



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA CIVIL**

**TEMA:**

**Desarrollo de código en Matlab para el diseño de vigas  
presforzadas, considerando diferentes códigos de diseño.**

**AUTOR:**

**Alvarado Campi, Gino Osvaldo**

**Trabajo de titulación previo a la obtención del título de  
INGENIERO CIVIL**

**TUTOR:**

**Ing. Ponce Vásquez, Guillermo Alfonso**

**Guayaquil, Ecuador**

**19 de septiembre del 2017**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
**FACULTAD DE INGENIERÍA**  
**CARRERA DE INGENIERÍA CIVIL**

**CERTIFICACIÓN**

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Alvarado Campi Gino Osvaldo**, como requerimiento para la obtención del título de **Ingeniero Civil**.

**TUTOR**

---

**Ing. Ponce Vásquez, Guillermo Alfonso. MSc**

**DIRECTORA DE LA CARRERA**

---

**Ing., Alcivar Bastidas, Stefany Esther. MSc**

**Guayaquil, 19 días del mes de septiembre del 2017**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE INGENIERÍA CIVIL  
CARRERA DE INGENIERÍA CIVIL**

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Alvarado Campi Gino Osvaldo**

**DECLARO QUE:**

El Trabajo de Titulación, **Desarrollo de código en Matlab para el diseño de vigas presforzadas, considerando diferentes códigos de diseño.**, previo a la obtención del título de **Ingeniero Civil**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

**Guayaquil, 19 de septiembre del 2017**

**EL AUTOR**

---

**Alvarado Campi, Gino Osvaldo**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE INGENIERÍA CIVIL

## AUTORIZACIÓN

Yo, **Alvarado Campi Gino Osvaldo**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Desarrollo de código en Matlab para el diseño de vigas presforzadas, considerando diferentes códigos de diseño.**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, 19 de septiembre del 2017**

**EL AUTOR:**

---

**Alvarado Campi, Gino Osvaldo**

## Urkund Analysis Result

**Analysed Document:** TRABAJO DE TITULO Gino Alvarado Campi..doc (D30373626)  
**Submitted:** 2017-09-05 03:23:00  
**Submitted By:** claglas@hotmail.com  
**Significance:** 9 %

### Sources included in the report:

1420470103\_CAPITULO II PRINCIPIOS BASICOS.docx (D12777825)  
CAPITULO1-6.pdf (D14030424)  
T.1.8A Hormigon Presforzado.docx (D11747340)  
2017-01-23 recopilacion.doc (D25375998)

## **AGRADECIMIENTO**

Agradezco principalmente a Dios, por todas bendiciones que me ha dado en la vida, por guiarme en el buen camino y ayudarme a ser una mejor persona.

A mis queridos padres Gino Xavier Alvarado Terán y Rosa Alexandra Campi Yépez, porque siempre creyeron en mí, me apoyaron en todas las decisiones que eh tomado en mi vida y siempre fueron mi ejemplo a seguir.

A mis hermanos Gino, Guadalupe y Kelly, que siempre están junto a mí y me brindan todo su apoyo y amor.

A mi novia Jael Castillo, por ser una persona muy especial e importante en mi vida durante esta etapa universitaria y por brindarme todo su apoyo y amor incondicional.

A todos mis profesores de la universidad, especialmente a mi director de tesis Ing. Guillermo Ponce por su confianza brindada al momento del desarrollo de esta tesis.

A las Ingenieras Lilia Valarezo, Stefany Alcivar, Clara Glas y Ana Camacho, por brindarme su apoyo en la mayoría de mi etapa universitaria.

## **DEDICATORIA**

Este logro se lo dedico a mis padres Gino Xavier Alvarado Terán y Rosa Alexandra Campi quienes me apoyaron en todo momento y me motivaron para seguir mis estudios universitarios, sin ustedes esto no hubiera sido posible. Este logro es de ustedes.

**Gino Alvarado Campi**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE INGENIERÍA**

**TRIBUNAL DE SUSTENTACIÓN**

---

**Ing. Valarezo Moreno, Lilia Marlene, MS.**

DECANA

---

**Ing. Hernández Barredo, Jaime Francisco, MSc.**

DOCENTE DE LA CARRERA

---

**Ing. Casal Rodríguez, Xavier Federico, Mgs.**

OPONENTE



# ÍNDICE

<b>1. INTRODUCCIÓN</b> .....	2
1.1    Objetivo.....	3
1.1.1 Objetivo General.....	3
1.1.2 Objetivo Especifico.....	3
1.2 Justificación.....	4
<b>2. MARCO TEÓRICO</b> .....	5
2.1 Antecedentes .....	5
2.2 Concise Beam.....	5
2.3 CivilCad3000.....	8
2.4 Fundamentos de la ingeniería.....	12
2.4.1 Definición de Presfuerzo.....	13
2.4.2 Tipos de vigas.....	17
2.4.2.1 Vigas tipo I.....	20
2.4.2.2 Vigas tipo T.....	21
2.4.2.3 Vigas Doble T.....	22
2.4.3 Materiales.....	24
2.4.4 Construcción.....	25
<b>3. MARCO METODOLÓGICO</b> .....	27
3.1 Descripción del programa.....	27
3.2 Funcionamiento.....	27
3.3 Manual de usuario.....	27
<b>4. Conclusiones</b> .....	43
<b>5. Limitaciones</b> .....	43
<b>ANEXOS</b> .....	44

## ÍNDICE DE GRÁFICOS

Gráfico 1. Presentación del programa Concise Beam.....	6
Gráfico 2. Ejemplo de Concise Beam en la parte de Editar Sección.....	7
Gráfico 3. Parámetros de diseño Concise Beam.....	8
Gráfico 4. Gráfico del presfuerzo en el programa Concise Beam.....	9
Gráfico 5. Ventana principal concise beam.....	10
Gráfico 6. Ventana de la orden de configuración.....	11
Gráfico 7. Selección del sistema de unidades.....	11
Gráfico 8. Selección de la normativa.....	12
Gráfico 9. Menú principal una vez entrada la información general.....	12
Gráfico 10. La ventana de geometría, .....	13
Gráfico 11. Distribución de esfuerzos.....	15
Gráfico 12. Viga de concreto.....	16
Gráfico 13. Momentos flexionantes.....	17
Gráfico 14. Representación de viga tipo I.....	20
Gráfico 15. Viga I (AASHTO).....	21
Gráfico 16. Viga tipo T (AASHTO).....	22
Gráfico 17. Viga Doble T (AASHTO).....	23
Gráfico 18. Software para diseño a flexión.....	27
Gráfico 19. Desplegable de secciones.....	28
Gráfico 20. Selección de secciones en el software.....	29
Gráfico 21. Cuadro de excentricidad en el software.....	29
Gráfico 22. Ventana de carga sobre impuesta.....	30
Gráfico 23. Ventana de carga muerta.....	31
Gráfico 24. Ventana de carga distribuida.....	32
Gráfico 25. Ventana de carga resultante del camión.....	33
Gráfico 26. Cuadro de datos de cargas.....	34
Gráfico 27. Ventana de longitud de la viga en pies.....	35
Gráfico 28. Ventana de resistencia del hormigón en lb/pies <sup>2</sup> .....	36
Gráfico 29. Ventana de resistencia del hormigón en transferencia .....	37
Gráfico 30. Ventana de Resistencia a flexión de los torones.....	39
Gráfico 31. Ventana de la relación de pretensado residual.....	39

Gráfico 32. Ventana del área de los torones.....	40
Gráfico 33. Cuadro de datos generales de la viga.....	40
Gráfico 34. Fórmulas para el cálculo de modulo de sección.....	42
Gráfico 35. Cuadro de verificación de sección.....	43
Gráfico 36. Cuadro de numero requerido de torones.....	44
Gráfico 37. Equilibrio de las fuerzas que actúan en la viga.....	45
Gráfico 38. Ecuaciones de esfuerzo en la parte superior.....	46
Gráfico 39. Cuadro de inserción de datos de torones.....	47
Gráfico 40. Ventana de ploteo .....	48
Gráfico 41. Cuadro de resultados del programa.....	49
Gráfico 42. Cuadro de resultados en exportados a Excel.....	49

## ÍNDICE DE TABLAS

Tabla 1. Conceptos, definiciones y unidades básicas.....	18
Tabla 2. Dimensiones para vigas I (AASTHO).....	20
Tabla 3. Dimensiones para vigas tipo T (AASTHO).....	21
Tabla 4. Dimensiones para vigas Doble T (AASTHO).....	22
Tabla 5. Propiedades del cable de 7 alambres sin revestimiento.....	25

## RESUMEN

En este proyecto de sustentación se procederá a realizar un Software en el programa Matlab el cual será muy útil al momento de realizar una revisión del diseño de vigas presforzadas simplemente apoyadas. Este software está programado para realizar revisiones de diseño cumpliendo todas las normas vigentes requeridas. De esta manera aseguramos el buen diseño de una viga presforzada y estaremos resguardando la seguridad de las personas que vayan a utilizar a futuro dichas estructuras lo cual es lo más importante al momento de construir una estructura. Este programa servirá para Ingenieros o estudiantes de ingeniería civil con conocimientos básicos de diseño, este software puede ayudar a futuros estudiantes de ingeniería a practicar al momento de diseñar vigas de hormigón presforzado. Esto es posible gracias a que esta herramienta trata de tener una interfaz de diseño cómoda y fácil para el usuario al momento de ingresar los datos garantizando un diseño seguro y funcional.

**Palabras clave:** Vigas, Hormigón, Presforzado, Software, Diseño, Herramienta, Interfaz.

## **ABSTRACT**

In this Project of thesis will proceed to realize a software that will be very useful for the design of reinforced beams with the help of the Matlab program. This software is programmed to perform design reviews complying with all current standards required. In this way we ensure the good design of reinforced beam and we will be safeguarding the safety of people who will use these structures in the future, which is the most important thing when you are building a structure. This program will be for engineers or students of civil engineering with basic knowledge of design, this software can help future engineering students to practice when designing reinforced concrete beams. This is possible thanks to the fact that this tool tries to have a design interface that is comfortable and easy for the user when entering data, guaranteeing a safe and functional design for the engineer.

**Key words:** Beam, Concrete, Tool, Design, software, reinforced, Interface.

# CAPITULO I

## 1. INTRODUCCIÓN

### 1.1 Antecedentes

Procederemos a realizar un Software de diseño de vigas presforzadas con la ayuda del programa MATLAB, elegimos este programa ya que tiene un fácil manejo y porque su método de programación es compatible con muchos otros programas de diseño, de esta manera podrá ser utilizado sin problemas en cualquiera de los programas compatibles y a futuro puede ser modificado para su mejora o adición de normas de diseño.

Matlab nos permite crear varias sub rutinas que una vez listas permitirán que el usuario solo tenga que ingresar los datos.

Este software ayudará a muchos Ingenieros civiles a realizar diseños de una manera más sencilla y amigable, ya que el software no es complicado de ejecutar.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo General.**

Crear un programa amigable para los diseñadores y estudiantes tanto en el ingreso de datos como en la programación de los mismos, así mismo que contenga una interfaz sencilla amigable y atractiva que sea fácil de aceptar por el usuario y que cumpla todos los requisitos y normas para el diseño de una viga presforzada.

### **1.2.2 Objetivo específico.**

- Revisar por medio del programa que se cumplan todos los requisitos vigentes para el diseño de vigas presforzadas.
- Realizar un software que permita agilizar el diseño de las vigas presforzadas.
- Presentar en el software los datos de una manera clara y de ser posible incluir los resultados de manera gráfica.
- Elaborar un manual de usuario para el fácil manejo del software.



## JUSTIFICACIÓN

En todo el mundo las construcciones civiles aumentan de gran manera y cada vez son obras de mayor magnitud por lo que se necesitan cumplir con mayor rigurosidad todas las normas de construcción, incluso en Ecuador las obras civiles son mayores y se necesitan agilizar el tiempo de diseño de las estructuras. Gracias a esto es muy importante implementar nuevas herramientas para facilitar y mejorar el trabajo al momento de diseñar por lo que este software ayudaría en reducción de costo al poder diseñar de una manera más eficaz, segura y rápida, casi al momento. El proceso para diseñar elementos presforzados a pesar de haber sido ya estudiado y presentando como un proceso bastante estandarizado, puede requerir de un tiempo importante dentro del proceso general de diseño de un proyecto estructural, por lo que se hace presente la necesidad de buscar un método que nos permita agilizar dicho proceso. Gracias a la necesidad de agilizar un proceso numérico surge como respuesta el desarrollo de softwares computacionales ya que estos influyen en el ahorro significativo de tiempo al momento de realizar los cálculos matemáticos.

## CAPITULO II

### 2.1 Antecedentes.

Hoy en día gracias a la tecnología se puede ir desarrollando programas o de igual manera mejorar programas ya existentes agregándoles normas actuales que permitan que el diseño sea mejor y cumpliendo todas las exigencias necesarias. Lo malo de muchos programas antiguos es que el usuario no se siente cómodo con su interfaz o siente complejidad al momento de ingresar los datos por lo cual optan por dejar atrás ese programa y probar nuevos a pesar de ser muy útiles el usuario siempre busca la comodidad o una interfaz agradable.

Existen programas muy buenos que al no ser modificables ni contar con actualizaciones con el paso del tiempo quedan obsoletos ya que las normas se van actualizando y mejorando con el tiempo, lo que hace que muchos de estos programas queden inservibles.

Se presentarán unos ejemplos de programas ya existentes para el diseño de vigas presforzadas.

### 2.2 Programas existentes

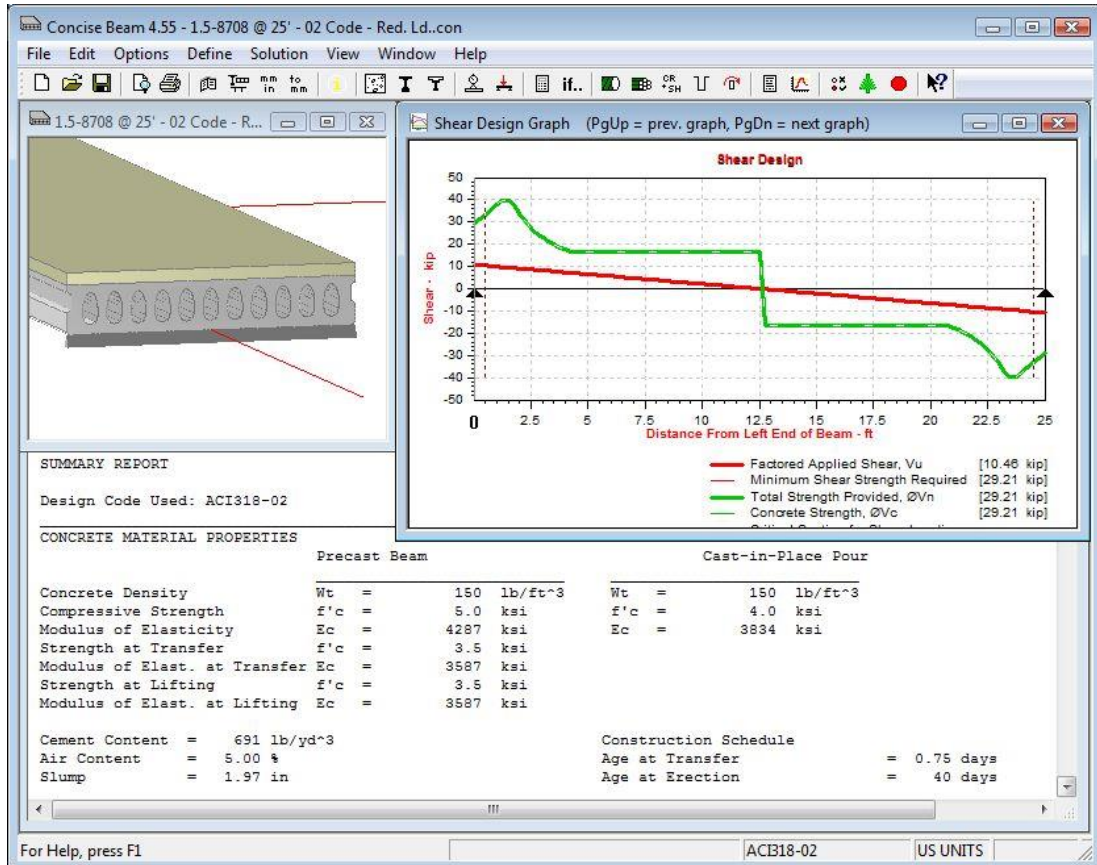
#### 2.2.1 Concise beam (2.2.1 Concise beam).

Es un Programa fácil de usar para el diseño de vigas de hormigón prefabricadas, se encarga de realizar análisis de carga y verificaciones de diseño de acuerdo a las normas vigentes. Los parámetros de código clave se pueden personalizar para simular otros códigos de diseño. Las vigas pueden ser reforzadas, parcial o totalmente pretensadas.

Se puede modelar cualquier sección transversal y permitirá que la sección varíe prismáticamente sobre la longitud de la viga. Un editor gráfico

permite al usuario describir cualquier sección transversal, incluyendo huecos.

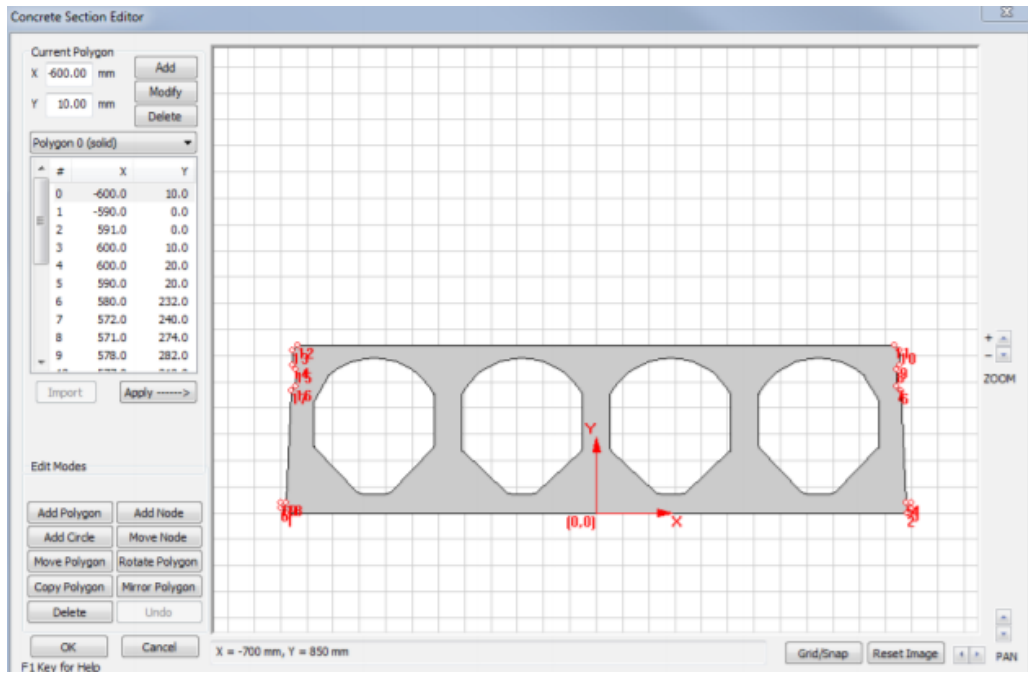
**Gráfico 1.** Presentación del programa Concise Beam (Blackmint, 2016).



