



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

---

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA**

**INSTITUTO DE INGENIERÍA**

**ADAPTACIÓN Y MODIFICACIÓN DE UN PROGRAMA  
DE ANÁLISIS DE PUENTES ATIRANTADOS**

**T E S I S**

QUE PARA OPTAR POR EL GRADO DE:

**MAESTRO EN INGENIERÍA**

INGENIERÍA CIVIL – ESTRUCTURAS  
P R E S E N T A :

**RAÚL SÁNCHEZ GARCÍA**

TUTOR:  
**DR. ROBERTO GÓMEZ MARTÍNEZ**

Enero del 2010



**INSTITUTO  
DE INGENIERÍA  
UNAM**

**JURADO ASIGNADO:**

Presidente: DR. JOSÉ ALBERTO ESCOBAR SÁNCHEZ  
Secretario: M. I. CARLOS JAVIER MENDOZA ESCOBEDO  
Vocal: DR. ROBERTO GÓMEZ MARTÍNEZ  
1<sup>er</sup>. Suplente: M.I. CARLOS ADÁN MONTOYA BELTRÁN  
2<sup>do</sup>. Suplente: M. I. GERARDO RODRÍGUEZ GUTIÉRREZ

Lugar donde se realizó la tesis:

INSTITUTO DE INGENIERÍA, UNAM.

**TUTOR DE TESIS:**

---

DR. ROBERTO GÓMEZ MARTÍNEZ

*Dedicado a mi esposa y a mi hija.*

## **Agradecimientos.**

A Isela y a Jazel por su amor y apoyo incondicional.

Al Dr. Roberto Gómez por la confianza y aliento en el desarrollo de esta tesis.

A los sinodales por sus valiosos comentarios y sugerencias hechas a este trabajo.

A todos los amigos del Instituto de Ingeniería por su amistad y apoyo.

Al Instituto de Ingeniería, UNAM, por el apoyo económico y espacio para la realización de este trabajo.

---

## CONTENIDO

<b>CAPÍTULO 1: INTRODUCCIÓN</b>	<b>1</b>
1.1 Antecedentes	2
1.2 Objetivo	3
<b>CAPÍTULO 2: MODELO DE ANÁLISIS ESTRUCTURAL</b>	<b>4</b>
2.1 Introducción	4
2.2 Idealización de la estructura	4
2.2.1 Elemento tipo barra	5
2.2.2 Elemento tipo tirante	6
2.2.3 Presfuerzo.	7
2.3 Análisis secuencial por etapas	8
2.3.1 Procedimiento de análisis no lineal en el tiempo	8
2.3.2 Ecuaciones fundamentales	11
2.3.3 Inclusión de las no linealidades geométricas	11
2.4 Comportamiento de los materiales	11
2.4.1 Concreto	11
2.4.2 Acero de refuerzo	18
2.4.3 Acero de presfuerzo	18
2.4.4 Acero de tirantes	18
<b>CAPÍTULO 3: PROGRAMA DE CÓMPUTO</b>	<b>21</b>
3.1 Estrategia de desarrollo	21
3.2 Herramientas de desarrollo	21
3.2.1 FORTRAN	21
3.2.2 Visual Basic 6.0	21
3.2.3 Api de Windows	22
3.2.4 Controles ActiveX	22
3.2.5 Activex Data Objects (ADO)	23
3.3 Modelo conceptual	23
3.4 Módulo de análisis	24
3.5 Módulo de presentación	26
3.5.1 Editor de comandos	26
3.5.2 Módulo de Análisis de Sintaxis ( <i>parser</i> )	27
3.5.3 Graficador 2D	27
3.5.4 Conexión entre procesos	28
3.5.5 Manejo de base de datos	30

---

<b>CAPÍTULO 4: EJEMPLO DE APLICACIÓN</b>	32
4.1 Descripción del modelo	32
4.2 Análisis de la secuencia constructiva en cantiliver	40
4.3 Uso del programa	41
4.4 Discusión de resultados	54
<b>CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES</b>	58
<b>REFERENCIAS</b>	61
<b>APÉNDICE</b>	64

