

## CAPITULO V

### 5. VISUAL BASIC

#### 5.1. PROGRAMA DE CÁLCULO

El desarrollo de aplicaciones de cómputo surge a partir de la necesidad de resolver problemas numéricos que involucran la realización de una cantidad considerable de cálculos numéricos.

En la ingeniería, como en otras disciplinas, los programas de computadora representan una de las herramientas de apoyo más importantes.

Por eso, son cada vez más los estudiantes, ingenieros y profesionistas en general quienes desarrollan aplicaciones de cómputo para ayudarse a resolver problemas.

Desde las tarjetas perforadas hasta nuestra fecha, la programación orientada a objetos ha tenido un notable desarrollo tecnológico y su atractivo es cada vez mayor.

Es por ello que un programa elaborado en Visual Basic 6.0 para el desarrollo de trabes armadas de acero representa una opción atractiva para todo aquél interesado en el tema, especialmente para el estudiante de licenciatura de ingeniería civil.

#### 5.2. INTRODUCCION A VISUAL BASIC

El lenguaje de programación BASIC (Beginner's All purpose Symbolic Instruction Code) nació en el año 1964 como una herramienta orientado a

principiantes, buscando una forma sencilla de realizar programas, empleando un lenguaje casi igual al usado cotidianamente y con instrucciones muy sencillas y escasas. Este lenguaje cubría casi todas las necesidades para la ejecución de programas.

La evolución del BASIC por los años 70 fue escasa, dado el auge que tomaron en aquella época lenguajes de alto nivel como el FORTRAN y el COBOL. Más adelante, con la aparición de los primeros ordenadores personales, a mediados de los ochenta, el BASIC resurgió como lenguaje de programación pensado para principiantes.

Sin embargo, fue gracias al entorno gráfico de Windows (Microsoft trademark), como VISUAL BASIC tomó ventajas sobre los otros lenguajes de cómputo y pudo satisfacer las necesidades que la programación requería.

Actualmente se han comercializado varias versiones de este producto desde su salida al mercado. Cada versión supera y mejora la anterior. Dados los buenos resultados a nivel profesional de este producto, Visual-Basic se ha convertido en la primera herramienta de desarrollo de aplicaciones en entorno Windows. En la mayor parte de las aplicaciones, las herramientas aportadas por VISUAL BASIC son mas que suficientes para lograr un programa fácil de realizar y de uso sencillo.

### **5.3. CARACTERISTICAS GENERALES DE VISUAL BASIC**

Visual Basic es una herramienta de diseño de aplicaciones para Windows, en la que estas se desarrollan a partir del diseño de una interfase gráfica. En una

aplicación Visual Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica. Es, por tanto, un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

La creación de un programa en Visual Basic lleva los siguientes pasos:

1. Creación de una interfase de usuario. Esta interfase será la principal vía de comunicación hombre-máquina, tanto para entrada de datos como para salida. Será necesario partir de una ventana (Formulario) a la que le iremos añadiendo los controles necesarios.
2. Definición de las propiedades de los controles (Objetos) que se hayan colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.
3. Generación del código asociado a los eventos que ocurran a estos objetos. A la respuesta a estos eventos (click, una tecla pulsada, etc.) se le llama Procedimiento. Deberá generarse de acuerdo a las necesidades del programa.
4. Generación del código del programa. Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. VISUAL BASIC ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos responden a un evento producido durante la ejecución del programa.

#### 5.4. USO DE VISUAL BASIC

La aplicación de VISUAL BASIC de Microsoft puede trabajar de dos modos distintos: en modo de diseño y en modo de ejecución. En modo de diseño el usuario construye interactivamente la aplicación, colocando controles en el formulario, definiendo sus propiedades y desarrollando funciones para gestionar los eventos.

La aplicación se prueba en modo ejecución. En ese caso el usuario actúa sobre el programa (introduce eventos) y prueba cómo responde el programa. Todas las propiedades de los objetos se establecen en modo de diseño.

Para iniciar Visual Basic 6.0, que es la versión utilizada en esta tesis, se pulsa sobre el botón Inicio/Programas/Microsoft Visual Basic 6.0 y se hace click sobre el icono del programa como se muestra en la Figura 5.1.



Figura 5.1 Acceso a Visual Basic.

En la ventana de Nuevo proyecto que aparece, se elige el icono EXE estándar, para crear un programa típico como se observa en la Figura 5.2.

El siguiente paso es la creación de la interfase gráfica y su respectiva vinculación con el programa de cálculo. El diseño de la interfase gráfica tiene como objeto básico al formulario. Los formularios conforman la base de las aplicaciones, dado que en ellos se posicionan los controles y objetos con los que

va a actuar el usuario. En las figuras siguientes se muestra el objeto formulario y sus respectivas propiedades (Figura 5.3 y Figura 5.4).



Figura 5.2 Creación de programa.

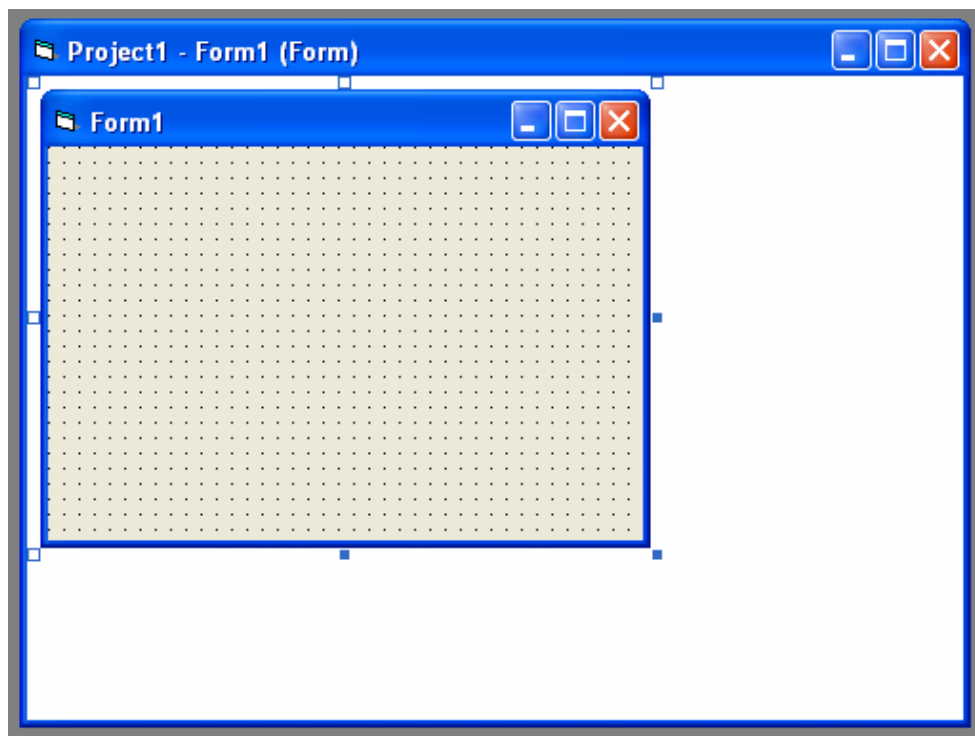


Figura 5.3 Objeto de Visual Basic.