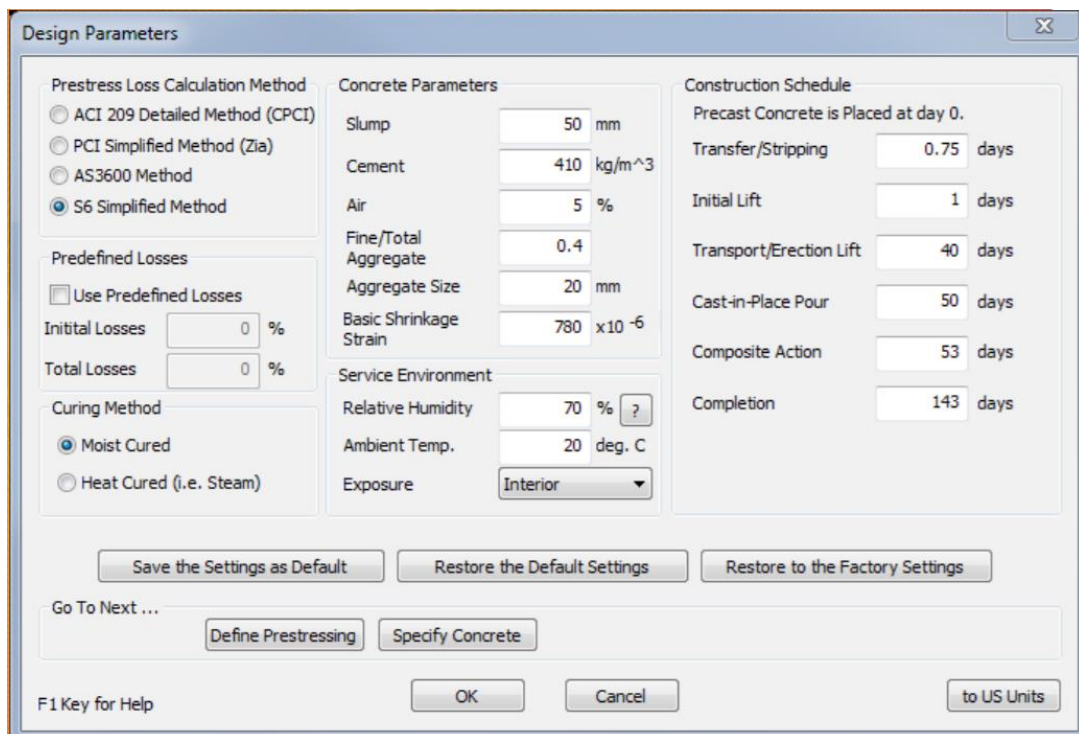
Concise Beam trabaja actualmente en dos dimensiones. El usuario puede trabajar en unidades métricas o estadounidenses con una variedad de unidades individuales como mm, cm, m y cambiarlas a voluntad del usuario.

Las bibliotecas de secciones transversales y materiales estándar de las industrias son proporcionadas por Concise Beam, estas pueden ser modificadas, ampliadas y personalizadas por el usuario.

**Gráfico 2.** Ejemplo de Concise Beam en la parte de Editar Sección (Blackmint, 2016).



**Gráfico 3.** Parámetros de diseño Concise Beam (Blackmint, 2016).



**Gráfico 4.** Gráfico del presfuerzo en el programa Concise Beam (Blackmint, 2016).

**Prestressing**

Strand Group Details

Number of Strands:   Epoxy Coated

Stress Ratio at Lock-off:  (fpj / fpu)

Strand Height at Left End:  mm

Strand End Type and Offset from End of Beam

Left: Fully Bonded  m

Right: Fully Bonded  m

Option 1: Code

Strand Type

Select from Library Name:  dia.:  mm fpu:  MPa

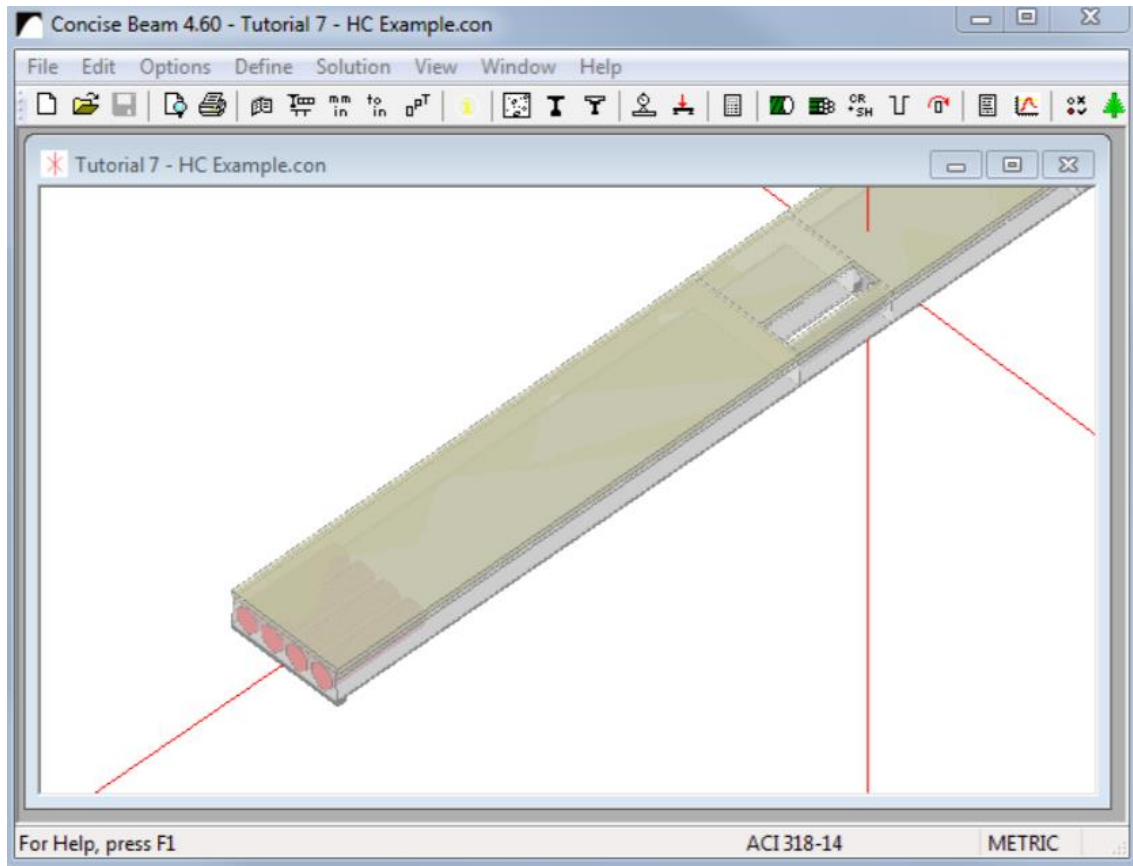
Low-Relaxation Strand A  mm<sup>2</sup> E:  MPa

Section	Grade (fpu)	Num.	L. Offset	R. Offset	Lock-Off	Method
<input checked="" type="checkbox"/> 12.7 (1/2)	1860 MPa	7	0 m	B 0 m	B 0.7 fpu	Code
<input checked="" type="checkbox"/> 12.7 (1/2)	1860 MPa	5	0 m	B 0 m	B 0.7 fpu	Code

Go To Next ...

F1 Key for Help

**Gráfico 5.** Ventana principal del programa Concise Beam (Blackmint, 2016).

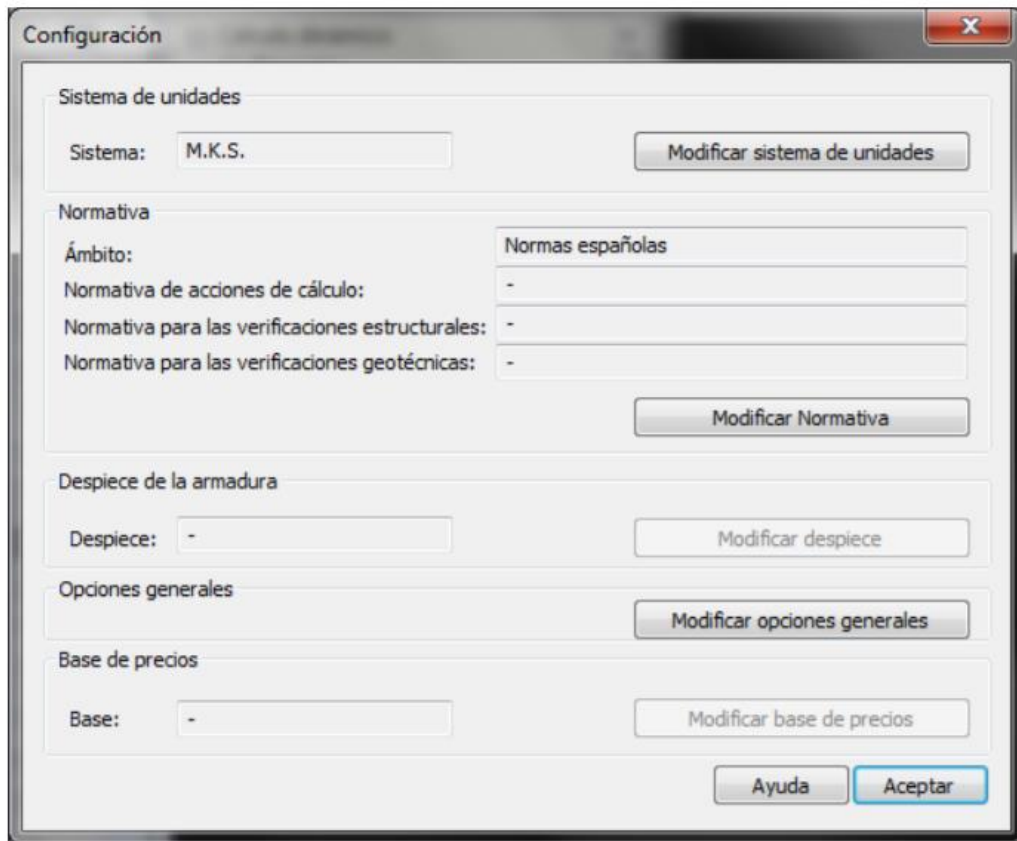


### 2.3. Civilcad3000 (2.3.2 Civilca3000)

Este programa permite tener en cuenta la presencia de apoyos elásticos en las estructuras, la cargas que se emplean en el cálculo del programa son cargas concentradas, de valor constante y se mueven sobre la estructura a velocidad constante.

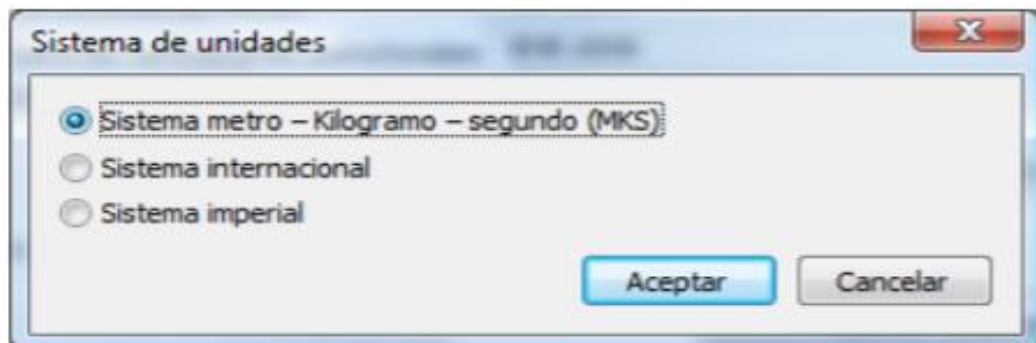
Este programa realiza 2 tipos de cálculos principales, Series temporales y cálculos envolventes. En el primero de ellos se determina la evolución temporal del desplazamiento, velocidad y aceleración vertical en cualquier nudo de la malla, además de los giros en los estribos, al transitar sobre la estructura un vehículo o tren de carga a ciertas velocidades de paso determinadas.

**Gráfico 6.** Ventana de la orden de configuración (CivilCad3000 2016).

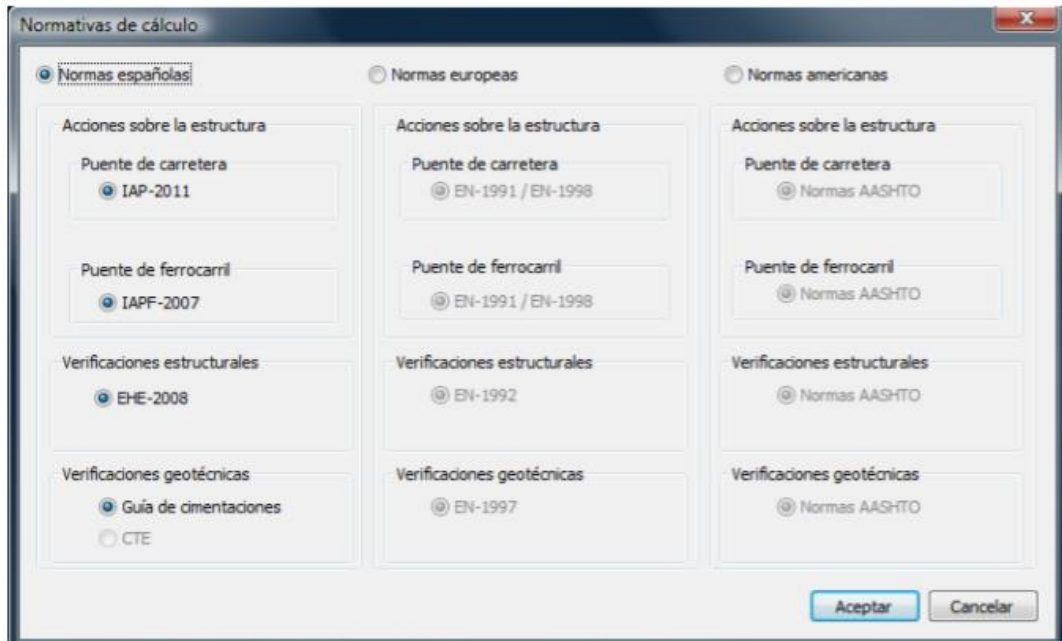


Por defecto aparecerán las opciones que estén activas en el menú general de civilcad3000. No obstante, el usuario siempre puede editar o modificar con los botones 'Modificar sistema de unidades' y 'modificar normativas'.

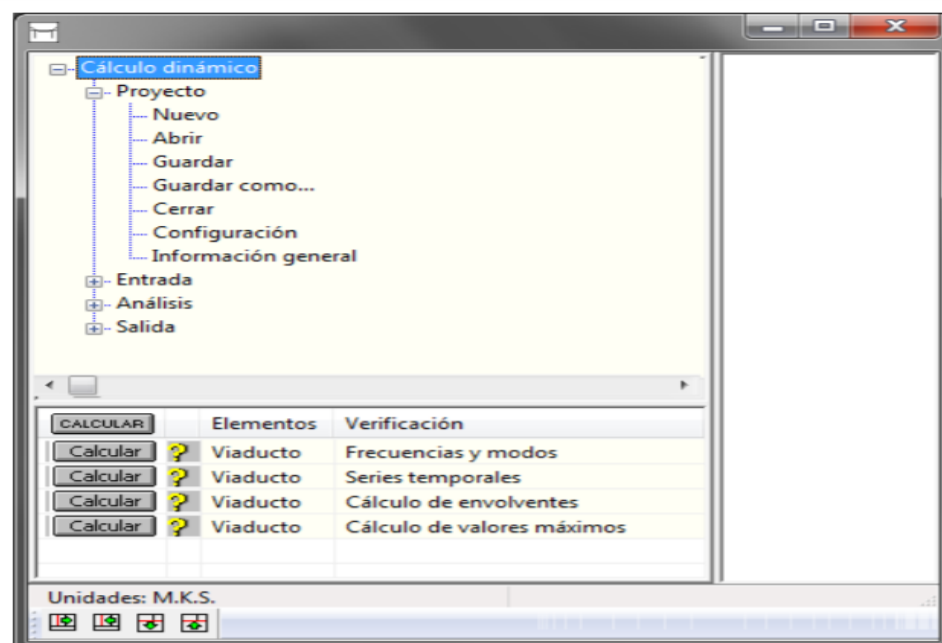
**Gráfico 7.** Selección del sistema de unidades (Civilcad3000).



**Gráfico 8.** Selección de la normativa (CivilCad3000, 2016).

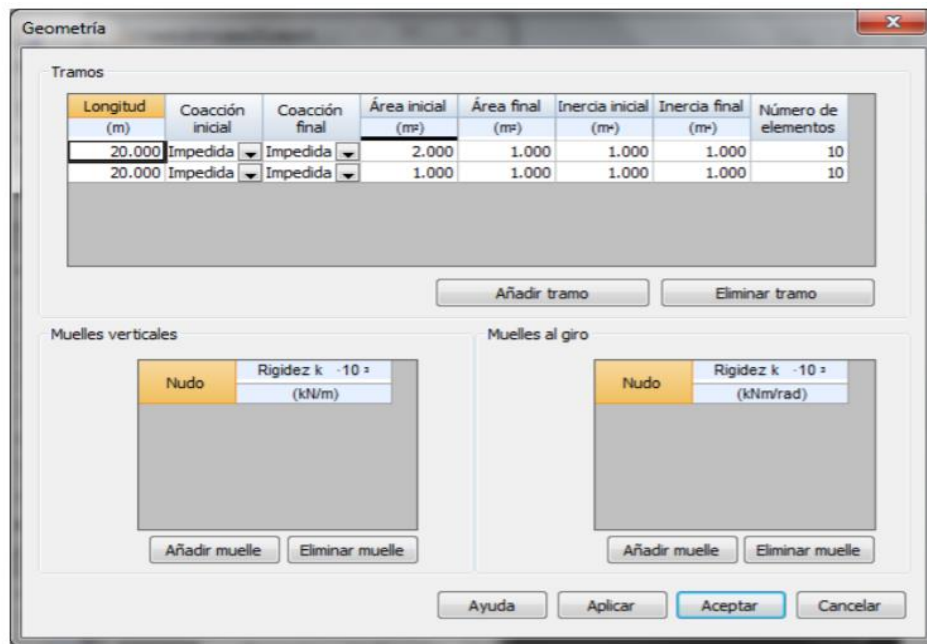


**Gráfico 9.** Menú principal una vez entrada la información general (CivilCad3000, 2016).





**Gráfico 10.** La ventana de geometría, empleada para la definición del modelo (CivilCad3000, 2016).



## 2.4 Fundamentos de la ingeniería

### 2.4.1 Definición de Presfuerzo

El presfuerzo significa la creación intencional permanente de esfuerzos en un conjunto de piezas o una estructura, ya que de esta manera se trata de mejorar su comportamiento y resistencia. El presfuerzo se las aplica a estructuras de muchos tipos y materiales, la aplicación más frecuente se da en el diseño de concreto estructural. El concepto del presfuerzo consiste en introducir suficiente pre compresión axial en las vigas para que se eliminen todos los esfuerzos de tensión que actuara en el concreto.

El ACI nos da las siguientes definiciones, el concreto presforzado es el concreto en el cual se introducen esfuerzos internos de tal distribución y magnitud que los esfuerzos resultantes debido a cargas externas se contrarrestan entre sí a un grado deseado. En elementos de concretos

reforzado el presfuerzo es introducido normalmente tensado el acero de refuerzo.

