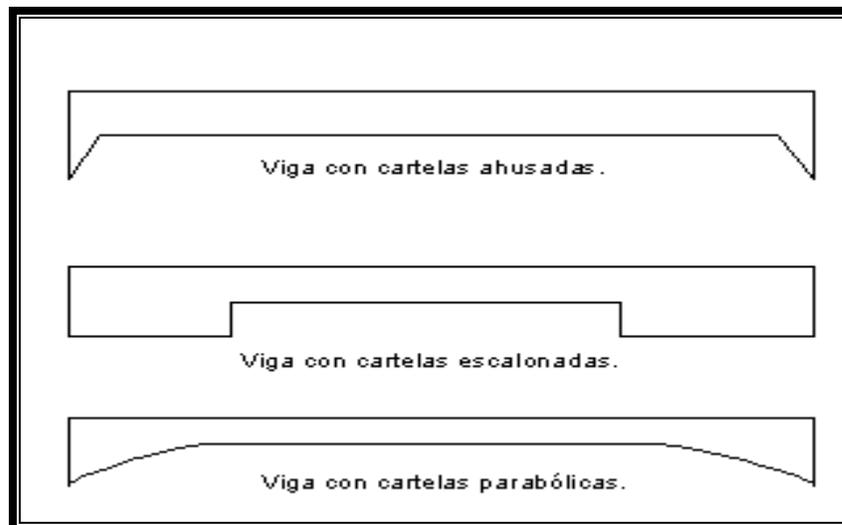
Es común observar, hoy estructuras con vigas acarteladas, pero en concreto armado, sin embargo, su elaboración cada vez resulta más costosa, por causa de la colocación de los aceros (cabillas), del encofrado y mano de obra.

Otra de las soluciones viables es realizar estructuras en acero estructural con perfiles laminados o tubulares. La solución de las secciones acarteladas también puede ser aplicada, y en forma más rápida que en concreto armado.

En este trabajo de grado se analizarán los pórticos tipo galpón, de perfiles laminados de acero estructural, en particular que influencia tienen las cartelas sobre estos pórticos. Con la ayuda del programa **Acartbeam V1.3** se logró analizar, los comportamientos de los perfiles con cartelas en vigas empotradas en sus extremos, estos resultados forman parte del anexo.

## II. PLANTEAMIENTO DEL PROBLEMA

Los miembros de sección variable, también son conocidos como miembros no prismáticos. Las formas más comunes son las cartelas escalonadas, ahusadas o las parabólicas.



II.1 Tipos de vigas con sección variable.

El cálculo de este tipo de miembros en concreto armado previo al desarrollo de las computadoras personales, requería de cálculos extremadamente laboriosos, por lo que desarrollaron tablas con las que se pueden determinar el valor de los momentos en los extremos, para cargas distribuidas y puntuales. Estas tablas están recogidas en el *Handbook of Frame Constants*, publicado por la Portland Cement Association.

En el caso de este trabajo especial de grado, el tipo de material a considerar son los perfiles laminados de acero estructural, como se acoto en el párrafo anterior la solución de estructuras con miembros prismáticos o de sección variable, sigue siendo difícil. La razón es que los métodos aplicables a estas estructuras, implican la solución de integrales que están en función de la inercia pero en el denominador.

$$\Delta = \int Mm/EI_{(x)} \quad (1)$$

$$\Delta = \int \partial MM/\partial PEI_{(x)} \quad (2)$$

La consecuencia de esto es que se tendría que resolver una integral que de forma cerrada sería imposible, más aun en el caso de secciones con perfiles de acero. Esta inercia variable, como se demuestra en el anexo de este trabajo especial de grado, resulta muy compleja.

Una forma para resolver este problema, podría ser la usar el principio de los trabajos virtuales o el teorema de Castigliano, con el fin de calcular las deflexiones, o utilizar el método de la viga conjugada.

Si bien es cierto, que en el mercado existe una gran cantidad de programas capaces de analizar y diseñar cualquier tipo de estructuras, no hay un programa que genere información de cómo influyen las cartelas en un pórtico tipo galpón. Esto se tradujo en la necesidad de crear un programa específico, que entre sus bondades esta la de calcular pórticos para galpones mostrando las resultantes, y los diagramas de fuerzas cortantes, axiales, y de momento flector como producto de las cargas que actúen en el sistema. Pero una de principales particularidades de este programa denominado ***Acartbeam V1.3***, es la de generar tablas para vigas con cartelas y empotradas en sus extremos de los perfiles HEA, HEB, IPN e IPE y cualquiera definido por el usuario. El objetivo de esto es que el usuario no requiera de resolver este tipo de viga cada vez que cambie la pendiente, la altura, la luz o la carga, sino que utilice estas tablas para calcular en forma rápida los momentos en los extremos.



Los objetivos generales y específicos planteados en el presente trabajo especial de grado son:

### **III.1 OBJETIVO GENERAL**

- Analizar estructuralmente la influencia de las vigas acarteladas en pórticos para galpones sometidas a diversas combinaciones de cargas, para obtener parámetros típicos dentro del comportamiento con relación a reacciones, cortes, momentos flectores y desplazamientos laterales.

### **III.2 OBJETIVOS ESPECÍFICOS**

- Preparar un programa general de análisis que considere las vigas de sección variable, correspondiente a los perfiles laminados comerciales. De tal manera que una vez conformada la base de datos de dichas vigas acarteladas se analice la influencia para las siguientes variables:
  - a) Altura del galpón.
  - b) Luces.
  - c) Inclinación de las vigas.
- Crear un patrón de comportamiento general con las variaciones encontradas.
- Concluir acerca del comportamiento general.



Este trabajo especial de grado conlleva el aprendizaje del lenguaje de programación Microsoft Visual Basic 6.0, con el objetivo de crear una aplicación, que permite el análisis de vigas acarteladas en pórticos para galpones, mediante el método de las rotaciones o método de la pendiente-deflexión.

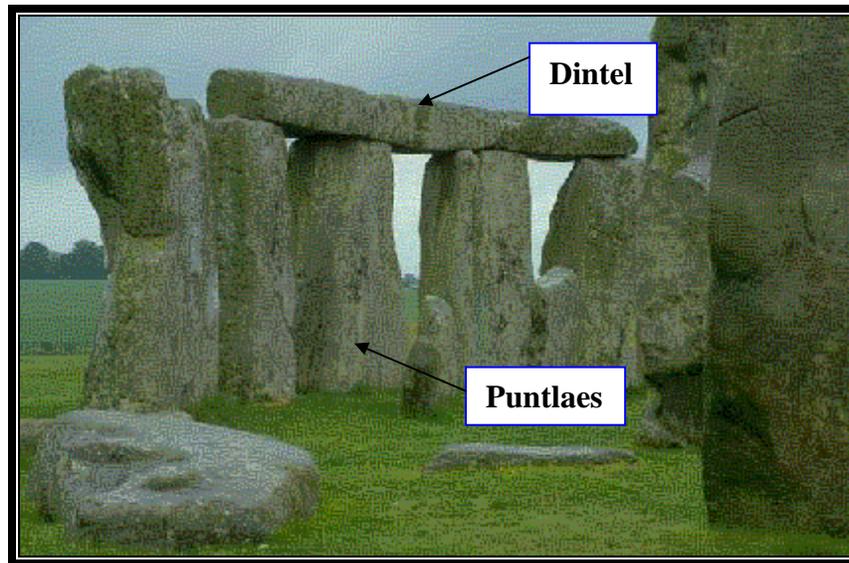
Crear la base de datos que contiene la información de las propiedades de los perfiles laminados comerciales más utilizados, los HEB, HEA, IPN e IPE.

Reproducir mediante la aplicación del programa, tablas que permitan calcular los momentos de empotramiento, de vigas acarteladas con perfiles laminados comerciales, para cargas distribuidas y puntuales.

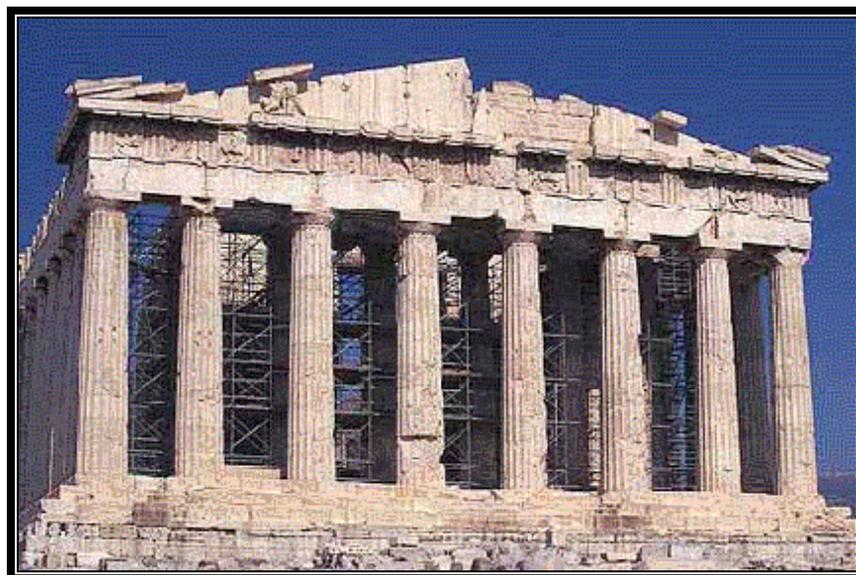
El desarrollo del manual de usuario del programa *AcartBeamV1.3* para el análisis de pórticos tipo galpón con vigas acarteladas.



El origen de los pórticos actuales, proviene del mundo antiguo. Las primeras estructuras datan del año 1800 a. C. en lo que se conoce con el nombre de los **trilitos del conjunto de Stonehenge**, en Inglaterra. Otro ejemplo lo representan las construcciones clásicas de los griegos como el **P Partenón**.



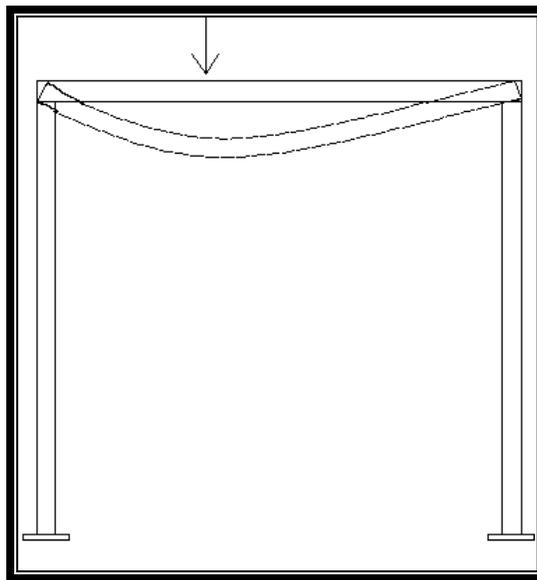
Trilitos del conjunto de Stonehenge Inglaterra. [www.olejarz.com/images/46%20Stonehenge.jpg](http://www.olejarz.com/images/46%20Stonehenge.jpg)



P Partenón Griego templo dedicado a la diosa [www.terra.com.mx/Turismo/ galeria\\_de\\_fotos/005...](http://www.terra.com.mx/Turismo/galeria_de_fotos/005...)

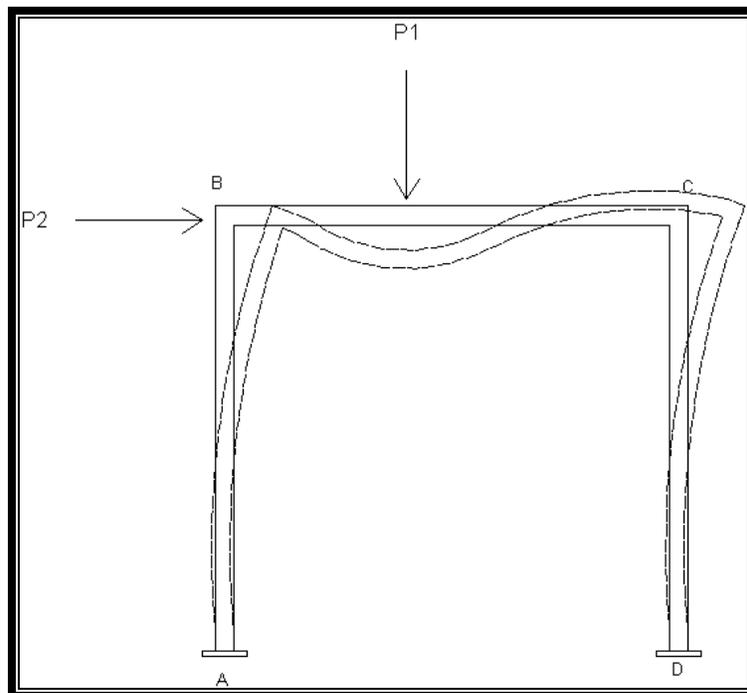
Entre las cosas que tenían en común estas estructuras encontramos los puntales, elementos verticales, los dinteles, elemento horizontal, y el material de construcción que eran rocas de gran tamaño.

En términos estructurales el conjunto puntal-dintel solo permite la flexión en el dintel por efectos de cargas verticales (V.3), en el caso del puntal este se flexiona si existe una carga horizontal que lo solicite. Esto es ocasionado porque no existe una unión rígida entre el dintel y los puntales. Ahora bien, si se forma una junta rígida entre el dintel y el pilar, se pasa a llamar viga y columna. Esto da origen al pórtico rígido simple.



V.3 conjunto dintel-puntal.

Como resultado de esta unión rígida entre las columnas y las vigas se logra una mayor resistencia y rigidez (V.4). La ganancia es que este sistema puede soportar cargas verticales y horizontales en forma simultánea. Estando sometido entonces a esfuerzos de tracción y compresión en las columnas y momentos flectores en las vigas.

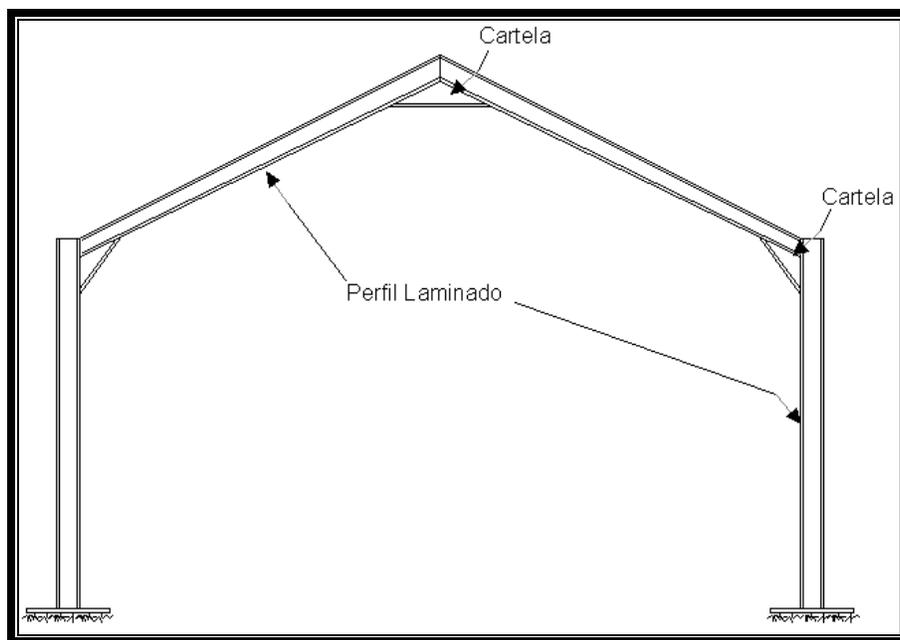


V.4 pórtico rígido simple.

Los materiales más utilizados en la actualidad son el concreto armado, la madera y el acero. Desde los inicios de los años 90 se ha ido incrementando la construcción de estructuras en acero estructural. Esto en parte a la sobreproducción del material a nivel mundial que ha provocado la baja de los precios, y la rapidez con la que se puede ejecutar una obra con perfiles de acero estructural.

No obstante, aun se sigue utilizando el concreto armado como material de construcción, incluso en lo que se conoce como concreto prefabricado.

El tipo de pórtico al que se refiere este trabajo especial de grado, es uno muy particular. Son pórticos para galpones con cartelas, de perfiles laminados de acero estructural a dos aguas (V.5).



El pórtico de acero es un sistema estructural capaz de soportar momentos en las juntas, con miembros trabajando a flexión. La base del éxito de este sistema son las uniones. Las uniones se pueden hacer bien sea de soldadura o utilizando planchas con soldadura o planchas con pernos.

Los galpones presentan ciertas características, espaciales que los hacen muy particulares tales como, luces muy grandes, pudiendo ser mayores a 40 metros, así mismo sus alturas sobrepasan los 4 metros. Dadas estas características los pórticos para galpones, realizados en acero estructural, pueden ser celosías, caracterizadas por estar sometidos a fuerzas axiales y cargas aplicadas en los nodos. O el pórtico de perfiles de acero laminado, a dos aguas que son objeto de estudio en este trabajo especial de grado. La mayoría de los galpones cumplen el objetivo de almacenar, maquinaria industrial, alimentos, equipos mecánicos, fabricas, hangares, gimnasios. Básicamente la de depósitos de gran volumen.

No es común el uso de galpones a varios niveles, por lo que las principales cargas a las que está sometido son: el viento, el peso propio, grúas internas para el transporte de materiales, en ocasiones los cambios de temperatura por fuego o nieve, y los sismos. No se hace mención del efecto del agua acumulada en la superficie dado que estos pórticos son inclinados.

Si bien es cierto que con los perfiles de acero es posible construir con mucha rapidez, podría resultar poco económico hacer los galpones con perfiles laminados de gran envergadura, para que resista el conjunto de fuerzas al cual se someterá. Para evitar esto una de las soluciones es agrandar la sección de la viga en los extremos, esto también es causado por la falta de existencia de perfiles de gran tamaño en el mercado nacional.

Este tipo de solución se conoce con el término de sección acartelada. El objetivo de esto es permitir que se absorba mayor momento en los nodos, con lo que se consigue disminuir el momento máximo en el centro de las vigas. Esto se traduce en ahorro de material y disminución de costos.

### **V.1. MÉTODO DE LAS ROTACIONES.**

Este método también es conocido como “método de las deformaciones angulares o método de la pendiente-deflexión”, es un método de desplazamiento dado que las incógnitas son rotaciones y desplazamientos. *“Fue desarrollado originalmente, por Heinrich Mandrela y Otto Mohr para efectuar el análisis de esfuerzos secundarios en celosías”* (Hibbeler, Russell, 1997.p. 518). En 1915 el profesor George A. Maney presenta una ampliación, de los trabajos de Mandrela y Mohr, pero esta vez el método es aplicado al análisis de vigas continuas y pórticos.

El nombre de pendiente-deflexión viene dado, por la razón *“de que los momentos en los extremos de los miembros de las estructuras estáticamente indeterminadas, se expresan en función de las rotaciones (o pendientes) y las deflexiones en los nodos”* (McCormac, Jack y Nelson, James, 2002. p. 465).

Este método, no considera las deformaciones por corte, ni por fuerza axial. Se considera las deformaciones por momento flector. *“Como el efecto de la deformación del cortante y la carga axial en el análisis de las tensiones de la mayoría de los pórticos indeterminados es muy pequeña, el error resultante del uso este método es también pequeño”* (Norris, Charles y Wilbur, John, 1969. p. 422).

Su enunciado dice así: “La ecuación fundamental de la deformación angular es una expresión del momento en un extremo de una barra en función de cuatro cantidades: el giro de la tangente en cada extremo de la barra, el giro de la cuerda que une los extremos de la elástica, y las cargas exteriores aplicadas en la barra”

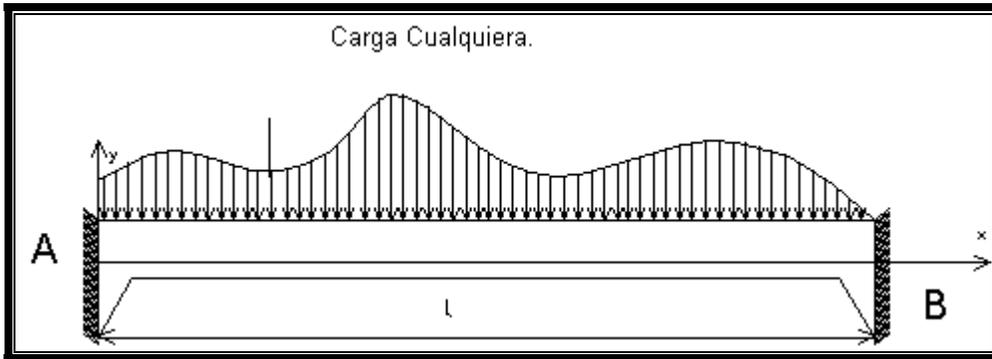
( Norris, Charles y Wilbur, John, p. 422).

Detrás de este método este el principio de superposición, ya se considera el efecto final como el resultado, de varios efectos por separado, tal cual lo refiere este enunciado.

Las ecuaciones que permiten aplicar este método de rotaciones como se ha mencionado son complejas, pero el Ingeniero Francés, Bresse (1854) desarrolló unas ecuaciones generales, para el cálculo de los desplazamientos de miembros de ejes curvos, o rectos, que contemplaba el hecho de tener secciones variables. La solución de estas ecuaciones cuya demostración esta en el apéndice de esta trabajo, se muestra a continuación.

$$\begin{aligned}
 K_1 &= \int_0^l \frac{x^2}{I} dx & K_2 &= \int_0^l \frac{(l-x)^2}{I} dx \\
 K_3 &= \int_0^l \frac{x(l-x)}{I} dx & & \\
 K_4 &= \int_0^l \frac{\mu x}{I} dx & K_5 &= \int_0^l \frac{\mu(l-x)}{I} dx
 \end{aligned} \tag{3}$$

Sea:

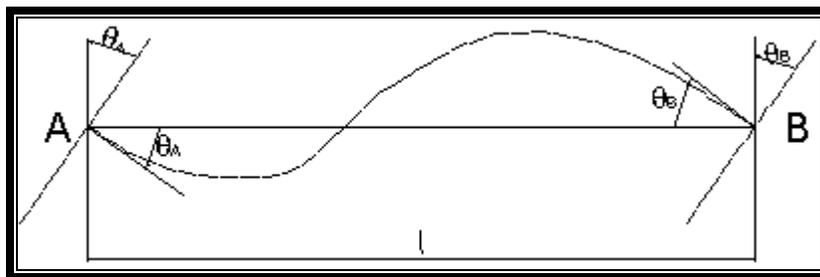


V.6 Viga empotrada, cargada de sección variable.