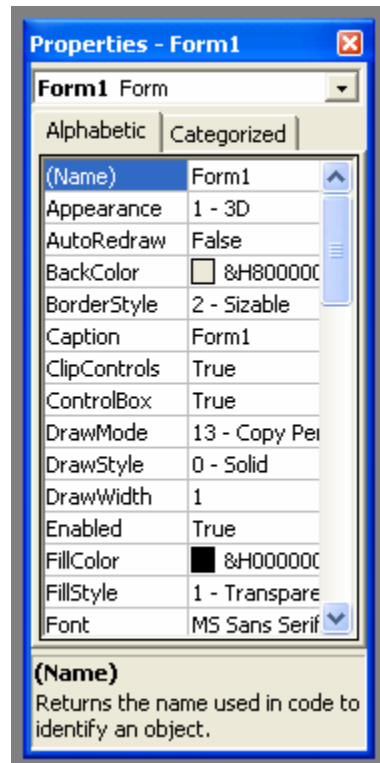


Figura 5.4 Cuadro de propiedades.

Cada uno de los controles con los que se trabajan en Visual Basic tienen sus correspondientes propiedades; estas se modifican con fines visuales y de función según la índole del mismo. En las propiedades se establecen aspectos como lo son el tipo de letra, el color del fondo, las dimensiones, la posición en el formulario, el tipo de borde, etc.

Uno de los objetos más importante y usados en el programa fue el botón comando. Este objeto tiene la facultad de llevar a cabo una función interna del programa, es decir, lleva a cabo una rutina de cálculo. Este comando es activado al darle click sobre el botón y desencadena una serie de instrucciones asignadas mediante código. En la Figura 5.5 se observa dónde se encuentra el elemento botón de comando.

Otro objeto que es de mucha utilidad en la construcción de programas de cálculo, ofreciendo espacio al programador para insertar información, es el control etiqueta o label. El control etiqueta se muestra en la Figura 5.6.

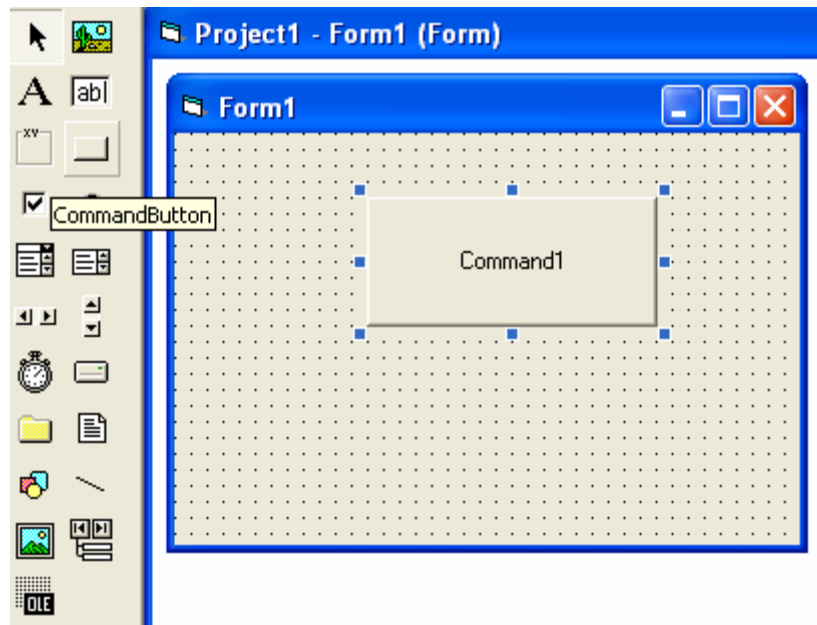


Figura 5.5 Botón Comando.

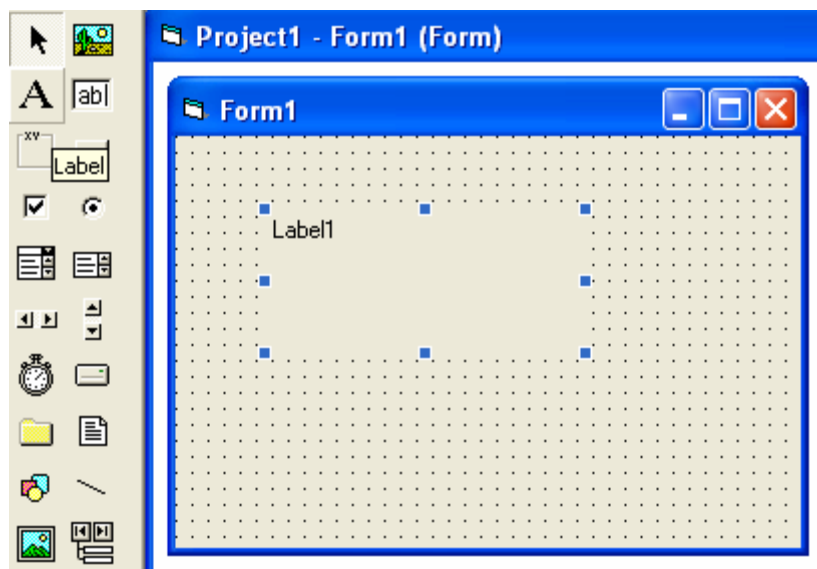


Figura 5.6 Label.

Uno de los controles más importantes de Visual Basic son las cajas de texto, ya que son éstos en los que el usuario introduce los valores o datos con los que va a operar el programa. Las cajas de texto sirven también para mostrar resultados, operaciones internas del programa, introducir información numérica o de texto, etc. La caja de texto se presenta en la Figura 5.7.