## *Resumen.*

Se presenta la modificación y adaptación de un programa que evalúa la respuesta no lineal de puentes atirantados de concreto reforzado a través de los rangos: elástico, agrietamiento, inelástico y último. El programa es una herramienta eficaz para analizar cómo cambia el estado de una estructura y su configuración geométrica sujetas a cargas que dependen del tiempo durante la operación de construcción secuencial por etapas; es decir, permite analizar la secuencia completa de la construcción de un puente en un solo funcionamiento en vez de hacer varios modelos diferentes, uno por cada etapa. Esta capacidad es importante puesto que los esfuerzos experimentados por la estructura durante el proceso de construcción pueden ser los más importantes que experimentará a lo largo de su vida. Se incorporó al programa una interfaz gráfica capaz de manipular los datos de entrada, ejecutar el análisis, de visualizar la configuración geométrica de cada etapa de análisis así como los resultados del análisis. Se presentan también un ejemplo de aplicación de un puente atirantado a manera de ejemplificar el uso del programa.

## Capítulo 1

# INTRODUCCIÓN

En las últimas décadas el número de puentes atirantados se ha incrementado considerablemente alrededor del mundo. En la actualidad estos puentes representan una alternativa que permite cubrir grandes claros y grandes alturas; además, este tipo de estructuración proporciona beneficios económicos, cualidades estéticas y versatilidad arquitectónica.

Sin embargo, para lograr la eficiencia que proporciona un puente atirantado y garantizar su funcionalidad es necesario un correcto diseño y una adecuada construcción del mismo, factores que están ligados a tratar de conocer su comportamiento estructural en sus diferentes fases constructivas. Este aspecto se torna un tanto complejo debido a que intervienen multitud de factores entre los que cabe destacar la variación de las propiedades de los materiales con el tiempo, las características geométricas, un bajo amortiguamiento y la gran flexibilidad de este tipo de estructuras.

La técnica de construcción segmental o por etapas, para puentes atirantados, resulta económicamente factible pero por otro lado puede representar para los ingenieros un problema substancial de análisis. Por ejemplo, la construcción por etapas involucra el uso de concreto colado en sitio o concreto precolado de diferentes edades y calidades, tendones de presfuerzo instalados en diferentes etapas de construcción, instalación y tensado de tirantes permanentes y auxiliares y cambios en la forma estructural debidos a cargas permanentes y a cargas que dependen del tiempo. Además, los puentes atirantados exhiben una respuesta estructural altamente no lineal, principalmente por la no linealidad geométrica de los tirantes, por la combinación de momento y carga axial en el tablero y pilones y por el comportamiento no lineal de los materiales. Entre los efectos de los materiales que dependen del tiempo son de importancia el flujo plástico, la contracción y el envejecimiento para el concreto y la relajación de los aceros de presfuerzo y de los tirantes.

De lo anterior se deduce la importancia de contar con herramientas que permitan analizar cómo cambia el estado de una estructura y su configuración geométrica sujeta a dichos efectos a corto y a largo plazo; a corto plazo, ya que las redistribuciones de esfuerzos que ocurren en las etapas intermedias de construcción pueden ser las más importantes que se encontrarán a lo largo de la vida útil del puente, y a largo plazo para conocer la evolución de la estructura después de terminada su construcción; así como el efecto de las operaciones de mantenimiento, tales como el reemplazo de un cable corroído.

Para resolver el problema anterior, y ya que las ecuaciones de equilibrio pueden ser no lineales, se requiere de una solución por iteraciones, lo que habla de la importancia del uso de programas de computadora en estos tipos de análisis. El uso de programas de computadora permite llevar a cabo el análisis para el diseño de estructuras de manera rápida, puede permitir el análisis de nuevas técnicas constructivas, la utilización de nuevos materiales o la combinación de ambos aspectos.

## 1.1 ANTECEDENTES

Desde los años 70 investigaciones sobre el análisis no lineal de estructuras de concreto presforzado se han desarrollado en la universidad de California, Berkeley, bajo la dirección del profesor A. C. Scordelis. Varios programas han emergido de esas investigaciones; a continuación se mencionan los de interés para este trabajo.