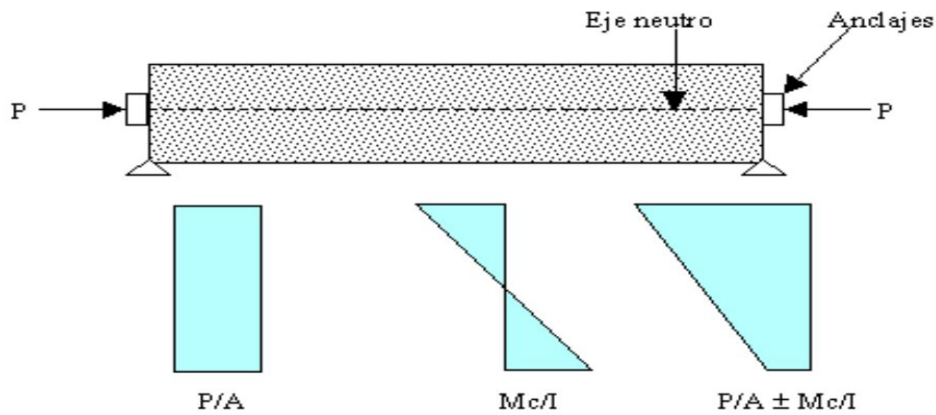
Es importante que el diseñador comprenda los 2 conceptos que pueden ser aplicados y analizados para el comportamiento del concreto presfuerzo, de esta manera pueda proporcionar y diseñar con inteligencia y eficacia.

El primer concepto es Preforzar para mejorar el comportamiento elástico del concreto, dicho concepto interpreta al concreto como un material elástico y este es el criterio que los ingenieros adoptan comúnmente. El concreto se comprime de tal manera que el mismo sea capaz de resistir los esfuerzos a tensión que se presenten. Desde este punto de vista el concreto está sujeto a 2 sistemas de fuerzas, presfuerzo interno y carga externa, con los esfuerzos de tensión debido a la carga externa contrarrestados por los esfuerzos de compresión debido al presfuerzo.

Similarmente, el agrietamiento del concreto debido a la carga es contrarrestado por la pre compresión producida por los tendones. Siempre y cuando no existan grietas en el hormigón las deformaciones, los esfuerzos y las deflexiones del concreto ya que los sistemas de fuerzas pueden ser considerados por separado o superpuestos en caso de necesitarlo.

Consideremos una viga rectangular con carga externa y presforzada por un tendón a través de su eje centroidal.

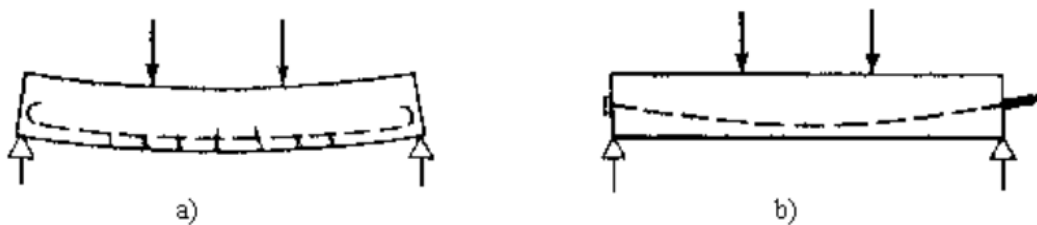
**Gráfico 11.** Distribución de esfuerzos a través de una sección de concreto presforzada concéntricamente (ConstruAprende, 2016).



Concepto 2 es, Preforzar para aumentar la resistencia última del elemento. Este concepto considera al concreto presforzado como una combinación de concreto y de acero, similar al concreto reforzado, con acero tomando tensión y concreto tomando compresión de tal manera que los dos materiales formen un solo elemento resistente con el momento externo.

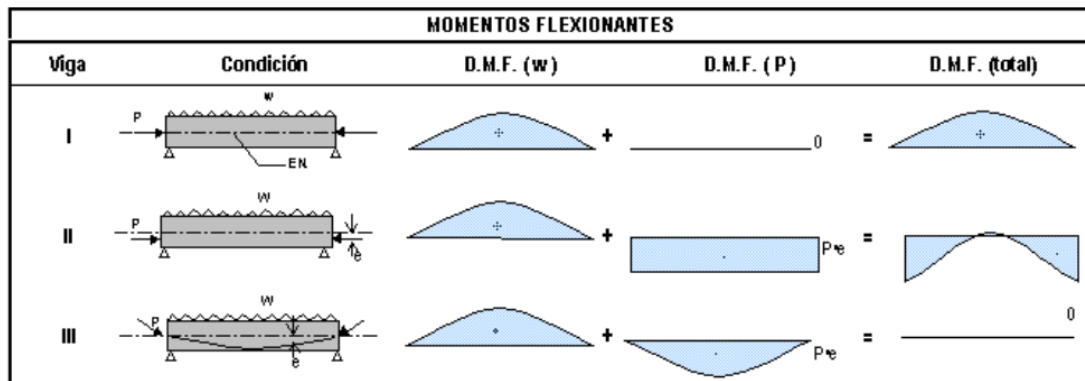
En el hormigón presforzado siempre se debe usar un acero de alta resistencia que así mismo debe fluir antes de que su resistencia llegue al máximo (siempre que la viga sea dúctil). Si el acero de alta resistencia es simplemente embebido en el concreto, este tendrá que agrietarse antes de que la resistencia total del acero se desarrolle.

**Gráfico 12.** Viga de concreto (ConstruAprende, 2016).



En la primera imagen podemos observar una viga simplemente reforzada con grietas y deflexiones excesivas, en la segunda vemos una viga presforzada sin grietas y con pequeñas deflexiones.

**Gráfico 13.** Momentos flexionantes a lo largo de una viga presforzada simplemente apoyada (ConstruAprende, 2016).



**Tabla 1.** Conceptos, definiciones y unidades básicas.

CONCEPTO	DEFINICIÓN	UNIDADES	
		MKS	SI
FUERZA	Acción capaz de producir un movimiento	Kg., Ton 1 Kg <sub>f</sub> = 9.81 N 1 Ton/ml x 9.81 = 9.81 KN/ml	Newton, KN 1N = 0.10197 Kg <sub>f</sub> Kg/ml x 0.00981=Kg/ml
AREA	Las dimensiones que quedan comprendidas dentro de un cuerpo.	cm. <sup>2</sup> , m <sup>2</sup> 1m <sup>2</sup> = 10,000 cm <sup>2</sup>	mm. <sup>2</sup> 1cm <sup>2</sup> = 100 mm <sup>2</sup>
DEFORMACIÓN	Cambio de forma, ley de Hooke	cm,m	mm.
RESISTENCIA Y ESFUERZO	Limite de capacidad. Fuerza por unidad de área	Kg. / cm. <sup>2</sup>  1 $\frac{Kg}{cm^2}$ = 0.0986 MPa	Pa, MPa 1Pa = $\frac{1N}{m^2}$ 1MPa = 1x10 <sup>6</sup> N/M <sup>2</sup> 1MPa = 10.2 $\frac{Kg}{cm^2}$
E	Módulo de elasticidad	Kg./cm <sup>2</sup>	Pa, MPa
I	Inercia	cm <sup>4</sup>	mm <sup>4</sup>
C	Centroide	cm	mm.
S	Módulo de Sección	cm <sup>3</sup>	mm <sup>3</sup>
M	Momento Flexiónante	Kg. cm, Kg.m, Ton.m	N mm , KN mL
V	Cortante	Kg, Ton.	N KN.

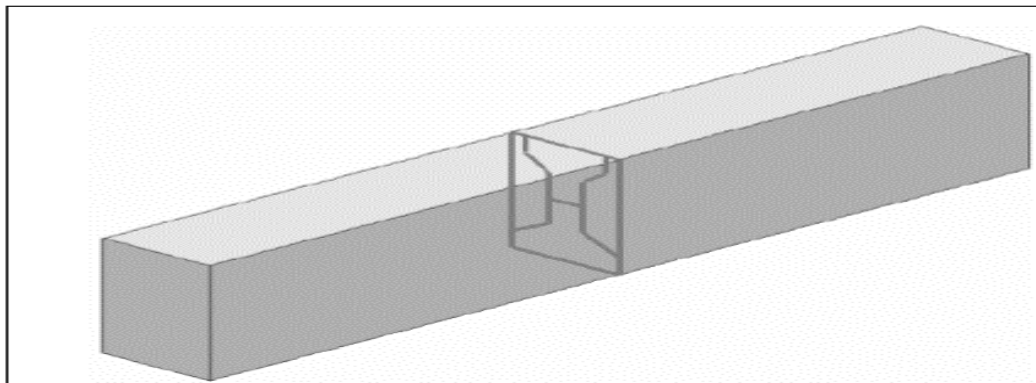
Elaborada por: El Autor

### 2.4.2 Tipos de viga

Las Vigas a inicios fueron de forma rectangular y luego con el paso del tiempo buscando optimizar recursos y el avance de los diseños esta forma rectangular quedo obsoleta.

Con el tiempo se fueron ofreciendo varios tipos de secciones debido a que el concreto como es un material moldeable no es complicado darle la forma que uno desee. Con el paso del tiempo la viga I fue una de las secciones que cumple en su mayoría un ahorro de material y mayor viabilidad a las diferentes cargas.

**Gráfico 14.** Representación de viga tipo I dentro de una viga rectangular (Nawy, 2009).



Al saber que en los puntos extremos de esta sección se generan los esfuerzos máximos es importante considerar sitios donde se necesite mayor cantidad de concreto y acero. Dichos sitios se les conoce como alma y patines de la viga, estos tienen la misma resistencia que una viga rectangular y se logra reducir el peso del elemento.

Cuando una viga tipo I se considera como viga presforzada aumentan las ventajas que brinda como elemento de construcción ya que representan una mejora en la parte técnica y son económicas.

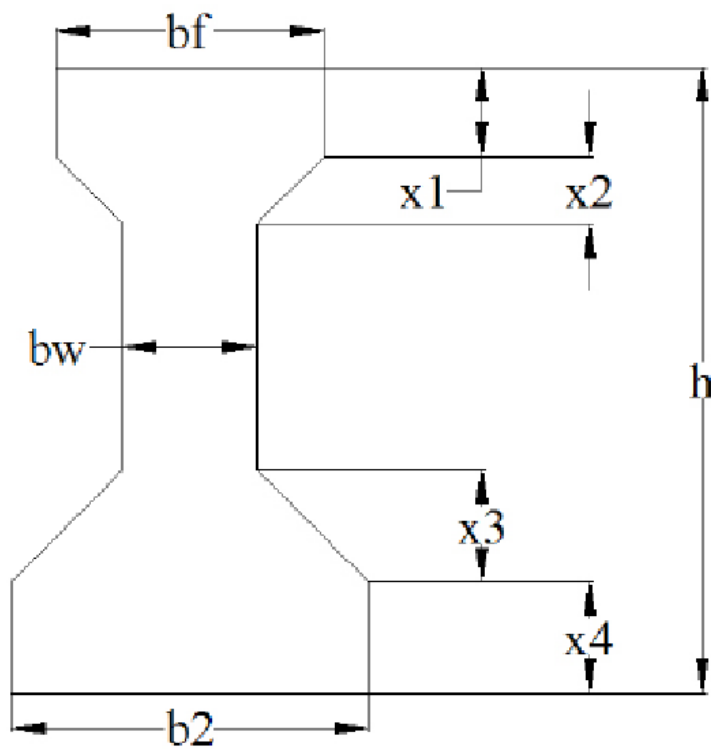
En la elaboración de varias estructuras como puentes, puentes peatonales etc.... se deben cumplir las normas respectivas que son las AASHTO. Este manual recomienda vigas tipo I estándar para puentes siempre y cuando el mismo no tenga claros mayores a 50 metros. En la siguiente tabla podemos observar los diferentes tipos de vigas que nos recomienda AASTHO.

### 2.4.2.1 Vigas tipo I

**Tabla 2.** Dimensiones para vigas (AASHTO).

Tipo	$b_f$ (mm)	$X_1$ (mm)	$X_2$ (mm)	$b_2$ (mm)	$X_3$ (mm)	$X_4$ (mm)	$b_w$ (mm)	$h$ (mm)
AASHTO 1	304.80	101.60	76.20	406.40	127.00	127.00	152.40	711.20
AASHTO 2	304.80	152.40	76.20	457.20	152.40	152.40	152.40	914.40
AASHTO 3	406.40	177.80	114.30	558.80	190.50	177.80	177.80	1143.00
AASHTO 4	508.00	203.20	152.40	660.40	228.60	203.20	203.20	1371.60
AASHTO 5	1066.80	127.00	177.80	711.20	254.00	203.20	203.20	1600.20
AASHTO 6	1066.80	127.00	177.80	711.20	254.00	203.20	203.20	1828.80

**Gráfico 15.** Viga I (AASHTO).

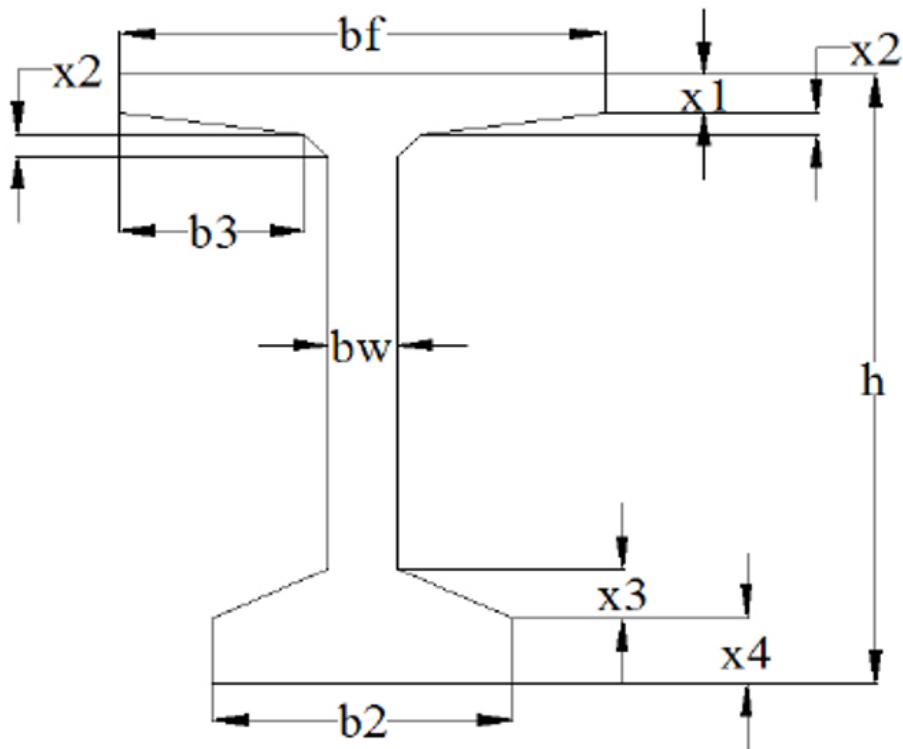


### 2.4.2.2 Dimensiones de Vigas tipo T

**Tabla 3.** Dimensiones para vigas tipo T (AASTHO).

Tipo	$b_f$ (mm)	$X_1$ (mm)	$X_2$ (mm)	$b_2$ (mm)	$b_3$ (mm)	$x_3$ (mm)	$x_4$ (mm)	$b_w$ (mm)	$h$ (mm)
BT-54	1066.80	88.90	50.80	660.40	406.40	114.30	152.40	152.40	1371.60
BT-63	1066.80	88.90	50.80	660.40	406.40	114.30	152.40	152.40	1600.20
BT-72	1066.80	88.90	50.80	660.40	406.40	114.30	152.40	152.40	1828.80

**Gráfico 16.** Viga tipo T (AASTHO).



### 2.4.2.3 Vigas Doble T

Gráfico 17. Viga Doble T (AASTHO).

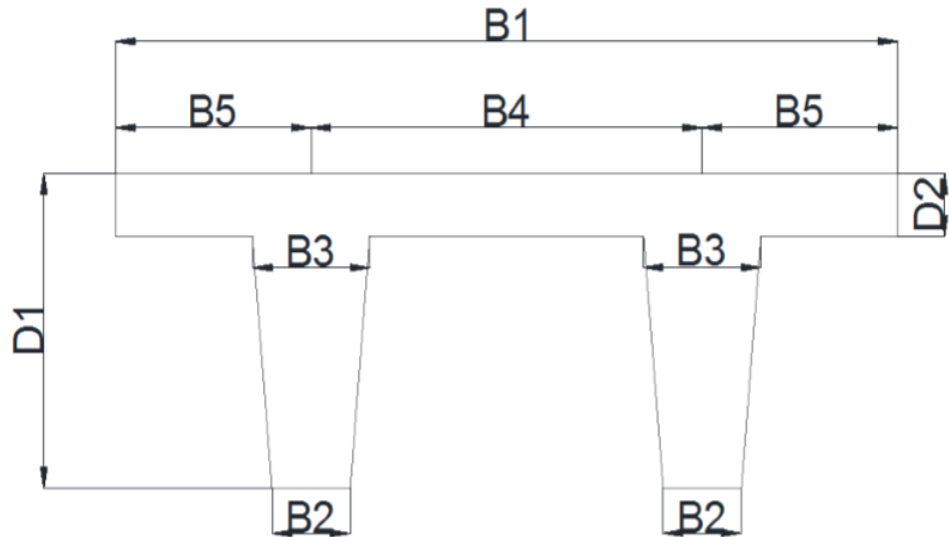


Tabla 4. Dimensiones para vigas Doble T (AASTHO).

Tipo	D1	D2	B1	B2	B3	B4	B5
DOBLE TEE9	86,40	10,00	457,20	16,50	22,90	228,60	114,30
DOBLE TEE6	86,40	10,00	365,80	12,00	19,70	183,00	91,40
DOBLE TEE8	76,20	10,00	457,20	17,50	22,90	228,60	114,30
DOBLE TEE4	86,40	10,00	304,80	12,00	19,70	152,40	76,20
DOBLE TEE2	86,40	10,00	244,00	12,00	19,70	122,00	61,00
DOBLE TEE5	76,20	10,00	365,80	12,00	19,70	183,00	91,40
DOBLE TEE7	66,00	10,00	457,20	18,40	22,90	228,60	114,30
DOBLE TEE3	66,00	10,00	304,80	9,50	14,60	152,40	76,20
DOBLE TEE1	66,00	10,00	244,00	9,50	14,60	122,00	61,00



### **2.4.3 Materiales**

El concreto que se utiliza en elementos presforzados comúnmente de una mayor resistencia que en el concreto reforzado.

En el concreto presforzado el acero de alta resistencia es uno de los materiales principales ya que el presfuerzo se logra principalmente gracias al acero. El acero permite garantizar la transmisión del esfuerzo compresivo al concreto, que es lo que se desea al momento de considerar el uso de elementos presforzados.

Hay 3 formas en las que utiliza el acero como tendones en el concreto reforzado, en torones, en varillas de acero de aleación y en alambres redondos.

Las varillas de aleación fluctúan entre 10200 y 11250 kg/cm<sup>2</sup> dependiendo del grado, el cable tensado y el alambre tiene aproximadamente 17600 kg/cm<sup>2</sup> y 19000 kg/cm<sup>2</sup> de resistencia a la tensión.