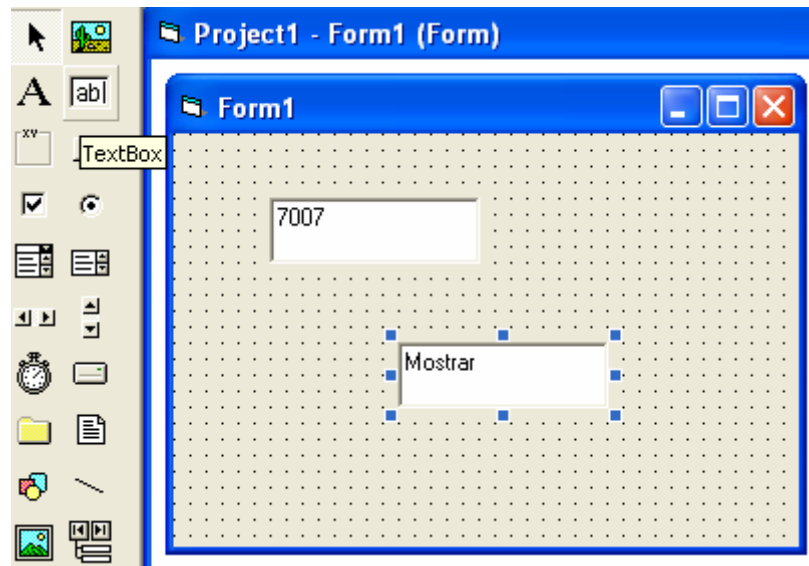


Figura 5.7 Caja de Texto.

Otro control que tiene como finalidad la de captar datos de inicio de forma similar al textbox, es el combobox. Este resulta una combinación de una lista desplegable y de una caja de texto donde la información que se requiere está ya definida, por lo que el usuario solo selecciona el valor numérico que necesita usar. En la Figura 5.8 se muestra la caja combinada.

Las imágenes son controles que pueden contener un gráfico en su interior. Es importante señalar que la imagen contenida no adopta el tamaño designada por el botón imagen, de modo que es importante dar el tamaño adecuado del



botón imagen para que ésta pueda visualizarse de una forma adecuada. En la Figura 5.9 se muestra el control imagen.

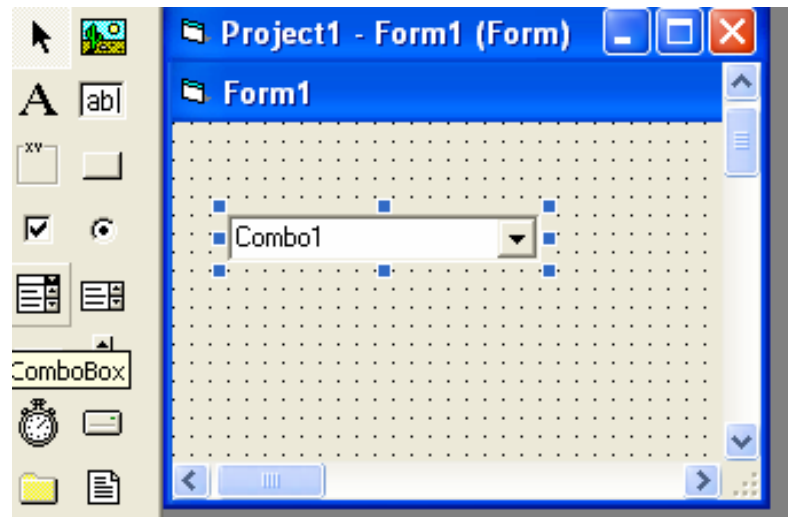


Figura 5.8 Caja Combinada.

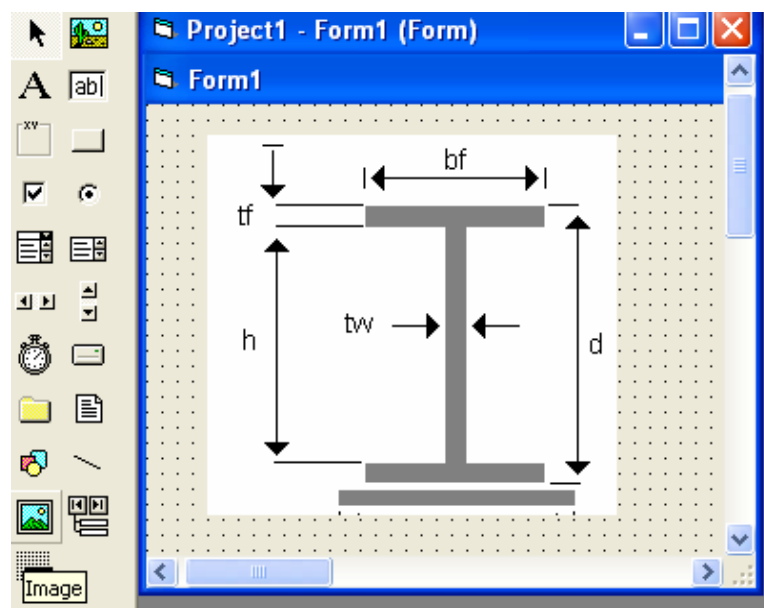


Figura 5.9 Imagen.

El control timer es un objeto que sólo utilizaremos en la introducción del programa, es decir, en la presentación del software. Este objeto tiene como

función ajustar el tiempo de ejecución de algún suceso dentro del programa, establecido en milisegundos. En la Figura 5.10 se muestra el objeto timer.

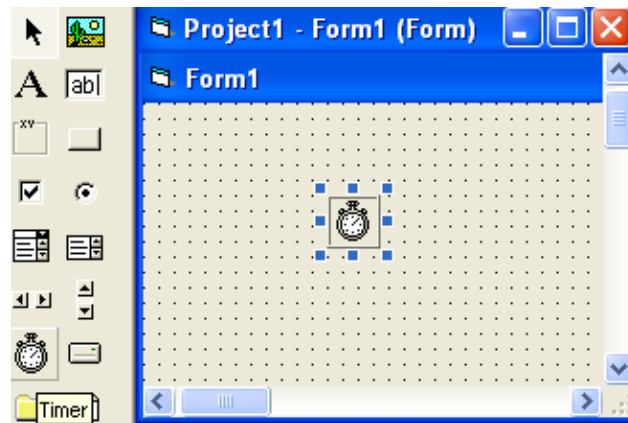


Figura 5.10 Timer.

Todos los objetos que se han utilizado en la construcción del programa se encuentran en la barra de controles de Visual Basic 6.0 al momento de crear un nuevo formulario como se muestra en la Figura 5.11.

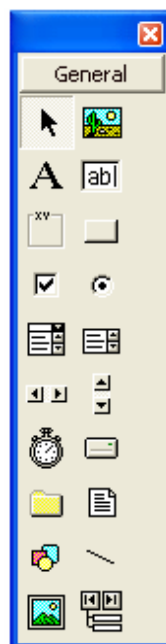


Figura 5.11 Controles.

Una de las aplicaciones de esta tesis en el uso de Visual Basic 6.0 radica en el uso de cajas de texto. En el momento en que una caja de datos es seleccionada, ésta cambiará de color ciertas imágenes.