Los momentos de empotramiento por efecto de las cargas externas son:

$$MF_{AB} = \frac{K_3 K_4 - K_1 K_5}{K_3^2 - K_1 K_2} l \quad MF_{BA} = \frac{K_2 K_4 - K_3 K_5}{K_3^2 - K_1 K_2} l \quad (4)$$

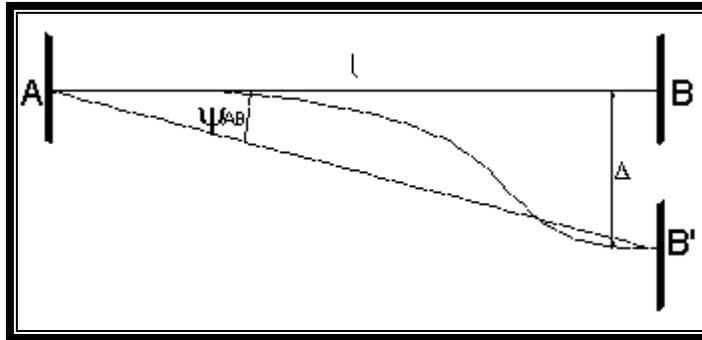
Los momentos por causa de las rotaciones en las juntas son (V.7):



V.7 Rotación de las juntas.

$$MF_{AB}' = \frac{K_3 E I^2}{K_3^2 - K_1 K_2} \left( \frac{K_1}{K_3} \theta_A + \theta_B \right) \quad MF_{BA}' = \frac{K_3 E I^2}{K_3^2 - K_1 K_2} \left( \theta_A + \frac{K_2}{K_3} \theta_B \right) \quad (5)$$

Por efecto de un desplazamiento  $\Delta$  en uno de los extremos de la viga se tienen los siguientes momentos:



V.8 Desplazamiento del extremo B.

$$M_{AB}' = \frac{K_1 + K_3}{K_1 K_2 - K_3^2} \Delta E I \quad M_{BA}' = \frac{K_2 + K_3}{K_1 K_2 - K_3^2} \Delta E I \quad (6)$$

$$\Psi_{AB} = \frac{\Delta}{l} \quad (7)$$

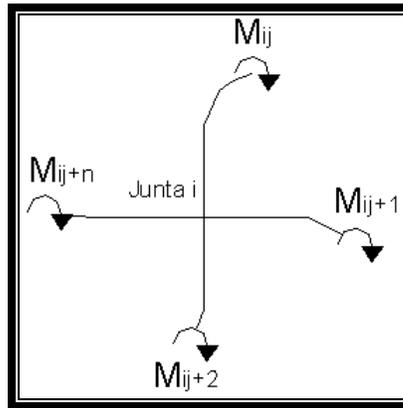
Aplicando el principio de superposición, que no es más sumar de los tres efectos se tiene que el momento final es:

$$\begin{aligned} M_{AB} &= MF_{AB} + MF'_{AB} + M'_{AB} \\ M_{BA} &= MF_{BA} + MF'_{BA} + M'_{BA} \end{aligned} \quad (8)$$

Sustituyendo por sus valores en 8 se tiene:

$$\begin{aligned} M_{AB} &= \frac{K_3 K_4 - K_1 K_5}{K_3^2 - K_1 K_2} l + \frac{K_3 E I^2}{K_3^2 - K_1 K_2} \left( \frac{K_1}{K_3} \theta_A + \theta_B \right) + \frac{K_1 + K_3}{K_1 K_2 - K_3^2} \Delta E I \\ M_{BA} &= \frac{K_2 K_4 - K_3 K_5}{K_3^2 - K_1 K_2} l + \frac{K_3 E I^2}{K_3^2 - K_1 K_2} \left( \theta_A + \frac{K_2}{K_3} \theta_B \right) + \frac{K_2 + K_3}{K_1 K_2 - K_3^2} \Delta E I \end{aligned}$$

En el caso de vigas de más de un tramo y pórticos se debe de cumplir que la suma de los momentos en las juntas debe de ser igual a cero, es decir:



V.9 Junta donde convergen n elementos.

$$\sum M_i = 0 \quad (10)$$

Esta ecuación de equilibrio en la junta permite establecer las relaciones entre los diferentes momentos con lo que establecen posteriormente el sistema de ecuaciones.

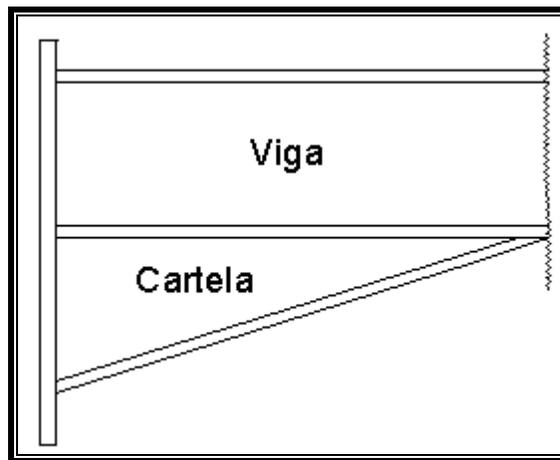
Para la aplicación de los de las ecuaciones, (Norris, Charles y Wilbur, John, p. 422 y 423) conviene fijar los criterios de signos de la siguiente forma:

1. Los momentos en los extremos de una barra son positivos si actúan en el sentido de las agujas del reloj.
2. Sea  $\theta$  el giro de la tangente a la elástica en el extremo de una barra, respecto a la posición original de la misma. El ángulo  $\theta$  es positivo cuando la tangente a la elástica ha girado en el sentido de las agujas del reloj, desde su dirección original.

3. Sea  $\Psi$  el giro de la cuerda que une los extremos de la elástica, respecto a la dirección original de la barra. El ángulo  $\Psi$  se considera positivo cuando la cuerda ha girado el sentido de las agujas del reloj, desde su dirección original.

## V.2 DEFINICIONES.

**Cartela:** elemento saliente a manera de ménsula de más altura que vuelo, sirve para sostener un cuerpo que sobresale (.inti.gov.ar/CIRSOC301) véase la figura V.10. Su función estructural es la adsorber más momento en los extremos, con lo que se puede conseguir ahorro de material.



V.10 unión viga-cartela.

**Inercia:** tendencia de un objeto a no cambiar su estado de movimiento. El movimiento puede ser traslacional o rotacional. En el primero se cumple que a mayor masa mayor inercia, en el segundo la inercia es función de la masa y de cómo se distribuye esta alrededor del eje de rotación, a mayor separación de la masa del eje de rotación mayor inercia (ciencianet.com).

**Momento de inercia:** *se entiende como una medida de la inercia rotacional, o tendencia de un cuerpo a resistirse al cambio en su movimiento rotacional (ciencianet.com).*

**Viga:** *elemento usualmente horizontal, recto y utilizado para soportar cargas verticales. Su diseño por lo general es para resistir flexión, sin embargo, si su luz es muy corta y soporta grandes cargas se pueden generar fuerzas internas cortantes considerables, pudiendo esto determinar el diseño (Hibbeler, Russell, p.3).*

**Columna:** *elementos que por lo general son verticales y se diseñan para resistir cargas axiales, tanto de tracción como de compresión. Estos elementos pueden también estar sometidos a cargas axiales y a momentos flectores. Estas columnas se conocen con el nombre de columnas flexocomprimidas (Hibbeler, Russell, p.6).*

**Pórtico o marco:** *sistema estructural capaz de soportar momentos en las juntas, con miembros trabajando a flexión. Esta formado por columnas y vigas. Las conexiones entre estos elementos, pueden ser rígidas o articuladas.*



Visual Basic es un lenguaje de programación desarrollado por Microsoft. Visual Basic es un lenguaje visual que desciende del lenguaje de programación Basic. Visual hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). Esta basado en objetos a diferencia de C++ o Java que están orientados a objetos.

Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma.

La versión de Visual Basic 6.0 esta orientada a la realización de programas para Windows, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús y otros. Este tipo de lenguaje es denominado lenguaje de cuarta generación, dado que una gran cantidad de tareas se realizan sin escribir código, simplemente con operaciones graficas con el ratón.

La mayoría de los elementos de interacción con el usuario de los que se disponen en Windows 95, 98, NT pueden ser programados en Visual Basic 6.0 de forma sencilla.

Visual tiene dos modos de trabajar una es el modo de diseño, en donde el usuario construye interactivamente la aplicación, colocando controles en el formulario, definiendo sus propiedades, y desarrollando funciones para gestionar los eventos. El otro modo es el de ejecución donde se ejecuta la aplicación desarrollada.

## **VI.1 Conceptos propios de Visual Basic.**

**VI.1.1 Formularios y Controles:** cualquier elemento gráfico es un tipo de control, los botones, las cajas de diálogo y de texto, las cajas de selección desplegables, los botones de opción y selección, las barras de desplazamiento horizontales y verticales, los gráficos, los menús, son controles de Visual Basic.

**VI.1.2 El formulario o *form*** es una ventana, que contiene los controles. Así una aplicación puede tener varios formularios, pero si es sencilla un solo formulario puede ser suficiente. Tanto los controles y los formularios deben de tener nombres asignados por el usuario.

**VI.1.3 Objetos y Propiedades:** los formularios y los controles son entidades genéricas denominadas *clase*. Mientras que un objeto es una entidad que tiene asociado un conjunto de métodos, eventos y propiedades.

Cada formulario y cada tipo de control tienen un conjunto de propiedades que definen su aspecto gráfico tales como el tamaño, el color, la posición de la ventana, tipo y tamaño de letra entre otros. Muchas de estas propiedades pueden ser cambiadas en el modo diseño, otras muy particulares en el modo ejecución.

**VI.1.4 Eventos:** cualquier acción que genere el usuario en el programa se le denomina *evento*. Por ejemplo el hacer clic sobre un botón, arrastrar un icono, pulsar una tecla o combinaciones de teclas, elegir una opción de un menú, escribir en una caja de texto, o simplemente mover el ratón.

**VI.1.5 Métodos:** son las funciones internas de un determinado objeto que permite realizar funciones sobre él o sobre otro objeto. Estas funciones ya vienen pre-programadas dentro del lenguaje

## **VI.2 Facilidades que ofrece Visual Basic.**

**VI.2.1-** Facilidad para el diseño de pantallas, se puede dibujar literalmente la interfaz de usuario, por medio de una o más ventanas, sobre las cuales se colocan los controles que permiten la interacción con la aplicación.

**VI.2.2-** Programación por eventos, en buena medida el código de un programa en Visual Basic es para indicar como responder ante eventos determinados. El orden en que se ejecuta el código es función de los eventos generados por las acciones del usuario en su interacción con la aplicación.

- VI.2.3-** La existencia de una gran cantidad de controles para el diseño de la interfaz.
  
- VI.2.4-** Los grandes programas se pueden diseñar mediante módulos lo que se traduce en un programa más fácil de manejar y menos sensible a los errores.
  
- VI.2.5-** Visual Basic posee un editor-interpretador que por lo general detecta e incluso sugiere los cambios para corregir errores tipográficos o de programación.
  
- VI.2.6-** Tiene un lenguaje denominado por los programadores como lenguaje de alto nivel, como herramienta de programación contiene un conjunto de potentes funciones de fácil uso, tales como funciones matemáticas incorporadas, funciones de manipulación de cadenas de caracteres, creación y utilización de gráficos, manejo de bases de datos, elementos multimedia y otras.
  
- VI.2.7-** Posee un extenso sistema de ayuda integrado en el paquete del programa.

**VI.2.8-** El modelo de objetos componentes (COM), que incluye automatización y la especificación de objetos ActiveX, hace posible que los componentes del programa, que el programador construye, funcionen con otros componentes de la aplicación. Del mismo modo permite que se creen herramientas para un programa ya existente.

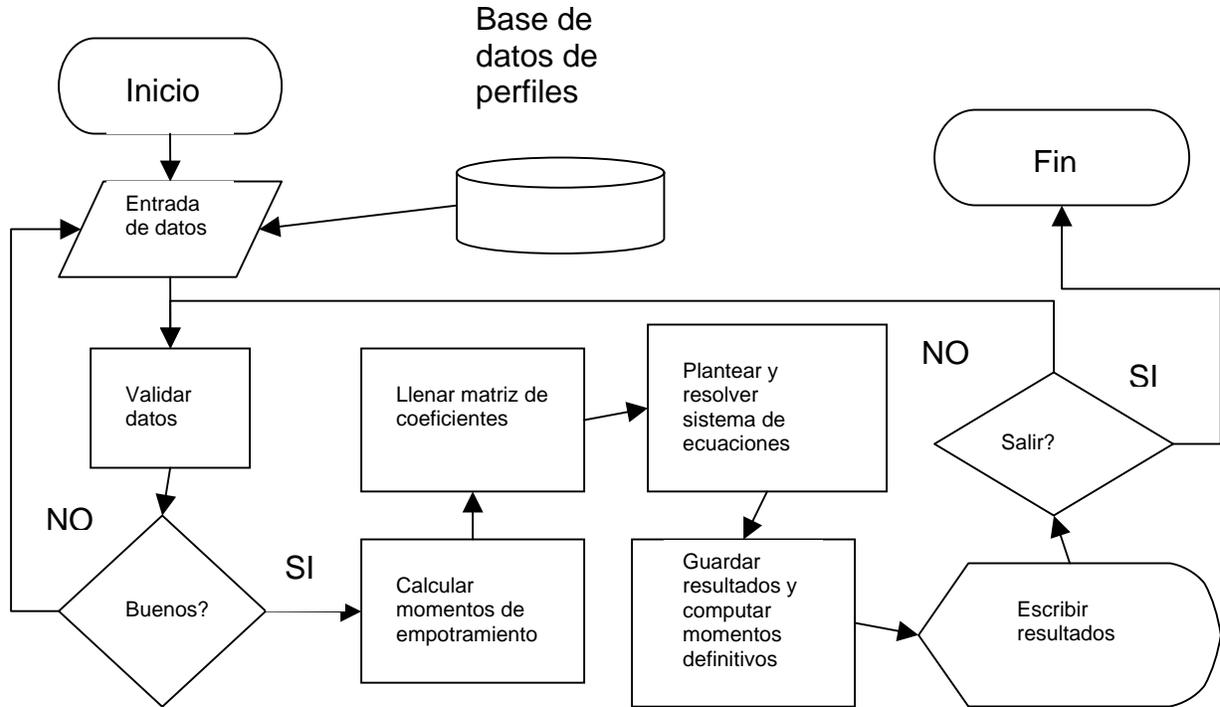


A continuación se explican los pasos que se siguieron para el desarrollo de los objetivos de este trabajo especial de grado.

1. **Aprender el lenguaje de Visual Basic:** se realizó un curso de Visual Basic 6.0 nivel básico, en el Centro de enseñanza asistido por computador CENEAC, en la facultad de Ciencias de la UCV.
2. **Creación de una base de datos:** al considerarse los perfiles laminados comerciales, se requirió crear una base de datos con la información de estos perfiles, que fuera compatible con Visual Basic.
3. **Análisis y fundamentos del problema:** se recopiló información preliminar por medio de la cual entender las variables que conforman el problema de las secciones variables en pórticos.
4. **Estructuración esquemática del sistema:** se crearon los diagramas de flujo preliminares para la estructura general del programa.
5. **Posicionamiento de los objetos en el formulario:** diseño de la interfaz, procurando hacerla cómoda para los usuarios finales del programa.

- 6. Creación de las subrutinas que forman el programa:** aprovechando que Visual permite desarrollar una aplicación en términos de módulos, se procedió a crear módulos con cada una de las funciones del programa.
- 7. Depuración de las subrutinas:** una vez finalizada cada subrutina se procedió a correr las mismas con el fin de determinar errores y proceder a corregirlos.
- 8. Integrar las subrutinas en el proyecto principal:** en esta fase se integraron todas las subrutinas o módulos independientes en el programa principal.
- 9. Depurar el proyecto final:** con la estructura final del programa que comprende la integración de todos los elementos al sistema principal, se realizaron corridas con el objeto para hallar posibles errores y corregirlos.
- 10. Validación de los resultados:** comparación de los resultados emitidos por el programa, con problemas existentes en la bibliografía consultada, en un principio considerando sección constante y posteriormente con cartelas siendo consistentes con los resultados que ofrecían los textos.

### VII.1 Flujoograma



### VII.1.1 Código fuente

En este subcapítulo se muestran las subrutinas correspondientes a las formas principales del programa, encargadas del cálculo de momentos, inercia de la sección, coeficientes  $K_i$ , entre otras aplicaciones destinadas a la entrada de datos.

#### VII.1.1.1 Código fuente del Form PorticoProp.

La forma PorticoProp genera resultados de momentos a partir de los datos que introduce el usuario. Para esto se tiene la opción de cargar cada elemento con sus propiedades, tipos de carga y de perfil, cuyas propiedades pueden ser dadas por el usuario o cargadas desde una base de datos, con las propiedades geométricas de los perfiles laminados más comunes, HEA, HEB, IPE e IPN.

Datos que ingresa el usuario en mm:

**Hac:** altura de la cartela en el extremo A.

**Hbc:** altura de la cartela en el extremo B.

**Xa:** longitud horizontal de la cartela en A.

**Xb:** longitud horizontal de la cartela en B.

**d:** altura del perfil.

**bf:** largo del ala.

**tf:** ancho del ala.

**tw:** ancho del alma

**P1 y P2:** cargas puntuales (kg)

Const Pi = 3.14159265359

Dim Lon, **Hac**, **Hbc**, **Xa**, **Xb**, **d**, **bf**, **tf**, **tw**, **P1**, **P2**, a, b As Double

Dim buenos As Boolean

Dim MatA(0 To 4, 0 To 5), MatB(0 To 4, 1) As Single

Dim Nm As Integer  
Dim Eg As Boolean

---

Private Sub CalcMom\_Click()

buenos = True

Call revisarDatos(buenos)

If buenos = True Then

E = Val(Text5.Text) \* 1000000 \* 10000

IntroDatPort

For elemento = 1 To 4

MomentoS(elemento, 1) = 0

MomentoS(elemento, 2) = 0

Next

BarraProg.Max = 20

BarraProg.Min = 1

n = 1

For elemento = 1 To 4

For carga = 1 To 5

Call CalcularMom

BarraProg.Value = n

n = n + 1

Next

Next

n = 0

BarraProg.Value = 1

For i = 1 To 4

For j = 1 To 2

momEmpot(n).Caption = Format(MomentoS(i, j), "##,##0.00") + " Kg-m"

n = n + 1

Next

Next

CONV\_COEF\_EN\_LETRAS

LLENAR\_MATRIZ\_SISTEMA

RESOLVER\_SISTEMA

CALCULAR\_MOM\_DEF

COLOCAR\_RESULTADOS

Else

MsgBox ("Faltan datos")

End If

End Sub

---

```
Private Sub CALCULAR_MOM_DEF()
Dim fiB, fiC, fiD, R1, R2 As Double
L = Val(Text1.Text)
```

```
'If Alfa = 0 Then
'Alfa = 0.000000000001
'End If
h = L / 2 * Tan(Alfa)
```

```
fiB = MatA(0, Nm)
fiC = MatA(1, Nm)
fiD = MatA(2, Nm)
R1 = MatA(3, Nm)
R2 = MatA(4, Nm)
```

```
MomDef(1, 1) = A1 + D1 * fiB + F1 * R1
MomDef(1, 2) = B1 + E1 * fiB + G1 * R1
```

```
MomDef(2, 1) = A2 + C2 * fiB + D2 * fiC + F2 * (R2 - R1) * L / (4 * h)
MomDef(2, 2) = B2 + D2 * fiB + E2 * fiC + G2 * (R2 - R1) * L / (4 * h)
```

```
MomDef(3, 1) = A3 + C3 * fiC + D3 * fiD + F3 * (R1 - R2) * L / (4 * h)
MomDef(3, 2) = B3 + D3 * fiC + E3 * fiD + G3 * (R1 - R2) * L / (4 * h)
```

```
MomDef(4, 1) = A4 + C4 * fiD + F4 * R2
MomDef(4, 2) = B4 + D4 * fiD + G4 * R2
```

```
End Sub
```

---

```
Private Sub COLOCAR_RESULTADOS()
Dim Unidad As String
For i = 1 To 5
'If Abs(MatA(i - 1, Nm)) < 0.00000001 Then
' desplaz(i - 1).Caption = 0
'Else
If i = 1 Or i = 2 Or i = 3 Then
Unidad = " rad"
Else
Unidad = " Mts."
End If
desplaz(i - 1).Caption = Str(MatA(i - 1, Nm)) + Unidad ' Scientific
'End If
Next
cont = 0
For i = 1 To 4
```

```

For j = 1 To 2
momEmpot(cont + 8).Caption = Str(MomDef(i, j))
cont = cont + 1
Next
Next
End Sub

```

---

```

'-----Algoritmo del calculo-----
Public Sub GaussJordan()
Dim k, j, i, L As Integer
Dim t As Single

On Error GoTo ErrE
'-----Transponer filas-----
If (MatA(0, 0) = 0) Then
For i = 0 To (Nm - 1)
    If (MatA(i, 0) <> 0) Then
        For j = 0 To Nm
            MatB(0, j) = MatA(0, j)
            MatA(0, j) = MatA(i, j)
            MatA(i, j) = MatB(0, j)
        Next
    End If
Next
End If
'-----
If Eg = True Then Exit Sub
For k = 0 To Nm - 1
    t = MatA(k, k)
    For j = 0 To Nm
        MatA(k, j) = MatA(k, j) / t
    Next j
    For i = 0 To Nm - 1
        If (i = k) Then GoTo Es
        t = MatA(i, k)
        For L = 0 To Nm
            MatA(i, L) = MatA(i, L) - t * MatA(k, L)
        Next L
    Es:
        Next i
Next k
'-----

Exit Sub
ErrE:

```

```
MsgBox "El sistema no tiene solución", vbCritical
End Sub
```

---

```
Private Sub LeerMatriz()
On Error GoTo ErrLec
  Dim i, j As Integer
  For i = 0 To Nm - 1
    For j = 0 To Nm
      'If Cuadric.TextMatrix(i + 1, j) = "" Then
      'Cuadric.TextMatrix(i + 1, j) = 0
      MatA(i, j) = MatrizAcalc(i + 1, j + 1)
    Next
  Next
  Eg = False
Exit Sub
ErrLec:
  MsgBox "Error al ingresar los datos (" & Err.Description & ")", vbExclamation
  Eg = True
End Sub
```

---

```
Private Sub RESOLVER_SISTEMA()
```

```
  For i = 1 To 5
    For j = 1 To 6
      MatrizAcalc(i, j) = MatrizA(i, j)
    Next
  Next
```

```
  Call LeerMatriz
  Call GaussJordan
End Sub
```