A la agrupación de alambres, sujetándose firmemente una forma helicoidal se le llama torón, comúnmente se fabrican en un conjunto de 7 alambres, en el cual 1 va en el centro y los otros 6 rodeándolo. Los torones tienen unas medidas que van desde 3/8" hasta 0.6 pulgadas de diámetro.

**Tabla 5.** Propiedades del cable de 7 alambres sin revestimiento.

<b>Grado 250</b>							
Diámetro Nominal		Resistencia a la Ruptura		Área Nominal del Cable		Carga Mínima- Elongación del 1%	
(in)	(mm)	(lb)	(Kg)	$in^2$	$mm^2$	(lb)	(kg)
0.250	6.350	9,000	4,082	0.036	23.22	7,650	3,469
0.313	7.940	14,500	6,577	0.058	37.42	12,300	5,579
0.375	9.530	20,000	9,071	0.080	51.61	17,000	7,711
0.438	11.110	27,000	12,246	0.108	69.68	23,000	10,432
0.500	12.700	36,000	16,329	0.144	92.90	30,600	13,879
0.600	15.240	54,000	24,493	0.216	139.35	45,900	20,819
<b>Grado 270</b>							
Diámetro Nominal		Resistencia a la Ruptura		Área Nominal del Cable		Carga Mínima- Elongación del 1%	
(in)	(mm)	(lb)	(Kg)	$in^2$	$mm^2$	(lb)	(kg)
0.375	9.53	23,000	10,432	0.085	54.84	19,550	8,867
0.438	11.11	31,000	14,061	0.115	74.19	26,350	11,952
0.500	12.70	41,300	18,733	0.153	98.71	35,100	15,921
0.600	15.24	58,600	26,580	0.217	140.00	49,800	22,588

Al momento de construir elementos de concreto presforzado el acero de refuerzo es muy útil. El uso del mismo nos sirve para poder aumentar la resistencia, la ductilidad, resistir esfuerzos de tensión y compresión, resistir a cortante, resistir a torsión, reducir deformaciones a largo plazo y confinar el concreto.

#### 2.4.4 Construcción

- Se selecciona y se procesa los materiales, seleccionar bien los agregados gruesos y finos de buena calidad.
- Se procede al mezclado de dichos materiales de una manera que permita que el hormigón obtenga la resistencia de diseño deseada y pueda resistir sin problemas los esfuerzos internos causados por el acero de refuerzo.
- Cuando la mezcla esta lista se procede a la colocación y vertido en el molde metálico que tenga las dimensiones requeridas por el diseño,

siempre se debe usar un vibrador que permita que la mezcla se distribuya por todo el encofrado.

- Luego se debe esperar el tiempo de fraguado y proceder a curar el hormigón mojándolo constantemente para lograr que no se creen grietas y lograr que el mismo gane resistencia.
- Por último, se procede a llevarlo al lugar deseado mediante maquinaria que permita transportar el gran peso de unas vigas de estas dimensiones.

## **2. CAPITULO III**

### **3.1 Descripción del programa y su funcionamiento**

Se creó un software mediante el uso del programa Matlab que contiene una interfaz muy sencilla para brindarle todas las comodidades y facilidades en cuanto a la inserción y obtención de datos, de tal manera que el usuario se sienta satisfecho al momento de utilizarlo, además es un programa que contiene muchas opciones de personalización, para que a futuro se pueda ingresar más opciones al momento de diseñar y muestras los resultados de manera clara.

### **3.2 Funcionalidad**

El software realizado tendrá como principal funcionalidad ayudar a cualquier usuario, ya sea a estudiantes, o a Ingenieros Civiles, debido a que muchas veces se requieren criterios adquiridos a través de estudios universitarios o por experiencia en el ejercicio de la profesión, pero en fin el software está dirigido al uso de múltiples usuarios.

### **3.3 Manual de usuario**

El enfoque del software es hacia el diseño de la viga con los torones a flexión.

**Gráfico 18.** Software para diseño a flexión de torones de viga presforzada.

**ANALISIS A FLEXIÓN DE VIGA PRESFORZADA**

SECCIONES

EXCENTRICIDAD  
 CONSTANTE  
 NO CONSTANTE

CARGAS  
W SD (plf)   
W CS (plf)   
W D (plf)   
W L (plf)   
CAMION (lb)

DATOS  
L (ft)   
f'c (psi)   
f'ci (psi)   
fpu (psi)   
Y   
At (in<sup>2</sup>)

DATOS DE LOSA  
f'c losa (psi)   
b losa (in)   
h losa (in)

VERIFICACION DE SECCION  
S (SECCION)      S (REQUERIDA)  
St (in<sup>3</sup>)  > St (in<sup>3</sup>)   
Sb (in<sup>3</sup>)  > Sb (in<sup>3</sup>)   
SECCION RECOMENDADA  
  
VERIFICAR

TORONES REQUERIDOS  
Pi calculado (lb)   
# REQ   
GRAFICAS DE MOMENTO

	# TORONES	Y (in)	SECCION	X (ft)
1				
2				
3				
4				

CALCULAR  
EXCENTRICIDAD  
PLOTTEAR  
EXPORTAR

Quitar filas      Agregar filas

**Elaborado por:** El Autor

El programa consta con una base de datos en excel para diferentes secciones de vigas presforzadas. En el archivo de excel se pueden añadir más secciones con sus respectivas propiedades geométricas y el programa leerá estas internamente, esto permite que con el tiempo el usuario pueda añadir las secciones que necesite y de esta manera tener una biblioteca de secciones más completa o con secciones personalizadas. Se nos presenta una opción para desplegar las secciones que se encuentran en la base de datos.

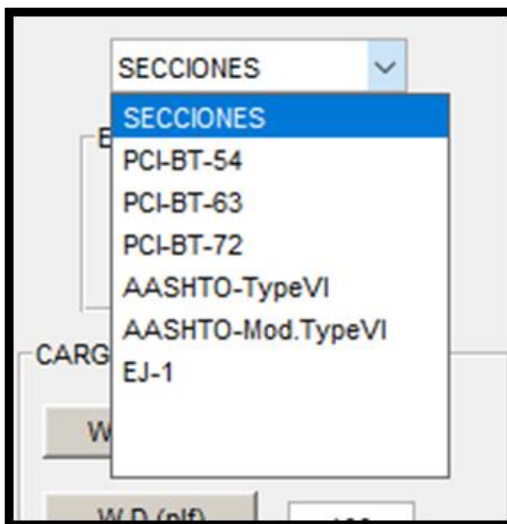
**Gráfico 19.** Desplegable de secciones.



**Elaborado por:** El Autor

Una vez abierta la opción de las secciones, se puede seleccionar una sección que el ingeniero, con su experiencia, crea que es satisfactoria.

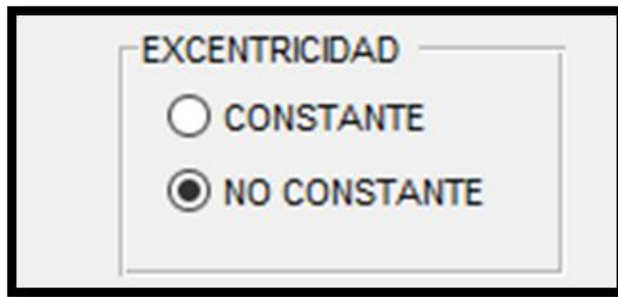
**Gráfico 20.** Selección de secciones en el software.



**Elaborado por:** El Autor

La opción para evaluar si la viga tiene configuración constante o no constante de torones influye en el cálculo del módulo de sección, que es con el cual vamos a evaluar si la sección es la adecuada para algún caso específico.

**Gráfico 21.** Cuadro de excentricidad en el software.

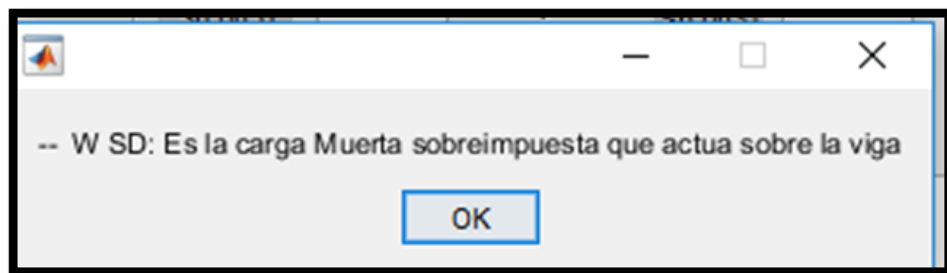


**Elaborado por:** El Autor

Los tipos de cargas que se requieren en el programa son:

- Carga sobre impuesta

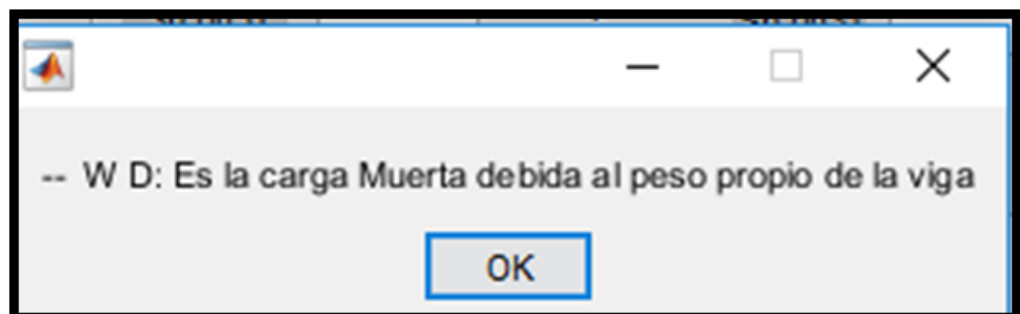
**Gráfico 22.** Ventana de carga sobre impuesta



**Elaborado por:** El Autor

- Carga muerta

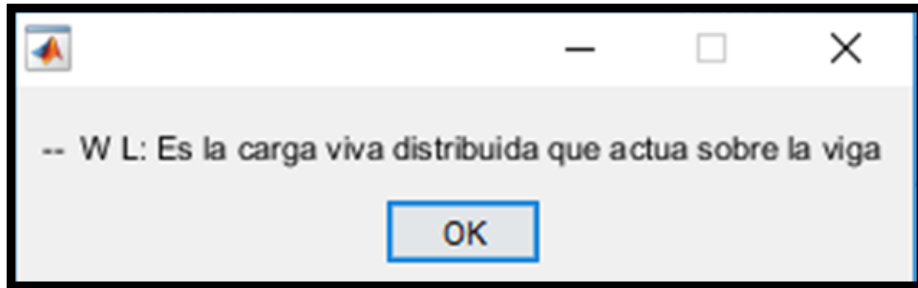
**Gráfico 23.** Ventana de carga muerta



**Elaborado por:** El Autor

- Carga viva distribuida

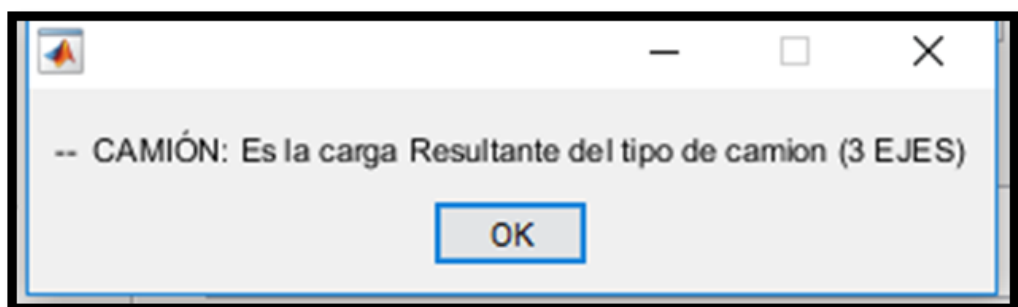
**Gráfico 24.** Ventana de carga distribuida



**Elaborado por:** El Autor

- Carga resultante del camión

**Gráfico 25.** Ventana de carga resultante del camión



**Elaborado por:** El Autor

Todas las cargas que se utilizan en este software están dadas en libras/pies. Como este software también está pensado para estudiantes al momento de ser usado y tener una interfaz amigable hicimos que el usuario pueda ver el significado de cada variable dando un "Clic" en cada una, de esta manera se despajara de cualquier duda que tenga con respecto a la mismas.



**Gráfico 26.** Cuadro de datos de cargas.



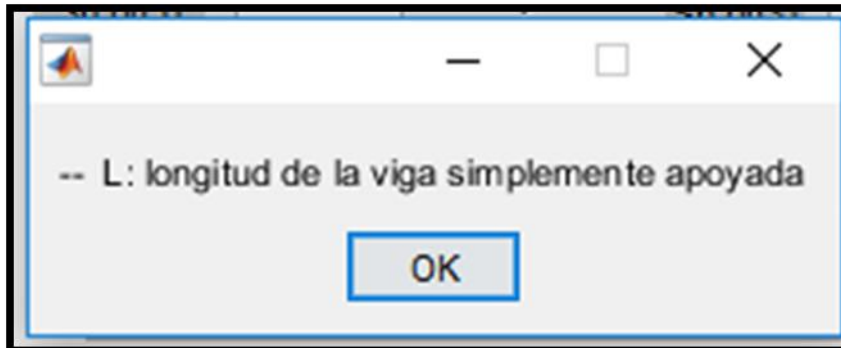
CARGAS	
W SD (plf)	1100
W D (plf)	100
W L (plf)	1000
CAMION (p)	2

**Elaborado por:** El Autor

Datos de la viga para el caso específico a analizar:

- Longitud de la viga en pies.

**Gráfico 27.** Ventana de longitud de la viga en pies.



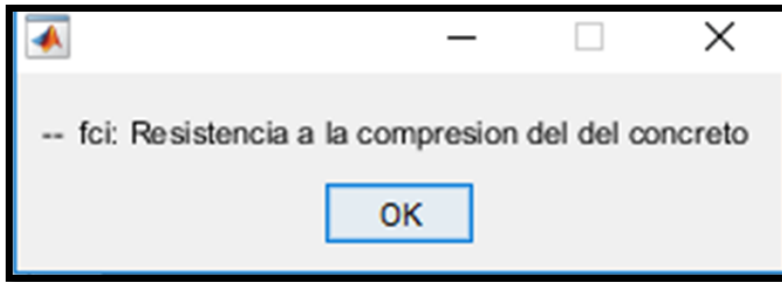
-- L: longitud de la viga simplemente apoyada

OK

**Elaborado por:** El Autor

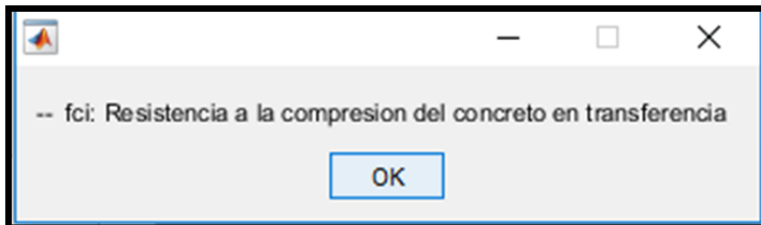
- Resistencia del hormigón a la compresión en libras/pies<sup>2</sup>.

**Gráfico 28.** Ventana de resistencia del hormigón en lb/pies<sup>2</sup>



**Elaborado por:** El Autor

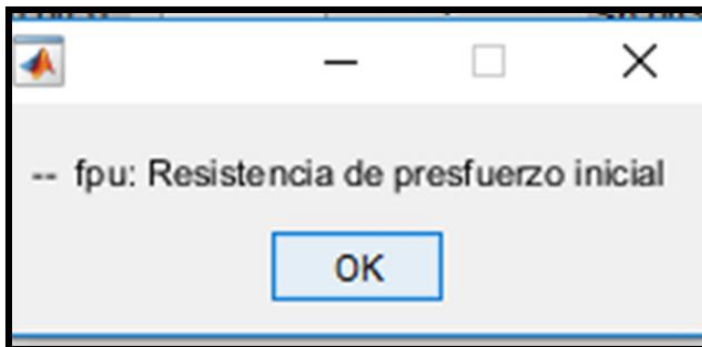
- Resistencia del hormigón en transferencia en libras/pies<sup>2</sup>.
- **Gráfico 29.** Ventana de resistencia del hormigón en transferencia en libras/pies<sup>2</sup>



**Elaborado por:** El Autor

- Resistencia a la flexión de los torones en libras/pies<sup>2</sup>.

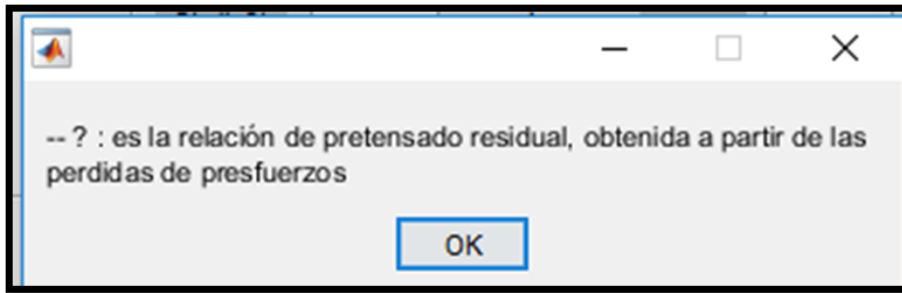
**Gráfico** Ventana de Resistencia a flexión de los torones en libras/pies<sup>2</sup>



**Elaborado por:** El Autor

- En este cuadro podemos observar la Relación de pretensado residual.

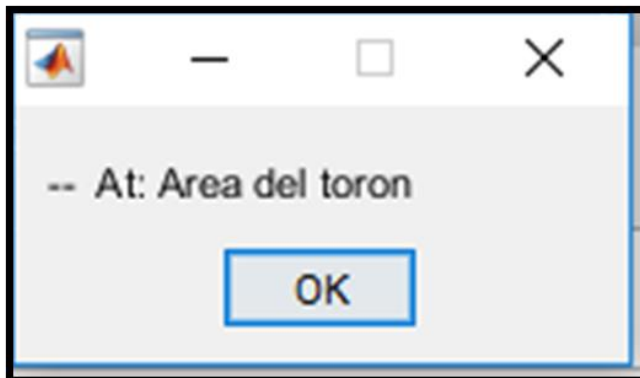
**Gráfico 31.** Ventana de la relación de pretensado residual.



**Elaborado por:** El Autor

- El área de torones

**Gráfico 32.** Ventana del área de los torones



**Elaborado por:** El Autor

**Gráfico 33.** Cuadro de datos generales de la viga.

DATOS	
L (ft)	60
f <sub>c</sub> (psi)	50000
f <sub>ci</sub> (psi)	3575
f <sub>pu</sub> (psi)	270000
Y	0.82
A <sub>t</sub> (in <sup>2</sup> )	0.153

**Elaborado por:** El Autor

El siguiente cuadro es para evaluar la viga con respecto al modulo de sección requerido por las características del proyecto. El modulo

de sección de la viga se lee a partir de la base de datos creada en excel, y el modulo de sección requerido se calcula a partir de las siguientes ecuaciones para la parte superior e inferior:

**Gráfico 34.** Formulas para el calculo del modulo de seccion (Nawi,2010)

$$S^t \geq \frac{M_D + M_{SD} + M_L}{\gamma f_{ti} - f_c}$$
$$S^b \geq \frac{M_D + M_{SD} + M_L}{f_t - \gamma f_{ci}}$$

**M<sub>D</sub>**: momento máximo por carga muerta.

**M<sub>SD</sub>**: momento máximo por carga sobre impuesta.

**M<sub>L</sub>**: momento máximo por carga viva.

**f<sub>t</sub>**: Máximo esfuerzo de tensión admisible en el hormigón con cargas de servicio después de perdidas.

**γ**: Relación de pretensado residual, obtenida a partir de las perdidas de presfuerzo.

**f<sub>ci</sub>**: Máximo esfuerzo de compresión admisible en el hormigón inmediatamente después de que se produzca la transferencia y antes de las pérdidas.