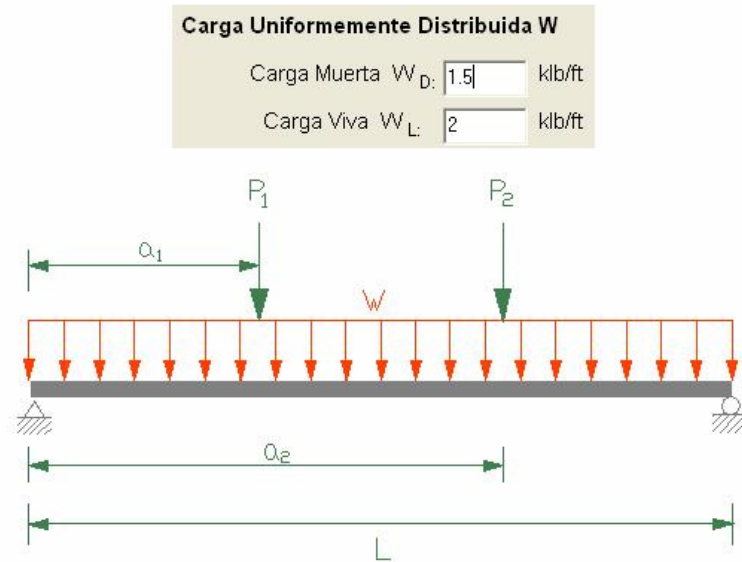


Figura 5.12 Carga uniformemente distribuida con diferente color .

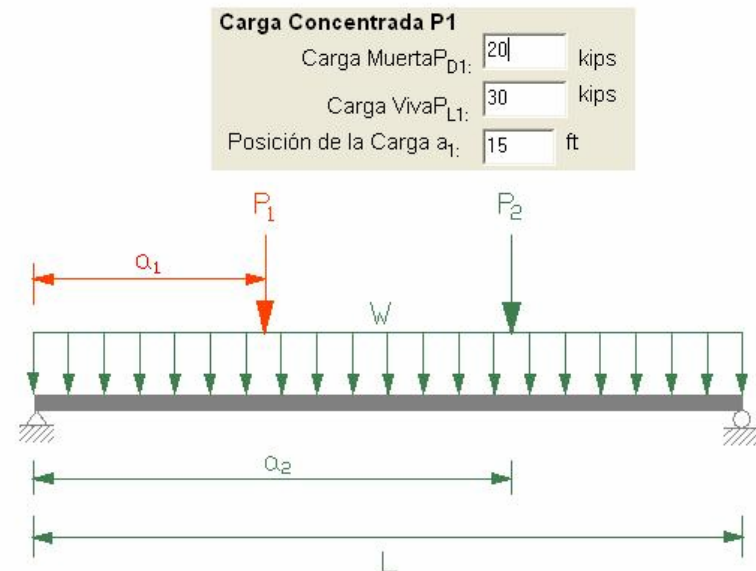


Figura 5.13 Carga concentrada con diferente color.

Por ejemplo; en caso de ser seleccionada alguna caja de datos inherente a la carga uniformemente distribuida, la imagen que aparecerá será la Figura 5.12.

En caso de seleccionar alguna caja de datos referente a una carga concentrada, la imagen se actualizará, como se observa en la Figura 5.13

Código del formulario1 *cajas de texto*:

```
Private Sub Text2_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga222.bmp")
```

```
End Sub
```

```
Private Sub Text3_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga222.bmp")
```

```
End Sub
```

```
Private Sub Text4_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga444.bmp")
```

```
End Sub
```

```
Private Sub Text5_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga444.bmp")
```

```
End Sub
```

```
Private Sub Text6_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga333.bmp")
```

```
End Sub
```

```
Private Sub Text7_GotFocus()
```

```
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga333.bmp")
```

```
End Sub
```

```
Private Sub Text9_GotFocus()  
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga444.bmp")  
End Sub  
  
Private Sub Text10_GotFocus()  
Picture1.Picture = LoadPicture(App.Path & "\dibujo-viga333.bmp")  
End Sub
```

Las imágenes de resultados también se actualizan dependiendo del tipo de cargas a las que la trabe esté sujeta; ya sea que esté sujeta a una carga distribuida, a una carga concentrada o dos cargas concentradas (ver Figura 5.14).

Código del formulario2 *trabe armada*:

```
Private Sub DatosR1()  
If P1 > 0 And P2 = 0 Then  
    If a1 < L / 2 Then  
        Picture1.Picture = LoadPicture(App.Path & "\RT4-R4-1.bmp")  
    ElseIf a1 = L / 2 Then  
        Picture1.Picture = LoadPicture(App.Path & "\RT3-R4-2.bmp")  
    Else  
        Picture1.Picture = LoadPicture(App.Path & "\RT7-R1.bmp")  
    End If  
Elseif P2 > 0 And P1 = 0 Then  
    If a2 < L / 2 Then  
        Picture1.Picture = LoadPicture(App.Path & "\RT5-R5-1.bmp")  
    ElseIf a2 = L / 2 Then
```

```
Picture1.Picture = LoadPicture(App.Path & "\RT10-R5-2.bmp")  
Else  
    Picture1.Picture = LoadPicture(App.Path & "\RT8-R2.bmp")  
End If  
Elseif P1 = 0 And P2 = 0 Then  
    Picture1.Picture = LoadPicture(App.Path & "\RT1-R3-2.bmp")  
Elseif P1 > 0 And P2 > 0 Then  
    If a1 = L - a2 Then  
        Picture1.Picture = LoadPicture(App.Path & "\RT2-R3-1.bmp")  
    Elseif a1 < L - a2 Then  
        Picture1.Picture = LoadPicture(App.Path & "\RT6-R6-1.bmp")  
    Else  
        Picture1.Picture = LoadPicture(App.Path & "\RT9-R6-2.bmp")  
    End If  
End If  
End If  
End Sub
```

En las imágenes donde aparecen resultados finales tenemos otra de las aportaciones importantes de Visual Basic. Los resultados numéricos aparecen dentro de las imágenes y también pueden ser impresos.

Todos los valores se colocan por medio de coordenadas cartesianas tomando como límites el tamaño de la imagen. Ver Figura 5.15.