---

```
Private Sub LLENAR_MATRIZ_SISTEMA()
Dim h, CoefVcd, CoefVcb, CoefHcd, CoefHcb, CoefVHcb, CoefVHcd As Double
Dim FactorMab, FactorMcb, FactorMcd, FactorMed As Double
```

```
h = L / 2 * Tan(Alfa)
L = Val(Text1.Text)
```

```
FactorMed = LI * Sin(Alfa) / (Lm * L / 2)
FactorMcd = 2 / L
FactorMab = FactorMed
FactorMcb = FactorMcd
```

```
CoefHcd = (MatrizDatos(4, 1) + MatrizDatos(4, 2) + MatrizDatos(4, 7)) * Lm / 2
```

$$\begin{aligned} \text{CoefHcd} &= \text{CoefHcd} + (\text{MatrizDatos}(4, 3) * (\text{Lm} - \text{MatrizDatos}(4, 4)) + \\ &\text{MatrizDatos}(4, 5) * (\text{Lm} - \text{MatrizDatos}(4, 6))) / \text{Lm} \\ \text{CoefHcd} &= \text{CoefHcd} + ((\text{MatrizDatos}(3, 1) + \text{MatrizDatos}(3, 2) + \text{MatrizDatos}(3, 7)) \\ &* \text{LI} / 2) * \text{Sin}(\text{Alfa}) \\ \text{CoefHcd} &= \text{CoefHcd} + (\text{MatrizDatos}(3, 3) * (\text{LI} - \text{MatrizDatos}(3, 4)) + \text{MatrizDatos}(3, \\ &5) * (\text{LI} - \text{MatrizDatos}(3, 6))) * \text{Sin}(\text{Alfa}) / \text{LI} \\ &'////////// \\ \text{CoefVcd} &= ((\text{MatrizDatos}(3, 1) + \text{MatrizDatos}(3, 2) + \text{MatrizDatos}(3, 7)) * \text{LI} ^ 2 / 2) \\ &* 2 / \text{L} \\ \text{CoefVcd} &= \text{CoefVcd} + (\text{MatrizDatos}(3, 3) * (\text{LI} - \text{MatrizDatos}(3, 4)) + \text{MatrizDatos}(3, \\ &5) * (\text{LI} - \text{MatrizDatos}(3, 6))) * 2 / \text{L} \\ &'////////// \\ \text{CoefHcb} &= (\text{MatrizDatos}(1, 1) + \text{MatrizDatos}(1, 2) + \text{MatrizDatos}(1, 7)) * \text{Lm} / 2 \\ \text{CoefHcb} &= \text{CoefHcb} + (\text{MatrizDatos}(1, 3) * \text{MatrizDatos}(1, 4) + \text{MatrizDatos}(1, 5) * \\ &\text{MatrizDatos}(1, 6)) / \text{Lm} \\ \text{CoefHcb} &= \text{CoefHcb} + (\text{MatrizDatos}(2, 1) + \text{MatrizDatos}(2, 2) + \text{MatrizDatos}(2, 7)) * \\ &\text{LI} * \text{Sin}(\text{Alfa}) / 2 \\ \text{CoefHcb} &= \text{CoefHcb} + (\text{MatrizDatos}(2, 3) * \text{MatrizDatos}(2, 4) + \text{MatrizDatos}(2, 5) * \\ &\text{MatrizDatos}(2, 6)) * \text{Sin}(\text{Alfa}) / \text{LI} \\ &'////////// \\ \text{CoefVcb} &= ((\text{MatrizDatos}(2, 1) + \text{MatrizDatos}(2, 2) + \text{MatrizDatos}(2, 7)) * \text{LI} ^ 2 / 2) \\ &* 2 / \text{L} \\ \text{CoefVcb} &= \text{CoefVcb} + (\text{MatrizDatos}(2, 3) * \text{MatrizDatos}(2, 4) + \text{MatrizDatos}(2, 5) * \\ &\text{MatrizDatos}(2, 6)) * 2 / \text{L} \\ &'////////// \\ \text{CoefVHcd} &= -\text{CoefVcd} + \text{CoefHcd} * \text{LI} * \text{Sin}(\text{Alfa}) * 2 / \text{L} \\ \text{CoefVHcb} &= \text{CoefVcb} - \text{CoefHcb} * \text{LI} * \text{Sin}(\text{Alfa}) * 2 / \text{L} \\ &'////////// \end{aligned}$$

$$\begin{aligned} \text{MatrizA}(1, 1) &= \text{E1} + \text{C2} \\ \text{MatrizA}(1, 2) &= \text{D2} \\ \text{MatrizA}(1, 3) &= 0 \\ \text{MatrizA}(1, 4) &= \text{G1} - \text{F2} * \text{L} / (4 * \text{h}) \\ \text{MatrizA}(1, 5) &= \text{F2} * \text{L} / (4 * \text{h}) \\ \text{MatrizA}(1, 6) &= -(\text{B1} + \text{A2}) \end{aligned}$$

$$\begin{aligned} \text{MatrizA}(2, 1) &= \text{D2} \\ \text{MatrizA}(2, 2) &= \text{E2} + \text{C3} \\ \text{MatrizA}(2, 3) &= \text{D3} \\ \text{MatrizA}(2, 4) &= (\text{F3} - \text{G2}) * \text{L} / (4 * \text{h}) \\ \text{MatrizA}(2, 5) &= (\text{G2} - \text{F3}) * \text{L} / (4 * \text{h}) \\ \text{MatrizA}(2, 6) &= -(\text{B2} + \text{A3}) \end{aligned}$$

$$\begin{aligned} \text{MatrizA}(3, 1) &= 0 \\ \text{MatrizA}(3, 2) &= \text{D3} \\ \text{MatrizA}(3, 3) &= \text{E3} + \text{C4} \end{aligned}$$

$$\text{MatrizA}(3, 4) = G3 * L / (4 * h)$$

$$\text{MatrizA}(3, 5) = F4 - G3 * L / (4 * h)$$

$$\text{MatrizA}(3, 6) = -(B3 + A4)$$

$$\text{MatrizA}(4, 1) = D1 + E1$$

$$\text{MatrizA}(4, 2) = 0$$

$$\text{MatrizA}(4, 3) = C4 + D4$$

$$\text{MatrizA}(4, 4) = F1 + G1$$

$$\text{MatrizA}(4, 5) = F4 + G4$$

$$\text{MatrizA}(4, 6) = -(A1 + B1 + A4 + B4 + Lm * (\text{CoefHcd} - \text{CoefHcb}))$$

$$\text{MatrizA}(5, 1) = \text{FactorMab} * (D1 + E1) - \text{FactorMcb} * (C2 + D2)$$

$$\text{MatrizA}(5, 2) = -\text{FactorMcb} * (D2 + E2) + \text{FactorMcd} * (C3 + D3)$$

$$\text{MatrizA}(5, 3) = -\text{FactorMed} * (C4 + D4) + \text{FactorMcd} * (D3 + E3)$$

$$\text{MatrizA}(5, 4) = \text{FactorMab} * (F1 + G1) + \text{FactorMcb} * (F2 + G2) * L / (4 * h) + \text{FactorMcd} * (F3 + G3) * L / (4 * h)$$

$$\text{MatrizA}(5, 5) = -\text{FactorMcb} * (F2 + G2) * L / (4 * h) - \text{FactorMed} * (F4 + G4) - \text{FactorMcd} * (F3 + G3) * L / (4 * h)$$

$$\text{MatrizA}(5, 6) = -(\text{CoefVHcb} + \text{FactorMab} * (A1 + B1) - \text{FactorMcb} * (A2 + B2) - \text{CoefVHcd} - \text{FactorMed} * (A4 + B4) + \text{FactorMcd} * (A3 + B3))$$

End Sub

---

Private Sub CONV\_COEF\_EN\_LETRAS()

$$A1 = \text{LeTras}(1, 1)$$

$$A2 = \text{LeTras}(2, 1)$$

$$A3 = \text{LeTras}(3, 1)$$

$$A4 = \text{LeTras}(4, 1)$$

$$B1 = \text{LeTras}(1, 2)$$

$$B2 = \text{LeTras}(2, 2)$$

$$B3 = \text{LeTras}(3, 2)$$

$$B4 = \text{LeTras}(4, 2)$$

$$C1 = \text{LeTras}(1, 3)$$

$$C2 = \text{LeTras}(2, 3)$$

$$C3 = \text{LeTras}(3, 3)$$

$$C4 = \text{LeTras}(4, 3)$$

$$D1 = \text{LeTras}(1, 4)$$

$$D2 = \text{LeTras}(2, 4)$$

$$D3 = \text{LeTras}(3, 4)$$

$$D4 = \text{LeTras}(4, 4)$$

$$E1 = \text{LeTras}(1, 5)$$

$$E2 = \text{LeTras}(2, 5)$$

E3 = LeTras(3, 5)  
E4 = LeTras(4, 5)

F1 = LeTras(1, 6)  
F2 = LeTras(2, 6)  
F3 = LeTras(3, 6)  
F4 = LeTras(4, 6)

G1 = LeTras(1, 7)  
G2 = LeTras(2, 7)  
G3 = LeTras(3, 7)  
G4 = LeTras(4, 7)

End Sub

---

```
Private Sub revisarDatos(buenos As Boolean)
If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or
Text5.Text = "" Then
buenos = False
End If
End Sub
```

---

```
Private Sub CalcularMom()
Dim K1, K2, K3, K4, K5 As Double
```

```
Call tomar_datos(Lon, Hac, Hbc, Xa, Xb, d, bf, tf, tw, P1, P2, a, b)
Call Calcular_Kas(K1, K2, K3, K4, K5)
```

$$Mab = (K3 * K4 - K1 * K5) * Lon / (K3^2 - K1 * K2)$$

$$Mba = (K2 * K4 - K3 * K5) * Lon / (K3^2 - K1 * K2)$$

```
MomentoS(elemento, 1) = Mab + MomentoS(elemento, 1)
MomentoS(elemento, 2) = Mba + MomentoS(elemento, 2)
```

```
' AQUI SE INTRODUCEN LOS COEFICIENTES "LETRAS" A LA MATRIZ
```

```
' COEFICIENTE A
```

```
LeTras(elemento, 1) = Mab
```

```
' COEFICIENTE B
```

```
LeTras(elemento, 2) = Mba
```

```
' COEFICIENTE C
```

```
LeTras(elemento, 3) = E * K1 * Lon ^ 2 / (K3 ^ 2 - K1 * K2)
```

```
' COEFICIENTE D
```

```
LeTras(elemento, 4) = E * K3 * Lon ^ 2 / (K3 ^ 2 - K1 * K2)
```

```
' COEFICIENTE E
```

```
LeTras(elemento, 5) = E * K2 * Lon ^ 2 / (K3 ^ 2 - K1 * K2)
```

' COEFICIENTE F

LeTras(elemento, 6) =  $(K1 + K3) * E * Lon / (K1 * K2 - K3 ^ 2)$

' COEFICIENTE G

LeTras(elemento, 7) =  $(K2 + K3) * E * Lon / (K1 * K2 - K3 ^ 2)$

End Sub

---

Private Sub Calcular\_Kas(K1, K2, K3, K4, K5 As Double)

Dim n As Long

Dim X As Double

K1 = 0

K2 = 0

K3 = 0

K4 = 0

K5 = 0

Progreso.Min = 1

Progreso.Max = Divisiones

For n = 1 To Divisiones

X = n \* (Lon / Divisiones) - (Lon / (2 \* Divisiones))

K1 = K1 +  $(X ^ 2 / Inercia(X))$

K2 = K2 +  $(Lon - X) ^ 2 / Inercia(X)$

K3 = K3 +  $X * (Lon - X) / Inercia(X)$

K4 = K4 +  $Miu(X) * X / Inercia(X)$

K5 = K5 +  $Miu(X) * (Lon - X) / Inercia(X)$

Progreso.Value = n

Next

K1 = K1 \* (Lon / Divisiones)

K2 = K2 \* (Lon / Divisiones)

K3 = K3 \* (Lon / Divisiones)

K4 = K4 \* (Lon / Divisiones)

K5 = K5 \* (Lon / Divisiones)

Progreso.Value = 1

End Sub

---

Function Inercia(X As Double) As Double

Dim A1, A2, A3, A4, A5, D1, D2, D3, D4, D5, I1, I2, I3, I4, I5, tfi, h, Area, Sumatoria

As Double

Dim SumarA4 As Boolean

SumarA4 = True

If X >= 0 And X <= Xa Then

h = Hac \* (1 - X / Xa)

tfi = tf \*  $Sqr(Xa ^ 2 + Hac ^ 2) / Xa$

```

End If
If X > Xa And X <= (Lon - Xb) Then
  h = 0
  SumarA4 = False
  A4 = 0
  A5 = 0
End If
If X > (Lon - Xb) And X <= Lon Then
  h = Abs(Hbc * (Lon - Xb - X) / Xb)
  tfi = tf * Sqr(Xa ^ 2 + Hac ^ 2) / Xb
End If

If h <= tfi Then
  tfi = h
  SumarA4 = False
  A4 = 0
End If

A1 = bf * tf
A2 = tw * (d - 2 * tf)
A3 = bf * tf

If SumarA4 = True Then
  A4 = tw * (h - tfi)
End If

If h > 0 Then
  A5 = bf * tfi
End If

D1 = (h + d - tf / 2)
D2 = h + d / 2
D3 = h + tf / 2
D4 = tfi + (h - tfi) / 2
D5 = tfi / 2

Area = A1 + A2 + A3 + A4 + A5
Sumatoria = A1 * D1 + A2 * D2 + A3 * D3 + A4 * D4 + A5 * D5
Y = Sumatoria / Area

I1 = bf * tf ^ 3 / 12
I2 = tw * (d - 2 * tf) ^ 3 / 12
I3 = bf * tf ^ 3 / 12
If SumarA4 = True Then
  I4 = tw * (h - tfi) ^ 3 / 12
Else
  I4 = 0

```

```

I5 = 0
End If
If h > 0 Then
  I5 = bf * tfi ^ 3 / 12
End If
Inercia = I1 + A1 * (D1 - Y) ^ 2
Inercia = I2 + A2 * (D2 - Y) ^ 2 + Inercia
Inercia = I3 + A3 * (D3 - Y) ^ 2 + Inercia
Inercia = I4 + A4 * (D4 - Y) ^ 2 + Inercia
Inercia = I5 + A5 * (D5 - Y) ^ 2 + Inercia

End Function

```

---

```

Function Miu(X As Double) As Double

```

```

If carga = 3 Or carga = 4 Then
  ' !!!!!Carga puntual!!!!
  If X > 0 And X <= a Then
    Miu = P1 * (Lon - a) * X / Lon
  End If
  If X > a And X < Lon Then
    Miu = P1 * a * (1 - X / Lon)
  End If
End If

If carga = 1 Or carga = 2 Or carga = 5 Then
  ' !!!!!Carga distribuida uniforme!!!!
  Miu = P1 * X * (Lon - X) / 2
End If
End Function

```

---

```

Private Sub tomar_datos(Lon, Hac, Hbc, Xa, Xb, d, bf, tf, tw, P1, P2, a, b As
Double)

```

```

  Lon = MatrizDatos(elemento, 12)
  Hac = MatrizDatos(elemento, 13)
  Hbc = MatrizDatos(elemento, 14)
  Xa = MatrizDatos(elemento, 15)
  Xb = MatrizDatos(elemento, 16)

  d = MatrizDatos(elemento, 8) / 1000
  bf = MatrizDatos(elemento, 9) / 1000
  tf = MatrizDatos(elemento, 10) / 1000
  tw = MatrizDatos(elemento, 11) / 1000

```

```
Select Case carga
Case 1
  P1 = MatrizDatos(elemento, 1)
Case 2
  P1 = MatrizDatos(elemento, 2)
Case 3
  P1 = MatrizDatos(elemento, 3)
  a = MatrizDatos(elemento, 4)
Case 4
  P1 = MatrizDatos(elemento, 5)
  a = MatrizDatos(elemento, 6)
Case 5
  P1 = MatrizDatos(elemento, 7)
End Select
```

```
End Sub
```

---

```
Private Sub IntroDatPort()
Dim de, Hbc, teefe As Double
Lm = Val(Text4.Text)
L = Val(Text1.Text)
```

```
If Val(Text2.Text) > 0 Then
Alfa = Val(Text2.Text) * Pi / 180
Else
Alfa = 0.000000001
End If
LI = L / (2 * Cos(Alfa))
MatrizDatos(1, 12) = Lm
MatrizDatos(4, 12) = Lm
MatrizDatos(2, 12) = LI
MatrizDatos(3, 12) = LI
Xcart = Val(Text3.Text)
de = MatrizDatos(2, 8)
teefe = MatrizDatos(2, 10)
```

```
If Xcart = 0 Then
Xb = 0
Else
```

```
Gamma = Atn(((de - teefe) / 1000) * Cos(Alfa) / Xcart)
Xb = ((de - teefe) / 1000) * Cos(Gamma) / Sin((Pi / 2) - Alfa - Gamma)
End If
```

```
If Alfa = 0 Then
```

```

Hbc = Xcart
Else
Hbc = Xb * Tan(Pi / 2 - Alfa - Gamma)
End If

' Hac
MatrizDatos(1, 13) = 0
MatrizDatos(2, 13) = de + (de - teefe)
MatrizDatos(3, 13) = de + ((de - teefe) / Cos(Alfa))
MatrizDatos(4, 13) = Hbc * 1000

' Hbc
MatrizDatos(1, 14) = Hbc * 1000
MatrizDatos(2, 14) = de + ((de - teefe) / Cos(Alfa))
MatrizDatos(3, 14) = de + (de - teefe)
MatrizDatos(4, 14) = 0

' Xa
  MatrizDatos(1, 15) = 0
  MatrizDatos(2, 15) = Xcart / Cos(Alfa)
If Alfa > 0 Then
  If (de - teefe) / Tan(Alfa) >= L / 4 Then
    MatrizDatos(3, 15) = 0
  Else
    MatrizDatos(3, 15) = (de - teefe) / Sin(Alfa)
  End If
End If
If Alfa = 0 Then
  MatrizDatos(3, 15) = 0
End If
  MatrizDatos(4, 15) = Xb

' Xb
  MatrizDatos(1, 16) = Xb
If Alfa > 0 Then
  If (de - teefe) / Tan(Alfa) >= L / 4 Then
    MatrizDatos(2, 16) = 0
  Else
    MatrizDatos(2, 16) = (de - teefe) / Sin(Alfa)
  End If
End If
If Alfa = 0 Then
  MatrizDatos(2, 16) = 0
End If

MatrizDatos(3, 16) = Xcart / Cos(Alfa)
MatrizDatos(4, 16) = 0

```

End Sub

---

```
Private Sub Command1_Click()  
ElemProp.Tag = 1  
ElemProp.Caption = "Propiedades del elemento 1"  
ElemProp.Show 1  
End Sub
```

---

```
Private Sub Command2_Click()  
ElemProp.Tag = 2  
ElemProp.Caption = "Propiedades del elemento 2"  
ElemProp.Show 1  
End Sub
```

---

```
Private Sub Command3_Click()  
ElemProp.Tag = 3  
ElemProp.Caption = "Propiedades del elemento 3"  
ElemProp.Show 1  
End Sub
```

---

```
Private Sub Command4_Click()  
ElemProp.Tag = 4  
ElemProp.Caption = "Propiedades del elemento 4"  
ElemProp.Show 1  
End Sub
```

---

```
Private Sub Command5_Click()  
VyN.Show 1  
End Sub
```

---

```
Private Sub EyC_Click()  
DB_Perfiles.Show 1  
End Sub
```

---

```
Private Sub Form_Load()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
Text5.Text = "2.1"  
Nm = 5  
End Sub
```

---

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
```

```
Dim Msg      ' Declara la variable.
If UnloadMode > 0 Then
    ' Si sale de la aplicación.
    Msg = "¿Realmente desea salir de la aplicación?"
Else
    ' Si sólo se cierra el formulario.
    Msg = "¿Realmente desea salir de la aplicación?" "" ¿Realmente desea cerrar el
formulario?"
End If
' Si el usuario hace clic en el botón No, se detiene QueryUnload.
If MsgBox(Msg, vbQuestion + vbYesNo, Me.Caption) = vbNo Then Cancel =
True
End Sub
```

---

```
Private Sub menuAcerca_Click()
Acercade.Show 1
End Sub
```

---

```
Private Sub menuCart_Click()
tablas.Show 1
End Sub
```

---

```
Private Sub menuGenerar_Click()
GeneraTabla.Show 1
End Sub
```

---

```
Private Sub menuMom_Click()
Mempot.Show 1
End Sub
```

---

```
Private Sub menuSalir_Click()

    If MsgBox("¿Realmente desea salir de la aplicación?", vbQuestion + vbYesNo,
Me.Caption) = vbNo Then
        Else
            End
        End If
    End Sub
```

---

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
    Case Asc("0") To Asc("9")
        If KeyAscii = Asc("0") Then
            If Text1.Text = "" Then
                KeyAscii = 0
            End If
        End If
    End Case
End Sub
```

```
End If
Case 8
Case Asc(".")
  If InStr(Text1.Text, ".") Then
    KeyAscii = 0
  End If
Case Else: KeyAscii = 0
End Select
End Sub
```

---

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
  Case Asc("0") To Asc("9")
    If KeyAscii = Asc("0") Then
      If Text2.Text = "0" Then
        KeyAscii = 0
      End If
    End If
    If Text2.Text = "0" Then
      KeyAscii = 0
    End If
  Case 8
  Case Asc(".")
    If InStr(Text2.Text, ".") Then
      KeyAscii = 0
    End If
  Case Else: KeyAscii = 0
End Select
End Sub
```

---

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
  Case Asc("0") To Asc("9")
    If KeyAscii = Asc("0") Then
      If Text3.Text = "0" Then
        KeyAscii = 0
      End If
    End If
    If Text3.Text = "0" Then
      KeyAscii = 0
    End If
  Case 8
  Case Asc(".")
    If InStr(Text3.Text, ".") Then
      KeyAscii = 0
    End If
  Case Else: KeyAscii = 0
End Select
End Sub
```

```
Case Else: KeyAscii = 0
End Select
End Sub
```

---

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
Case Asc("0") To Asc("9")
If KeyAscii = Asc("0") Then
If Text4.Text = "" Then
KeyAscii = 0
End If
End If
Case 8
Case Asc(".")
If InStr(Text4.Text, ".") Then
KeyAscii = 0
End If
Case Else: KeyAscii = 0
End Select
End Sub
```

---

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
Case Asc("0") To Asc("9")
If KeyAscii = Asc("0") Then
If Text5.Text = "0" Then
KeyAscii = 0
End If
End If
If Text5.Text = "0" Then
KeyAscii = 0
End If
Case 8
Case Asc(".")
If InStr(Text5.Text, ".") Then
KeyAscii = 0
End If
Case Else: KeyAscii = 0
End Select
End Sub
```

### VII.1.1.2 Código fuente del Form Mempot

La forma Mempot se encarga de calcular los momentos de empotramiento debido a las cargas aplicadas en el elemento, a partir de las variables posibles para el cálculo, tales como la luz de la viga, tipo de perfil y dimensiones de la cartela. Se tiene la opción de escoger entre carga distribuida uniforme y carga puntual.