Ademas este software tiene implementado un cuadro de texto que se encuentra ubicado en la parte inferior de los módulos de secciones evaluados, donde después de verificar si todo cumple nos recomienda el tipo de viga que se debería usar para ese tramo con esas características, solo si es que la viga que se escogió al principio no cumple con lo necesario, en

caso de que todo cumple en el cuadro nos indicara que la viga que escogimos es adecuada para continuar con el diseño.

**Gráfico 35.** Cuadro de verificación de sección.

VERIFICACION DE SECCION

S (SECCION)			S (REQUERIDA)		
St (in3)	15421	>	St (in3)	524.571	OK
Sb (in3)	14915	>	Sb (in3)	3831.59	OK

SECCION RECOMENDADA

Seleccione una seccion I de acuerdo al sb y st requerido

VERIFICAR

**Elaborado por:** El Autor

Este cuadro se genera el valor que los cables deberían generar en teoría para las cargas aplicadas, además también se da el número de cables requeridos aproximados.

**Gráfico 36.** Cuadro de número requerido de torones

Panel

Pi calculado	738961
# REQ	25.5546

**Elaborado por:** El Autor

El siguiente cuadro es para la inserción de los datos de los torones. Los resultados se pueden calcular en el programa, y para tener una idea de la ubicación de los torones con respecto a su altura en la viga está el botón plotear que genera un breve vistazo a los torones.

Una vez efectuado el cálculo se pueden exportar a Excel los resultados. El programa evalúa la sección en transferencia y en servicio.

El programa evalúa 7 secciones cada vez que corre, por lo tanto, una limitación es que usuario está obligado a ingresar todas las secciones a lo largo de la viga. Como recomendación se pueden ingresar en 0 metros, al final de la viga, en la mitad de la viga, y 4 cortes más en cualquier posición de la viga, para asegurarnos que cumple los esfuerzos actuantes con los admisibles.

Número de torones (#TORONES): se ingresan los torones por grupo, de acuerdo a la posición con respecto a la altura de la viga.

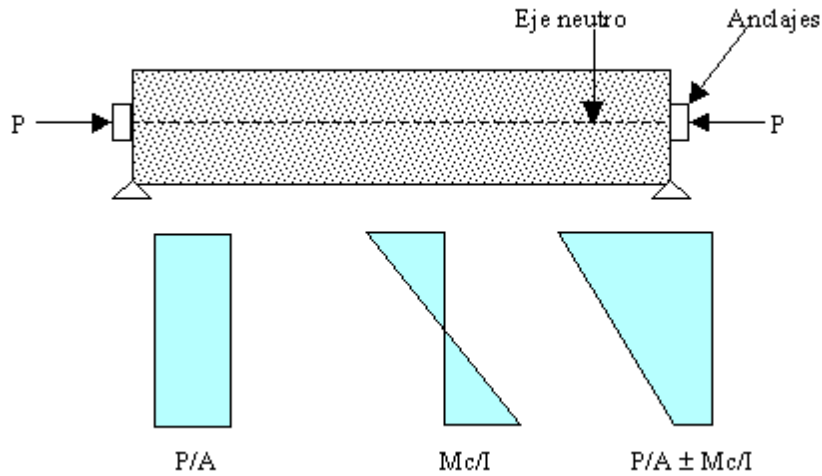
Posición de torones (Y): se especifica la altura en la que se encuentra el grupo de torones que se ha ingresado.

Sección a evaluar (SECCIÓN): se especifica a que sección pertenecen el grupo de torones ingresados.

Posición de secciones a evaluar (X): se especifica a los cuantos metros se encuentra la sección que se va a evaluar con la configuración de torones ingresada.

El programa se basa en las ecuaciones de equilibrio de las fuerzas que actúan en la viga presforzada.

**Gráfico 37.** Equilibrio de las fuerzas que actúan en la viga presforzada (construaprende,2010).



La viga debe evaluarse apenas empieza a transferirse la fuerza de los torones, y cuando ya comienza a trabajar (en servicio).

**Gráfico 38.** Ecuaciones de esfuerzo en la parte superior e inferior de la viga (Nawi,2010)

**Effective Stresses after Losses**

$$f^t = -\frac{P_e}{A_c} \left( 1 - \frac{ec_t}{r^2} \right) - \frac{M_D}{S^t} \leq f_t$$

$$f_b = -\frac{P_e}{A_c} \left( 1 + \frac{ec_b}{r^2} \right) + \frac{M_D}{S_b} \leq f_c$$

**Service-load Final Stresses**

$$f^t = -\frac{P_e}{A_c} \left( 1 - \frac{ec_t}{r^2} \right) - \frac{M_T}{S^t} \leq f_c$$

$$f_b = -\frac{P_e}{A_c} \left( 1 + \frac{ec_b}{r^2} \right) + \frac{M_T}{S_b} \leq f_t$$

**Gráfico 39.** Cuadro de inserción de datos de torones.

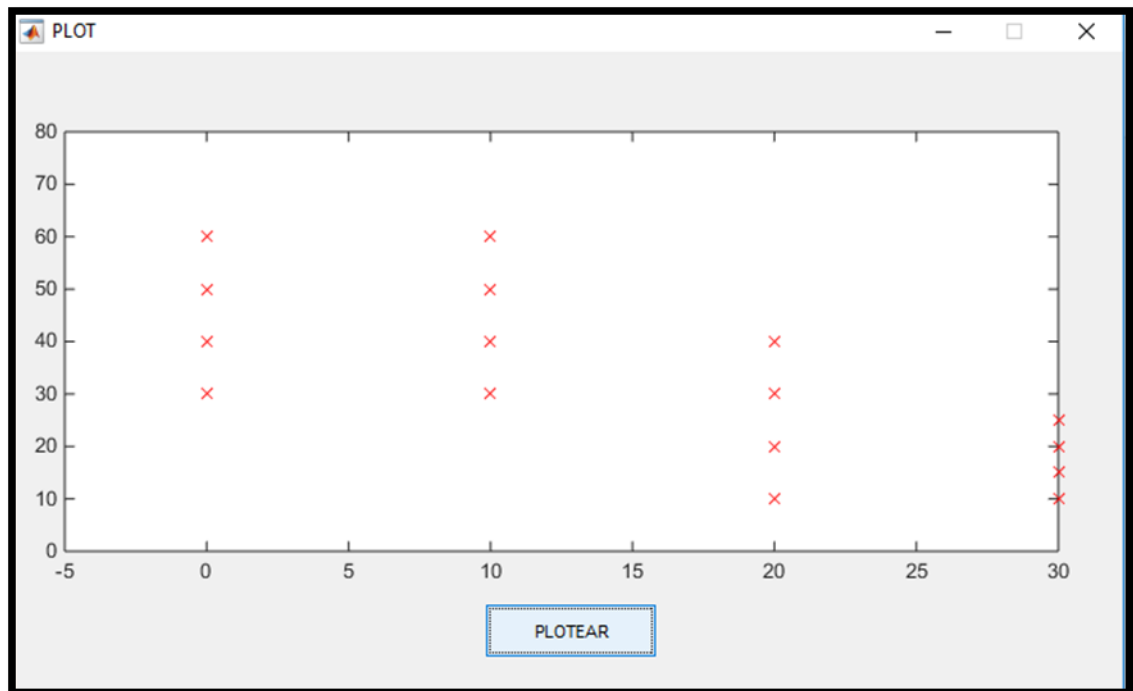
	# TORONES	Y (in)	SECCIÓN	X (ft)
1	2	50	1	10
2	2	40	1	10
3	2	30	1	10
4	4	20	1	10
5	2	45	2	20
6	2	35	2	20
7	2	25	2	20
8	4	20	2	20

**Elaborado por:** El Autor

En esta ventana se debe ingresar una configuración de los torones de acuerdo a la experiencia del usuario, ya que debe lograr que la excentricidad necesaria se logre mediante la configuración de los torones, esta luego calculara los esfuerzos actuantes contra los esfuerzos admisibles y si los actuantes son menores que los admisibles los cálculos estarán Ok, caso contrario el software enviara un mensaje de No Cumple.



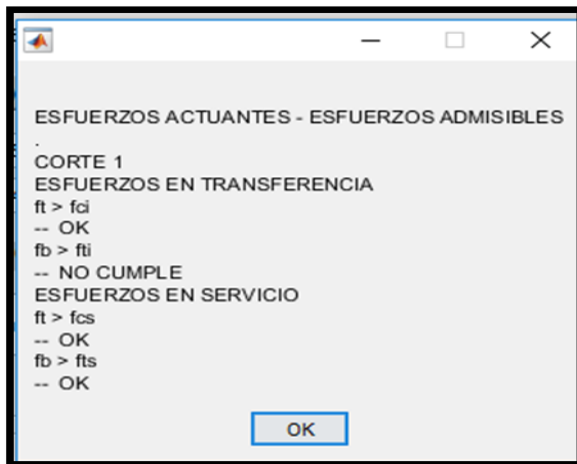
**Gráfico 40.** Ventana de ploteo para tener una aproximación de la ubicación de los torones.



**Elaborado por:** El Autor

Los resultados se presentan en un cuadro de texto, donde se especifica que si los torones no generan esfuerzos más allá de los permisibles. Además, el programa puede generar los resultados en Excel a partir del botón exportar. Si la excentricidad es constante se generarán torones a lo largo de la viga al inicio, al final, al centro y aleatoriamente en los puntos restantes hasta cumplir los 7 cortes necesarios.

**Gráfico 41.** Cuadro de resultados del programa.



Elaborado por: El Autor

**Gráfico 42.** Cuadro de resultados exportados a Excel.

CORTE 1	CORTE 2	CORTE 3	CORTE 4	CORTE 5	CORTE 6	CORTE 7
ESFUERZOS EN TRANSFERENCIA						
ft > fci -- OK	ft > fci -- OK	ft > fci -- OK	ft > fci -- OK	ft > fci -- OK	ft > fci -- OK	ft > fci -- OK
fb > fti -- NO CUMPLE	fb > fti -- NO CUMPLE	fb > fti -- NO CUMPLE	fb > fti -- NO CUMPLE	fb > fti -- OK	fb > fti -- OK	fb > fti -- OK
ESFUERZOS EN SERVICIO						
ft > fcs -- OK	ft > fcs -- OK	ft > fcs -- OK	ft > fcs -- OK	ft > fcs -- OK	ft > fcs -- OK	ft > fcs -- OK
fb > fts -- OK	fb > fts -- OK	fb > fts -- OK	fb > fts -- NO CUMPLE	fb > fts -- OK	fb > fts -- OK	fb > fts -- OK

Elaborado por: El Autor

## CAPITULO IV

### 4. CONCLUSIONES

A partir de la ejecución del Software se puede concluir lo siguiente:

1. El Software creado es capaz de cumplir satisfactoriamente con el diseño vigas presforzadas, tomando como referencia las normativas de diseño vigente.
2. La interfaz gráfica es amigable con el usuario puesto que permite la inserción de información de una manera sencilla y fácil, de tal forma que esto permite realizar múltiples procesos iterativos sin problema alguno.
3. El Software ejecutado servirá para el uso de estudiantes y profesionales, ya que es necesario tener ciertos conocimientos Elementos estructurales.

#### 4.1 Limitaciones.

- La base de datos se debe ingresar en Excel, pero debe ingresar el nombre de la sección en Matlab
- El usuario está obligado a evaluar 7 secciones a lo largo de la viga.
- Los torones deben ingresarse por grupos de acuerdo a su posición con respecto a la altura de la viga en la tabla de Excel presente en el programa.

## ANEXOS

### Código del Gui Presforzado.

```
function varargout = VIGAPRESFORZADA(varargin)
% VIGAPRESFORZADA MATLAB code for VIGAPRESFORZADA.fig

% Last Modified by GUIDE v2.5 19-Sep-2017 16:30:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @VIGAPRESFORZADA_OpeningFcn, ...
                  ...
                  'gui_OutputFcn',  @VIGAPRESFORZADA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function VIGAPRESFORZADA_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;

axes(handles.logo)

handles.imagen=imread('ucsg.jpg');
imagesc(handles.imagen)
axis off

guidata(hObject, handles);

function varargout = VIGAPRESFORZADA_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function menudesecciones_Callback(hObject, eventdata, handles)

%ES PARA LEER LA
SECCION UTILIZADA DEL DESPLEABLE DEL GUI
```

```

contenido=get(hObject,'String');           %GET ES PARA OBTENER
TODA LA MATRIZ QUE SE GENERA DE LAS SECCIONES EN TIPO STRING QUE ES
TEXTO
a=get(hObject,'Value');                   %SE GENERA VARIABLE
LLAMA A PARA SELEC EL PUESTO DE LA SECCION ELEGIDA
texto=contenido(a);                       %SE BUSCA LA SECCION DE
ACUERDO AL PUESTO
seccionfinal=texto;                      %RENOMBRAR LA VARIABLE
TEXTO POR SECCION FINAL
save ('seccion','seccionfinal');         %GUARDAR LA SECCION
SELECCIONADA

```

```

function menudecciones_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function wsd_Callback(hObject, eventdata, handles)

```

```

function wsd_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function wl_Callback(hObject, eventdata, handles)

```

```

function wl_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function wd_Callback(hObject, eventdata, handles)

```

```

function wd_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function wsds_Callback(hObject, eventdata, handles)

function wsds_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function VERIFICAR1_Callback(hObject, eventdata, handles)

%AQUI VA A FUNCIONAR LA PRIMERA PARTE DEL PROGRAMA
%1 PRIMERO SE VAN A LEER LOS DATOS
%2 CALCULAR LOS MOMENTOS A LO LARGO DE LA VIGA = GENERANDO FORMULA
%3 SE VA A LEER EL MAXIMO MOMENTO ::PARA CADA CASO, MD, MSL (QUE SON
CARGAS
%DISTRIBUIDAS), Y ML (QUE ES LA DE DISTRIBUIDA MAS PUNTUAL DE
CAMION)
%4 CARGAR Y UTILIZAR LA SECCION CON LOS DATOS TRAIOS DE EXCEL

%5 VERIFICAR LA SECCION

%lectura de datos
wsd11=str2double(get(handles.wsd, 'String')); %CARGA
SOBREIMPUESTA
wsd=(wsd11(:,1));

wl11=str2double(get(handles.wl, 'String'));%CARGA VIVA
wl=(wl11(:,1));

wd11=str2double(get(handles.wd, 'String'));%CARGA MUERTA
wd=(wd11(:,1));

wsds11=str2double(get(handles.wcs, 'String'));%CARGA LOSA
wsds=(wsds11(:,1));

cam111=str2double(get(handles.cam1, 'String'));%% ES EL CAMION , LA
CARGA RESULTANTE
cam1=(cam111(:,1));

luz11=str2double(get(handles.luz, 'String'));%LONGITUD DE VIGA
luz=(luz11(:,1));

```

```

fc11=str2double(get(handles.fc, 'String'));%ESFUERZO A COMPRESION
fc=(fc11(:,1));

save('longitud','luz');

fc11=str2double(get(handles.fci, 'String'));%ESFUERO A COMPRESION
INICIAL
fci=(fc11(:,1));

fp11=str2double(get(handles.fpu, 'String'));%ESFUERZO A TENSION DE
TORONES
fpu=(fp11(:,1));

yyy11=str2double(get(handles.yyy, 'String'));%FACTOR GAMA
yyy=(yyy11(:,1));

diamtoron11=str2double(get(handles.attd, 'String'));%AREA DEL TORON
diamtoron=(diamtoron11(:,1));
At=diamtoron;

%LECTURA DE DATOS DE LA LOSA
fclosa11=str2double(get(handles.fclosa, 'String'));%ESFUERZO A
COMPRESION DEL CONCRETO DE LA LOSA
fclosa=(fclosa11(:,1));

hlosa11=str2double(get(handles.hlosa1, 'String'));%ALTURA DE LA LOSA
hlosa=(hlosa11(:,1));

blosa11=str2double(get(handles.blosa1, 'String'));%ANCHO DE LA LOSA
blosa=(blosa11(:,1));

%aqui SE LEE LA SECCION QUE SE HA ELEGIDO;

load ('seccion','seccionfinal');
seccionfinal;

%2da parte: CALCULAR MOMENTOS A LO LARGO DE LA VIGA
long=[0:0.01:luz]';
qq=length(long);
D=long;

%MOMENTOS Y CORTANTE

Mwd=-((wd/2).*long.*(luz-long))*12; %MOMENTO POR
CARGA MUERTA
Vwd=- (wd*luz/2)+(wd*long);
LM=[Mwd,D,Vwd];

Mwsd=-((wsd/2).*long.*(luz-long))*12; %MOMENTO
CARGA SOBREPUESTA
Vwsd=- (wsd*luz/2)+(wsd*long);
LF=[Mwsd,D,Vwsd];

Mwsds=-((wsds/2).*long.*(luz-long))*12; %MOMENTO
CARGA LOSA
Vwsds=- (wsds*luz/2)+(wsds*long); %carga LOSA

```

```

LS=[Mwsds,D,Vwsds];

%CARGA VIVA MOMENTO

Mw11=cell(qq,1); %CARGA PUNTUAL DE CAMION
%CREA CELL(CELDA) CON LA CANTIDAD DE VALORES CORRESPONDIENTES A LONG
for j=1:qq %CONDICIONAL DE FOR SE
CALCULAN VALORES DEDE 1 HASTA qq
    if long(j)<(luz/2); %SI LA LONGITUD ES IGUAL
A 1
        Mw11{j}=(cam1/2).*long(j));
    else %CONDICIONAL EN CASO DE
NO COMPLIRSE EL IF
        Mw11{j}=(cam1/2)*(luz-long(j));
    end
end

Mw11=(cell2mat(Mw11))*(-1)*12; %TRANSFORMA DE CELDAS A
VALOR NUMERICO Y SE INVIERTE EL MOMENTO PRA QUE EL GRAFICO SALGA
HACIA ABAJO Y SE MULTIPLICA POR 12 PARA PARASALO A PULGADAS

Mw12=-((wl/2).*long.*(luz-long))*12; %CARGA DISTRUBUIDA DE
CAMION

Mw1=Mw11+Mw12;
Vw1=-((wl*luz/2)+(wl*long));
LL=[Mw1,D,Vw1]; %MATRIZ DONDE SE GUARDA
MOMENTO POSICION Y CORTANTE

save('losadato','fclosa','blosa','hlosa','fc')
save('momentos','Mwd','Mwsd','Mwsds','Mw1','LM','LF','LS','LL');

%3RA PARTE: EVALUAR LOS MAXIMOS MOMENTOS

Mwdmax=max(abs(Mwd));
Mwsdmax=max(abs(Mwsd));
% Mwsdsmax=max(abs(Mwsds));
Mw1max=max(abs(Mw1));

%4TA PARTE: LEER LA SECCION CON SUS DATOS

nombredearchivoexcel='SECCIONES.xlsx';
nombredehoja='SECCION';
[ num,txt ] = xlsread( nombredearchivoexcel,nombredehoja );
aa=num;
bb=txt;

%SEPARO CADA COLUMNA DE SUS MATRICES NUMERICAS O DE TEXTO PARA
AGRUPARLAS EN
%UNA SOLA MATRIZ TIPO CELDA, ESTO ES PARA DESPUES LEER CADA FILA
m=bb(:,1); %representa la SECCION
c2=aa(:,1);c3=aa(:,2);c4=aa(:,3);c5=aa(:,4);c6=aa(:,5);c7=aa(:,6);
c8=aa(:,7);c9=aa(:,8);c10=aa(:,9);c11=aa(:,10);