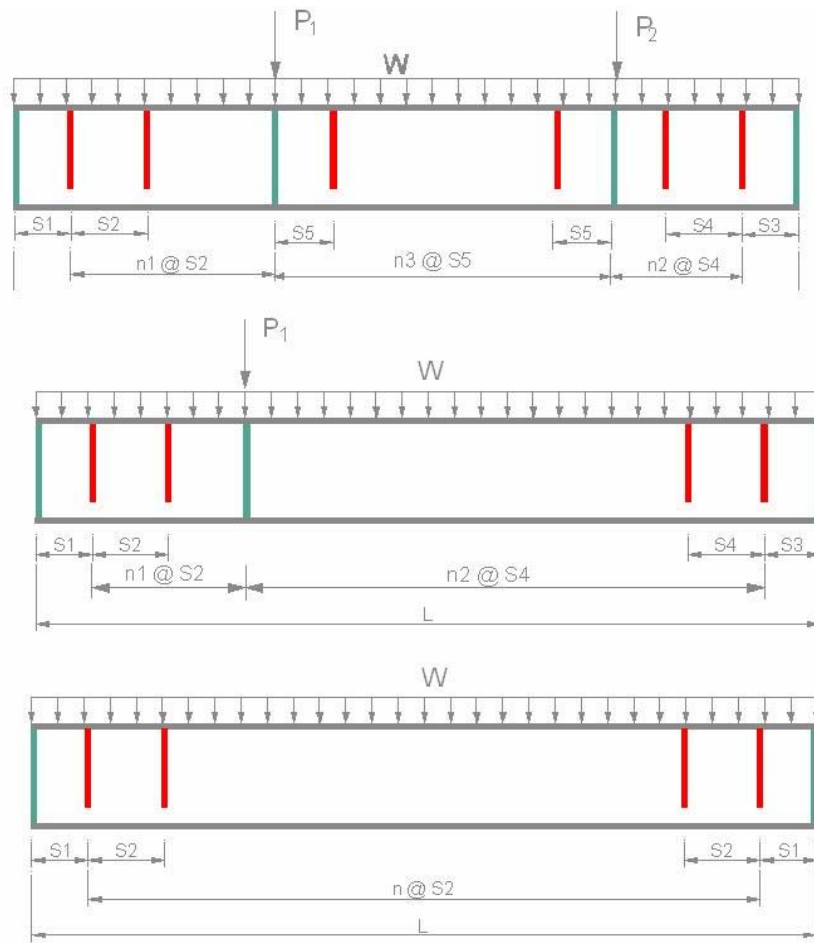


Figura 5.14 Trabe armada con diferentes cargas aplicadas.

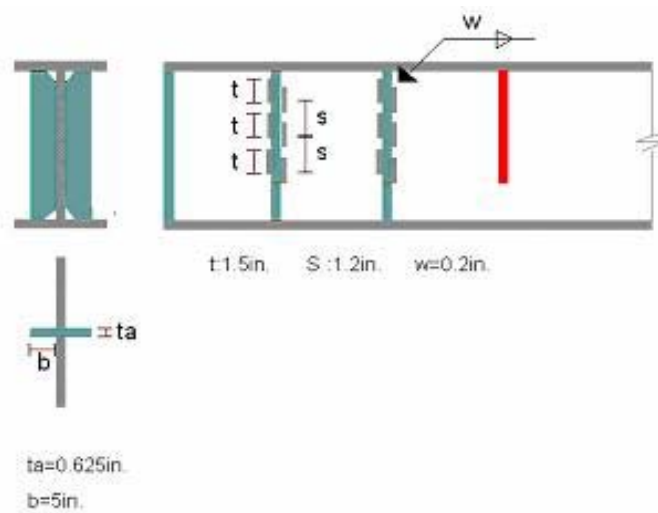


Figura 5.15 Resultados numéricos dentro de la imagen.

---

Código del formulario2 *atiesador de apoyo*:

Private Sub **DatosR3()**

Picture3.Font = "Arial"

Picture3.FontSize = 13

Picture3.CurrentX = 200

Picture3.CurrentY = 220

Picture3.Print "t:" & Round(t, 1) & "in."

Picture3.CurrentX = 280

Picture3.CurrentY = 220

Picture3.Print "S :" & Round(s\_apoyo, 1) & "in."

Picture3.CurrentX = 370

Picture3.CurrentY = 220

Picture3.Print "w=" & Round(w, 1) & "in."

Picture3.CurrentX = 50

Picture3.CurrentY = 390

Picture3.Print "ta=" & Round(t\_x, 1) & "in."

Picture3.CurrentX = 50

Picture3.CurrentY = 420

Picture3.Print "b=" & Round(b\_x, 0) & "in."

**EndSub**

El software de traves armadas de acero de sección I, es el resultado de la aplicación de elementos gráficos, controles, formularios y objetos que se han presentado anteriormente. A continuación se muestra cómo se va construyendo el programa a partir de todos estos elementos.



En primer lugar se tiene el formulario de presentación, en el cual se colocó un timer y un control imagen. El control timer se encarga de mostrar el formulario durante un tiempo determinado y ocultarlo, una vez transcurrido. El control imagen sirve para contener el escudo de la Universidad (Figura 5.16).



Figura 5.16 Portada.

Una vez transcurrido el tiempo establecido para el timer, se entra en el formulario denominado MDIForm1 donde se encuentran dos menús: Calcular y Créditos (Figura 5.17).

En el menú de Créditos se encuentran los datos genéricos de los desarrolladores (ver Figura 5.18).

El menú "Calcular" nos llevará al formulario 1 que es dónde se efectuarán todos los cálculos del programa. En este formulario se encuentran las cajas de texto donde el usuario tendrá que introducir los datos de inicio para la ejecución del programa.



Figura 5.17 Formulario "MDIForm1".



Figura 5.18 Cuadro de Créditos.

También están contenidas dos imágenes; la primera nos muestra la forma de las cargas a la que la trabe armada estará sometida, mientras que la segunda ilustra la forma que tiene una trabe armada de sección transversal I.

Finalmente, se tienen tres controles comandos: Iniciar, Resultados y Cerrar.

(Figura 5.19).