SPCFRAME [12]. Este programa, que se basa en dos programas antecesores: PCFRAME desarrollado por Kang [11] y SFRAME desarrollado por Ketchum [14], permite realizar análisis no lineal de marcos planos de concreto presforzado, incluyendo efectos que dependen del tiempo debidos a la historia de carga, temperatura, flujo plástico, contracción, edad del concreto y relajación del presfuerzo. El programa se basa en el método del elemento finito, es capaz de predecir la respuesta para dicho tipo de estructuras a través de varias etapas de construcción, cargas de servicio, así como, en los rangos elástico, agrietamiento, inelástico y último. Cambios en las condiciones de apoyos, cargas, instalación y remoción de elementos barra y de tendones de presfuerzo también pueden ser incorporadas como operaciones de construcción en el análisis por etapas.

NASCAB [1]. Este programa está escrito sobre las capacidades del SPCFRAME, antes mencionado. Al igual que su antecesor, permite realizar análisis no lineal de marcos planos de concreto presforzado a través de varias etapas constructivas, incluyendo efectos que dependen del tiempo. Pero además incluye el manejo de elementos tipo tirante (cables) los cuales pueden ser instalados, tensados, re-tensados y removidos en cualquier etapa de análisis. Modelos constitutivos no lineales para los elementos tipo tirante son incluidos, al igual que para el concreto, acero de refuerzo y acero de presfuerzo; se consideran también modelos refinados para el flujo plástico del concreto; y efectos no lineales geométricos pueden incluirse en todos los elementos estructurales.

El programa NASCAB permite hacer análisis no lineal de puentes atirantados. Está programado en Fortran 77 y es de tipo consola, es decir, utiliza una línea de comando para pedir datos al usuario, también se le conoce como interfaz de solo texto. En este programa se proporcionan los datos de entrada por medio de un archivo de texto para generar un único archivo de salida o de resultados. El archivo de entrada hace uso de una serie de comandos para definir las características y las etapas constructivas del puente a analizar. En esta situación, los archivos, el de entrada y el de salida, tienen que ser manejados por un editor de texto externo. Para presentar los resultados por medio de un gráfico, se debe buscar los datos en el archivo de salida y exportarlos a un programa externo que permita la graficación. Esto representa una desventaja debido a la pérdida de agilidad en el manejo de los resultados; además, dependiendo del tamaño de la estructura, el archivo de salida podría ser de gran tamaño y no cualquier editor de texto lo podría manejar con agilidad. Otra desventaja es que la geometría de la estructura no se puede visualizar a menos que se importen los datos en algún programa externo con motor gráfico.

## **1.2 OBJETIVO**

El objetivo central de esta tesis es la creación de un programa (UNASCAB) en ambiente Windows que haga uso de las subrutinas de cálculo del programa NASCAB y desde el cual se pueda generar y ejecutar el archivo de entrada, ver el proceso de cálculo en tiempo real y desplegar el archivo de salida; pero además, que pueda visualizar la configuración geométrica de la estructura para cualquier etapa de análisis y de manera directa acceder a sus resultados por medio del uso de la tecnología de base de datos y desplegarlos en forma tabular o por medio de una gráfica, de tal manera que se pueda agilizar el análisis de dichas estructuras. Se busca que este programa sea una herramienta de apoyo en la práctica profesional del análisis y diseño de puentes atirantados.

## Capítulo 2

# MODELO DE ANÁLISIS ESTRUCTURAL

## 2.1 INTRODUCCIÓN

Las dificultades en el análisis de estructuras que incorporan concreto son, entre otros factores, el modelado del comportamiento conjunto de dos o más materiales con propiedades diferentes (como el concreto y el acero de refuerzo), la historia de cargas, la no linealidad de la respuesta debida a la fisuración, la influencia de los efectos diferidos provocados por el flujo plástico y la contracción del concreto y la relajación de los aceros activos. En los casos de análisis por etapas constructivas habrá que añadir los cambios de las características resistentes de la sección, la influencia del proceso constructivo y operaciones de mantenimiento.

Una formulación precisa acerca del comportamiento de estructuras de concreto a lo largo del tiempo, como puede ser un puente atirantado de concreto presforzado, es esencial para predecir los esfuerzos y deflexiones de la estructura en cualquier momento de su construcción y a lo largo de su vida útil. Es por esto que surge la necesidad de utilizar modelos de análisis en los que se contemplen de una manera racional los principales fenómenos mencionados anteriormente.

En este capítulo se hace una descripción breve del modelado de puentes de concreto atirantados y de los modelos matemáticos de los diferentes efectos antes mencionados y que están incorporados en el programa objeto de esta tesis, el cual se basó en las capacidades del programa NASCAB [1].