```



```

%matriz de secciones con sus propiedades geometricas
v=[m, num2cell(c2), num2cell(c3), num2cell(c4), num2cell(c5),
num2cell(c6), num2cell(c7), num2cell(c8), num2cell(c9),
num2cell(c10), num2cell(c11)];

%SELECCIONO LA FILA DE LAS SECCIONES
sec=seccionfinal;
tf=ismember(v(:,1), (sec));
seccionf=v(tf,:); %aqui estan todos los datos
de la seccion elegida

H=cell2mat(seccionf(:,2)); %altura de la viga
e=cell2mat(seccionf(:,3)); %espesor del alma
A=cell2mat(seccionf(:,4)); %area de la viga
I=cell2mat(seccionf(:,5)); %ineria de la seccion
yt=cell2mat(seccionf(:,6)); %dist del eje neutro al tope de la
seccion
yb=cell2mat(seccionf(:,7)); %dist del eje neutro al botton de la
seccion
sts=cell2mat(seccionf(:,8)); %modulo de seccion t
sbs=cell2mat(seccionf(:,9)); %modulo de seccion b
p=cell2mat(seccionf(:,10));
xa=cell2mat(seccionf(:,11));
r2=I/A;
save('secciondatos','H','e','A','I','yt','yb','sts','sbs','r2');
set(handles.stseccion,'String',sts);
set(handles.sbseccion,'String',sbs);

%5ta PARTE: EVALUAR LA SECCION. YA LOS CALCULOS DEL EJERCICIO

%5.1 Calculos de esfuerzos permisibles en tranferencia

fcil=-0.6*fc; %ESF TRANSF
fti=6*((fci)^(1/2)); %ESF TRANSF
fc1=-0.45*fc; %ESF SEVICIOS
ft=12*((fc)^(1/2)); %ESF SEVICIOS
%o se puede incrementar hasta 12 en sistamas de losas en una
direccion

save('esfpermisibles','fcil','fti','fc1','ft');
%5.2 ¿seccion constante? ESTA PARTE ES PARA LA SECCION REQUERIDA

load ('constantesino','ccl1');
ccl1;

%5.3 condicional para elegir seccion

st1= (Mwdmax+Mwsdmax+Mwlmx)/(((yyy)*(fti))-fc1);
sb1= (Mwdmax+Mwsdmax+Mwlmx)/(ft-((yyy)*(fcil)));
st2= (((1-yyy)*(Mwdmax))+Mwsdmax+Mwlmx)/(((yyy)*(fti))-fc1);
sb2= (((1-yyy)*(Mwdmax))+Mwsdmax+Mwlmx)/(ft-((yyy)*(fcil)));

if ccl1==1
    st=st1;
    sb=sb1;
else ccl1=0;
    st=st2;
    sb=sb2;

```

```

end

%MOSTRAR EN EL GUI EL ST Y SB REQUERIDO
set(handles.strequerida,'String',st);
set(handles.sbrequerida,'String',sb);

%PRIMER OK DE VERIFICACION DE SECCION
if sts>st
    set(handles.stok,'String','OK');
else
    set(handles.stok,'String','ELEGIR OTRA SECCION');
end
if sbs>sb
    set(handles.sbok,'String','OK');
else
    set(handles.sbok,'String','ELEGIR OTRA SECCION');
end
end
%RECOMENDACION DE LA SECCION

if ((st-sb)/st)>0.5
    set(handles.edit16,'String','Seleccione una seccion T de acuerdo
al sb y st requerido');
else
    if luz<30
        set(handles.edit16,'String','Seleccione una seccion
RECTANGULAR de acuerdo al sb y st requerido');
    else
        set(handles.edit16,'String','Seleccione una seccion I de
acuerdo al sb y st requerido');
    end
end
end

%AQUI SE manda al PANEL los datos PARA DAR EL NUMERO DE TORONES
REQUERIDOS Y LA FUERZA
%NECESARIA PARA ESOS TORONES

fcip=fti-(yt/H)*(fti-fci1);
Pir=A*(abs(fcip));set(handles.edit21,'String',Pir);%lb
fpi=0.7*fpu;
Apreq=Pir/fpi;
numtorreq=Apreq/At;set(handles.edit54,'String',numtorreq);
%Pireal=fpi*At;set(handles.edit22,'String',Pireal);

%LEER LA TABLA DE TORONES

TAB=(get(handles.uitable5, 'Data'));
numtoron=str2double(TAB(:,1));

function stseccion_Callback(hObject, eventdata, handles)

function stseccion_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function sbseccion_Callback(hObject, eventdata, handles)

function sbseccion_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function strequerida_Callback(hObject, eventdata, handles)

function strequerida_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sbrequerida_Callback(hObject, eventdata, handles)

function sbrequerida_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function stok_Callback(hObject, eventdata, handles)

function stok_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function sbok_Callback(hObject, eventdata, handles)
```

```
function sbok_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit16_Callback(hObject, eventdata, handles)
```

```
function edit16_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function luz_Callback(hObject, eventdata, handles)
```

```
function luz_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function fc_Callback(hObject, eventdata, handles)
```

```
function fc_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```

function fci_Callback(hObject, eventdata, handles)

function fci_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fpu_Callback(hObject, eventdata, handles)

function fpu_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function yyy_Callback(hObject, eventdata, handles)

function yyy_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fclosa_Callback(hObject, eventdata, handles)

function fclosa_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function hlosa_Callback(hObject, eventdata, handles)

tex1='-- h losa: espesor de losa ';
msgbox({tex1});

```

```
function hlosa_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function blosa_Callback(hObject, eventdata, handles)
```

```
tex1='-- blosa: ancho de la porcion actuante de losa sobre la viga  
presforzada  ';  
msgbox({tex1});
```

```
function blosa_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
function edit36_Callback(hObject, eventdata, handles)
```

```
function edit36_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit37_Callback(hObject, eventdata, handles)
```

```
function edit37_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit38_Callback(hObject, eventdata, handles)
```

```
function edit38_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit39_Callback(hObject, eventdata, handles)
```

```
function edit39_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit40_Callback(hObject, eventdata, handles)
```

```
function edit40_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit41_Callback(hObject, eventdata, handles)
```

```
function edit41_CreateFcn(hObject, eventdata, handles)
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton2_Callback(hObject, eventdata, handles)

function edit23_Callback(hObject, eventdata, handles)

function edit23_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit24_Callback(hObject, eventdata, handles)

function edit24_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit25_Callback(hObject, eventdata, handles)

function edit25_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit26_Callback(hObject, eventdata, handles)

function edit26_CreateFcn(hObject, eventdata, handles)

```



```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit27_Callback(hObject, eventdata, handles)
```

```
function edit27_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit28_Callback(hObject, eventdata, handles)
```

```
function edit28_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit30_Callback(hObject, eventdata, handles)
```

```
function edit30_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit31_Callback(hObject, eventdata, handles)
```

```
function edit31_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

function edit32_Callback(hObject, eventdata, handles)

function edit32_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit33_Callback(hObject, eventdata, handles)

function edit33_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit34_Callback(hObject, eventdata, handles)

function edit34_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit35_Callback(hObject, eventdata, handles)

function edit35_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)

function edit21_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit22_Callback(hObject, eventdata, handles)

function edit22_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function AGREGARFILAS_Callback(hObject, eventdata, handles)

%AGREGAR FILA
data=get(handles.uitable5, 'data');
a1=size(data,1)+1;
data(end+1,:)=cell(1,4);
data(a1,2)={' '};
set(handles.uitable5,'data', data);

function QUITARFILAS_Callback(hObject, eventdata, handles)

%QUITAR FILA
oldDat = get(handles.uitable5,'data');
nRows = size(oldDat,1)-1;
data = oldDat(1:nRows,:);
set(handles.uitable5,'data',data)

function CALCULAR_Callback(hObject, eventdata, handles)
% Admisible vs Actuantes

load('secciondatos','H','e','A','I','yt','yb','sts','sbs','r2');
load('momentos','Mwd','Mwsd','Mwsds','Mwl','LM','LF','LS','LL');
load('esfpermisibles','fcil','fti','fcl','ft');
load('constantesino','ccl');
load('losadato','fclosa','blosa','hlosa','fc');

fcil;
fti;
fcl;
ft;

%datos de losa para calcular seccion compuesta

nfg=fclosa/fc;
beq=blosa*nfg;
ybc=A*yb+(beq*hlosa*(H-(hlosa/2))/(A+beq*hlosa));
ytc=- (H-ybc);
ytg=- (H+hlosa-ybc);
Ic=I+A*((ybc-yb)^2)+((beq*(hlosa^3))/12)+beq*hlosa*((H+(hlosa/2)-
ybc)^2);
Stg=Ic/ytg;
Stc=Ic/ytc;

```

```

Sbc=Ic/ycb;

att=str2double(get(handles.attd, 'String'));
ato=(att(:,1));%area de toron
fpuu=str2double(get(handles.fpu, 'String'));
fpu=(fpuu(:,1));%fpu
fpi=0.7*fpu;%fpi
fpe=0.82*fpi;

%LEER LA TABLA
TAB=(get(handles.uitable5, 'Data'));
save('tablatorones','TAB');
numtoron=str2double(TAB(:,1));
Y=str2double(TAB(:,2));
corte=str2double(TAB(:,3));
X=str2double(TAB(:,4));

[qq,~]=size(numtoron);          %CALCULAR CUANTAS FILAS TIENE PARA
LUEGO CORRER ALGUN CONDIDICONAL FOR

%


---



%Condicional Para leer los valores de configuracion de torones

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==1;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y1=n;
X1=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==2;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y2=n;
X2=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==3;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
end

```

```

n=cell2mat(n);
m=cell2mat(m);
Y3=n;
X3=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==4;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y4=n;
X4=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==5;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y5=n;
X5=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==6;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y6=n;
X6=m;

n=cell(qq,1);
m=cell(qq,1);
for i=1:qq;
    if corte(i)==7;
        n{i}=Y(i);
        m{i}=X(i);
    end
end
n=cell2mat(n);
m=cell2mat(m);
Y7=n;
X7=m;

save('ubicaciondetorones','Y1','X1','Y2','X2','Y3','X3','Y4','X4','Y5',
'X5','Y6','X6','Y7','X7')

```

```

%PLOT
%-----
%-----
%-----

%Ventana de Texto Calcular
%CORTE 1 AA*

if ccl==0

n=cell(qq,1); %Celdas para guardar
datos en un siguiente for
for i=1:qq;
    if corte(i)==1;
        n{i}=numtoron(i);
    end
end
n=cell2mat(n); %numero de torones en
fila
nuntotal=sum(n); %suma de los torones de
acuerdo al corte especificado
SAe=nuntotal*ato; %sumatoria de las areas
toron= numero total de torones* area de toron

Ae=cell(qq,1); %Area del toron por
posicion en el corte especificado (1)
for i=1:qq;
    if corte(i)==1;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
Ae=cell2mat(Ae); %area de torn por
seccion en guardado *la posicion en Y
Aet=sum(Ae); %sumatoria del las areas
de torones por seccion

% rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;
el=e;

for i=1:qq;
    if corte(i)==1;
        pos=X(i);
    end
end
pos;

pp=pos;

```

```

tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);
MUERTA DE VIGA %=== Md=====pp carga

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);
Mds=====SOBREIMPUESTA %===

tf=ismember(LS(:,2),(pp));
Mwsdsx=LS(tf,:);
%=== Mdss=====pp LOSA

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);
%=== Mdss=====carga VIVA

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe; %FPI Y FPE SON PORCENTJES DE FPU

%Esfuerzos en Transferencia

ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts); %esfuerzo en la
transferencia en el tope

fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs); %esf en la transferencia
en la parte baja

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);

fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);

%Esfuerzos en seccion compuesta

ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

%TEXTO EN VENTANA DE CALCULAR

tex00='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
tex000='.';
tex01='CORTE 0';
tex0000='ESFUERZOS EN TRANSFERENCIA';
tex00000='ft > fci';

%MENSAJE 1ER PUNTO

```

```

if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
    tex1='-- NO CUMPLE';
else
    tex1='-- OK';
end

tex22='fb > fti';

if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO
    tex2='-- NO CUMPLE';
else
    tex2='-- OK';
end

tex33='ESFUERZOS EN SERVICIO';
tex333='ft > fcs';

%MENSAJE 1ER PUNTO

if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
    tex3='-- NO CUMPLE';
else
    tex3='-- OK';
end
tex444='fb > fts';
%MENSAJE 1ER PUNTO
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
    tex4='-- NO CUMPLE';
else
    tex4='-- OK';
end

%MENSAJE SECCION COMPUESTA 1ER PUNTO

%MENSAJE 1ER PUNTO COMP
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
    tex4444='-- NO CUMPLE';
else
    tex4444='-- OK';
end

%MENSAJE 1ER PUNTO b

if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
    tex44444='-- NO CUMPLE';
else
    tex44444='-- OK';
end

msgbox({tex00,tex000,tex01,tex0000,tex00000,tex1,tex22,tex2,tex33,te
x333,tex3,tex444,tex4,'ESFUERZOS EN SECCION
COMPUESTA','fc>ftc',tex4444,'ft>tbc',tex44444})

```



```

save('es1','tex00','tex000','tex01','tex0000','tex00000','tex1','tex
22','tex2','tex33','tex333','tex3','tex444','tex4','tex4444','tex444
44')

```

```

%AQUI TERMINA LA EL CORTE 1
AA*****+

```

```

%AQUI EMPIEZA LA EL CORTE 2
AA*****+

```

```

n=cell(qq,1);                                     %genera unas celdas para
guardar datos del siguiente for
for i=1:qq;
    if corte(i)==2;
        n{i}=numtoron(i);
    end
end

```

```

n=cell2mat(n);                                     %numero de torones en fila
nuntotal=sum(n);                                  %suma de los torones de
acuerdo al corte especificado                    %sumatoria de las areas
SAe=nuntotal*ato;                                 %sumatoria de las areas
toron= numero total de torones* area de toron

```

```

Ae=cell(qq,1);                                     %Area del toron por
posicion en el corte especificado (1)
for i=1:qq;
    if corte(i)==2;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end

```

```

Ae=cell2mat(Ae);                                   %area de torn por seccion
en guardado *la posicion en Y
Aet=sum(Ae);                                       %sumatoria del las areas
de torones por seccion

```

```

rec=1;                                           %2 pulg de recubrimiento
variar

```

```

yr=Aet/SAe;
e=yr;
e2=e;%:
:

```

```

for i=1:qq;
    if corte(i)==2;
        pos=X(i);
    end
end
pos;

```

```

pp=pos;
tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2),(pp));
Mwsdx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);

fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);

%ESFUERZO SECCION COMUESTA

ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

tex55='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
tex55='.';
tex05='CORTE 1';
tex5555='ESFUERZOS EN TRANSFERENCIA';
tex55555='ft > fci';
%MENSAJE 1ER PUNTO
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
    tex5='-- NO CUMPLE';
else
    tex5='-- OK';
end

tex66='fb > fti';
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO

```

```

        tex6='-- NO CUMPLE';
    else
        tex6='-- OK';
    end

tex77='ESFUERZOS EN SERVICIO';
tex777='ft > fcs';
%MENSAJE 1ER PUNTO
if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
    tex7='-- NO CUMPLE';
else
    tex7='-- OK';
end
tex88='fb > fts';
%MENSAJE 1ER PUNTO
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
    tex8='-- NO CUMPLE';
else
    tex8='-- OK';
end

%MENSAJE SECCION COMPUESTA
%MENSAJE 1ER PUNTO
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
    tex888='-- NO CUMPLE';
else
    tex888='-- OK';
end

%MENSAJE 1ER PUNTO
if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
    tex8888='-- NO CUMPLE';
else
    tex8888='-- OK';
end

msgbox({tex55,tex555,tex05,tex5555,tex55555,tex5,tex66,tex6,tex77,te
x777,tex7,tex88,tex8,'ESFUERZOS EN SECCION
COMPUESTA','fc>ftc',tex888,'ft>tbc',tex8888})
save('es2','tex55','tex555','tex05','tex5555','tex55555','tex5','tex
66','tex6','tex77','tex777','tex7','tex88','tex8','tex888','tex8888'
)
%AQUI TERMINA LA EL CORTE 2
AA*****+

%AQUI EMPIEZA LA EL CORTE 3
AA*****+

n=cell(qq,1);%genera unas celdas para guardar datos del siguiente
for
for i=1:qq;
    if corte(i)==3;
        n{i}=numtoron(i);
    end
end
n=cell2mat(n);%numero de torones en fila
nuntotal=sum(n);%suma de los torones de acuerdo al corte
especificado

```

```

SAe=nuntotal*ato; %sumatoria de las areas toron= numero toral de
torones* area de toron

Ae=cell(qq,1);%Area del toron por posicion en el corte especificado
(1)
for i=1:qq;
    if corte(i)==3;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
Ae=cell2mat(Ae);%area de torn por seccion en guardado *la posicion
en Y
Aet=sum(Ae);%sumatoria del las areas de torones por seccion

rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;
e3=e;%:.....
:.....

for i=1:qq;
    if corte(i)==3;
        pos=X(i);
    end
end
pos;

pp=pos;
tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2),(pp));
Mwsdx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

```

```
fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja
```

```
%Esfuerzos en Servicio
```

```
ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);
```

```
fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);
```

```
%ESFUERZOS SECCION COMPUESTA
```

```
ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);
```

```
fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);
```

```
tex99='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
```

```
tex999='.';
```

```
tex09='CORTE 2';
```

```
tex9999='ESFUERZOS EN TRANSFERENCIA';
```

```
tex99999='ft > fci';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
```

```
    tex9='-- NO CUMPLE';
```

```
else
```

```
    tex9='-- OK';
```

```
end
```

```
tex1010='fb > fti';
```

```
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO
```

```
    tex10='-- NO CUMPLE';
```

```
else
```

```
    tex10='-- OK';
```

```
end
```

```
tex1212='ESFUERZOS EN SERVICIO';
```

```
tex121212='ft > fcs';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
```

```
    tex12='-- NO CUMPLE';
```

```
else
```

```
    tex12='-- OK';
```

```
end
```

```
tex1313='fb > fts';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
```

```
    tex13='-- NO CUMPLE';
```

```
else
```

```
    tex13='-- OK';
```

```
end
```

```
%MENSAJE SECCION COMPUESTA
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
```

```
    tex1333='-- NO CUMPLE';
```

```
else
```

```
    tex1333='-- OK';
```

```
end
```

```

%MENSAJE 1ER PUNTO
if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
    tex13333='-- NO CUMPLE';
else
    tex13333='-- OK';
end

msgbox({tex99,tex999,tex09,tex9999,tex99999,tex9,tex1010,tex10,tex12
12,tex121212,tex12,tex1313,tex13,'ESFUERZOS EN SECCION
COMPUESTA','fc>ftc',tex1333,'ft>tbc',tex13333})
save('es3','tex99','tex999','tex09','tex9999','tex99999','tex9','tex
1010','tex10','tex1212','tex121212','tex12','tex1313','tex13','tex13
33','tex13333')
%AQUI TERMINA LA DEL CORTE
3AA*****+

%AQUI EMPIEZA LA DEL CORTE 4
AA*****+

n=cell(qq,1);%genera unas celdas para guardar datos del siguiente
for
for i=1:qq;
    if corte(i)==4;
        n{i}=numtoron(i);
    end
end
n=cell2mat(n);%numero de torones en fila
nuntotal=sum(n);%suma de los torones de acuerdo al corte
especificado
SAe=nuntotal*ato;%sumatoria de las areas toron= numero total de
torones* area de toron

Ae=cell(qq,1);%Area del toron por posicion en el corte especificado
(1)
for i=1:qq;
    if corte(i)==4;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
Ae=cell2mat(Ae);%area de torn por seccion en guardado *la posicion
en Y
Aet=sum(Ae);%sumatoria del las areas de torones por seccion

rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;
e4=e;%:
:
:

for i=1:qq;
    if corte(i)==4;
        pos=X(i);
    end
end
pos;