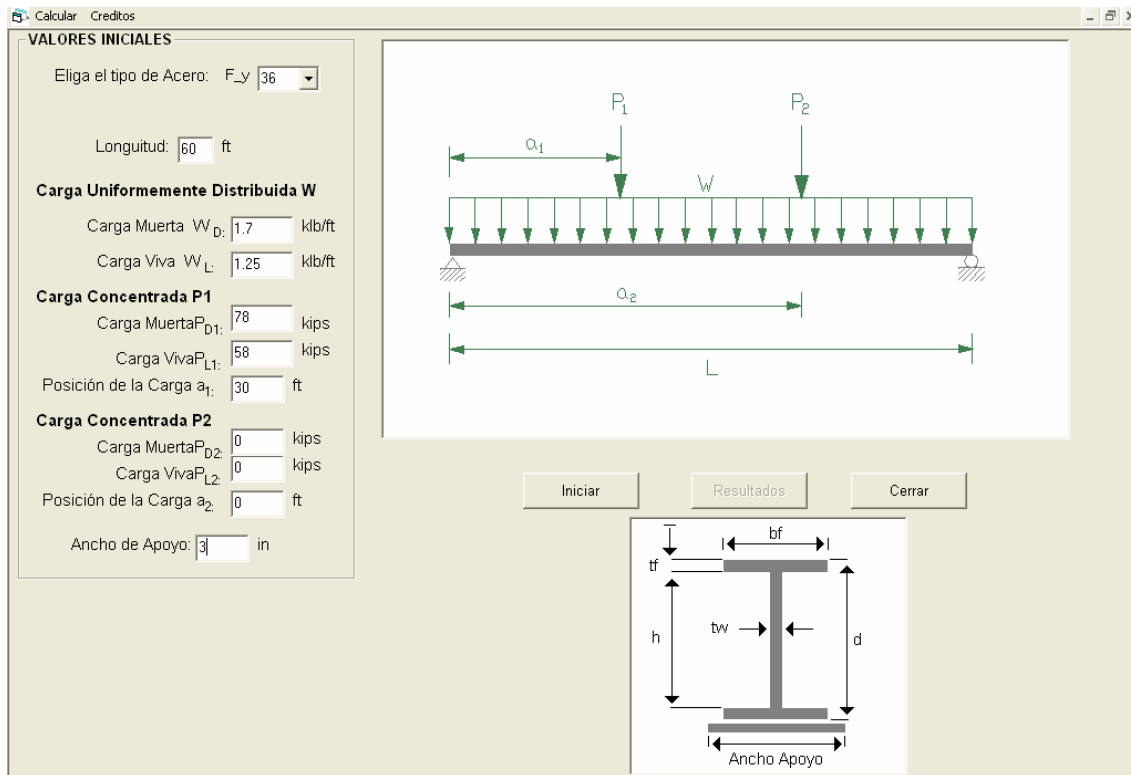


Figura 5.19 Pantalla principal de Cálculo.

Las cajas de texto permiten que cambie el gráfico que contiene el control imagen mediante código. Esto se hace a través del suceso “got focus”. Además, cada caja de texto cuenta con un objeto de validación; de esa forma se evita que se introduzcan valores no numéricos o que se deje la caja vacía.

En las Figura 5.20, Figura 5.21 y Figura 5.22 se muestran, de manera independiente, los elementos que conforman el formulario 1; las cajas de texto, la trabe en forma de I y el tipo de cargas a la que puede estar sujeta la trabe armada.

**VALORES INICIALES**

Eliga el tipo de Acero: F<sub>y</sub>

Longitud:  ft

**Carga Uniformemente Distribuida W**

Carga Muerta W<sub>D</sub>:  klb/ft

Carga Viva W<sub>L</sub>:  klb/ft

**Carga Concentrada P1**

Carga Muerta P<sub>D1</sub>:  kips

Carga Viva P<sub>L1</sub>:  kips

Posición de la Carga a<sub>1</sub>:  ft

**Carga Concentrada P2**

Carga Muerta P<sub>D2</sub>:  kips

Carga Viva P<sub>L2</sub>:  kips

Posición de la Carga a<sub>2</sub>:  ft

Ancho de Apoyo:  in

Figura 5.20 Datos de entrada.

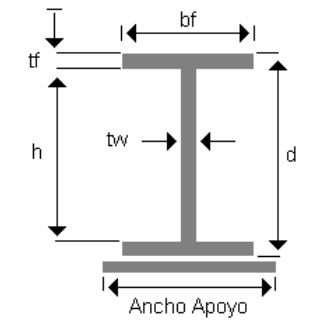


Figura 5.21 Sección I.

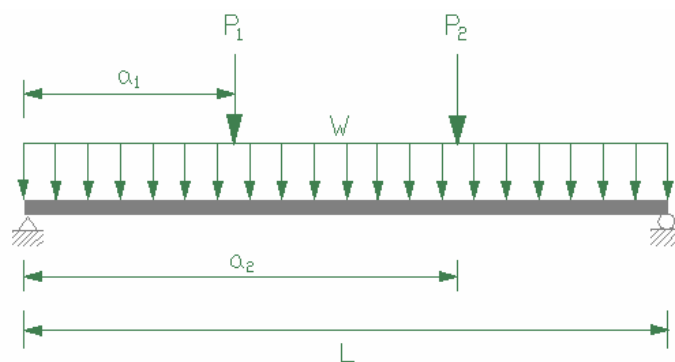


Figura 5.22 Representación gráfica del problema.

Una vez que han sido introducidos todos los datos iniciales, entra en acción el botón Iniciar. En este control se encuentra el código para la ejecución de todas las operaciones matemáticas necesarias para el diseño de la trabe armada (Figura 5.23).

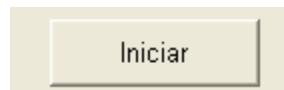


Figura 5.23 Botón Iniciar

El último control del Formulario1 es el botón de Cerrar (Figura 5.24). De ser requerido este botón nos llevará al formulario denominado MDIForm1, donde encontraremos los Menús de Calcular (que nos lleva al formulario 1) y Créditos (nos muestra los datos genéricos de los tesisas).

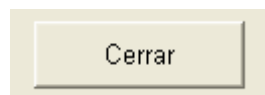


Figura 5.24 Botón Cerrar.

A su vez, el botón Resultados de la Figura 5.25 será el que nos muestre los resultados numéricos y gráficos que desarrolló internamente el control Iniciar.

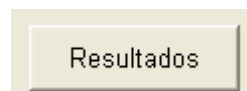


Figura 5.25 Botón Resultados.

Los resultados que se obtienen a partir de las operaciones matemáticas se presentarán en una carpeta que contendrá cuatro secciones al momento de dar clic al botón Resultados.

En la primera sección aparecerá una imagen de la trabe armada de sección transversal I con sus correspondientes dimensiones: peralte, ancho de patín, espesor del patín, espesor del alma, etc, además de mostrar gráficamente la separación de los atiesadores intermedios y de apoyo en la trabe (Figura 5.26).



Figura 5.26 Pantalla Resultados, Geometría de la Trabe Armada.

Esta sección cuenta con el control comando Imprimir, Recalcular y Cerrar, además de cuatro cajas de texto situadas del lado izquierdo.

El botón Imprimir mandará la orden a la impresora de imprimir la imagen de la trabe armada con todo lo que contiene, es decir, con todos los valores numéricos que se observan en la imagen. Este botón aparecerá en las cuatro secciones de resultados, mientras que el botón Cerrar ejecuta la acción de regresar al Formulario 1 (Figura 5.27).

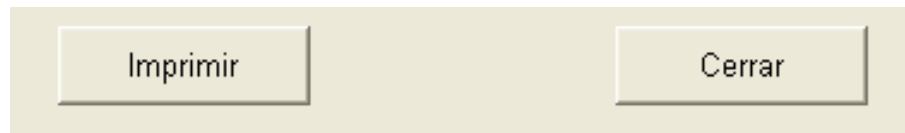


Figura 5.27 Botón Imprimir-Cerrar.