Datos de entrada

L: luz del galpón.  
 Hac: altura de la cartela en A.  
 Hbc: altura de la cartela en B.  
 Xa: longitud horizontal de la cartela A.  
 Xb: longitud horizontal de la cartela B.

```
Dim db As Database
Dim td As TableDef
Dim tablaabierta As Recordset
Dim Mab, Mba, L, Hac, Hbc, Xa, Xb, d, bf, tf, tw, P1, P2, a, b As Double
```

---

```
Private Sub BdD_Click()
```

```
propiedades.Caption = "Escoja perfil"
```

```
Tipo.Visible = True
```

```
Denom.Visible = True
```

```
    dText.Enabled = 0
```

```
    bfText.Enabled = 0
```

```
    tfText.Enabled = 0
```

```
    twText.Enabled = 0
```

```
cargardb
```

```
,
```

```
    'PROCEDIMIENTO PARA LLENAR EL COMBO BOX CON LAS  

    'TABLAS QUE EXISTAN EN LA BASE DE DATOS
```

```
,
```

```
Tipo.Clear
```

```
On Error GoTo cancelaerror
```

```
Set db = OpenDatabase(App.Path + "\" + "PerfAcero.mdb")
```

```
cancelaerror:
```

```

If Err.Number = 3024 Then
Resume Next
End If
'ExisteTabla = False
On Error GoTo cancela
For Each td In db.TableDefs
'Sólo las tablas con atributo igual a CERO,
'son tablas normales
If td.Attributes = 0 Then
Tipo.AddItem td.Name
'If td.Name = nombreperfil1.Text Then
'ExisteTabla = True
'End If
End If
Next
db.Close
Set db = Nothing
cancela:
If Err.Number = 91 Then
Resume Next
End If
ReSort Tipo
Tipo.Text = Tipo.List(0)
Denom.Text = Denom.List(0)
End Sub

```

---

```

Sub cargardb()
If FileExists(App.Path + "\PerfAcero.mdb") Then
'La base de datos existe, se puede abrir para modificar
Ruta = App.Path + "\PerfAcero.mdb"
Set db = OpenDatabase(Ruta)
MsgBox "Base de datos cargada", vbInformation, "Aviso"
Else
'crear base de datos
'Set MiWorkspace = DBEngine.Workspaces(0)
'Ruta = App.Path + "\PerfAcero.mdb"
'Set db = MiWorkspace.CreateDatabase(Ruta, dbLangGeneral, dbVersion30)
'Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla: "
'Set db = OpenDatabase(Ruta)
MsgBox "La Base de datos no existe", vbExclamation, "Alerta"
MsgBox "Debe crear una base de datos para perfiles, denominada --PerfAcero--
o introducir los datos para el cálculo", vbInformation, "Solución"
Usuario.Value = True
Usuario.SetFocus
'nombreperfil1.Enabled = True
'Crear.Enabled = True

```

```
'Frame1.Enabled = True
'Tipo.SetFocus
End If
End Sub
```

---

```
Private Sub Calcular_Click()
Dim K1, K2, K3, K4, K5 As Double

Call tomar_datos(L, Hac, Hbc, Xa, Xb, d, bf, tf, tw, P1, P2, a, b)
Call Calcular_Kas(K1, K2, K3, K4, K5)

Mab = (K3 * K4 - K1 * K5) * L / (K3 ^ 2 - K1 * K2)
Mba = (K2 * K4 - K3 * K5) * L / (K3 ^ 2 - K1 * K2)

MabLabel.Caption = "Mab= " + Str(Mab)
MbaLabel.Caption = "Mba= " + Str(Mba)

End Sub
```

---

```
Private Sub Calcular_Kas(K1, K2, K3, K4, K5 As Double)
Dim n As Long
Dim X As Double
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
Progreso.Min = 1
Progreso.Max = Divisiones

For n = 1 To Divisiones
X = n * (L / Divisiones) - (L / (2 * Divisiones))
K1 = K1 + (X ^ 2 / Inercia(X))
K2 = K2 + (L - X) ^ 2 / Inercia(X)
K3 = K3 + X * (L - X) / Inercia(X)
K4 = K4 + Miu(X) * X / Inercia(X)
K5 = K5 + Miu(X) * (L - X) / Inercia(X)
Progreso.Value = n
Next
K1 = K1 * (L / Divisiones)
K2 = K2 * (L / Divisiones)
K3 = K3 * (L / Divisiones)
K4 = K4 * (L / Divisiones)
K5 = K5 * (L / Divisiones)
Progreso.Value = 1
```

End Sub

---

Function Inercia(X As Double) As Double

Dim A1, A2, A3, A4, A5, D1, D2, D3, D4, D5, I1, I2, I3, I4, I5, tfi, h, Area, Sumatoria  
As Double

Dim SumarA4 As Boolean

SumarA4 = True

```

If X >= 0 And X <= Xa Then
  h = Hac * (1 - X / Xa)
  tfi = tf * Sqr(Xa ^ 2 + Hac ^ 2) / Xa
End If
If X > Xa And X <= (L - Xb) Then
  h = 0
  SumarA4 = False
  A4 = 0
  A5 = 0
End If
If X > (L - Xb) And X <= L Then
  h = Abs(Hbc * (L - Xb - X) / Xb)
  tfi = tf * Sqr(Xa ^ 2 + Hac ^ 2) / Xb
End If

```

If h <= tfi Then

```

tfi = h
SumarA4 = False
A4 = 0
End If

```

```

A1 = bf * tf
A2 = tw * (d - 2 * tf)
A3 = bf * tf

```

```

If SumarA4 = True Then
  A4 = tw * (h - tfi)
End If

```

```

If h > 0 Then
  A5 = bf * tfi
End If

```

D1 = (h + d - tf / 2)

$$D2 = h + d / 2$$

$$D3 = h + tf / 2$$

$$D4 = tfi + (h - tfi) / 2$$

$$D5 = tfi / 2$$

$$\text{Area} = A1 + A2 + A3 + A4 + A5$$

$$\text{Sumatoria} = A1 * D1 + A2 * D2 + A3 * D3 + A4 * D4 + A5 * D5$$

$$Y = \text{Sumatoria} / \text{Area}$$

$$I1 = bf * tf^3 / 12$$

$$I2 = tw * (d - 2 * tf)^3 / 12$$

$$I3 = bf * tfi^3 / 12$$

If SumarA4 = True Then

$$I4 = tw * (h - tf)^3 / 12$$

Else

$$I4 = 0$$

$$I5 = 0$$

End If

If h > 0 Then

$$I5 = bf * tfi^3 / 12$$

End If

$$\text{Inercia} = I1 + A1 * (D1 - Y)^2$$

$$\text{Inercia} = I2 + A2 * (D2 - Y)^2 + \text{Inercia}$$

$$\text{Inercia} = I3 + A3 * (D3 - Y)^2 + \text{Inercia}$$

$$\text{Inercia} = I4 + A4 * (D4 - Y)^2 + \text{Inercia}$$

$$\text{Inercia} = I5 + A5 * (D5 - Y)^2 + \text{Inercia}$$

End Function

---

Function Miu(X As Double) As Double

Dim PendQ, Fact, Av, Bv, Pb As Double

If Option2.Value = True And Option3.Value = True Then

' !!!!!Carga puntual!!!!'

If X >= 0 And X <= a Then

$$\text{Miu} = P1 * (L - a) * X / L$$

End If

If X > a And X < L Then

$$\text{Miu} = P1 * a * (1 - X / L)$$

End If

End If

If Option1.Value = True And Option3.Value = True Then

' !!!!!Carga distribuida uniforme!!!!'

```
Miu = P1 * X * (L - X) / 2  
End If
```

```
End Function
```

---

```
Private Sub tomar_datos(L, Hac, Hbc, Xa, Xb, d, bf, tf, tw, P1, P2, a, b As Double)
```

```
    'AQUI SE PODRIAN VALIDAR LOS DATOS  
    '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
    L = Val(LText.Text)  
    Hac = Val(HacText.Text) / 1000  
    Hbc = Val(HbcText.Text) / 1000  
    Xa = Val(XaText.Text)  
    Xb = Val(XbText.Text)
```

```
    d = Val(dText.Text) / 1000  
    bf = Val(bfText.Text) / 1000  
    tf = Val(tfText.Text) / 1000  
    tw = Val(twText.Text) / 1000
```

```
    P1 = Val(P1text.Text)  
    P2 = Val(P2text.Text)  
    a = Val(aText.Text)  
    b = Val(bText.Text)
```

```
End Sub
```

---

```
Private Sub cont_Click()  
Unload Memptot  
End Sub
```

---

```
Private Sub Denom_Click()  
Set db = OpenDatabase(App.Path + "\" + "PerfAcero.mdb")  
  
Set tablaabierta = db.OpenRecordset(Tipo.Text, dbOpenDynaset)  
tablaabierta.FindFirst "designacion=" & Str(Denom.ItemData(Denom.ListIndex))  
    dText.Text = tablaabierta!d  
    bfText.Text = tablaabierta!bf  
    tfText.Text = tablaabierta!tf  
    twText.Text = tablaabierta!tw  
End Sub
```

---

```
Private Sub Drive1_Change()  
  
On Error GoTo ManejadorError  
Dir1.Path = Drive1.Drive
```

ManejadorError:

```
If Err.Number = 68 Then
    resp = MsgBox("El dispositivo no está preparado", vbAbortRetryIgnore)
    If resp = vbAbort Then
        Drive1.Drive = Dir1.Path
    End If
    If resp = vbRetry Then
        Resume
    End If
    If resp = vbIgnore Then
        Resume Next
    End If
End If
End Sub
```

---

```
Private Sub Form_Load()
'Divisiones = 1000
Option1.Value = True
Option3.Value = True
Usuario.Value = True
Tipo.Visible = False
Denom.Visible = False
Image3.Visible = False
Image1.Picture = LoadPicture(App.Path + "\distribuida.jpg")

End Sub
```

---

```
Private Sub Image2_Click()
Frame1.Visible = False
Frame2.Visible = False
Frame4.Visible = False
Calcular.Visible = False
cont.Visible = False
Image3.Visible = True
Progreso.Visible = False
End Sub
```

---

```
Private Sub Image3_Click()
Frame1.Visible = True
Frame2.Visible = True
Frame4.Visible = True
Calcular.Visible = True
cont.Visible = True
Image3.Visible = False
Progreso.Visible = True
```

End Sub

---

```
Private Sub Option1_Click()
Option4.Visible = False
Option3.Value = True
Option3.Caption = "Uniforme"
Option4.Caption = "Trapezoidal"
bText.Visible = False

' For i = 12 To 13
' Label4(i).Visible = True
' Next
For i = 12 To 17
Label4(i).Visible = False
Next
Image1.Picture = LoadPicture(App.Path + "\distribuidatotal.jpg")
aText.Visible = False
' bText.Visible = True
P2text.Visible = False
```

End Sub

---

```
Private Sub Option2_Click()
Option4.Visible = True
Option3.Value = True
Option3.Caption = "Una sola carga"
Option4.Caption = "Dos cargas a 1/3 de L"
For i = 14 To 17
Label4(i).Visible = False
Next
For i = 12 To 13
Label4(i).Visible = True
Next
Image1.Picture = LoadPicture(App.Path + "\puntual.jpg")

aText.Visible = True
P2text.Visible = False
bText.Visible = False
```

End Sub

---

```
Private Sub Option3_Click()
If Option1.Value = True Then
P2text.Visible = False
Label4(14).Visible = False
Label4(15).Visible = False
Image1.Picture = LoadPicture(App.Path + "\distribuida.jpg")
```

End If

```
If Option2.Value = True Then
  For i = 14 To 17
    Label4(i).Visible = False
  Next
  For i = 12 To 13
    Label4(i).Visible = True
  Next
  aText.Visible = True
  P2text.Visible = False
  bText.Visible = False
  Image1.Picture = LoadPicture(App.Path + "\puntual.jpg")
End If
End Sub
```

---

```
Private Sub Option4_Click()
  If Option1.Value = True Then
    P2text.Visible = True
    Label4(14).Visible = True
    Label4(15).Visible = True
    Image1.Picture = LoadPicture(App.Path + "\trapezoidal.jpg")
  End If
```

```
If Option2.Value = True Then
  For i = 12 To 17
    Label4(i).Visible = False
  Next
  aText.Visible = False
  bText.Visible = False
  P2text.Visible = False
  Image1.Picture = LoadPicture(App.Path + "\untercio.jpg")
End If
```

End Sub

---

```
Private Sub Tipo_Change()
  Denom.Clear
  llenardetablas
End Sub
```

---

```
Private Sub Tipo_Click()
  Denom.Clear
  llenardetablas
End Sub
```

---

```
Private Sub llenardetablas()  
  
Set db = OpenDatabase(App.Path + "\PerfAcero.mdb")  
Set tablaabierta = db.OpenRecordset(Tipo.Text, dbOpenDynaset)  
  
If Not tablaabierta.EOF Then tablaabierta.MoveFirst  
Do While Not tablaabierta.EOF  
    Denom.AddItem tablaabierta!Designacion  
    Denom.ItemData(Denom.NewIndex) = Str(tablaabierta!Designacion)  
    tablaabierta.MoveNext  
Loop  
db.Close  
Set db = Nothing  
ReSort Denom  
Denom.ListIndex = 0  
End Sub
```

---

```
Private Sub Usuario_Click()  
propiedades.Caption = "Propiedades del perfil"  
Tipo.Visible = False  
Denom.Visible = False  
    dText.Enabled = 1  
    bfText.Enabled = 1  
    tfText.Enabled = 1  
    twText.Enabled = 1  
Denom.Text = Denom.List(0)  
    dText.Text = 1  
    bfText.Text = 1  
    tfText.Text = 1  
    twText.Text = 1  
  
End Sub
```

### **VII.1.1.3 Código fuente del Form DB\_Perfiles**

La forma DB\_Perfiles está diseñada para proveer información acerca de las propiedades de los perfiles laminados más comunes, además de poder crear, modificar o leer una base de datos, con las características de cada tipo de perfil y su designación.

```
Dim mydb As Database
Dim MiWorkspace As Workspace
Dim Mitabla As TableDef
Dim Ruta As String
Dim MisCampos(32) ' campos MisCampos(0) y MisCampos(1)
Dim tablaabierta As Recordset
```

---

```
Function FileExists(Filename As String) As Boolean
    Dim TempAttr As Integer
    If (Len(Filename) = 0) Or (InStr(Filename, "**") > 0) Or (InStr(Filename, "?") > 0)
    Then
        FileExists = False
        Exit Function
    End If
    On Error GoTo ErrorFileExist

    TempAttr = GetAttr(Filename)

    FileExists = ((TempAttr And vbDirectory) = 0)
    GoTo ExitFileExist
```

ErrorFileExist:

```
FileExists = False
Resume ExitFileExist
```

ExitFileExist:

```
On Error GoTo 0
End Function
```

---

Private Sub Agregar\_Click()

```
If valor.Text = "Valor" Then
    valor.Text = ""
Else
    If Caracteristica.ListIndex = -1 Then
        MsgBox "debe seleccionar un valor de caracteristica"
    Else
        If valor.Text = "" Then
            MsgBox "Debe colocar algun valor para ser agregado", vbExclamation,
            "Atención!"
        Else
            Select Case Caracteristica.ListIndex
            Case 0: registro
```

```
        Case 1: aregistro
        Case Else: verificarprop
    End Select
End If
End If
End If
labelregistro
valor = ""
End Sub
```

---

```
Sub verificarprop()
    If Propiedad.ListIndex = -1 Then
        MsgBox "debe seleccionar un valor en propiedad"
    Else
        If valor.Text = "" Then
            MsgBox "Debe colocar algun valor para ser agregado", vbExclamation,
                "Atención!"
        Else
            aregistro
        End If
    End If
End Sub
```

---

```
Private Sub AñadirDes_Click()
    Dim existeregistro As Boolean

    existeregistro = False
    If design = "Designación" Then
        design = ""
    Else
        For n = 0 To Designacion.ListCount - 1
            If Val(Designacion.ItemData(n)) = Val(design.Text) Then
                existeregistro = True
            End If
        Next n
    End If

    If existeregistro Then
        If design <> "" Then
            MsgBox "Este registro ya existe, debe colocar otra designación para un nuevo perfil", vbInformation, "Atención!"
        Else
            MsgBox "Debe colocar una designación para cargar el nuevo perfil",
                vbInformation, "Atención!"
        End If
    End If
```

```
Else
If design.Text = "Designación" Then
  design.Text = ""
Else
  If Val(design.Text) > 32767 Then
    MsgBox "Debe colocar un valor menor o igual a 32767", vbExclamation,
    "Atención!"
    design.Text = ""
  Else
    Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
    If design = "" Then
      MsgBox "Debe escribir un número de designación para el perfil",
      vbExclamation, "Atención!"
    Else
      tablaabierta.AddNew
      tablaabierta!Designacion = design.Text

      LlenarConCeros
      'aquí se llenan los demás campos de la matriz con ceros
      tablaabierta.Update
      Designacion.Clear
      Llenarlistas
      For n = 0 To Designacion.ListCount - 1
        If Designacion.ItemData(n) = design Then
          Designacion.ListIndex = n
        End If
      Next n
      labelregistro
    End If
  End If
End If
End Sub
```

---

```
Function existeregistro(a As Control, variable As Integer) As Boolean
  For n = 0 To a.ListCount
    If a.ItemData = variable Then
      existeregistro = True
    End If
  Next n
End Function
```

---

```
Private Sub BorrarDes_Click()
```

```
If MsgBox("Desea eliminar el registro para el perfil (" + nombreperfil1.Tag + "-" +  
Str(Designacion.ItemData(Designacion.ListIndex)) + ")", vbYesNo, "Atención!") =  
vbYes Then  
If Designacion.ListIndex = -1 Then  
MsgBox "Debe seleccionar un elemento de la lista", vbInformation, "Atención!"  
Else  
If Designacion.ListCount > 1 Then  
tablaabierta.Delete  
Designacion.Clear  
llenarlistas  
Else  
MsgBox "Debe existir al menos un elemento en la lista", vbInformation,  
"Atención!"  
End If  
End If  
End If  
End Sub
```

---

```
Private Sub Caracteristica_Click()  
  
Select Case Caracteristica.ListIndex  
  
Case 0  
desactivarprop  
Unidad.Caption = "[kgf/m]"  
Case 1  
desactivarprop  
Unidad.Caption = "[cm2]"  
Case 2  
activarprop  
Propiedad.AddItem "Altura (d)"  
Propiedad.AddItem "Alas (bf)"  
Propiedad.AddItem "Alas (tf)"  
Propiedad.AddItem "Alma (tw)"  
Propiedad.ListIndex = 0  
Case 3  
activarprop  
Propiedad.AddItem "Ix"  
Propiedad.AddItem "Sx"  
Propiedad.AddItem "rx"  
Propiedad.AddItem "Iy"  
Propiedad.AddItem "Sy"  
Propiedad.AddItem "ry"  
Propiedad.ListIndex = 0  
Case 4  
activarprop
```

```
Propiedad.AddItem "rt"  
Propiedad.AddItem "d/Af"  
Propiedad.ListIndex = 0  
Case 5  
  activarprop  
  Propiedad.AddItem "Alas"  
  Propiedad.AddItem "Alma"  
  Propiedad.ListIndex = 0  
Case 6  
  activarprop  
  Propiedad.AddItem "J"  
  Propiedad.AddItem "Cw"  
  Propiedad.ListIndex = 0  
Case 7  
  activarprop  
  Propiedad.AddItem "Zx"  
  Propiedad.AddItem "Zy"  
  Propiedad.ListIndex = 0  
Case 8  
  activarprop  
  Propiedad.AddItem "a"  
  Propiedad.AddItem "T"  
  Propiedad.AddItem "k"  
  Propiedad.AddItem "k1"  
  Propiedad.AddItem "g1"  
  Propiedad.AddItem "c"  
  Propiedad.AddItem "m"  
  Propiedad.AddItem "n"  
  Propiedad.AddItem "g"  
  Propiedad.AddItem "g2"  
  Propiedad.ListIndex = 0  
End Select
```

End Sub

---

```
Sub desactivarprop()  
Propiedad.Clear  
Propiedad.Enabled = False  
End Sub
```

---

```
Sub activarprop()  
Propiedad.Clear  
Propiedad.Enabled = True  
End Sub
```

---

```
Private Sub cont_Click()
```

```
Unload DB_Perfiles  
End Sub
```

---

```
Private Sub Designacion_GotFocus()  
Caracteristica.ListIndex = 0  
End Sub
```

---

```
Private Sub Caracteristica_KeyPress(KeyAscii As Integer)  
KeyAscii = 0  
End Sub
```

---

```
Private Sub Cargar_Click()  
Dim db As Database  
Dim td As TableDef  
  
nombre.Tag = nombre  
Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla:"  
Frame1.Enabled = False  
cargardb
```

```
'  
'PROCEDIMIENTO PARA LLENAR EL COMBO BOX CON LAS  
'TABLAS QUE EXISTAN EN LA BASE DE DATOS  
'
```

```
nombreperfil1.Clear
```

```
Set db = OpenDatabase(App.Path + "\" + nombre.Tag + ".mdb")  
ExisteTabla = False  
For Each td In db.TableDefs  
Sólo las tablas con atributo igual a CERO,  
son tablas normales  
If td.Attributes = 0 Then  
nombreperfil1.AddItem td.Name  
If td.Name = nombreperfil1.Text Then  
ExisteTabla = True  
End If  
End If  
Next  
db.Close  
Set db = Nothing
```

```
ReSort nombreperfil1  
End Sub
```

---

```
Private Sub Crear_Click()
```

```

design.Tag = "primera vez"
nombreperfil1.Tag = nombreperfil1
If nombreperfil1 <> "" Then
    creatable
    'If tablaabierta.CacheSize <> 0 Then
    If Crear.Tag <> "existe" Then
        Designacion.Clear
        llenarlistas
    Else
        Designacion.Clear
        llenarlistas
    End If
Else
    MsgBox "Debe introducir el nombre del perfil", vbInformation, "Atención!"
End If

'
'VOLVER A LLENAR EL COMBO BOX CON LAS TABLAS
'QUE EXISTEN EN LA BASE DE DATOS

nombreperfil1.Clear

Set db = OpenDatabase(App.Path + "\" + nombre.Tag + ".mdb")
'ExisteTabla = False
For Each td In db.TableDefs
    'Sólo las tablas con atributo igual a CERO,
    'son tablas normales
    If td.Attributes = 0 Then
        nombreperfil1.AddItem td.Name
        'If td.Name = nombreperfil1.Text Then
        ' ExisteTabla = True
        'End If
    End If
Next
db.Close
Set db = Nothing
ReSort nombreperfil1
End Sub

```

---

```

Sub llenarlistas()
Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)

If Not tablaabierta.EOF Then tablaabierta.MoveFirst
Do While Not tablaabierta.EOF
    Designacion.AddItem tablaabierta!Designacion
    Designacion.ItemData(Designacion.NewIndex) = Str(tablaabierta!Designacion)

```

```
    tablaabierta.MoveNext
Loop
ReSort Designacion
Designacion.ListIndex = 0
labelregistro
End Sub
```

---

```
Sub cargardb()
If FileExists(App.Path + "\" + nombre.Tag + ".mdb") Then
    'La base de datos existe, se puede abrir para modificar
    If MsgBox("Este archivo ya existe, ¿desea modificarlo?", vbYesNo, "Atención!") =
vbYes Then
        Ruta = App.Path + "\" + nombre.Tag + ".mdb"
        Set mydb = OpenDatabase(Ruta)
        Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla:"
        nombreperfil1.Enabled = True
        Crear.Enabled = True
        Frame1.Enabled = False
        nombreperfil1.SetFocus
    Else
        MsgBox "Debe crear o cargar una base de datos para ejecutar el programa",
vbExclamation, "Atención!"
    End If
Else
    'crear base de datos
    Set MiWorkspace = DBEngine.Workspaces(0)
    Ruta = App.Path + "\" + nombre.Tag + ".mdb"
    Set mydb = MiWorkspace.CreateDatabase(Ruta, dbLangGeneral, dbVersion30)
    Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla: "
    Set mydb = OpenDatabase(Ruta)
    MsgBox "Base de datos creada y cargada", vbInformation, "Felicitaciones..."
    nombreperfil1.Enabled = True
    Crear.Enabled = True
    Frame1.Enabled = True
    nombreperfil1.SetFocus
End If
End Sub
```

---

```
Private Sub design_Click()
    borrar design
End Sub
```

---

```
Private Sub design_GotFocus()
    borrar design
End Sub
```

---

```
Sub borrardesign()  
  If design.Tag = "primera vez" Then  
    design.Tag = "no primera vez"  
    design.Text = ""  
  End If  
End Sub
```

---

```
Private Sub design_KeyPress(KeyAscii As Integer)  
  borrardesign  
  Select Case KeyAscii  
    Case Asc("0") To Asc("9")  
    Case 8  
    Case 13: AñadirDes_Click  
    Case Else: KeyAscii = 0  
  End Select  
End Sub
```

---

```
Private Sub Designacion_Click()  
  labelregistro  
  'valor.SetFocus  
End Sub
```

---

```
Private Sub Designacion_KeyPress(KeyAscii As Integer)  
  If KeyAscii = 13 Then design.SetFocus  
End Sub
```

---

```
Private Sub Explorar_Click()  
  nombrevar = nombre.Text  
  tablaperfil = nombreperfil1.Text  
  EnumTabla.Show  
End Sub
```

---

```
Private Sub Form_Load()  
  
  For n = 0 To 29  
    ValDato(n) = ""  
  Next n  
End Sub
```

---

```
Private Sub nombre_KeyPress(KeyAscii As Integer)  
  If KeyAscii = 13 Then  
    Cargar_Click  
    If KeyAscii = Asc(vbCr) Then  
      KeyAscii = 0  
    End If  
  End If  
End Sub
```