## 2.2 IDEALIZACIÓN DE LA ESTRUCTURA

El programa permite el análisis plano de estructuras por medio del ensamble de elementos tipo barra, rectos unidimensionales, y elementos tipo tirantes; en un plano global  $X$ - $Y$  interconectados por puntos denominados nodos, Fig. 2.01. Los nodos en los extremos de cada barra tienen tres grados de libertad, desplazamiento en  $X$  y  $Y$  y rotación en  $Z$ , mientras que los elementos de tipo tirante o cable tienen dos grados de libertad, desplazamiento en  $X$  y  $Y$ .

La configuración de la estructura, las cargas aplicadas y las condiciones de frontera pueden ser cambiadas en cualquier etapa de solución, lo que permite realizar análisis por etapas constructivas. Los cambios en la configuración de la estructura consisten en la instalación o eliminación de elementos tipos barra, instalación de carros de colado, tensado y retensado de los tendones de presfuerzo y de los elementos tipo cable.

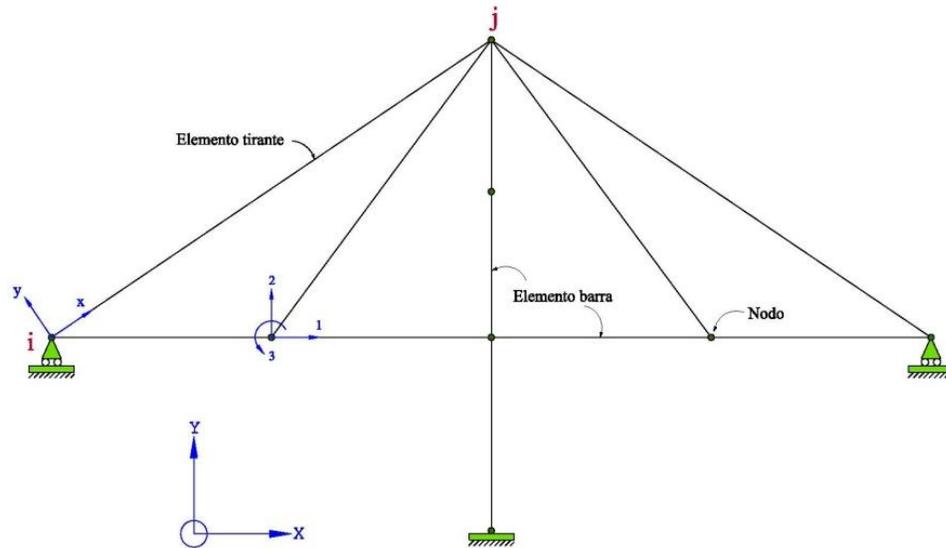
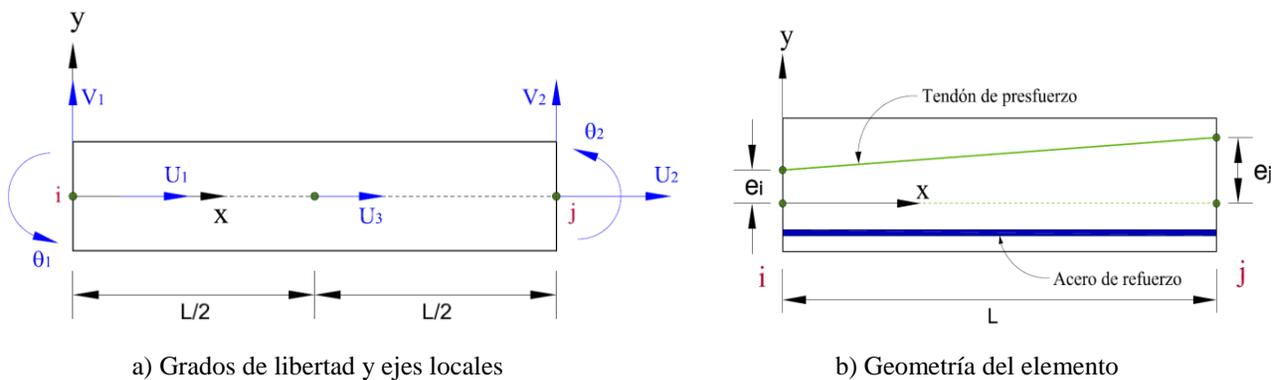