Las cajas de texto y el comando Recalcular tiene el propósito de convertir este programa en un software interactivo. Su uso consiste en lo siguiente: el usuario podrá introducir valores de  $b_f$ ,  $t_f$ ,  $t_w$  y  $h$  distintos a los que obtuvo el programa y que se muestran en la imagen de la sección uno.

Una vez que se introducen los nuevos valores de  $b_f$ ,  $t_f$ ,  $t_w$  y  $h$  y se da clic al comando Recalcular, el software calculará la resistencia de la trabe armada con los nuevos valores y los resultados aparecerán a la derecha de las cajas de texto (Figura 5.28)

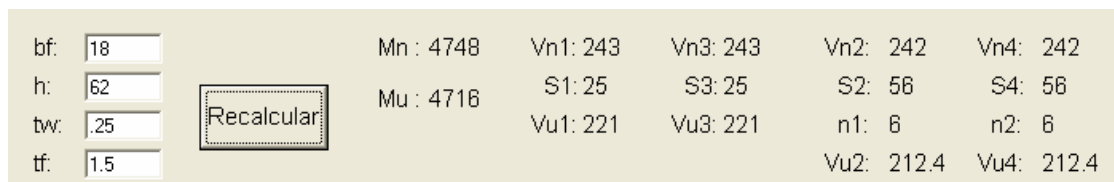


Figura 5.28 Valores Recalculados.

Como resultados finales, el programa mostrará la distancia a la cual deberían estar los atiesadores intermedios de contar con las nuevas dimensiones propuestas de la trabe. En caso de que la trabe armada no cumpliera con la resistencia a cortante, el programa dará aviso de ello con un mensaje como el de la Figura 5.29.

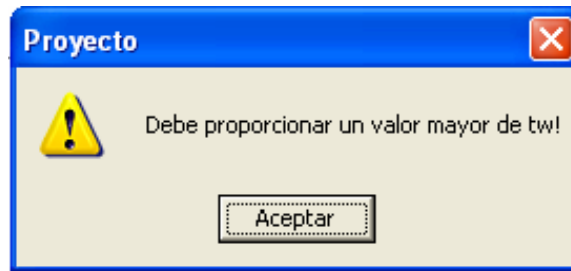


Figura 5.29 Cuadro de diálogo.

En la segunda sección de resultados se encontrarán dos controles principales; el primero es una imagen referente a los atiesadores intermedios y el segundo es una lista de información (Figura 5.30).

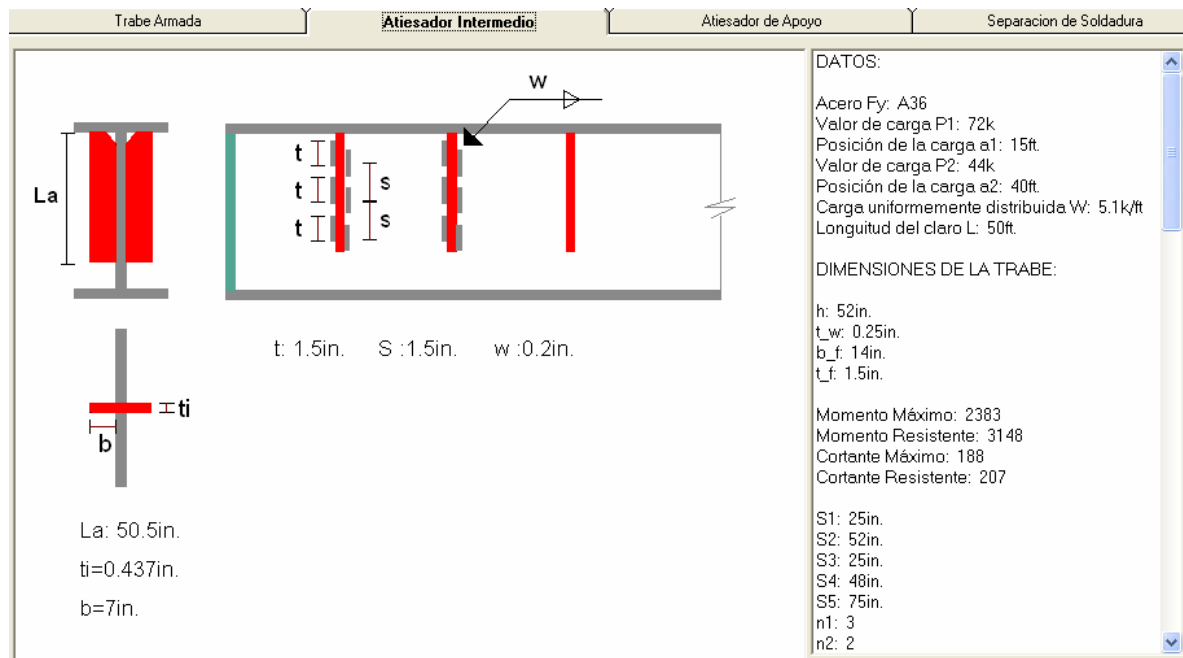


Figura 5.30 Pantalla de Resultados, Atiesadores intermedios.

La Figura 5.31, que corresponde a la segunda sección de resultados, muestra todos los datos necesarios para la construcción de los atiesadores intermedios: longitud del atiesador, ancho, espesor y espaciamiento de soldadura. Esta imagen se podrá imprimir por medio del control Imprimir.



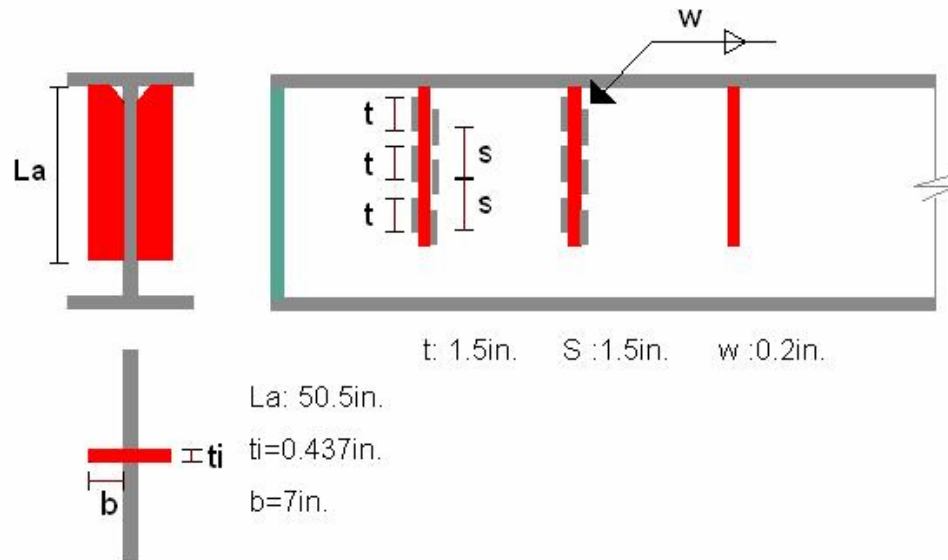


Figura 5.31 Atiesadores intermedios.