End Sub

---

```
Sub creartabla()
Dim n As Integer
Dim campo As String

If ExisteTabla Then
    If MsgBox("¿La tabla ya existe, desea trabajar en ella?", vbOKCancel,
"Atención!") = vbCancel Then
        nombreperfil1.Text = ""
        Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla:"
        Frame1.Enabled = False
        Crear.Tag = "existe"
    Else
        Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla: " +
nombreperfil1.Tag
        Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
        Frame1.Enabled = True
        ' aqui se debe llenar de nuevo la lista
        Designacion.Clear
        llenarlistas
        Designacion.SetFocus
    End If
Else
    Set Mitabla = mydb.CreateTableDef(nombreperfil1.Tag)
    For n = 0 To 30
        Select Case n
            Case 0
                campo = "P"
            Case 1
                campo = "Area"
            Case 2
                campo = "d"
            Case 3
                campo = "bf"
            Case 4
                campo = "tf"
            Case 5
                campo = "tw"
            Case 6
                campo = "lx"
            Case 7
                campo = "Sx"
            Case 8
                campo = "rx"
            Case 9
```

```
    campo = "ly"  
Case 10  
    campo = "Sy"  
Case 11  
    campo = "ry"  
Case 12  
    campo = "rT"  
Case 13  
    campo = "dAf"  
Case 14  
    campo = "bf2tf"  
Case 15  
    campo = "dtw"  
Case 16  
    campo = "j"  
Case 17  
    campo = "Cw"  
Case 18  
    campo = "Zx"  
Case 19  
    campo = "Zy"  
Case 20  
    campo = "a"  
Case 21  
    campo = "T"  
Case 22  
    campo = "k"  
Case 23  
    campo = "k1"  
Case 24  
    campo = "g1"  
Case 25  
    campo = "c"  
Case 26  
    campo = "m"  
Case 27  
    campo = "n"  
Case 28  
    campo = "g"  
Case 29  
    campo = "g2"  
Case 30  
    campo = "designacion"  
End Select
```

```
If n <> 30 Then  
    Set MisCampos(n) = Mitabla.CreateField(campo, dbText)
```

```
MisCampos(n).Size = 30
Mitable.Fields.Append MisCampos(n)
Crear.Tag = "existe"
Else
Set MisCampos(n) = Mitable.CreateField(campo, dbInteger)
MisCampos(n).Size = 30
Mitable.Fields.Append MisCampos(n)
Crear.Tag = "existe"
End If
Next n
mydb.TableDefs.Append Mitable
MsgBox "La tabla fue creada exitosamente", vbInformation, "Felicitaciones..."
design.Tag = "primera vez"
Frame1.Enabled = True
Frame1.Caption = "Nombre de archivo en uso: " + nombre.Tag + " Tabla: " +
nombreperfil1.Tag
Designacion.SetFocus
Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
```

```
tablaabierta.AddNew
```

```
tablaabierta!P = "0"
tablaabierta!Area = "0"
tablaabierta!d = "0"
tablaabierta!bf = "0"
tablaabierta!tf = "0"
tablaabierta!tw = "0"
tablaabierta!lx = "0"
tablaabierta!Sx = "0"
tablaabierta!rx = "0"
tablaabierta!ly = "0"
tablaabierta!Sy = "0"
tablaabierta!ry = "0"
tablaabierta!rt = "0"
tablaabierta!dAf = "0"
tablaabierta!bf2tf = "0"
tablaabierta!dtw = "0"
tablaabierta!j = "0"
tablaabierta!Cw = "0"
tablaabierta!Zx = "0"
tablaabierta!Zy = "0"
tablaabierta!a = "0"
tablaabierta!t = "0"
tablaabierta!k = "0"
tablaabierta!K1 = "0"
tablaabierta!G1 = "0"
tablaabierta!C = "0"
```

```
tablaabierta!m = "0"  
tablaabierta!n = "0"  
tablaabierta!g = "0"  
tablaabierta!G2 = "0"  
tablaabierta!Designacion = 0  
    tablaabierta.Update
```

```
End If
```

```
End Sub
```

---

```
Sub LlenarConCeros()
```

```
    'Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
```

```
'tablaabierta.Edit
```

```
tablaabierta!P = "0"  
tablaabierta!Area = "0"  
tablaabierta!d = "0"  
tablaabierta!bf = "0"  
tablaabierta!tf = "0"  
tablaabierta!tw = "0"  
tablaabierta!lx = "0"  
tablaabierta!Sx = "0"  
tablaabierta!rx = "0"  
tablaabierta!ly = "0"  
tablaabierta!Sy = "0"  
tablaabierta!ry = "0"  
tablaabierta!rt = "0"  
tablaabierta!dAf = "0"  
tablaabierta!bf2tf = "0"  
tablaabierta!dtw = "0"  
tablaabierta!j = "0"  
tablaabierta!Cw = "0"  
tablaabierta!Zx = "0"  
tablaabierta!Zy = "0"  
tablaabierta!a = "0"  
tablaabierta!t = "0"  
tablaabierta!k = "0"  
tablaabierta!K1 = "0"  
tablaabierta!G1 = "0"  
tablaabierta!C = "0"  
tablaabierta!m = "0"  
tablaabierta!n = "0"  
tablaabierta!g = "0"  
tablaabierta!G2 = "0"
```

End Sub

---

```
Function ExisteTabla() As Boolean
Dim td As TableDef
ExisteTabla = False
For Each td In mydb.TableDefs
'Sólo las tablas con atributo igual a CERO,
'son tablas normales
If td.Attributes = 0 And UCase(td.Name) = UCase(nombreperfil1.Text) Then
ExisteTabla = True
End If
Next
End Function
```

---

```
Sub labelregistro()
' esto sirve para leer el registro y colocar en las etiquetas el valor
' correspondiente a cada campo

Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
tablaabierta.FindFirst "designacion=" &
Str(Designacion.ItemData(Designacion.ListIndex))
ValDato(0).Caption = tablaabierta!P + " kgf/m"
ValDato(1).Caption = tablaabierta!Area + " cm2"
ValDato(2).Caption = tablaabierta!d + " mm"
ValDato(3).Caption = tablaabierta!bf + " mm"
ValDato(4).Caption = tablaabierta!tf + " mm"
ValDato(5).Caption = tablaabierta!tw + " mm"
ValDato(6).Caption = tablaabierta!lx + " cm4"
ValDato(7).Caption = tablaabierta!Sx + " cm3"
ValDato(8).Caption = tablaabierta!rx + " cm"
ValDato(9).Caption = tablaabierta!ly + " cm4"
ValDato(10).Caption = tablaabierta!Sy + " cm3"
ValDato(11).Caption = tablaabierta!ry + " cm"
ValDato(12).Caption = tablaabierta!rt + " cm"
ValDato(13).Caption = tablaabierta!dAf + "cm(-1)"
ValDato(14).Caption = tablaabierta!bf2tf
ValDato(15).Caption = tablaabierta!dtw
ValDato(16).Caption = tablaabierta!j + " cm4"
ValDato(17).Caption = tablaabierta!Cw + " cm6"
ValDato(18).Caption = tablaabierta!Zx + " cm3"
ValDato(19).Caption = tablaabierta!Zy + " cm3"
ValDato(20).Caption = tablaabierta!a + " mm"
ValDato(21).Caption = tablaabierta!t + " mm"
ValDato(22).Caption = tablaabierta!k + " mm"
ValDato(23).Caption = tablaabierta!K1 + " mm"
ValDato(24).Caption = tablaabierta!G1 + " mm"
```

```
ValDato(25).Caption = tablaabierta!C + " mm"  
ValDato(26).Caption = tablaabierta!m + " mm"  
ValDato(27).Caption = tablaabierta!n + " mm"  
ValDato(28).Caption = tablaabierta!g + " mm"  
ValDato(29).Caption = tablaabierta!G2 + " mm"
```

```
End Sub
```

---

```
Private Sub nombreperfil1_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then  
        Crear_Click  
        If KeyAscii = Asc(vbCr) Then  
            KeyAscii = 0  
        End If  
    End If  
End Sub
```

---

```
Private Sub Propiedad_Click()  
    Select Case Caracteristica.ListIndex  
    Case 2  
        Unidad.Caption = "[mm]"  
    Case 3  
        Select Case Propiedad.ListIndex  
        Case 0, 3  
            Unidad.Caption = "[cm4]"  
        Case 1, 4  
            Unidad.Caption = "[cm3]"  
        Case 2, 5  
            Unidad.Caption = "[cm]"  
        End Select  
    Case 4  
        Select Case Propiedad.ListIndex  
        Case 0  
            Unidad.Caption = "[cm]"  
        Case 1  
            Unidad.Caption = "[cm(-1)]"  
        End Select  
    Case 5  
        Unidad.Caption = "[adim]"  
    Case 6  
        Select Case Propiedad.ListIndex  
        Case 0  
            Unidad.Caption = "[cm4]"  
        Case 1  
            Unidad.Caption = "[cm6]"  
        End Select  
    End Select
```

```
End Select
Case 7
    Unidad.Caption = "[cm3]"
Case 8
    Unidad.Caption = "[mm]"

End Select
End Sub
```

---

```
Private Sub valor_Click()
borrarvalor
End Sub
```

---

```
Private Sub valor_GotFocus()
borrarvalor
End Sub
Sub borrarvalor()
    If valor.Tag = "borrar" Then
        valor.Text = ""
        valor.Tag = ""
    End If
End Sub
```

---

```
Private Sub valor_KeyPress(KeyAscii As Integer)
'
'este condicional, es para eliminar el
'beep cuando se presiona enter, pero cancela la ejecución de enter,
'por eso está después de que se toma en cuenta el hecho que el usuario
'presione enter....
'
'If KeyAscii = Asc(vbCr) Then
' KeyAscii = 0
'End If

If Len(valor) = 0 Then
    Select Case KeyAscii
        Case Asc("0") To Asc("9")
            Case 8
            Case Else: KeyAscii = 0
    End Select
End If

If InStr(valor, ".") > 0 Then
    Select Case KeyAscii
        Case Asc("0") To Asc("9")
            Case 8
```

```

Case 13
  Agregar_Click
  If KeyAscii = Asc(vbCr) Then
    KeyAscii = 0
  End If
  Case Else: KeyAscii = 0
End Select
Else
  Select Case KeyAscii
  Case Asc("0") To Asc("9")
  Case 8
  Case Asc(".")
  Case 13
  Agregar_Click
  If KeyAscii = Asc(vbCr) Then
    KeyAscii = 0
  End If
  Case Else: KeyAscii = 0
End Select
End If
End Sub

```

---

```

Sub registro()
Dim carac As String
Dim prop As String
carac = Caracteristica.ListIndex
prop = Propiedad.ListIndex
Set tablaabierta = mydb.OpenRecordset(nombreperfil1.Tag, dbOpenDynaset)
tablaabierta.FindFirst "designacion=" &
Str(Designacion.ItemData(Designacion.ListIndex))
'
'aqui se busca el registro que esta seleccionado en designación
'

```

```

Select Case carac
  Case 0
    tablaabierta.Edit
    tablaabierta!P = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 1
    tablaabierta.Edit
    tablaabierta!Area = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 2

```

```
Select Case prop
  Case 0
    tablaabierta.Edit
    tablaabierta!d = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 1
    tablaabierta.Edit
    tablaabierta!bf = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 2
    tablaabierta.Edit
    tablaabierta!tf = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 3
    tablaabierta.Edit
    tablaabierta!tw = valor.Text
    tablaabierta.Update
    Call siguiente

End Select
Case 3
Select Case prop
  Case 0
    tablaabierta.Edit
    tablaabierta!lx = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 1
    tablaabierta.Edit
    tablaabierta!Sx = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 2
    tablaabierta.Edit
    tablaabierta!rx = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 3
    tablaabierta.Edit
    tablaabierta!ly = valor.Text
    tablaabierta.Update
    Call siguiente
  Case 4
    tablaabierta.Edit
```

```
tablaabierta!Sy = valor.Text
tablaabierta.Update
Call siguiente
Case 5
tablaabierta.Edit
tablaabierta!ry = valor.Text
tablaabierta.Update
Call siguiente
```

```
End Select
Case 4
Select Case prop
Case 0
tablaabierta.Edit
tablaabierta!rt = valor.Text
tablaabierta.Update
Call siguiente
Case 1
tablaabierta.Edit
tablaabierta!dAf = valor.Text
tablaabierta.Update
Call siguiente
```

End Select

```
Case 5
Select Case prop
Case 0
tablaabierta.Edit
tablaabierta!bf2tf = valor.Text
tablaabierta.Update
Call siguiente
Case 1
tablaabierta.Edit
tablaabierta!dtw = valor.Text
tablaabierta.Update
Call siguiente
End Select
```

```
Case 6
Select Case prop
Case 0
tablaabierta.Edit
tablaabierta!j = valor.Text
tablaabierta.Update
Call siguiente
Case 1
```

```
    tablaabierta.Edit
    tablaabierta!Cw = valor.Text
    tablaabierta.Update
    Call siguiente
End Select
Case 7
Select Case prop
Case 0
    tablaabierta.Edit
    tablaabierta!Zx = valor.Text
    tablaabierta.Update
    Call siguiente
Case 1
    tablaabierta.Edit
    tablaabierta!Zy = valor.Text
    tablaabierta.Update
    Call siguiente
End Select
Case 8
Select Case prop
Case 0
    tablaabierta.Edit
    tablaabierta!a = valor.Text
    tablaabierta.Update
    Call siguiente
Case 1
    tablaabierta.Edit
    tablaabierta!t = valor.Text
    tablaabierta.Update
    Call siguiente
Case 2
    tablaabierta.Edit
    tablaabierta!k = valor.Text
    tablaabierta.Update
    Call siguiente
Case 3
    tablaabierta.Edit
    tablaabierta!K1 = valor.Text
    tablaabierta.Update
    Call siguiente
Case 4
    tablaabierta.Edit
    tablaabierta!G1 = valor.Text
    tablaabierta.Update
    Call siguiente
Case 5
    tablaabierta.Edit
```

```
    tablaabierta!C = valor.Text
    tablaabierta.Update
    Call siguiente
Case 6
    tablaabierta.Edit
    tablaabierta!m = valor.Text
    tablaabierta.Update
    Call siguiente
Case 7
    tablaabierta.Edit
    tablaabierta!n = valor.Text
    tablaabierta.Update
    Call siguiente
Case 8
    tablaabierta.Edit
    tablaabierta!g = valor.Text
    tablaabierta.Update
    Call siguiente
Case 9
    tablaabierta.Edit
    tablaabierta!G2 = valor.Text
    tablaabierta.Update
    Call siguiente
End Select
```

End Select

End Sub

---

Sub siguiente()

If Not (Caracteristica.ListIndex = 8 And Propiedad.ListIndex = 9) Then

Select Case Caracteristica.ListIndex

Case 0, 1

Caracteristica.ListIndex = Caracteristica.ListIndex + 1

If Caracteristica.ListIndex >= 1 Then

Caracteristica\_Click

End If

Case Else

If Propiedad.ListIndex = Propiedad.ListCount - 1 Then

Caracteristica.ListIndex = Caracteristica.ListIndex + 1

Propiedad.ListIndex = 0

Caracteristica\_Click

Else

Propiedad.ListIndex = Propiedad.ListIndex + 1

End If  
End Select  
End If  
End Sub

## **VII.2 Creación del manual del usuario**

Una vez finalizado y depurado el programa *AcartBeam V1.3*, se procedió a la redacción del manual del usuario con el fin de explicar el funcionamiento y uso del mismo, teniendo como premisa el hecho que el usuario final pudiese ejecutarlo sin mayor dificultad, a través de imágenes y descripciones de las ventanas emergentes que conforman este programa. Además, se ofrecen ayudas dinámicas incluidas dentro del código del programa.



Una vez finalizado este trabajo especial de grado y después de desarrollar y utilizar el programa ***AcartBeam V1.3*** se ha llegado a las siguientes conclusiones:

La utilización de un sistema computarizado para cálculo de estructuras hace más eficiente la toma de decisiones al momento del diseño, dado que los resultados pueden ser obtenidos con mayor precisión y rapidez que el proceso manual. Pero para el desarrollo de aplicaciones se requiere resolver el problema en un principio manualmente, esto con el objetivo de determinar las variables que intervienen en el en la estructura considerada en el estudio.

Las cartelas, según se pudo observar en las tablas de vigas acarteladas horizontales, cargadas uniformemente que forman parte del anexo, tienen una distancia óptima de 0,35 L siendo ésta la distancia en la que el elemento acartelado absorbe más momento. Para longitudes mayores a 0,35 L los momentos disminuyen a medida que crece la cartela, con lo que comienza a disminuir su eficiencia, generándose pérdida de material.

Mediante uso del programa ***AcartBeam V1.3*** se puede tener una idea preliminar sobre el tipo de perfil a utilizar en el diseño del pórtico. Esto se traduce en ahorro de tiempo, debido al fácil manejo del programa. La influencia que representan las vigas acarteladas en los pórticos para galpones esta íntimamente ligado al ahorro de material, y resolver el problema de la no existencia en el mercado nacional de todos los perfiles HEA, HEB, IPN, e IPE.

El método de las rotaciones, es una manera versátil de resolver problemas de pórticos tipo galpón, debido a que se contempla primordialmente los desplazamientos como incógnitas de un sistema de ecuaciones y no las redundantes hiperestáticas, que generan un número mayor de incógnitas por junta.



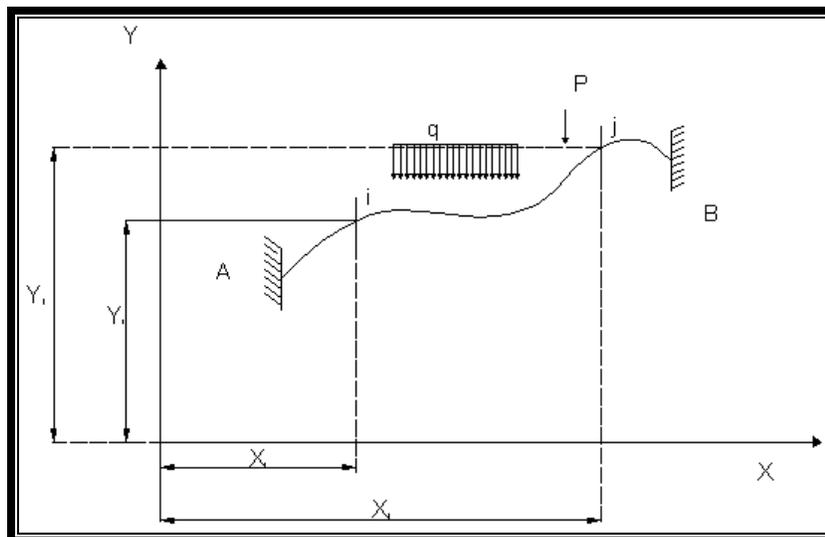
- El desarrollo de módulos que permitan considerar las deformaciones por cargas axiales, corte y cambios de temperaturas.
- Desarrollar otras aplicaciones incluyan vinculaciones tales como rotulas, resortes, empotramientos móviles entre otros. Del mismo modo que se consideren cargas trapezoidales y triangulares en los miembros.
- Crear un modulo que se apoye en herramientas CAD con el fin de dibujar en forma tridimensional el pórtico, que a su vez considere el efecto de cargas perpendiculares al plano de la estructura.
- Crear una aplicación que permita representar gráficamente la deformada del pórtico.
- Desarrollar una aplicación grafica, con el fin de mostrar los diagramas de cuerpo libre, de cargas axiales, cargas cortantes y momento flector en cada miembro del pórtico.



Las demostraciones que aquí se hacen en buena medida son apuntes tomados de las clases de estructuras dictadas por el profesor Tomas Osers durante octubre de 2002 y marzo de 2003 en la facultad de ingeniería de la U.C.V.

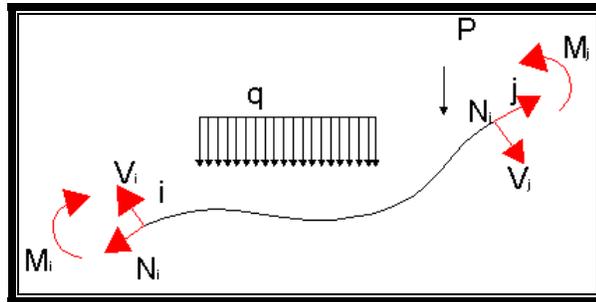
Formulas de Bresse.

Sea, un elemento estructural de eje recto o curvo, con inercia variable o constante empotrado en sus extremos, (X.1).



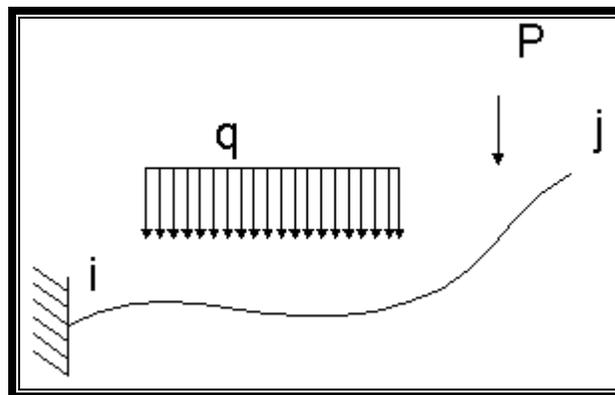
X.1 Miembro cualquiera sometido a cargas externas.

Despreciando las deformaciones por carga axial y de corte, se tiene tres posibles desplazamientos en cada extremo y tres reacciones tal como se muestra en la figura (X.2). Recordando que en el plano un cuerpo tiene tres posibles desplazamientos. Horizontal, vertical y una rotación.



X.2 Reacciones internas en evidencia.

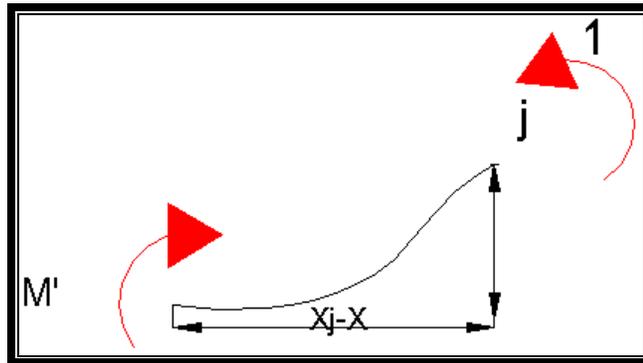
Mediante el principio de superposición se sumaran los efectos de las cargas externas y los efectos de las cargas unitarias tal que  $P_{final}=P+P'$



X.3 Sistema primario (P).

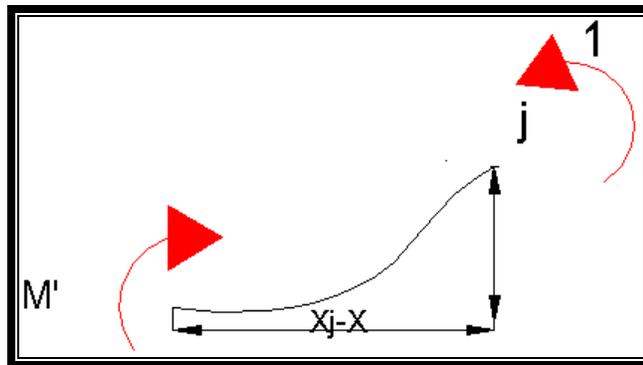
Como puede verse en X.3 se ha empotrado el extremo i y se ha dejado libre el extremo j siendo este el sistema primario P donde actúan solo las cargas externas. Como se observa este sistema no es indeterminado. De aquí se calcula el momento M que posteriormente formara parte del sistema de ecuaciones.