```

```

pp=pos;
tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2),(pp));
Mwsdsx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);

fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);

%ESFUERZOS SECCION COMPUESTA
ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

texaa='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
texaaa='.';
tex0a='CORTE 3';
texaaaa='ESFUERZOS EN TRANSFERENCIA';
texaaaaa='ft > fci';
%MENSAJE 1ER PUNTO
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
    texa='-- NO CUMPLE';
else
    texa='-- OK';
end

texbb='fb > fti';
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO

```

```

        texb='-- NO CUMPLE';
    else
        texb='-- OK';
    end

    texcc='ESFUERZOS EN SERVICIO';
    texccc='ft > fcs';
    %MENSAJE 1ER PUNTO
    if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
        texc='-- NO CUMPLE';
    else
        texc='-- OK';
    end
    texdd='fb > fts';
    %MENSAJE 1ER PUNTO
    if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
        texd='-- NO CUMPLE';
    else
        texd='-- OK';
    end

    %MENSAJE SECCION COMPUESTA
    %MENSAJE 1ER PUNTO
    if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
        texddd='-- NO CUMPLE';
    else
        texddd='-- OK';
    end

    %MENSAJE 1ER PUNTO
    if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
        texdddd='-- NO CUMPLE';
    else
        texdddd='-- OK';
    end

    msgbox({texaa,texaaa,tex0a,texaaaa,texaaaaa,texa,texbb,texb,texcc,te
    xccc,texc,texdd,texd,'ESFUERZOS EN SECCION
    COMPUESTA','fc>ftc',texddd,'ft>tbc',texdddd})
    save('es4','texaa','texaaa','tex0a','texaaaa','texaaaaa','texa','tex
    bb','texb','texcc','texccc','texc','texdd','texd','texddd','texdddd'
    )
    %AQUI TERMINA LA DEL CORTE 4
    AA*****

    %AQUI empieza LA DEL CORTE 5
    AA*****

    n=cell(qq,1);%genera unas celdas para guardar datos del siguiente
    for
    for i=1:qq;
        if corte(i)==5;
            n{i}=numtoron(i);
        end
    end
    n=cell2mat(n);%numero de torones en fila
    nuntotal=sum(n);%suma de los torones de acuerdo al corte
    especificado

```



```

SAe=nuntotal*ato; %sumatoria de las areas toron= numero toral de
torones* area de toron

Ae=cell(qq,1);%Area del toron por posicion en el corte especificado
(1)
for i=1:qq;
    if corte(i)==5;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
Ae=cell2mat(Ae);%area de torn por seccion en guardado *la posicion
en Y
Aet=sum(Ae);%sumatoria del las areas de torones por seccion

rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;
e5=e;%:.....:
:.....:

for i=1:qq;
    if corte(i)==5;
        pos=X(i);
    end
end
pos;

pp=pos;
tf=ismember(LM(:,2), (pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2), (pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2), (pp));
Mwsdsx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2), (pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

```

```
fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja
```

```
%Esfuerzos en Servicio
```

```
ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);
```

```
fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);
```

```
%ESFUERZO SECCION COMPUESTA
```

```
ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);
```

```
fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);
```

```
texee='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
```

```
texeee='.';
```

```
tex0e='CORTE 4';
```

```
texeeee='ESFUERZOS EN TRANSFERENCIA';
```

```
texeeeee='ft > fci';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
```

```
    texe='-- NO CUMPLE';
```

```
else
```

```
    texe='-- OK';
```

```
end
```

```
texff='fb > fti';
```

```
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO
```

```
    texf='-- NO CUMPLE';
```

```
else
```

```
    texf='-- OK';
```

```
end
```

```
texgg='ESFUERZOS EN SERVICIO';
```

```
texggg='ft > fcs';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
```

```
    texg='-- NO CUMPLE';
```

```
else
```

```
    texg='-- OK';
```

```
end
```

```
texhh='fb > fts';
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
```

```
    texh='-- NO CUMPLE';
```

```
else
```

```
    texh='-- OK';
```

```
end
```

```
%MENSAJE SECCION COMPUESTA
```

```
%MENSAJE 1ER PUNTO
```

```
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
```

```
    texhhh='-- NO CUMPLE';
```

```
else
```

```
    texhhh='-- OK';
```

```
end
```



```

pp=pos;
tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2),(pp));
Mwsdsx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);

fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);

%ESFUERZOS SECCION COMPUESTA
ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

texAA='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
texAAA='.';
tex0A='CORTE 5';
texAAAA='ESFUERZOS EN TRANSFERENCIA';
texAAAAA='ft > fci';
%MENSAJE 1ER PUNTO
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
    texA='-- NO CUMPLE';
else
    texA='-- OK';
end

```

```

texBB='fb > fti';
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO
    texB='-- NO CUMPLE';
else
    texB='-- OK';
end

texCC='ESFUERZOS EN SERVICIO';
texCCC='ft > fcs';
%MENSAJE 1ER PUNTO
if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
    texC='-- NO CUMPLE';
else
    texC='-- OK';
end
texDD='fb > fts';
%MENSAJE 1ER PUNTO
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
    texD='-- NO CUMPLE';
else
    texD='-- OK';
end
%MENSAJE SECCION COMPUESTA
%MENSAJE 1ER PUNTO
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
    texDDD='-- NO CUMPLE';
else
    texDDD='-- OK';
end

%MENSAJE 1ER PUNTO
if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
    texDDDD='-- NO CUMPLE';
else
    texDDDD='-- OK';
end

msgbox({texAA,texAAA,tex0A,texAAAA,texAAAAA,texA,texBB,texB,texCC,te
xCCC,texC,texDD,texD,'ESFUERZOS EN SECCION
COMPUESTA','fc>ftc',texDDD,'ft>tbc',texDDDD})
save('es6','texAA','texAAA','tex0A','texAAAA','texAAAAA','texA','tex
BB','texB','texCC','texCCC','texC','texDD','texD','texDDD','texDDDD'
)

%CORTE 7

n=cell(qq,1);%genera unas celdas para guardar datos del siguiente
for
for i=1:qq;
    if corte(i)==7;
        n{i}=numtoron(i);
    end
end
n=cell2mat(n);%numero de torones en fila
nuntotal=sum(n);%suma de los torones de acuerdo al corte
especificado
SAe=nuntotal*ato; %sumatoria de las areas toron= numero toral de
torones* area de toron

```

```

Ae=cell(qq,1);%Area del toron por posicion en el corte especificado
(1)
for i=1:qq;
    if corte(i)==7;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
Ae=cell2mat(Ae);%area de torn por seccion en guardado *la posicion
en Y
Aet=sum(Ae);%sumatoria del las areas de torones por seccion

rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;
e7=e;%:.....:
:.....:

for i=1:qq;
    if corte(i)==7;
        pos=X(i);
    end
end
pos;

pp=pos;
tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:);%ya esta!!!!!! === Md

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:);%ya esta!!!!!! === Mds

tf=ismember(LS(:,2),(pp));
Mwsdsx=LS(tf,:);%ya esta!!!!!! === Mdss

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:);%ya esta!!!!!! === Mdss

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia
ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts);%esfuerzo en la transferencia
en el tope

fbq=(-(Pi/A))*(1-((e*yb)/r2))-(Md/sbs);%esf en la transferencia en
la parte baja

```

```

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt/sts);

fbz=(-(Pe/A))*(1-((e*yb)/r2))-(Mt/sbs);

%ESFUERZOS SECCION COMPUESTA
ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1-((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

texEE='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
texEEE='.';
tex0E='CORTE FINAL';
texEEEE='ESFUERZOS EN TRANSFERENCIA';
texEEEEEE='ft > fci';
%MENSAJE 1ER PUNTO
if abs(ftq)>=abs(fci1)%PONER EN ABSOLUTO
    texE='-- NO CUMPLE';
else
    texE='-- OK';
end

texFF='fb > fti';
if abs(fbq)>=abs(fti)%PONER EN ABSOLUTO
    texF='-- NO CUMPLE';
else
    texF='-- OK';
end

texGG='ESFUERZOS EN SERVICIO';
texGGG='ft > fcs';
%MENSAJE 1ER PUNTO
if abs(ftz)>=abs(fc1)%PONER EN ABSOLUTO
    texG='-- NO CUMPLE';
else
    texG='-- OK';
end
texHH='fb > fts';
%MENSAJE 1ER PUNTO
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
    texH='-- NO CUMPLE';
else
    texH='-- OK';
end

%MENSAJE SECCION COMPUESTA
%MENSAJE 1ER PUNTO
if abs(ftm)>=abs(fc1)%PONER EN ABSOLUTO
    texHHH='-- NO CUMPLE';
else
    texHHH='-- OK';
end

%MENSAJE 1ER PUNTO
if abs(fbm)>=abs(ft)%PONER EN ABSOLUTO
    texHHHH='-- NO CUMPLE';

```

```

else
    texHHHH='-- OK';
end

msgbox({texEE,texEEE,tex0E,texEEEE,texEEEEEE,texE,texFF,texF,texGG,te
xGGG,texG,texHH,texH,'ESFUERZOS EN SECCION
COMPUESTA','fc>ftc',texHHH,'ft>tbc',texHHHH})

save('es7','texEE','texEEE','tex0E','texEEEE','texEEEEEE','texE','tex
FF','texF','texGG','texGGG','texG','texHH','texH','texHHH','texHHHH'
)

%GUARDAR LA EXCENTICIDAD EN CADA CORTE para usarse en el boton luego

save('exc','e1','e2','e3','e4','e5','e6','e7');

%TERMINA CONDICIONAL DE de NO Constante

%=====
=====

%ventana de texto CALCULAR
%Ahora condicional de que si ES CONSTANTE

else

    n=cell(qq,1); %genera unas celdas para guardar datos del
siguiente for
for i=1:qq;
    if corte(i)==1;
        n{i}=numtoron(i);
    end
end
n=cell2mat(n); %numero de torones en fila
nuntotal=sum(n); %suma de los torones de acuerdo al corte
especificado
SAe=nuntotal*ato; %sumatoria de las areas toron= numero total de
torones * area de toron

Ae=cell(qq,1); %Area del toron por posicion en el corte
especificado (1)
for i=1:qq;
    if corte(i)==1;
        Ae{i}=(numtoron(i).*ato)*Y(i);
    end
end
end

```



```

Ae=cell2mat(Ae);%area de torn por seccion en guardado *la posicion
en Y
Aet=sum(Ae);%sumatoria del las areas de torones por seccion

rec=1;%2 pulg de recubrimiento variar

yr=Aet/SAe;
e=yr;

e1=e;

for i=1:qq;
    if corte(i)==1;
        pos=X(i);
    end
end
pos;

pp=pos;

tf=ismember(LM(:,2),(pp));
Mwdx=LM(tf,:); %=== Md=====pp carga MUERTA DE
VIGA

tf=ismember(LF(:,2),(pp));
Mwsdx=LF(tf,:); %=== Mds=====SOBREIMPUESTA

tf=ismember(LS(:,2),(pp));
Mwsdsx=LS(tf,:); %=== Mdss===== pp LOSA

tf=ismember(LL(:,2),(pp));
Mlx=LL(tf,:); %=== Mdss=====carga VIVA

Ml=abs(Mlx(1));
Mdss=abs(Mwsdsx(1));
Mds=abs(Mwsdx(1));
Md=abs(Mwdx(1));

Mo=Mdss+Md;
Mt=Mds+Ml;

Mt1=Mds+Ml+Md

Pi=(ato*nuntotal)*fpi;
Pe=(ato*nuntotal)*fpe;

%Esfuerzos en Transferencia

ftq=(-(Pi/A))*(1-((e*yt)/r2))-(Md/sts); %esfuerzo en la
transferencia en el tope

fbq=(-(Pi/A))*(1+((e*yb)/r2))+(Md/sbs); %esf en la
transferencia en la parte baja

```

```

%Esfuerzos en Servicio

ftz=(-(Pe/A))*(1-((e*yt)/r2))-(Mt1/sts);

fbz=(-(Pe/A))*(1+((e*yb)/r2))+(Mt1/sbs);

%Esfuerzos en seccion compuesta

ftm=(nfg*Pi/A)*(1-((e*yt)/r2))+(Mo/sts)+(Mt/Stg);

fbm=(nfg*Pi/A)*(1+((e*yb)/r2))+(Mo/sbs)+(Mt/Sbc);

tex00='ESFUERZOS ACTUANTES - ESFUERZOS ADMISIBLES';
tex000='.';
tex01='CORTE 0';
tex0000='ESFUERZOS EN TRANSFERENCIA';
tex00000='ft > fci';

%MENSAJE 1ER PUNTO

if abs(ftq)>=abs(fti)%PONER EN ABSOLUTO
    tex1='-- NO CUMPLE';
else
    tex1='-- OK';
end

tex22='fb > fti';
if abs(fbq)>=abs(fci1)%PONER EN ABSOLUTO
    tex2='-- NO CUMPLE';
else
    tex2='-- OK';
end

tex33='ESFUERZOS EN SERVICIO';
tex333='ft > fcs';
%MENSAJE 1ER PUNTO
if abs(ftz)>=abs(fcl)%PONER EN ABSOLUTO
    tex3='-- NO CUMPLE';
else
    tex3='-- OK';
end
tex444='fb > fts';
%MENSAJE 1ER PUNTO
if abs(fbz)>=abs(ft)%PONER EN ABSOLUTO
    tex4='-- NO CUMPLE';
else
    tex4='-- OK';
end

%MENSAJE SECCION COMPUESTA 1ER PUNTO

```

```

%MENSAJE 1ER PUNTO COMP
if abs(ftm)>=abs(ft)%PONER EN ABSOLUTO
    tex4444='-- NO CUMPLE';
else
    tex4444='-- OK';
end

%MENSAJE 1ER PUNTO
if abs(fbm)>=abs(fc1)%PONER EN ABSOLUTOfc1
    tex44444='-- NO CUMPLE';
else
    tex44444='-- OK';
end

msgbox({tex00,tex000,tex01,tex0000,tex00000,tex1,tex22,tex2,tex33,te
x333,tex3,tex444,tex4,'ESFUERZOS EN SECCION
COMPUESTA','ftc>fc',tex4444,'fbc>ft',tex44444})
save('es1','tex00','tex000','tex01','tex0000','tex00000','tex1','tex
22','tex2','tex33','tex333','tex3','tex444','tex4','tex4444','tex444
44')
%TERMINA EL CORTE 1 AA*****+

%-----
-----

%Excel TODOS LOS CORTES SEAN IGUALES AL CORTE 1

tex55=tex00;tex99=tex00;texaa=tex00;texee=tex00;texAA=tex00;texEE=te
x00;

tex555=tex000;tex999=tex000;texaaa=tex000;texeee=tex000;texAAA=tex00
0;texEEEE=tex000;

tex05=tex01;tex09=tex01;tex0a=tex01;tex0e=tex01;tex0A=tex01;tex0E=te
x01;

tex5555=tex0000;tex9999=tex0000;texaaaa=tex0000;texeeee=tex0000;texA
AAA=tex0000;texEEEE=tex0000;

tex55555=tex00000;tex99999=tex00000;texaaaaa=tex00000;texeeeeee=tex00
000;texAAAAA=tex00000;texEEEEEE=tex00000;

tex5=tex1;tex9=tex1;texa=tex1;texe=tex1;texA=tex1;texE=tex1;

tex66=tex22;tex1010=tex22;texbb=tex22;texff=tex22;texBB=tex22;texFF=
tex22;

tex6=tex2;tex10=tex2;texb=tex2;texf=tex2;texB=tex2;texF=tex2;

tex77=tex33;tex1212=tex33;texcc=tex33;texgg=tex33;texCC=tex33;texGG=
tex33;

```

```
tex777=tex333;tex1212=tex333;texccc=tex333;texggg=tex333;texCCC=te  
x333;texGGG=tex333;
```

```
tex7=tex3;tex12=tex3;texc=tex3;texg=tex3;texC=tex3;texG=tex3;
```

```
tex88=tex444;tex1313=tex444;texdd=tex444;texhh=tex444;texDD=tex444;t  
exHH=tex444;
```

```
tex8=tex4;tex13=tex4;texd=tex4;texh=tex4;texD=tex4;texH=tex4;
```

```
tex888=tex4444;tex1333=tex4444;texddd=tex4444;texhhh=tex4444;texDDD=  
tex4444;texHHH=tex4444;
```

```
tex8888=tex44444;tex13333=tex44444;texdddd=tex44444;texhhhh=tex44444  
;texDDDD=tex44444;texHHHH=tex44444;
```

```
%DE CADA PUNTO SE HACE UN SAVE
```

```
save('es2','tex55','tex555','tex05','tex5555','tex55555','tex5','tex  
66','tex6','tex77','tex777','tex7','tex88','tex8','tex888','tex8888'  
)
```

```
save('es3','tex99','tex999','tex09','tex9999','tex99999','tex9','tex  
1010','tex10','tex1212','tex121212','tex12','tex1313','tex13','tex13  
33','tex13333')
```

```
save('es4','texaa','texaaa','tex0a','texaaaa','texaaaaa','texa','tex  
bb','texb','texcc','texccc','texc','texdd','texd','texddd','texdddd'  
)
```

```
save('es5','texee','texeee','tex0e','texeeee','texeeeeee','texe','tex  
ff','texf','texgg','texggg','texg','texhh','texh','texhhh','texhhhh'  
)
```

```
save('es6','texAA','texAAA','tex0A','texAAAA','texAAAAA','texA','tex  
BB','texB','texCC','texCCC','texC','texDD','texD','texDDD','texDDDD'  
)
```

```
save('es7','texEE','texEEE','tex0E','texEEEE','texEEEEEE','texE','tex  
FF','texF','texGG','texGGG','texG','texHH','texH','texHHH','texHHHH'  
)
```

```
%Grabar LA EXCENTRICIDAD EN EL CASO CONSTANTE
```

```
e2=e1;e3=e1;e4=e1;e5=e1;e6=e1;e7=e1;
```

```
save('exc','e1','e2','e3','e4','e5','e6','e7');  
end
```

```
function edit48_Callback(hObject, eventdata, handles)
```

```
function edit48_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit49_Callback(hObject, eventdata, handles)
```

```
function edit49_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function CAMIONP_Callback(hObject, eventdata, handles)
```

```
tex1='-- CAMIÓN: Es la carga Resultante del tipo de camion (3 EJES)  
';  
msgbox({tex1});
```

```
function cam1_Callback(hObject, eventdata, handles)
```

```
function cam1_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function cam2_Callback(hObject, eventdata, handles)
```

```
function cam2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function cam3_Callback(hObject, eventdata, handles)
```

```

function cam3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function FCLOSAB_Callback(hObject, eventdata, handles)

function HLOSAB_Callback(hObject, eventdata, handles)

function BASELOSAB_Callback(hObject, eventdata, handles)

function LUZB_Callback(hObject, eventdata, handles)

tex1='-- L: longitud de la viga simplemente apoyada ';
msgbox({tex1});

function FCB_Callback(hObject, eventdata, handles)
tex1='-- fci: Resistencia a la compresion del del concreto ';
msgbox({tex1});

function FCIB_Callback(hObject, eventdata, handles)
tex1='-- fci: Resistencia a la compresion del concreto en
transferencia ';
msgbox({tex1});

function FPUB_Callback(hObject, eventdata, handles)
tex1='-- fpu: Resistencia de presfuerzo inicial ';
msgbox({tex1});

function YB_Callback(hObject, eventdata, handles)
tex1='-- ? : es la relación de pretensado residual, obtenida a
partir de las perdidas de presfuerzos ';
msgbox({tex1});

function WSDq_Callback(hObject, eventdata, handles)
tex1='-- W SD: Es la carga Muerta sobreimpuesta que actua sobre la
viga ';
msgbox({tex1});

function WLq_Callback(hObject, eventdata, handles)