En la lista de la Figura 5.32 se hallan todos los resultados numéricos que obtiene el programa a partir de los datos de inicio y que después se exponen en las cuatro secciones de resultados. Por lo tanto, en la lista pueden verse los valores numéricos de cada una de las secciones.

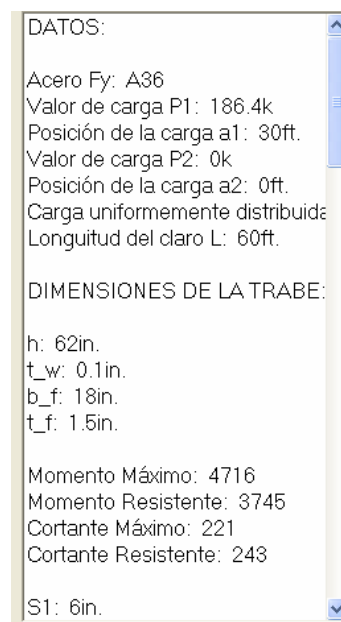


Figura 5.32 Resumen de datos y resultados.

Mediante un botón comando llamado Guardar se presenta una caja de diálogo con el cual es posible guardar el archivo de resultados en la carpeta que el usuario desee (Figura 5.33).

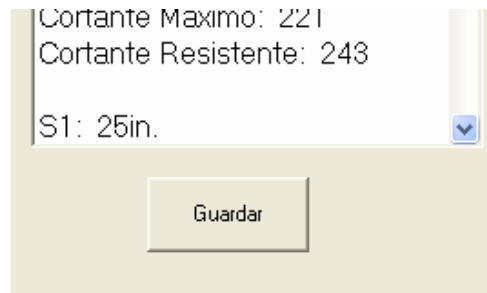


Figura 5.33 Botón Guardar.

En la tercera sección de resultados se encontrarán un control de imagen en la cual puede observarse toda la información referente a los atiesadores de apoyo: ancho del atiesador, espesor, distancia de soldadura, etc. Esta imagen también se podrá imprimir mediante el control Imprimir (Figura 5.34). El botón Cerrar nos llevará al Formulario 1.

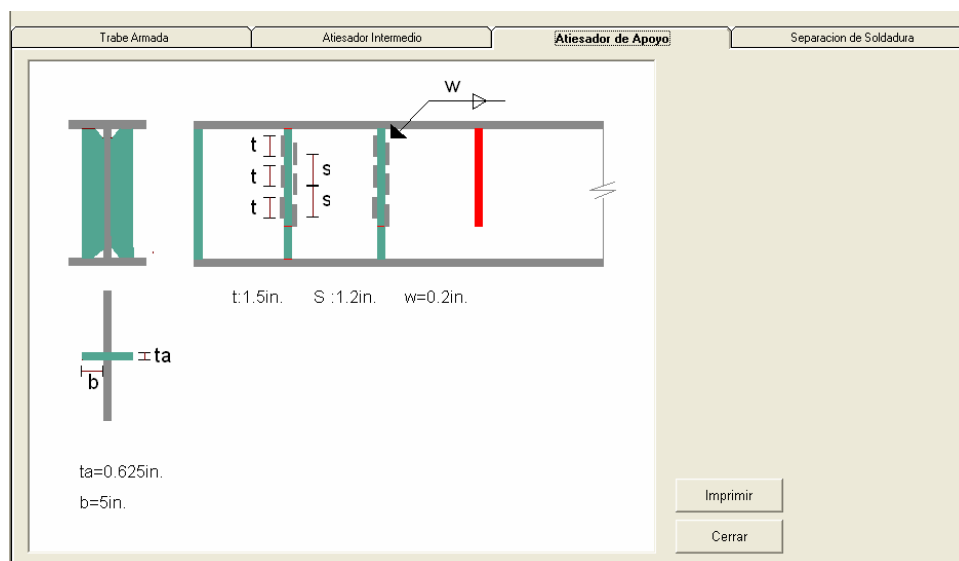


Figura 5.34 Pantalla de Resultados, Atiesadores de Apoyo.

Finalmente, la cuarta sección de resultados se puede observar en la Figura 5.35:

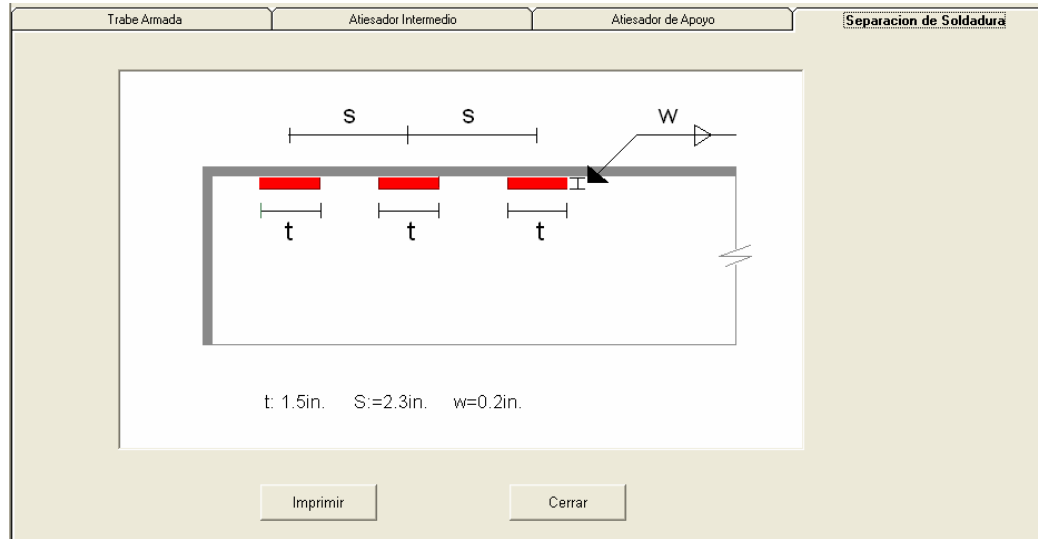


Figura 5.35 Pantalla de Resultados, Soldaduras.

En esta sección se encontrará la información referente a la soldadura que une al alma con el patín de la trabe armada; se visualiza el espesor y la separación entre soldadura.

El control Imprimir y Cerrar funcionan de la misma forma que los controles de las secciones pasadas.