Tomemos como primera incógnita la rotación del extremo  $j$ , ( $\theta_j$ ). Para tal fin se aplica un momento unitario en  $j$  como se ve en la figura X.4, que define el sistema  $P'$  donde solo actúa la carga unitaria en el lugar que se desea saber el valor del desplazamiento.



X.4 Sistema P'.

Si se toma una sección cualquiera  $m$  y se considera el elemento entre  $m$  y  $j$  nos queda lo siguiente.

X.5 Sección comprendida entre  $m$  e  $j$ .

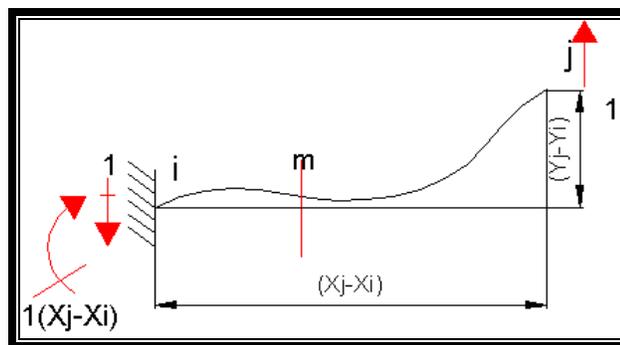
Aplicando el principio del trabajo virtual para cuerpos deformables (P.T.V.C.D.), tenemos las siguientes ecuaciones:

$$1\theta_j - 1\theta_i = \int_s \frac{M' M}{EI} ds + \int_s \frac{\alpha + \Delta(\Delta t) M'}{h} ds \quad (1)$$

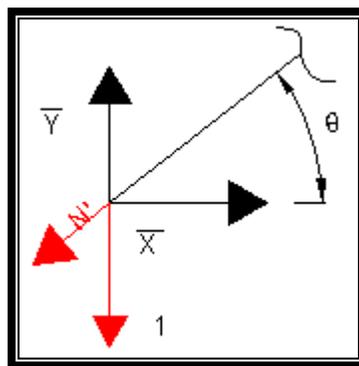
$$\theta_j = \theta_i + \int_s \left[ \frac{M}{EI} + \frac{\alpha_i + \Delta(\Delta t)}{h} \right] ds \quad (2)$$

La ecuación 2 permite determinar el desplazamiento en el extremo  $j$ , es decir, la rotación  $\theta_j$  en función del momento que generan las cargas del sistema  $P$ , la rotación del extremo  $i$ , y los de cambios de temperatura existentes en el miembro.

Aplicando el mismo procedimiento antes expuesto se determina el desplazamiento vertical del extremo  $j$  ( $v_j$ ), para eso se define otro sistema  $P'$  sometido a una carga unitaria en el extremo  $j$ .

X.6 Sistema  $P'$ .

Al tomar nuevamente la parte del miembro comprendida entre  $m$  y  $j$  aparece una carga axial  $N'$ , que es función de la carga unitaria y se proyecta en el eje axial del miembro. Véase en la figura X.7.

X.7 Proyección de la carga unitaria en  $N'$ .

De las figuras X.6 y X.7 se desprende que:

$$\begin{aligned} N' &= -\bar{X} \cos \theta - \bar{Y} \operatorname{sen} \theta & (3) \\ \bar{Y} &= -1 \quad \bar{X} = 0 & \Rightarrow N' = \operatorname{sen} \theta \\ M' &= 1(x_j - x) \end{aligned}$$

Por *P.T.V.C.D.*

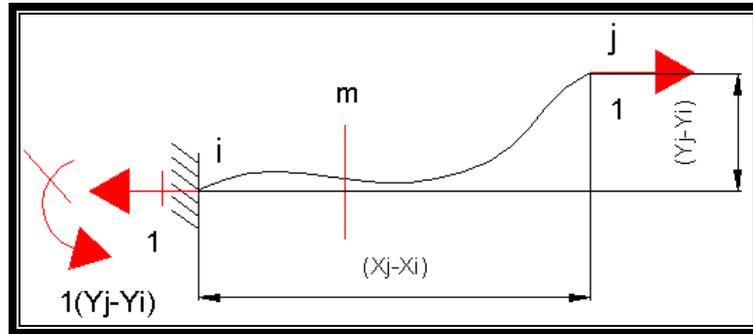
$$-\theta_i M' - 1v_i + 1v_j = \int_s \frac{M' M}{EI} ds + \int_s \frac{\alpha_t \Delta(\Delta t) M'}{h} ds + \int_s N' \alpha_t \Delta t_0 ds \quad \text{reagrupando} \quad (4)$$

$$v_j = v_i + \theta_i M' + \int_s M' \left[ \frac{M}{EI} + \frac{\alpha_t \Delta(\Delta t)}{h} \right] ds + \int_s N' \alpha_t \Delta t_0 ds \quad \text{Sustituyendo M'y N'}$$

$$v_j = v_i + \theta_i (X_j - X_i) + \int_s (X_j - X) \left[ \frac{M}{EI} + \frac{\alpha_t \Delta(\Delta t)}{h} \right] ds + \int_s \operatorname{sen} \theta \alpha_t \Delta t_0 ds \quad (5)$$

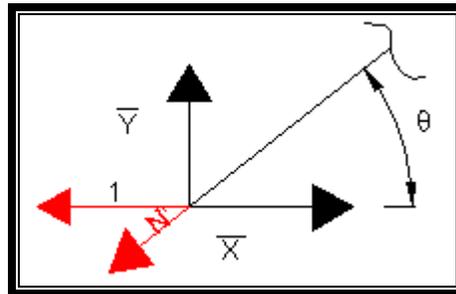
La ecuación 5 permite determinar el desplazamiento vertical del extremo j en términos de la rotación de i, el momento, por las cargas externas, el alargamiento o acortamiento del miembro en dirección **Y**. Representado por el  $\operatorname{sen} \theta$  ya que el alargamiento total de la barra es  $\Delta L$  y  $\Delta L \operatorname{sen} \theta = \Delta L_y$ .

El procedimiento se repite pero con el fin de determinar la ecuación que permite hallar el desplazamiento en la dirección horizontal  $\mu$ .



X.8 Sistema P'

Aplicando una carga horizontal unitaria en j con el fin de determinar el desplazamiento  $\mu_j$ . Se genera un momento en el extremo i,  $M' = -(Y_j - Y)$ .



X.9

De la ecuación 3:

$$N' = -\bar{X} \cos \theta - \bar{Y} \sin \theta$$

$$\bar{Y} = 0 \quad \bar{X} = -1 \quad \Rightarrow \quad N' = \cos \theta$$

$$M' = -(Y_j - Y)$$

Por P.T.V.C.D.

$$1\mu_j - 1\mu_i + M'\theta_i = \int_s \frac{M'M}{EI} ds + \int_s \frac{M'\alpha_i \Delta(\Delta t)}{h} ds + \int_s N'\alpha_i \Delta t ds \tag{6}$$

$$\mu_j = \mu_i - (Y_j - Y)\theta_i - \int_s (Y_j - Y) \left[ \frac{M}{EI} + \frac{\alpha_i \Delta(\Delta t)}{h} \right] ds + \int_s \cos \theta \alpha_i \Delta t ds \tag{7}$$

Bajo el mismo razonamiento que en la ecuación 5 el último término de la ecuación 7 representa el alargamiento que sufre el miembro en la dirección  $X$ .

$$\Delta L \cos \theta = \Delta L_x .$$

Hay que hacer una consideración importante en la forma final de las ecuaciones. En el caso particular de este trabajo especial de grado, no se consideran las deformaciones por cambios de temperatura. En consecuencia los términos que tienen cambios de temperatura son nulos.

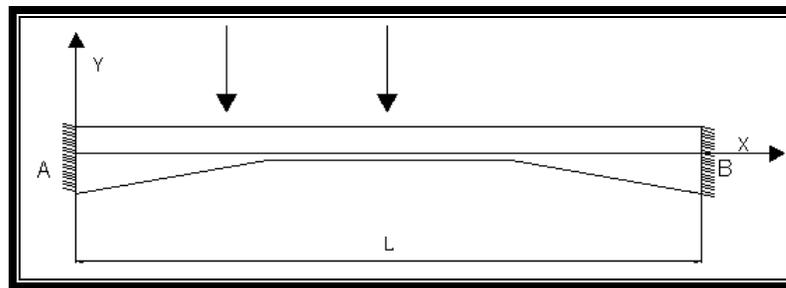
$$\theta_j = \theta_i - \int_s \frac{M}{EI} ds \quad (8)$$

$$\theta_i = \theta_j + \int_s \frac{M}{EI} ds \quad (9)$$

$$v_j = v_i + \theta_i (X_j - X_i) + \int_s (X_j - X) \frac{M}{EI} ds \quad (10)$$

$$\mu_j = \mu_i - \theta_i (Y_j - Y_i) - \int_s \frac{(Y_j - Y)M}{EI} ds \quad (11)$$

Las ecuaciones 8, 9, 10 y 11 un resultado particular de las formulas de Bresse, ya como se menciono antes se están despreciando los cambios de temperatura. Estas ecuaciones permiten determinar los desplazamientos que sufre un elemento sometido a cargas externas. Para la aplicación de estas formulas se utilizara una viga de sección variable empotrada en sus extremos, sometida a un grupo de cargas, tal cual lo muestra la figura X.9.



X.10

Las condiciones de borde son:

$$v_i = v_j = \theta_i = \theta_j = 0 \quad \text{y} \quad x_i = 0 \quad x_j = L$$

Por las ecuaciones 9 y 10 se tiene que:

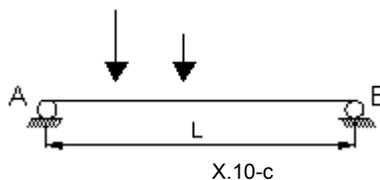
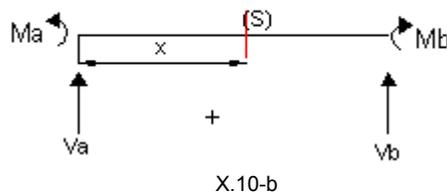
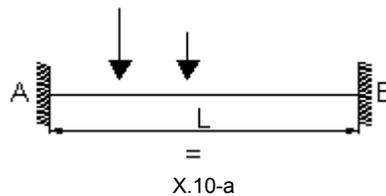
$$\int_0^l \frac{Ml}{EI} dx - \int_0^l \frac{Mx}{EI} dx = 0 \quad \int_0^l \frac{M}{EI} dx = 0$$

El sistema es equivalente es el que sigue.

$$\int_0^l \frac{M(l-x)}{I} dx \quad \int_0^l \frac{Mx}{I} dx = 0 \quad \text{Al multiplicar por } E \text{ a ambos lados de}$$

la igualdad, E (módulo de elasticidad del material) se simplifica.

En todo elemento horizontal cargado, para el caso de esta demostración, se cumple que por estática la reacción vertical es función de los momentos en los extremos y de la longitud. Aplicando el principio de superposición es posible determinar como varia el momento a lo largo de todo el elemento. Esto se ve con claridad en las figuras X.10-a, X.10-b y X.10-c.



La figura X.10-b representa el diagrama de cuerpo libre de la viga X.10-a. Como en la viga no hay cargas horizontales aplicadas, la representación en el diagrama de cuerpo libre de estas fuerzas no tiene sentido. Esto se ha hecho con el motivo de simplificar la demostración.

La figura X.10-c representa la misma viga X.10-a, sometida a las mismas cargas, con la diferencia de que esta simplemente apoya, de esta viga se obtiene la variación del momento, en términos del tipo de carga. Esta función por comodidad se le denomina  $\mu$ . Se obtiene como resultado de haber integrado la carga dos veces  $C=q(x)$ , donde  $q(x)$  es una carga cualquiera, que esta aplicada tanto en la viga X.10-a y X.10-c, puede ser carga puntual, distribuida y combinaciones de ellas. Si a C se le integra se consigue la función del corte  $V(x)$ .

$$V_{(x)} = \int q_{(x)} dx \quad (12)$$

Si ahora se integra  $V(x)$  se determina la función  $\mu(x)$ , que no es más que la función del momento  $M(x)$ .

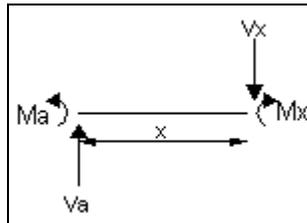
$$\mu_{(x)} = M_{(x)} = \int V_{(x)} dx \quad (13)$$

Si se toma momento en el extremo B de la viga X.10-b se puede calcular el valor de la reacción vertical en  $V_A$ . Suponiendo positivos los giros en el sentido de las agujas del reloj.

$$\otimes \sum M_{(B)} = 0 - M_A + M_B + LV_A = 0 \quad \text{despejando } V_{(A)}$$

$$\boxed{V_{(A)} = \frac{Ma - Mb}{L}} \quad (14)$$

Si se toma una parte de la viga comprendida entre A y S a una distancia  $x$  tal como lo muestran las vigas X.10-b y X.10-d y se aplica el mismo razonamiento se llega a otra ecuación general.



X. 10-d

$$\otimes \sum M_{(x)} = 0 - M_A + M_x + LV_A = 0 \quad \text{despejando } M_{(x)} \text{ y sustituyendo el valor de}$$

la ecuación 14, se tiene que:

$$M_{(x)} = M_A + \frac{M_B - M_A}{L} x \quad (15)$$

Por el principio de superposición sumamos los efectos de las vigas X.10-c y X.10-d, considerando la convención de signos del método de las rotaciones

$$M_a = -M_{ab} \quad \text{y} \quad M_b = M_{ba}$$

Por comodidad se denomina  $M=M_{(x)}$ , sustituyendo en 15 y sumando la función  $\mu$ .

$$M = \mu - M_{BA} + \frac{M_{AB} + M_{AB}}{L} x \Leftrightarrow M = \mu - M_{AB} \frac{(l-x)}{l} + M_{BA} \frac{x}{l}$$

$$M = \mu - M_{AB} \frac{(l-x)}{l} + M_{BA} \frac{x}{l} \quad (16)$$

Retomando las ecuaciones de la viga X.9

$$\int_0^l \frac{M(l-x)}{I} dx - \int_0^l \frac{Mx}{I} dx = 0, \text{ sustituyendo el valor de } M$$

$$\int_0^l \left\{ \mu - M_{AB} \frac{(l-x)}{l} + M_{BA} \frac{x}{l} \right\} \frac{(l-x)}{I} dx = 0 \quad (17)$$

$$\int_0^l \left\{ \mu - M_{AB} \frac{(l-x)}{l} + M_{BA} \frac{x}{l} \right\} \frac{x}{I} dx = 0 \quad (18)$$

$$\int_0^l \frac{\mu(l-x)}{I} dx - \int_0^l M_{AB} \frac{(l-x)^2}{Il} dx + \int_0^l M_{BA} \frac{x(l-x)}{Il} dx = 0 \quad (17,1)$$

$$\int_0^l \frac{\mu x}{I} dx - \int_0^l M_{AB} \frac{(l-x)}{Il} dx + \int_0^l M_{BA} \frac{x^2}{Il} dx = 0 \quad (18.1)$$

Con el fin de resolver el sistema de ecuaciones y hacerlo más sencillo se definen

los valores de las  $K_i$ , según *P., Charon. 1953, p.216.*

$$\begin{aligned} K_1 &= \int_0^l \frac{x^2 dx}{I} & K_2 &= \int_0^l \frac{(l-x)^2 dx}{I} \\ K_3 &= \int_0^l \frac{x(l-x) dx}{I} & & \\ K_4 &= \int_0^l \frac{\mu x dx}{I} & K_5 &= \int_0^l \frac{\mu(l-x) dx}{I} \end{aligned} \quad (19)$$

Resolviendo el sistema se determinan los momentos de empotramiento  $M_{AB}$  y  $M_{BA}$ .

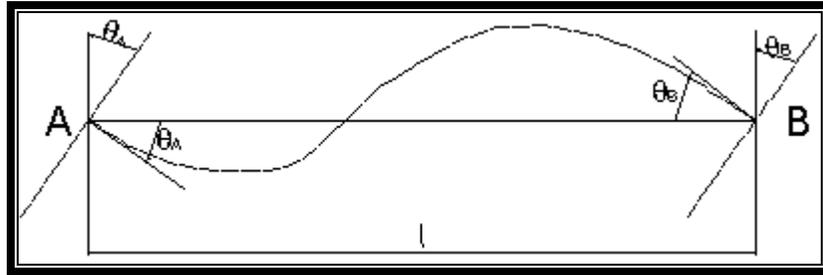
$$M_{AB} = \frac{K_3 K_4 - K_1 K_5}{K_3^2 - K_1 K_2} l \quad M_{BA} = \frac{K_2 K_4 - K_3 K_5}{K_3^2 - K_1 K_2} l \quad (20)$$

Estos momentos son los denominados momentos de empotramiento perfecto según *McCormac y Nelson* y esto se debe a que no hay rotación de los extremos. Como se ha visto la inercia es variable y por lo tanto no sale del integrado.

Para la determinación de los momentos  $MF'_{ij}$  (rotación de los extremos) las condiciones de borde son:

$$v_i = v_j = 0 \quad x_i = 0 \quad x_j = l$$

$$\theta_i = \theta_A \quad \theta_j = \theta_B$$



X.11

Sustituyendo en las ecuaciones 8 y 10 tenemos:

$$\theta_B = \theta_A - \int_0^l \frac{M}{EI} dx$$

$$-l\theta_A + l \int_0^l \frac{M}{EI} dx - \int_l^0 \frac{Mx}{EI} dx = 0 \text{ sustituyendo } \theta_A \text{ queda:}$$

$$-l\theta_B - \int_0^l \frac{Mx}{EI} dx, \text{ como } M = -M_{AB} + \frac{M_{AB} + M_{BA}}{l} x$$

$$l\theta_B + \frac{M_{AB}}{E} \int_0^l \frac{x}{I} dx - \frac{(M_{AB} + M_{BA})}{EI} \int_0^l \frac{x^2}{I} dx = 0 \quad (21)$$

$$\theta_A - \theta_B - \int_0^l \frac{M}{EI} dx$$

$$\theta_A - \theta_B + \frac{M_{AB}}{E} \int_0^l \frac{dx}{I} - \frac{(M_{AB} + M_{BA})}{EI} \int_0^l \frac{x}{I} dx = 0 \quad (22)$$

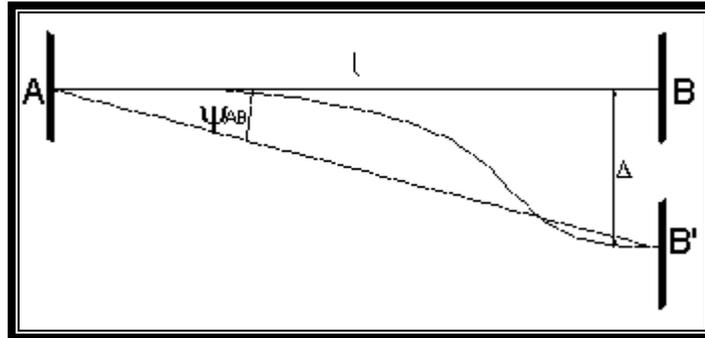
Sustituyendo las ecuaciones 19 y resolviendo el sistema se hallan los momentos

$MF'$

$$MF'_{AB} = \frac{K_3 El^2}{K_3^2 - K_1 K_2} \left( \frac{K_1}{K_3} \theta_A + \theta_B \right) \quad MF'_{BA} = \frac{K_3 El^2}{K_3^2 - K_1 K_2} \left( \theta_A + \frac{K_2}{K_3} \theta_B \right) \quad (23)$$

Las condiciones de borde en el caso de un desplazamiento vertical  $\Delta$  en el extremo B son:

$$\begin{aligned} v_0 &= 0 & v_l &= -\Delta \\ x_0 &= 0 & x_l &= l \\ \theta_0 &= 0 & x_l &= 0 \end{aligned}$$



X.12

De las ecuaciones 8 y 10:

$$l \int_0^l \frac{M}{EI} dx = 0$$

$$\Delta = - \int_0^l \frac{Mx}{EI} dx \quad \text{sustituyendo M por su valor:}$$

$$\Delta = - \frac{M_{AB}}{E} \int_0^l \frac{x}{I} dx + \frac{(M_{AB} + M_{BA})}{E} \int_0^l \frac{x^2}{I} dx \quad (24)$$

Sustituyendo las ecuaciones 19 y resolviendo el sistema se determinan los  $M'$ .

$$M_{AB}' = \frac{K_1 + K_3}{K_1 K_2 - K_3^2} \Delta EI \quad M_{BA}' = \frac{K_2 + K_3}{K_1 K_2 - K_3^2} \Delta EI \quad (25)$$

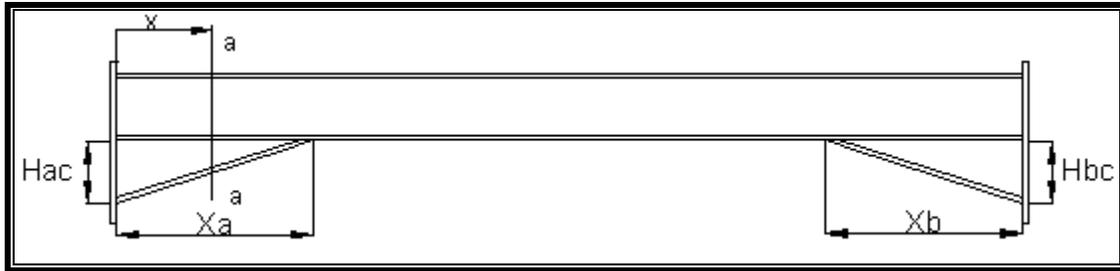
Las ecuaciones 19, 20, 23 y 25 han sido tomadas de *P. Charon, p.216, 217 y 219.*

El sistema final es:

$$M_{AB} = \frac{K_3 K_4 - K_1 K_5}{K_3^2 - K_1 K_2} l + \frac{K_3 E l^2}{K_3^2 - K_1 K_2} \left( \frac{K_1}{K_3} \theta_A + \theta_B \right) + \frac{K_1 + K_3}{K_1 K_2 - K_3^2} \Delta EI$$

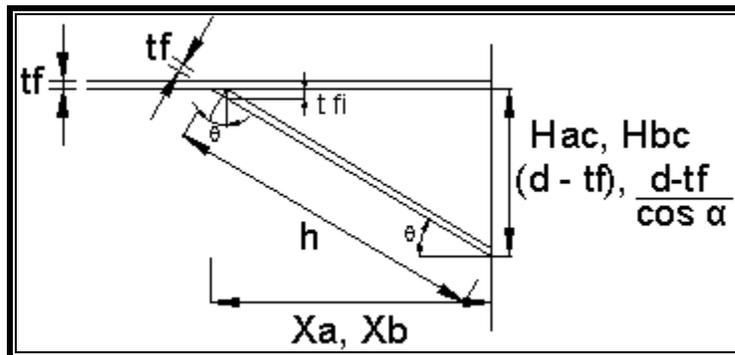
$$M_{BA} = \frac{K_2 K_4 - K_3 K_5}{K_3^2 - K_1 K_2} l + \frac{K_3 E l^2}{K_3^2 - K_1 K_2} \left( \theta_A + \frac{K_2}{K_3} \theta_B \right) + \frac{K_2 + K_3}{K_1 K_2 - K_3^2} \Delta EI$$

A continuación se presenta el cálculo de la inercia tal cual ha sido considerada en el programa.