```

```
tex1='-- W L: Es la carga viva distribuida que actua sobre la viga
';
msgbox({tex1});
```

```
function WDq_Callback(hObject, eventdata, handles)
tex1='-- W D: Es la carga Muerta debida al peso propio de la viga
';
msgbox({tex1});
```

```
function WSDSq_Callback(hObject, eventdata, handles)

tex1='-- W sds: es la carga muerta superimpuesta compuesta
adicional en transferencia. ';
msgbox({tex1});
```

```
function attt_Callback(hObject, eventdata, handles)
```

```
function attt_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function DIAB_Callback(hObject, eventdata, handles)
tex1='-- At: Area del toron ';
msgbox({tex1});
```

```
function edit54_Callback(hObject, eventdata, handles)
```

```
function edit54_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function constante_Callback(hObject, eventdata, handles)
```

```

function SSECCIOON_Callback(hObject, eventdata, handles)

function SREQUERIDA_Callback(hObject, eventdata, handles)

function uibuttongroup1_SelectionChangedFcn(hObject, eventdata,
handles)

%seleccionar SI es constante o no constante
%CONSTANTE 1 Y EL OTRO EN 0

if hObject==handles.constante
    cc1=1;
else hObject==handles.noconstante;
    cc1=0;
end
save ('constantesino','cc1');
%se grabo el valor si es o no constante, si es constante la
excentricidad el
%valor de cc1 es igual a 1, caso contrario el valor de cc1=0

function At_Callback(hObject, eventdata, handles)

function At1_Callback(hObject, eventdata, handles)

function At1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton24_Callback(hObject, eventdata, handles)

%EN ESTA PARTE SE EXPORTA A EXCEL-----

load('ubicaciondetorones','Y1','X1','Y2','X2','Y3','X3','Y4','X4','Y
5','X5','Y6','X6','Y7','X7')
load('tablatorones','TAB');

```



```

load('es1','tex00','tex000','tex01','tex0000','tex00000','tex1','tex
22','tex2','tex33','tex333','tex3','tex444','tex4','tex4444','tex444
44')
load('es2','tex55','tex555','tex05','tex5555','tex55555','tex5','tex
66','tex6','tex77','tex777','tex7','tex88','tex8','tex888','tex8888'
)
load('es3','tex99','tex999','tex09','tex9999','tex99999','tex9','tex
1010','tex10','tex1212','tex121212','tex12','tex1313','tex13','tex13
33','tex13333')
load('es4','texaa','texaaa','tex0a','texaaaa','texaaaaa','texa','tex
bb','texb','texcc','texccc','texc','texdd','texd','texddd','texddd'
)
load('es5','texee','texeee','tex0e','texeeee','texeeeeee','texe','tex
ff','texf','texgg','texggg','texg','texhh','texh','texhhh','texhhhh'
)
load('es6','texAA','texAAA','tex0A','texAAAA','texAAAAA','texA','tex
BB','texB','texCC','texCCC','texC','texDD','texD','texDDD','texDDDD'
)
load('es7','texEE','texEEE','tex0E','texEEEE','texEEEEEE','texE','tex
FF','texF','texGG','texGGG','texG','texHH','texH','texHHH','texHHHH'
)

TEX00='ft > fcs';
TEX01='fb > fts';

%AQUI GENERO UNAS MATRICES EN FILA PARA UBICARLAS EN EXCEL
filename='resultados.xlsx';
a={tex00};
b={'_', 'CORTE 1', '_', '_', 'CORTE 2', '_', '_', 'CORTE 3', '_', '_', 'CORTE
4', '_', '_', 'CORTE 5', '_', '_', 'CORTE 6', '_', '_', 'CORTE 7', '_', '_'};
c={tex00000, tex1, '_', tex00000, tex5, '_', tex00000, tex9, '_', tex00000, te
xa, '_', tex00000, texe, '_', tex00000, texA, '_', tex00000, texE, '_'};
d={tex22, tex2, '_', tex22, tex6, '_', tex22, tex10, '_', tex22, texb, '_', tex2
2, texf, '_', tex22, texB, '_', tex22, texF, '_'};
e={tex333, tex3, '_', tex777, tex7, '_', tex333, tex12, '_', tex333, texc, '_',
tex333, texg, '_', tex333, texC, '_', tex333, texG, '_'};
f={tex444, tex4, '_', tex444, tex8, '_', tex444, tex13, '_', tex444, texd, '_',
tex444, texh, '_', tex444, texD, '_', tex444, texH, '_'};

g={TEX00, tex4444, '_', TEX00, tex888, '_', TEX00, tex1333, '_', TEX00, texddd
, '_', TEX00, texhhh, '_', TEX00, texDDD, '_', TEX00, texHHH, '_'};
h={TEX00, tex44444, '_', TEX00, tex8888, '_', TEX00, tex13333, '_', TEX00, tex
dddd, '_', TEX00, texhhhh, '_', TEX00, texDDDD, '_', TEX00, texHHHH, '_'};

% T1=table(X1,Y1);
% T2=table(X2,Y2);
% T3=table(X3,Y3);
% T4=table(X4,Y4);
% T5=table(X5,Y5);
% T6=table(X6,Y6);
% T7=table(X7,Y7);

%ESCOJO LA HOJA DE EXCEL
%ESCOJO LOS RANGOS DE DONDE COMIENZA A ESCRIBIR

sheet = 1;
xlRange1 = 'A1'; %TITULO
xlRange2 = 'A3'; %corte

```

```

xlRange3 = 'A6';           %1er ok
xlRange4 = 'A8';           %2do ok
xlRange5 = 'A11';          %3er ok
xlRange6 = 'A13';          %4TO ok
xlRange7 = 'A16';          %COMP
xlRange8 = 'A18';          %COMP

```

```
%MODIFICAR EL FORMATO YA QUE MATLAB MANDA SOLO EL TEXTO
```

```

xlswrite(filename,a,sheet,xlRange1);
xlswrite(filename,b,sheet,xlRange2);
xlswrite(filename,c,sheet,xlRange3);
xlswrite(filename,d,sheet,xlRange4);
xlswrite(filename,e,sheet,xlRange5);
xlswrite(filename,f,sheet,xlRange6);
xlswrite(filename,g,sheet,xlRange7);
xlswrite(filename,h,sheet,xlRange8);

```

```

%CON ESTE COMANDO SE ABRE EL ARCHIVO DE EXCEL
winopen( 'resultados.xlsx' );

```

```

function pushbutton25_Callback(hObject, eventdata, handles)
%ESTE BOTON ES PARA QUE LA GUI DE PLOTEAR SE ABRA AL PRESIONAR EL
BOTON
PLOT

```

```
function pushbutton26_Callback(hObject, eventdata, handles)
```

```
function pushbutton27_Callback(hObject, eventdata, handles)
```

```

tex1='-- # TORON: En esta columna se ubican los torones por fila en
la viga ';
msgbox({tex1});

```

```
function pushbutton28_Callback(hObject, eventdata, handles)
```

```

tex1='-- Y: posicion de los torones con respecto a la altura de la
viga ';
msgbox({tex1});

```

```
function pushbutton29_Callback(hObject, eventdata, handles)
```

```

tex1='-- SECCIÓN: El número de corte al que corresponde la fila de
torones ';
msgbox({tex1});

function pushbutton30_Callback(hObject, eventdata, handles)

tex1='-- X: posicion en la que se analiza la sección con respecto a
la longitud de la viga ';
msgbox({tex1});

function uibuttongroup1_CreateFcn(hObject, eventdata, handles)

function edit56_Callback(hObject, eventdata, handles)

function edit56_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton31_Callback(hObject, eventdata, handles)

tex1='-- fc losa: resistencia a la compresion de la losa que se va
a usar ';
msgbox({tex1});

function edit57_Callback(hObject, eventdata, handles)

function edit57_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton32_Callback(hObject, eventdata, handles)

```

```

tex1='-- b losa: ancho efectivo de la losa sobre la viga
presforzada  ';
msgbox({tex1});

function edit58_Callback(hObject, eventdata, handles)

function edit58_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton33_Callback(hObject, eventdata, handles)

tex1='-- h losa: Espesor de la losa sobre la viga presforzada  ';
msgbox({tex1});

function wcs_Callback(hObject, eventdata, handles)

function wcs_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton34_Callback(hObject, eventdata, handles)

tex1='-- W CS: Es la carga del peso propio de la losa.  ';
msgbox({tex1});

function pushbutton36_Callback(hObject, eventdata, handles)
%ESTO ES PARA MANDAR la gui de los GRAFICOS DE MOMENTOS AL PRESIONAR
EL BOTON
MOMENTOGRAF

function blosa1_Callback(hObject, eventdata, handles)

```

```
function blosal_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function hlosal_Callback(hObject, eventdata, handles)
```

```
function hlosal_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function EXCENTRICIDAD_Callback(hObject, eventdata, handles)
%ESTE BOTON ES PARA GENERAR EL GUI DE EXCENTRICIDAD
EXCENTRICIDAD
```

## Codigo del Gui de PLOT.

```
function varargout = PLOT(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @PLOT_OpeningFcn, ...
                  'gui_OutputFcn',  @PLOT_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

function PLOT_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = PLOT_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function axes1_CreateFcn(hObject, eventdata, handles)

function PLOT_Callback(hObject, eventdata, handles)

%Grafico se vea desde antes de 0 los torones
clear
x=-2;
load('secciondatos','H','e','A','I','yt','yb','sts','sbs','r2')
load('ubicaciondetorones','Y1','X1','Y2','X2','Y3','X3','Y4','X4','Y
5','X5','Y6','X6','Y7','X7')
load('constantesino','cc1');
load('longitud','luz');

%Grafico sea mayor a lo ploteado
y=Y1+3;
x1=0;
y1=0;

%si es NO CONSTANTE SE DIBUJAN TODOS LOS 7 PUNTOS solo si es que SE
ESCRIBIERON
if cc1==0

plot(x,y,x1,y1,x1,H,X1,Y1,'rx',X2,Y2,'rx',X3,Y3,'rx',X4,Y4,'rx',X5,Y
5,'rx',X6,Y6,'rx',X7,Y7,'rx');

    %SI ES CONSTANTE SE REPITE EL Y1 EN LOS OTROS 6 PUNTOS
else
    X2=X1-X1+luz;Y2=Y1;Y3=Y1;Y4=Y1;Y5=Y1;Y6=Y1;Y7=Y1;
    X3=X2/2;
    X4=X1-X1;
    X5=X2/3;
    X6=X2*3./4;
    X7=X2*0.9;

plot(x,y,x1,y1,x1,H,X1,Y1,'rx',X2,Y2,'rx',X3,Y3,'rx',X4,Y4,'rx',X5,Y
5,'rx',X6,Y6,'rx',X7,Y7,'rx');
end

```

## Codigo del GUI MOMENTOS.

```
function varargout = MOMENTOGRAF(varargin)

% Edit the above text to modify the response to help MOMENTOGRAF

% Last Modified by GUIDE v2.5 19-Sep-2017 14:44:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @MOMENTOGRAF_OpeningFcn, ...
                  'gui_OutputFcn',   @MOMENTOGRAF_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

function MOMENTOGRAF_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = MOMENTOGRAF_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
%CARGAR LOS LAS MATRICES DE TODOS LOS MOMENTOS PARAGRAFICARLAS

load('momentos', 'Mwd', 'Mwsd', 'Mwsds', 'Mwl', 'LM', 'LF', 'LS', 'LL');

Mwdmax=(max(abs(Mwd)));%carga muerta
Mwsdmax=max(abs(Mwsd));%sobre impuesta
Mwsdsmax=max(abs(Mwsds));%de losa
Mwlmax=max(abs(Mwl));%viva

% format short
% Mwdmax;
% Mwsdmax;
% Mwsdsmax;
% Mwlmax;

set(handles.a, 'String', Mwdmax);
set(handles.aa, 'String', Mwsdmax);
set(handles.aaa, 'String', Mwsdsmax);
set(handles.aaaa, 'String', Mwlmax);

%GRAFICAR MOMENTOS

axes(handles.axes1);
x1 = LM(:,2);
x2 = LM(:,1);
plot(x1,x2,'red','LineWidth', 2.5);

axes(handles.axes2);
x1 = LF(:,2);
x2 = LF(:,1);
plot(x1,x2,'LineWidth', 2.5);

axes(handles.axes3);
x1 = LS(:,2);
x2 = LS(:,1);
plot(x1,x2,'green','LineWidth', 2.5);

axes(handles.axes4);
x1 = LL(:,2);
x2 = LL(:,1);

```



```

plot(x1,x2,'yellow','LineWidth', 2.5);

function pushbutton2_Callback(hObject, eventdata, handles)
function pushbutton3_Callback(hObject, eventdata, handles)

function pushbutton4_Callback(hObject, eventdata, handles)

function pushbutton5_Callback(hObject, eventdata, handles)

function a_Callback(hObject, eventdata, handles)

function a_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function aa_Callback(hObject, eventdata, handles)

function aa_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function aaa_Callback(hObject, eventdata, handles)

function aaa_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function aaaa_Callback(hObject, eventdata, handles)

function aaaa_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

## Codigo del GUI EXCENTRICIDAD.

```

function varargout = EXCENTRICIDAD(varargin)
% EXCENTRICIDAD MATLAB code for EXCENTRICIDAD.fig
%     EXCENTRICIDAD, by itself, creates a new EXCENTRICIDAD or
raises the existing
%     singleton*.
%
%     H = EXCENTRICIDAD returns the handle to a new EXCENTRICIDAD
or the handle to
%     the existing singleton*.
%
%     EXCENTRICIDAD('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in EXCENTRICIDAD.M with the given
input arguments.
%
%     EXCENTRICIDAD('Property','Value',...) creates a new
EXCENTRICIDAD or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before EXCENTRICIDAD_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to EXCENTRICIDAD_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help EXCENTRICIDAD

% Last Modified by GUIDE v2.5 19-Sep-2017 15:49:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...

```

```

        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @EXCENTRICIDAD_OpeningFcn, ...
        'gui_OutputFcn', @EXCENTRICIDAD_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function EXCENTRICIDAD_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = EXCENTRICIDAD_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function e1_Callback(hObject, eventdata, handles)

function e1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton1_Callback(hObject, eventdata, handles)

%PARA VER LA EXCENTRICIDAD
%EN CADA PUNTO

load('exc','e1','e2','e3','e4','e5','e6','e7');

e1;
e2;
e3;
e4;
e5;

```

```

e6;
e7;

set(handles.e1,'String',e1);
set(handles.e2,'String',e2);
set(handles.e3,'String',e3);
set(handles.e4,'String',e4);
set(handles.e5,'String',e5);
set(handles.e6,'String',e6);
set(handles.e7,'String',e7);
function e2_Callback(hObject, eventdata, handles)

function e2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function e3_Callback(hObject, eventdata, handles)

function e3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function e4_Callback(hObject, eventdata, handles)

function e4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function e5_Callback(hObject, eventdata, handles)

function e5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function e6_Callback(hObject, eventdata, handles)

```

```
function e6_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function e7_Callback(hObject, eventdata, handles)

function e7_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end.
```

## **DECLARACIÓN Y AUTORIZACIÓN**

Yo, **Alvarado Campi Gino Osvaldo**, con CC.: # **1205867607** autor/a del trabajo de titulación: **Desarrollo de código en Matlab para el diseño de vigas presforzadas, considerando diferentes códigos de diseño**, previo a la obtención del título de **ingeniero civil** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **19 de septiembre de 2017**

---

Nombre: **Alvarado Campi, Gino Osvaldo**

C.C: **1205867607**



<b>REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA</b>			
<b>FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN</b>			
<b>TEMA Y SUBTEMA:</b>	<b>Desarrollo de código en Matlab para el diseño de vigas presforzadas, considerando diferentes códigos de diseño.</b>		
<b>AUTOR(ES)</b>	<b>ALVARADO CAMPI, GINO OSVALDO</b>		
<b>REVISOR(ES)/TUTOR(ES)</b>	<b>PONCE VÁSQUEZ, GUILLERMO ALFONSO</b>		
<b>INSTITUCIÓN:</b>	UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL		
<b>FACULTAD:</b>	FACULTAD DE INGENIERÍA		
<b>CARRERA:</b>	<b>INGENIERÍA CIVIL</b>		
<b>TÍTULO OBTENIDO:</b>	<b>INGENIERO CIVIL</b>		
<b>FECHA DE PUBLICACIÓN:</b>	<b>19 DE SEPTIEMBRE DEL 2017</b>	<b>No. DE PÁGINAS:</b>	<b>96</b>
<b>ÁREAS TEMÁTICAS:</b>	ESTRUCTURAS		
<b>PALABRAS CLAVES/KEYWORDS:</b>	<b>Matlab, vigas, software, hormigón, presforzadas, hormigón.</b>		
<b>RESUMEN/ABSTRACT:</b>			
<p>En este proyecto de sustentación se procederá a realizar un Software en el programa Matlab el cual será muy útil al momento de realizar una revisión del diseño de vigas presforzadas simplemente apoyadas. Este software está programado para realizar revisiones de diseño cumpliendo todas las normas vigentes requeridas. De esta manera aseguramos el buen diseño de una viga presforzada y estaremos resguardando la seguridad de las personas que vayan a utilizar a futuro dichas estructuras lo cual es lo más importante al momento de construir una estructura. Este programa servirá para Ingenieros o estudiantes de ingeniería civil con conocimientos básicos de diseño, este software puede ayudar a futuros estudiantes de ingeniería a practicar al momento de diseñar vigas de hormigón presforzado. Esto es posible gracias a que esta herramienta trata de tener una interfaz de diseño cómoda y fácil para el usuario al momento de ingresar los datos garantizando un diseño seguro y funcional.</p>			
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-987330606	<b>E-mail:</b> ginoalvarado@outlook.com	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::</b>	<b>Nombre: ING. CLARA GLAS CEVALLOS</b>		
	<b>Teléfono: +593-4-206950</b>		
	<b>E-mail: claglas@hotmail.com</b>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			