X.13

De la figura X.13 previo al corte es importante hacer un detalle más cercano de la cartela y la viga según la figura X.14.



X.14 Detalle de la cartela y la viga

Como puede verse  $t_{fi}$  cuando se acerca al borde inferior de la viga es variable, este efecto no se ha despreciado, y ha sido considerado en el programa. Se muestra su valor según las ecuaciones siguientes.

$$\operatorname{tag} \theta = \frac{\operatorname{sen} \theta}{\operatorname{cos} \theta} = \frac{H_{AC}}{X_A} \quad (26)$$

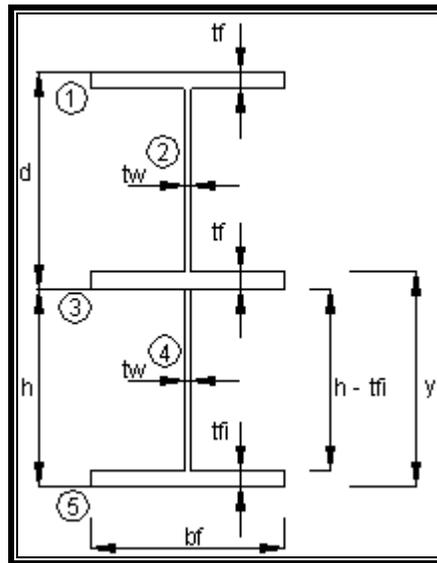
$$\operatorname{cos} \theta = \frac{t_f}{t_{fi}} \quad (27)$$

$$h = \sqrt{X_A^2 + H_{AC}^2} \quad (28)$$

De 27 se tiene que 
$$t_{fi} = \frac{t_f}{\cos \theta} \quad (29)$$

De la ecuación 26  $\cos \theta = \frac{\text{sen } \theta}{H_{AC}} X_A$  sustituyendo en 29

$$t_{fi} = \frac{t_f}{\text{sen } \theta * X_A} H_{AC} \Leftrightarrow t_{fi} = \frac{t_f \sqrt{H_{AC}^2 + X_A^2}}{X_A} \quad (30)$$



X.15 Cálculo de la inercia en la sección.

La figura X.15 representa un corte transversal de la viga en la parte que contiene cartela. Para determinar el centro de gravedad de la sección hay que hacer unas consideraciones.

La sección esta conformada por varias áreas, que tienen un brazo respecto a un eje horizontal que pasa por bf.

$$A_1 = bf * tf \quad ; \quad d_1 = h + d - \frac{tf}{2}$$

$$A_2 = tw * (d - 2tf) \quad ; \quad d_2 = h + \frac{d}{2}$$

$$A_3 = bf * tf \quad ; \quad d_3 = h + \frac{tf}{2}$$

$$A_4 = tw * (h - tfi) \quad ; \quad d_4 = tfi + \frac{h - tfi}{2}$$

$$A_5 = bf * tfi \quad ; \quad d_5 = \frac{tfi}{2}$$

$$S = A_1d_1 + A_2d_2 + A_3d_3 + A_4d_4 + A_5d_5 \quad (26)$$

$$A_T = \sum_{i=1}^n A_i \quad (27)$$

La ordenada del centro de gravedad es  $\bar{Y}$

$$\bar{Y} = \frac{S}{A_T} \quad (29)$$

Por definición la inercia de una sección rectangular es:

$$I = \frac{bH^3}{12} \quad ; \text{donde } b \text{ es la base y } H \text{ es la altura del elemento pero en nuestro caso}$$

hay que aplicar el teorema de los ejes paralelos puesto que hay que trasladar todas las inercias al centro de gravedad. Su ecuación es la siguiente:

$I_c = I_{cg} + MD^2$  ; donde M es la masa que cada una de las áreas y D es la distancia de la masa al centro de gravedad,  $I_{cg}$  es la inercia propia del elemento.

$$I_1 = \frac{bf * tf^3}{12} ; I_2 = \frac{tw(d - 2tf)^3}{12} ; I_3 = \frac{bf * tf^3}{12} ; I_4 = \frac{tw * (h - tf)^3}{12} ; I_5 = \frac{bf * tfi^3}{12}$$

La inercia final es:

$$I = \sum_{i=1}^n (I_i + A_i * (d_i - \bar{Y})^2) \quad (30)$$

Planteamiento del sistema de ecuaciones:

Se debe de cumplir que la suma de momentos en cada junta sea nula, tal que:

$$\text{Junta } B \Rightarrow M_{BA} + M_{BC} = 0$$

$$\text{Junta } C \Rightarrow M_{CB} + M_{CD} = 0$$

$$\text{Junta } D \Rightarrow M_{DC} + M_{DE} = 0 \quad \text{y que en la junta } C \text{ las fuerzas cortantes y}$$

axiales deben ser iguales.

$$H_{CB} = H_{CD}$$

$$V_{CB} = V_{CD}$$

Una vez planteado el sistema se deben de sustituir los valores de  $M_{ij}$  y proceder a resolver el sistema.

Todas las demostraciones que en este capítulo se han hecho forman parte del programa **AcartBeam V1.3**



## XI.1 Momentos de empotramiento para una carga uniformemente distribuida.

HEA	Relación de inercia	Relación entre tamaño de la cartela y longitud de la viga								
		0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
100	0,229	1,100	1,135	1,162	1,181	1,192	1,195	1,189	1,174	1,149
120	0,229	1,102	1,137	1,163	1,182	1,193	1,196	1,189	1,174	1,184
140	0,227	1,104	1,138	1,165	1,187	1,194	1,197	1,190	1,174	1,148
160	0,226	1,105	1,140	1,167	1,185	1,196	1,198	1,191	1,174	1,148
180	0,226	1,107	1,142	1,168	1,186	1,197	1,198	1,191	1,174	1,148
200	0,224	1,108	1,143	1,170	1,188	1,198	1,199	1,192	1,175	1,148
220	0,223	1,109	1,145	1,171	1,190	1,199	1,200	1,193	1,175	1,148
240	0,222	1,110	1,147	1,173	1,191	1,200	1,201	1,193	1,176	1,148
260	0,221	1,111	1,148	1,175	1,192	1,202	1,202	1,194	1,177	1,148
280	0,220	1,112	1,150	1,177	1,194	1,203	1,204	1,195	1,177	1,149
300	0,218	1,113	1,151	1,178	1,195	1,205	1,205	1,196	1,178	1,149
320	0,217	1,113	1,151	1,179	1,197	1,206	1,206	1,198	1,179	1,150
340	0,216	1,114	1,152	1,180	1,198	1,207	1,207	1,198	1,179	1,150
360	0,212	1,114	1,153	1,182	1,200	1,210	1,210	1,201	1,181	1,152
400	0,208	1,116	1,156	1,185	1,204	1,213	1,213	1,204	1,184	1,154
450	0,203	1,116	1,158	1,188	1,207	1,217	1,217	1,207	1,187	1,156
500	0,198	1,117	1,159	1,190	1,210	1,220	1,220	1,210	1,190	1,158
550	0,193	1,118	1,161	1,192	1,213	1,223	1,224	1,213	1,192	1,160
600	0,189	1,119	1,163	1,195	1,216	1,227	1,227	1,217	1,195	1,162
650	0,184	1,120	1,164	1,197	1,219	1,230	1,230	1,220	1,198	1,164
700	0,179	1,121	1,166	1,199	1,222	1,232	1,233	1,223	1,201	1,166
800	0,171	1,122	1,168	1,203	1,226	1,238	1,239	1,228	1,205	1,170
900	0,165	1,123	1,170	1,206	1,230	1,242	1,244	1,233	1,210	1,172
1000	0,158	1,125	1,172	1,208	1,233	1,247	1,248	1,238	1,214	1,175

HEB	Relación de inercia	Relación entre tamaño de la cartela y longitud de la viga								
		0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
100	0,230	1,100	1,134	1,161	1,181	1,192	1,195	1,190	1,175	1,150
120	0,299	1,101	1,136	1,163	1,182	1,193	1,196	1,190	1,175	1,149
140	0,299	1,103	1,138	1,164	1,183	1,194	1,196	1,190	1,175	1,149
160	0,227	1,105	1,140	1,166	1,185	1,195	1,198	1,191	1,175	1,150
180	0,226	1,106	1,141	1,168	1,186	1,197	1,199	1,192	1,176	1,150
200	0,225	1,107	1,142	1,169	1,188	1,198	1,200	1,193	1,176	1,150
220	0,224	1,108	1,144	1,171	1,189	1,199	1,201	1,193	1,177	1,150
240	0,223	1,109	1,146	1,172	1,191	1,200	1,202	1,194	1,177	1,150
260	0,222	1,110	1,147	1,174	1,192	1,202	1,203	1,195	1,178	1,150
280	0,220	1,111	1,148	1,176	1,194	1,203	1,204	1,196	1,178	1,151
300	0,219	1,111	1,150	1,177	1,195	1,205	1,205	1,197	1,179	1,151
320	0,217	1,112	1,150	1,178	1,197	1,206	1,207	1,198	1,180	1,152
340	0,215	1,112	1,151	1,180	1,198	1,208	1,208	1,200	1,182	1,153
360	0,213	1,113	1,152	1,181	1,200	1,209	1,210	1,201	1,183	1,154
400	0,208	1,114	1,154	1,184	1,203	1,213	1,213	1,204	1,185	1,156
450	0,203	1,115	1,156	1,187	1,206	1,216	1,217	1,208	1,188	1,158
500	0,199	1,116	1,158	1,189	1,209	1,220	1,220	1,211	1,191	1,160
550	0,193	1,117	1,160	1,192	1,213	1,223	1,224	1,214	1,194	1,162
600	0,189	1,118	1,162	1,194	1,215	1,226	1,227	1,217	1,196	1,164
650	0,184	1,119	1,163	1,196	1,218	1,229	1,230	1,220	1,199	1,165
700	0,180	1,120	1,164	1,198	1,221	1,232	1,233	1,223	1,202	1,168
800	0,172	1,121	1,167	1,202	1,225	1,238	1,239	1,229	1,207	1,171
900	0,165	1,122	1,169	1,205	1,229	1,242	1,244	1,234	1,211	1,174
1000	0,159	1,123	1,171	1,207	1,232	1,246	1,248	1,238	1,215	1,177

IPE Designación	Relación de inercia	Relación entre tamaño de la cartela y longitud de la viga								
		0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
80	0,213	1,101	1,137	1,166	1,186	1,198	1,202	1,197	1,181	1,156
100	0,212	1,103	1,139	1,167	1,187	1,200	1,203	1,197	1,182	1,156
120	0,211	1,105	1,141	1,169	1,189	1,200	1,204	1,198	1,182	1,156
140	0,210	1,107	1,143	1,170	1,190	1,201	1,204	1,198	1,182	1,156
160	0,209	1,109	1,144	1,172	1,191	1,203	1,205	1,199	1,183	1,156
180	0,209	1,110	1,146	1,173	1,193	1,204	1,206	1,200	1,183	1,156
200	0,208	1,111	1,148	1,175	1,194	1,205	1,207	1,200	1,184	1,156
220	0,207	1,112	1,149	1,177	1,195	1,206	1,208	1,201	1,184	1,156
240	0,207	1,113	1,151	1,178	1,197	1,207	1,209	1,201	1,184	1,156
270	0,204	1,115	1,153	1,181	1,199	1,209	1,211	1,203	1,185	1,156
300	0,202	1,117	1,155	1,183	1,202	1,212	1,213	1,205	1,187	1,157
330	0,200	1,118	1,157	1,185	1,204	1,214	1,215	1,206	1,188	1,158
360	0,198	1,118	1,158	1,187	1,206	1,216	1,216	1,208	1,189	1,159
400	0,195	1,120	1,160	1,190	1,209	1,219	1,219	1,210	1,191	1,160
450	0,190	1,120	1,162	1,192	1,213	1,223	1,223	1,214	1,194	1,162
500	0,186	1,122	1,164	1,195	1,216	1,226	1,226	1,217	1,196	1,164
550	0,182	1,123	1,166	1,198	1,219	1,229	1,230	1,220	1,199	1,166
600	0,178	1,123	1,167	1,200	1,221	1,232	1,232	1,222	1,201	1,167
IPN Designación	Relación de inercia	Relación entre tamaño de la cartela y longitud de la viga								
		0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
80	0,212	1,101	1,137	1,166	0,186	1,199	1,203	1,198	1,183	1,158
100	0,212	1,103	1,139	1,167	1,187	1,199	1,203	1,198	1,183	1,157
120	0,210	1,105	1,140	1,168	1,189	1,201	1,204	1,199	1,186	1,158
140	0,209	1,106	1,142	1,170	1,190	1,202	1,205	1,200	1,184	1,158
160	0,208	1,108	1,144	1,172	1,191	1,203	1,206	1,200	1,185	1,159
180	0,206	1,109	1,145	1,173	1,193	1,204	1,207	1,201	1,185	1,159
200	0,205	1,110	1,147	1,175	1,194	1,206	1,208	1,202	1,186	1,159
220	0,204	1,111	1,148	1,176	1,196	1,207	1,209	1,203	1,187	1,160
240	0,203	1,112	1,150	1,178	1,197	1,208	1,210	1,204	1,187	1,160
260	0,202	1,113	1,151	1,179	1,199	1,209	1,212	1,205	1,188	1,161
280	0,200	1,114	1,152	1,181	1,200	1,210	1,213	1,206	1,189	1,161
300	0,199	1,114	1,153	1,182	1,201	1,212	1,214	1,207	1,190	1,162
320	0,197	1,115	1,154	1,183	1,203	1,214	1,216	1,208	1,191	1,163
340	0,196	1,115	1,155	1,184	1,204	1,215	1,217	1,209	1,192	1,163
360	0,194	1,116	1,156	1,186	1,205	1,216	1,218	1,210	1,193	1,164
380	0,193	1,116	1,157	1,187	1,207	1,218	1,219	1,212	1,194	1,165
400	0,192	1,117	1,158	1,188	1,208	1,219	1,220	1,213	1,195	1,165
425	0,190	1,117	1,158	1,189	1,209	1,220	1,222	1,214	1,196	1,166
450	0,188	1,117	1,159	1,190	1,211	1,222	1,223	1,215	1,197	1,167
475	0,187	1,118	1,160	1,191	1,212	1,223	1,225	1,216	1,198	1,168
500	0,185	1,118	1,160	1,192	1,213	1,224	1,226	1,218	1,199	1,168
550	0,183	1,118	1,161	1,193	1,215	1,227	1,228	1,220	1,200	1,169
600	0,179	1,118	1,162	1,195	1,217	1,229	1,231	1,224	1,203	1,171

$Relación\ de\ inercia = (inercia\ de\ sección\ a\ L/2) / (inercia\ de\ sección\ en\ el\ extremo)$

$M_{ab} = -M_{ba} = X \cdot Q \cdot L^2 / 12$ , donde  $Q$  es el valor de la carga distribuida uniforme y  $L$  la longitud de la viga.

## **XI.2 Memoria fotográfica.**

Hangar en Aeropuerto Metropolitano, Ocumare del Tuy, Edo. Miranda fotos

XI.2.1, XI.2.2, XI.2.3, XI.2.4, XI.2.5 y XI.2.6



XI.2.1



XI.2.2



XI.2.3



XI.2.4



XI.2.5



XI.2.6

Galpón vía Punta de Piedras – Porlamar. Edo Nueva Esparta fotos XI.2.7 y XI.2.8



XI.2.7



XI.2.8

MANUAL DEL USUARIO

# AcartBeam V1.3

## INTRODUCCIÓN.

En las siguientes páginas se explicara como debe ser utilizado el programa **AcartBeam V1.3**. herramienta útil en el análisis estructural de pórticos que considera vigas acarteladas. Se contemplan los perfiles laminados de acero estructural como los elementos estructurales del pórtico. Sin embargo, ciertas consideraciones se puede utilizar como material el concreto armado.

Esta aplicación se ha desarrollado bajo el lenguaje de Visual Basic 6.0, por lo que presenta características típicas de Windows, tales como ventanas, menús desplegables, botones, comunes en los programa de uso comercial de Microsoft.

Con le programa **AcartBeam V1.3** se pueden determinar los desplazamientos en los nodos, las rotaciones de las juntas, los momentos en los extremos y por supuesto las reacciones. Las deformaciones por cargas axiales, corte y cambios de temperatura no se han considerado, pero esto no resta validez a los resultados ya que las deformaciones por carga axial, corte y cambios de temperatura son muy pequeñas en este tipo de pórticos para galpones.

La versión 1.0 fue mejorada hasta la actual versión 1,3 que incorpora mejores ayudas y gráficos que permiten comprender mejor el funcionamiento del programa de tal forma de poder analizar los resultados en forma correcta.

Para instalar el programa **AcartBeam V1.3** correctamente el usuario deberá seguir las instrucciones dadas por el programa de instalación, el cual se presentara como se muestra en figura 1.



Figura 1

Si el usuario hace clic en el botón aceptar el sistema procederá a indicar una serie de instrucciones, la instalación del programa no tardara más de tres minutos si el usuario cuenta con un computador equivalente a un Pentium cuatro, con Windows XP. En caso de que decida suspender la instalación con solo hacer clic en el botón salir se cancelara la acción.

Si el programa se ha instalado correctamente el usuario podrá visualizar en el **menú inicio**, renglón **todos los programas** el sistema **vigas acarteladas en galpones**.

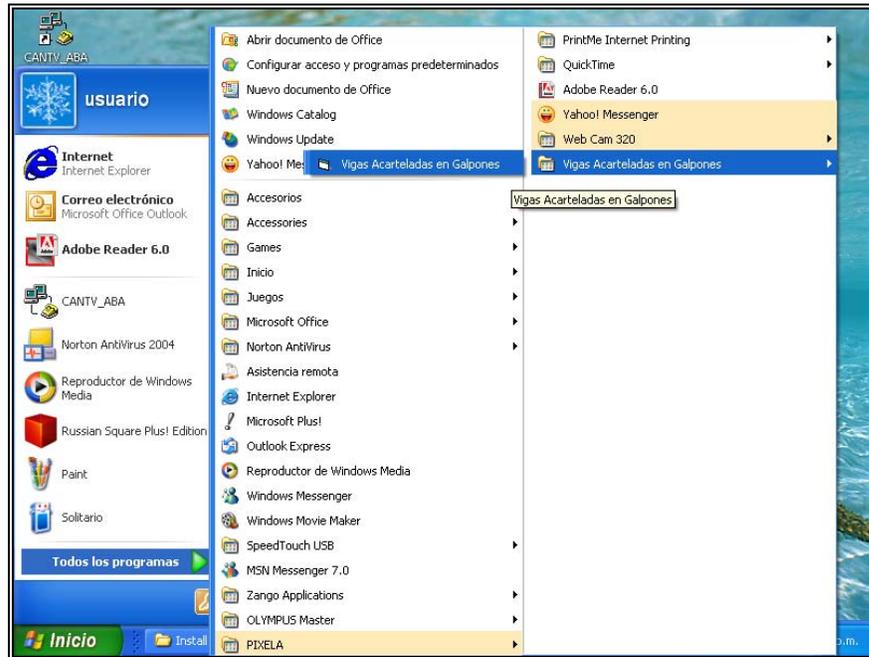


Figura 2

Al usuario hacer doble clic en el renglón identificado con el nombre del programa activara la carga del mismo donde se mostrara en la pantalla del computador la presentación del programa.



Figura 3

Al hacer click en el botón continuar de la pantalla de inicio del programa el usuario activara la carga de la página principal de la aplicación, desde la cual tendrá la posibilidad de suspender la activación del sistema o crear bases de datos, cuyos valores serán utilizados para los cálculos que el operador requiera. (Figura 4)

El usuario podrá introducir en la página principal del sistema las características principales del pórtico que desea estudiar, como son:

- Valor del ángulo formado entre sus elementos en grados siempre  $>0$
- Longitud de la cartela, expresada en metros
- Altura de los elementos, expresada en metros
- Luz del pórtico, expresada en metros

Propiedades de pórtico y cargas aplicadas - AcartBeam V1.3

Archivo Consulta Cálculo Acerca de...

Módulo de Elasticidad  x 10<sup>6</sup> kg/cm<sup>2</sup>  Nombre del archivo de texto

Rotación Junta B = --Vacio--  
 Rotación Junta C = --Vacio--  
 Rotación Junta D = --Vacio--

Desplaz. horizontal en B = --Vacio--  
 Desplaz. horizontal en D = --Vacio--  
 Desplaz. Vertical en C = --Vacio--  
 Desplaz. Horizontal en C = --Vacio--

Ver valores de cortante

Elem.1 Elem.2 Elem.3 Elem.4

Ángulo en grados   
 Longitud de Cartela Mts.   
 Altura Mts.   
 Luz Mts.

Momentos de empotramiento	[Kg-m]	Momentos definitivos
Mab= --Vacio--		Mab= --Vacio--
Mba= --Vacio--		Mba= --Vacio--
Mbc= --Vacio--		Mbc= --Vacio--
Mcb= --Vacio--		Mcb= --Vacio--
Mcd= --Vacio--		Mcd= --Vacio--
Mdc= --Vacio--		Mdc= --Vacio--
Mde= --Vacio--		Mde= --Vacio--
Med= --Vacio--		Med= --Vacio--

Calcular

Figura 4

Como se puede observar en la figura 5 el usuario podrá suspender la actividad iniciada en el programa al momento que él lo desee con solo dirigir el Mouse al renglón de archivo y seleccionar la opción **salir**. Lo que permite no tener que operar el sistema completo para poder abandonar el uso del mismo.

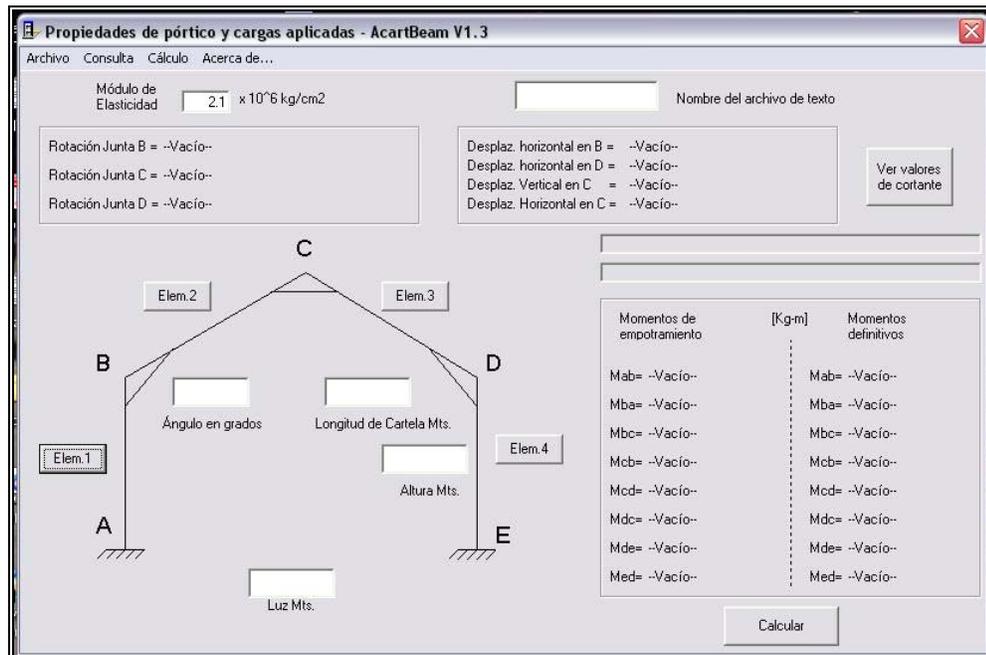


Figura 5

Si el usuario dirige el cursor del ratón al menú de **base de datos** y hace click en **edición y consulta de propiedades**, podrá constatar que el sistema carga una nueva pantalla en la cual podrá visualizar propiedades de un archivo ya existente o crear uno nuevo, como puede verse en la figura 6.

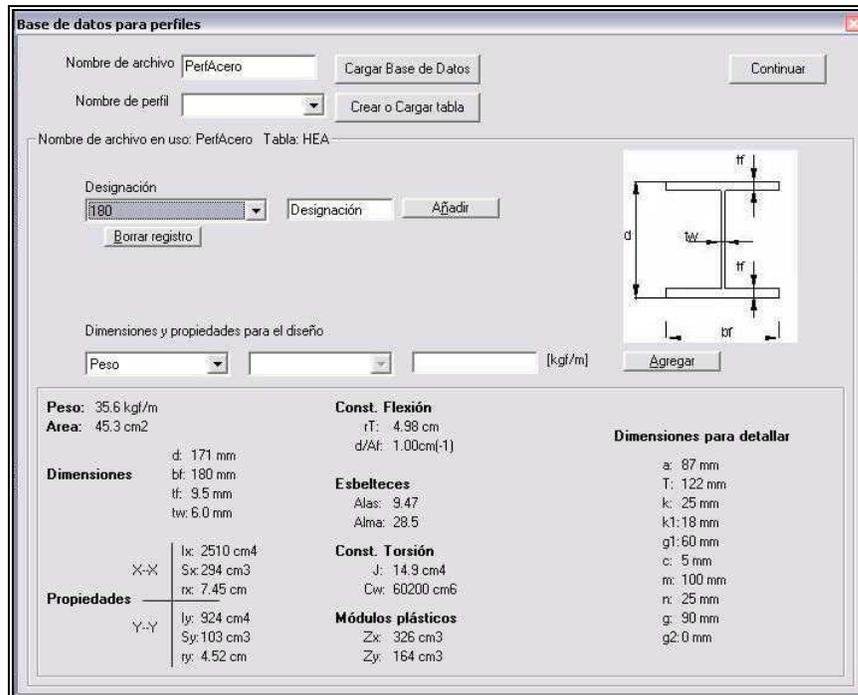


Figura 6

También podrá observarse que en el menú existe otro campo denotado con el nombre de (1) **“Cálculo”**, al hacer clic en este menú el programa procederá a hacer los cálculos necesarios para obtener los valores de las rotaciones en las juntas, los desplazamientos de los nodos **B**, **C**, **D**, los momentos de empotramiento y momentos finales por efecto de las cargas colocadas en los elementos.

El número **3** representa el conjunto de menús a los que se puede tener acceso. El número **4** muestra el lugar donde el usuario debe de colocar el nombre del archivo con el fin de poder ver los resultados impresos en un archivo de texto (.txt) que será guardado en la carpeta del programa, véase la figura 7.

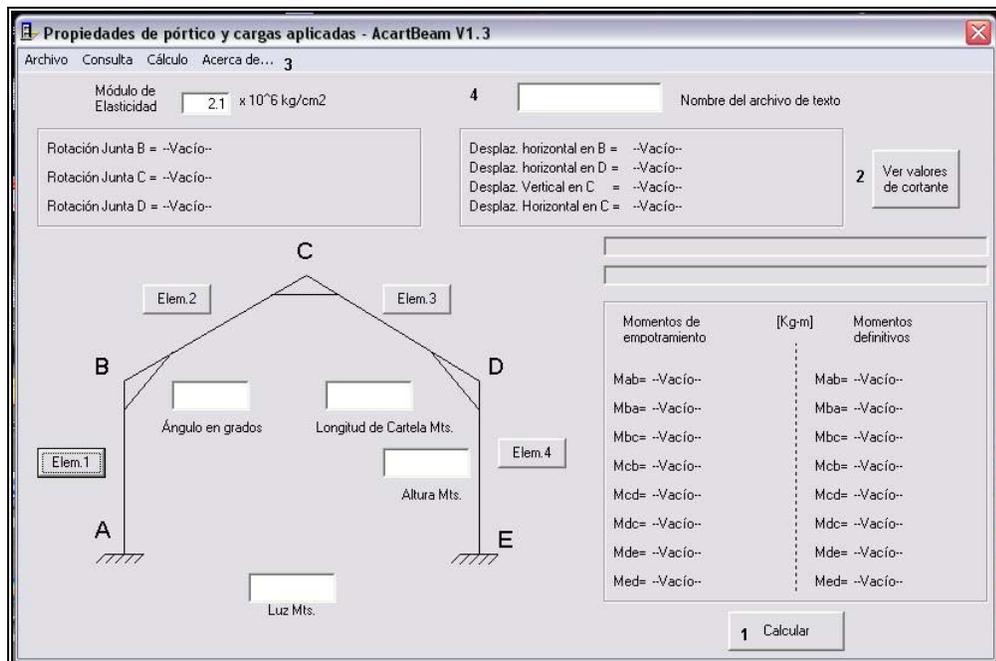


Figura 7

Estudiando con un poco más de detalle la pantalla principal del sistema:

Botón (2) “**ver valores de cortantes**”:

al hacer click sobre el botón el sistema mostrara una pantalla (figura 8) donde se puede visualizar los elementos del pórtico con sus respectivas cargas axiales y cortantes, en términos de solicitaciones y no de diagramas de cuerpo libre.

Es sumamente importante que el usuario este consiente de que los resultados son solicitaciones en los extremos por causa de las cargas aplicadas, pero no diagramas de cuerpo libre. Se considera positivo el corte si sube y la carga axial si sale del elemento en ambos extremos del elemento.

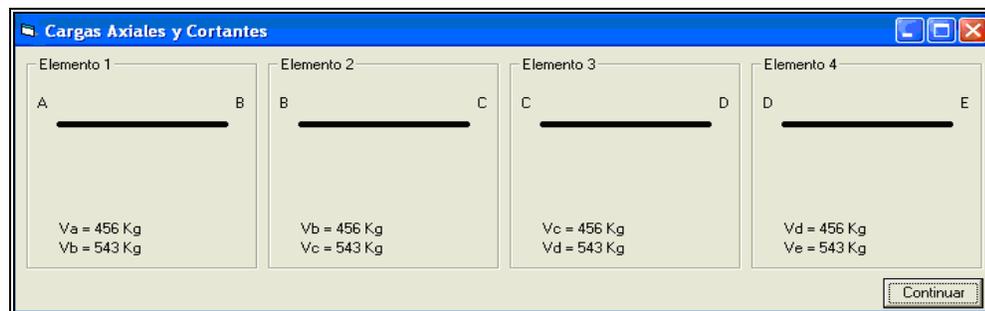


Figura 8

Menú “**base de datos**”: al hacer click sobre este menú el sistema procederá a cargar una nueva pantalla en la cual podrá crearse una nueva base de datos de perfiles o modificar alguna existente.

Base de datos para perfiles

Nombre de archivo: PerfAcero Cargar Base de Datos Continuar

Nombre de perfil: [ ] Crear o Cargar tabla

Nombre de archivo en uso: PerfAcero Tabla: HEA

Designación: I180 Designación: [ ] Agregar

Borrar registro

Dimensiones y propiedades para el diseño

Peso: [ ] [kgf/m] Agregar

<b>Peso:</b> 35.6 kgf/m	<b>Const. Flexión</b>	
<b>Area:</b> 45.3 cm <sup>2</sup>	rT: 4.98 cm	
	d/AF: 1.00cm(-1)	<b>Dimensiones para detallar</b>
<b>Dimensiones</b>	<b>Esbelleces</b>	a: 87 mm
d: 171 mm	Alas: 9.47	T: 122 mm
bf: 180 mm	Alma: 28.5	k: 25 mm
tf: 9.5 mm		k1: 18 mm
tw: 6.0 mm	<b>Const. Torsión</b>	g1: 60 mm
	J: 14.9 cm <sup>4</sup>	c: 5 mm
<b>Propiedades</b>	Cw: 60200 cm <sup>6</sup>	m: 100 mm
Ix: 2510 cm <sup>4</sup>	<b>Módulos plásticos</b>	n: 25 mm
Sx: 294 cm <sup>3</sup>	Zx: 326 cm <sup>3</sup>	g: 90 mm
rx: 7.45 cm	Zy: 164 cm <sup>3</sup>	g2: 0 mm
Iy: 924 cm <sup>4</sup>		
Sy: 103 cm <sup>3</sup>		
ry: 4.52 cm		

Figura 9

También contiene un submenú (ver figura 12) “**Tablas para cartelas**”: en el cual se muestran los factores para hallar los momentos de empotramiento  $M = \frac{Ql^2}{12} X$  de los perfiles HEA, HEB, IPE e IPN donde x es valor por el cual se debe multiplicar con el fin de saber el valor del momento en el extremo de la viga considerando la cartela hecha del mismo tipo de perfil que el de la viga. En la figura 13 se presentan las tablas para los perfiles arriba mencionados.

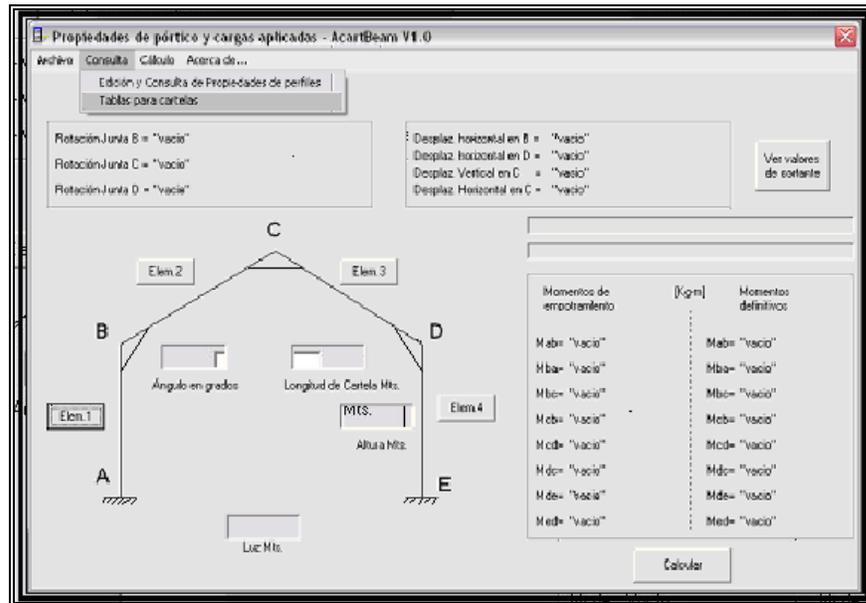


Figura 10

Momentos de empotramiento para una carga uniformemente distribuida

$$M_{ab} = -M_{ba} = X * P_1 * L^2 / 12$$

Valores de la tabla: X

HEA		HEB			IPN			IPE		
Designación	Relación de inercia	Relación entre tamaño de la cartela y longitud de la viga								
		0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
100	0,229	1,100	1,135	1,162	1,181	1,192	1,195	1,189	1,174	1,149
120	0,229	1,102	1,137	1,163	1,182	1,193	1,196	1,189	1,174	1,184
140	0,227	1,104	1,138	1,165	1,187	1,194	1,197	1,190	1,174	1,148
160	0,226	1,105	1,140	1,167	1,185	1,196	1,198	1,191	1,174	1,148
180	0,226	1,107	1,142	1,168	1,186	1,197	1,198	1,191	1,174	1,148
200	0,224	1,108	1,143	1,170	1,188	1,198	1,199	1,192	1,175	1,148
220	0,223	1,109	1,145	1,171	1,190	1,199	1,200	1,193	1,175	1,148
240	0,222	1,110	1,147	1,173	1,191	1,200	1,201	1,193	1,176	1,148
260	0,221	1,111	1,148	1,175	1,192	1,202	1,202	1,194	1,177	1,148
280	0,220	1,112	1,150	1,177	1,194	1,203	1,204	1,195	1,177	1,149
300	0,218	1,113	1,151	1,178	1,195	1,205	1,205	1,196	1,178	1,149
320	0,217	1,113	1,151	1,179	1,197	1,206	1,206	1,198	1,179	1,150
340	0,216	1,114	1,152	1,180	1,198	1,207	1,207	1,198	1,179	1,150
360	0,212	1,114	1,153	1,182	1,200	1,210	1,210	1,201	1,181	1,152
400	0,208	1,116	1,156	1,185	1,204	1,213	1,213	1,204	1,184	1,154
450	0,203	1,116	1,158	1,188	1,207	1,217	1,217	1,207	1,187	1,156
500	0,198	1,117	1,159	1,190	1,210	1,220	1,220	1,210	1,190	1,158
550	0,193	1,118	1,161	1,192	1,213	1,223	1,224	1,213	1,192	1,160
600	0,189	1,119	1,163	1,195	1,216	1,227	1,227	1,217	1,195	1,162
650	0,184	1,120	1,164	1,197	1,219	1,230	1,230	1,220	1,198	1,164
700	0,179	1,121	1,166	1,199	1,222	1,232	1,233	1,223	1,201	1,166
800	0,171	1,122	1,168	1,203	1,226	1,238	1,239	1,228	1,205	1,170
900	0,165	1,123	1,170	1,206	1,230	1,242	1,244	1,233	1,210	1,172
1000	0,158	1,125	1,172	1,208	1,233	1,247	1,248	1,238	1,214	1,175

Figura 11

Al introducir datos incorrectos o fuera de los límites permitidos por el programa el sistema mostrara un mensaje de error permitiéndole al usuario colocar únicamente datos reales.

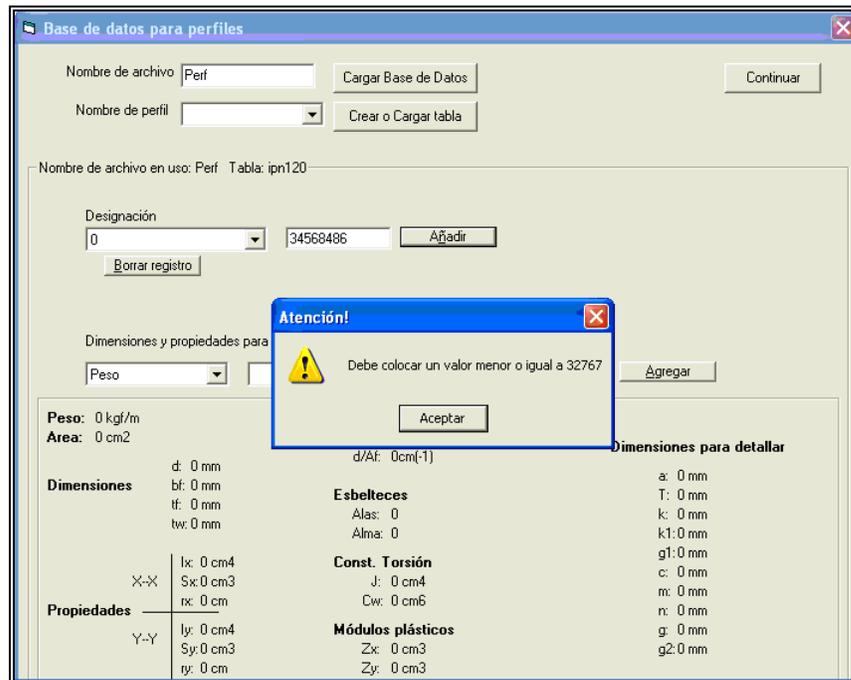


Figura 12

Se puede notar que el sistema requiere de una serie de datos que corresponden a dimensiones y propiedades necesarias para el diseño. Al presentar cada uno de estos requerimientos el programa muestra las unidades en las que debe introducirse cada una de ellas para evitar al usuario errores que puedan reflejar valores falsos como se ve en la figura 12.

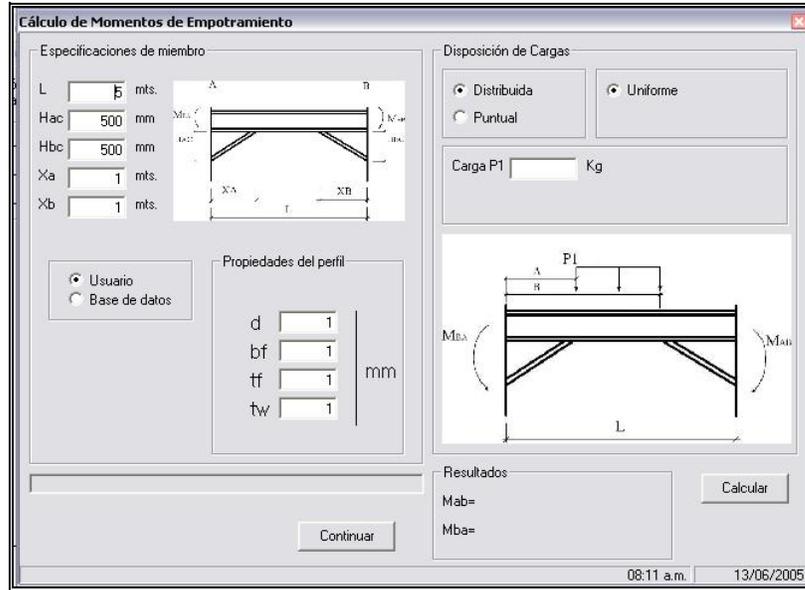


Figura 13

Menú “Cálculo”. Despliega un submenú denominado “Mom. De empotramiento” se hace click en él se abre otra ventana, donde se pueden determinar los valores de los momentos en los extremos de una viga empotrada, con cartelas y cargada uniformemente o con cargas puntuales (ver figura 13).

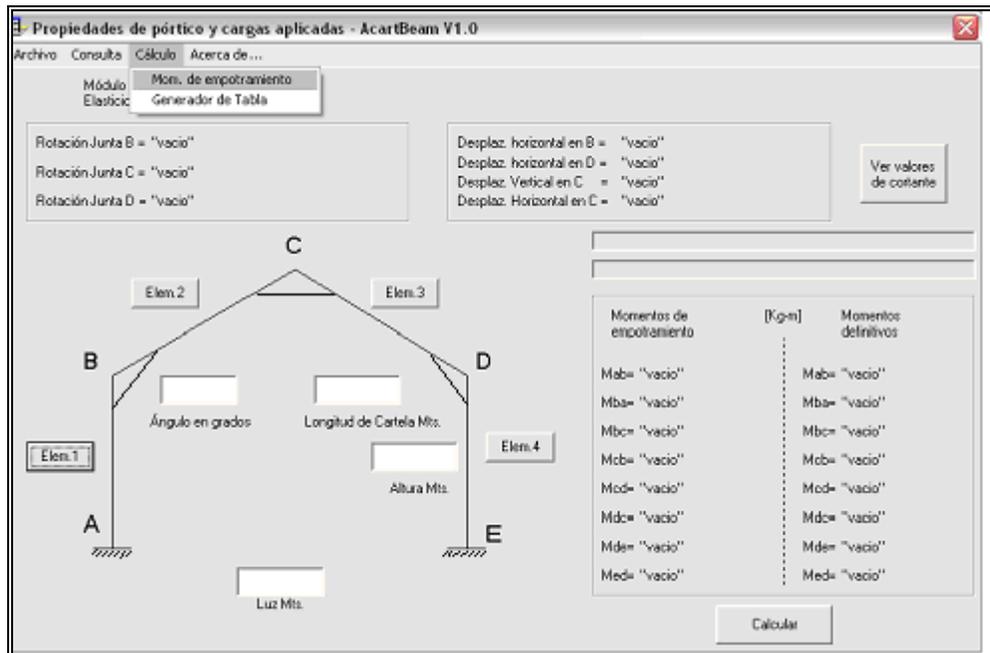


Figura 14

Menú **Generador de tablas**: se abre un submenú donde el usuario podrá examinar los perfiles ya existentes en la base de datos o crear el suyo con el fin de calcular en forma rápida los momentos de empotramiento de una viga cargada uniformemente para diferentes tipos de relaciones entre el tamaño de la cartela y la luz de la viga. Bajo la consideración de que la cartela es del mismo tipo de perfil que el propuesto en la viga por el usuario.

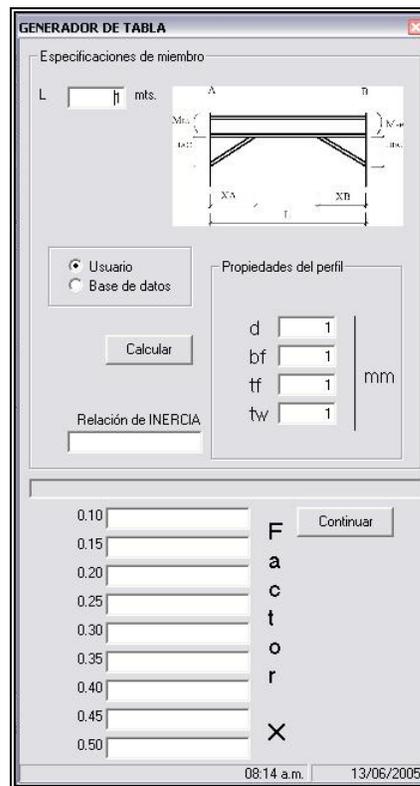


Figura 15

Luego que el sistema evalúe toda la información suministrada por el operador presentara una nueva pantalla en la cual se observaran las especificaciones iniciales dadas por el usuario, las propiedades del perfil, como también presenta la opción de introducir cargas tanto puntuales como distribuidas sobre el pórtico permitiendo así un estudio real de la influencia de la cartela que se escoja. Es importante aclarar que la tabla es valida para una viga horizontal, no inclinada.

Los resultados podrán ser guardados en un archivo de texto, al entrar al submenú “guardar archivo” previo deberá dar un nombre de lo contrario dará un mensaje de error. Este archivo de guardará en la carpeta del programa.

Al hacer click sobre el botón continuar se abrirá nuevamente la pantalla principal del programa permitiendo al usuario realizar cualquier tipo de modificación.

Es importante destacar que el sistema permitirá realizar estos cálculos sobre perfiles ya existentes en alguna base de datos o sobre algún perfil cuyas propiedades sean creadas por el usuario con lo que también es posible calcular los pórticos considerando el concreto armado como material. Las modificaciones para llevar a cabo este cálculo son:

- a) Modificar el módulo de elasticidad.
- b) Crear un perfil donde  $t_w$  sea igual  $d_f$ , y  $d$  igual a  $t_f$ .

Cabe la pena destacar que se podrán realizar tantos cálculos como requiera variando el tipo de carga y su distribución en el pórtico sin salir de la pantalla.

## XII. REFERENCIAS BIBLIOGRÁFICAS

Norris, C.H., y Wilbur, J. B. Análisis elemental de estructuras. Libros McGraw-Hill, México, 1973

McCormac, J. C., y Nelson, J. K. Análisis de estructuras método clásico y matricial. Alfaomega Grupo Editor, México 2002

Hibbeler, Russel C. Análisis estructural. Prentice-Hall Hispanoamericana, S.A. México, 1997

Wang, C.-K. Statically indeterminate structures. McGraw-Hill Book Company, Inc, Nueva York, 1953

Charon, P. La méthode de Cross et le calcul pratique des constructions hyperstatiques. Éditions Eyrolles, Paris, 1953

<http://inti.gov.ar/cirsoc/CIRSOC301/Reglamento/glosario>.

<http://ciencianet.com/p9.html>

<http://www.virtual.unal.edu.co>

Centro de enseñanza asistida por computadora. Programación con visual basic. Caracas, Universidad Central de Venezuela, 2004

García de J, J, Rodríguez, J. I. y Brazales, A. Aprenda visual basic 6.0. Escuela superior de ingenieros industriales de la Universidad de Navarra. España, 1999