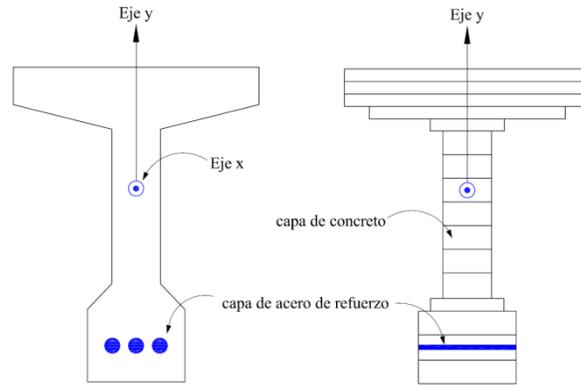
Figura 2.01.- Estructura y coordenadas.

### 2.2.1 Elemento tipo barra

El modelo de barra propuesto por Kang [12] es el usado para modelar los segmentos del tablero y los segmentos de las pilas y pilones. Este modelo se muestra en el Fig. 2.02. La geometría del elemento queda definida con respecto a un sistema local de coordenadas donde el eje *x* está definido por los nodos extremos *i* y *j* del elemento, el eje *y* es normal al eje *x* y coincide con el eje de simetría de la sección transversal del elemento. El elemento tipo barra tiene siete grados de libertad como se muestra en la Fig. 2.02 (a), denominados  $U_1, V_1, \theta_1, U_2, V_2, \theta_2$  y un grado de libertad axial interno,  $U_3$ , a la mitad de la longitud del elemento. Éste último es necesario para el correcto modelado de la rigidez a flexión a fin de tener en cuenta el desplazamiento del eje neutro, debido al agrietamiento y otras no linealidades del material en una viga de concreto.

La sección transversal de cada elemento tipo barra es dividida en un número discreto de capas de concreto y de capas de acero de refuerzo, Fig. 2.02 (c). La geometría de cada capa es definida por medio de su área y su posición con respecto al eje local *x*. Cada capa de concreto y de acero de refuerzo se encuentra en un estado de esfuerzo dado definido por relaciones esfuerzo-deformación no lineal. Se pueden incorporar diversos tipos de concreto u otros materiales para una misma sección transversal.





c) Sección transversal discretizada

Figura 2.02.- Elemento tipo barra.

### 2.2.2 Elemento tipo tirante

La rigidez axial de un cable depende de dos factores: la flecha del cable y la deformación axial del cable. La flecha tiene un efecto de ablandamiento en la rigidez del cable, lo que resulta en una relación desplazamiento-fuerza axial no lineal. Así, cuando existen valores relativamente grandes de flecha, el cable tiene baja rigidez. Si la flecha decrece, la rigidez del cable aumenta y el comportamiento del cable se aproxima al de una barra articulada en tensión.

Para considerar los efectos anteriores, la flecha y efectos elásticos, se considera que el cable real puede modelarse mediante dos elementos diferentes conectados en serie. Este modelo es propuesto por Abbas [1] para modelar los tirantes. El elemento tipo tirante consta de un cable inextensible conectado en serie con una barra articulada. El cable tiene una longitud  $l$ , un módulo de elasticidad infinito y se asume que tiene un perfil parabólico; la barra tiene longitud  $l$  con módulo elástico y sección transversal igual al del cable real. De esta manera, la deformación axial total está dada por la suma de las deformaciones de cada parte.

$$\delta = \delta_c + \delta_b \quad (2.01)$$

donde  $\delta_c$  y  $\delta_b$  son las elongaciones individuales del cable y de la barra, respectivamente, cuando están sujetas a un cambio de esfuerzo de  $\sigma_1$  a  $\sigma_2$ .

### Formulación matricial

EL elemento cable se muestra en la Fig. 2.03. El eje local  $\mathbf{x}$  está definido por los nodos extremos  $\mathbf{i}$  y  $\mathbf{j}$ ; el eje local  $\mathbf{y}$  es perpendicular al eje local  $\mathbf{x}$ . Con referencia a las coordenadas locales, el elemento tiene dos grados de libertad en cada nodo; ambos son de traslación en  $\mathbf{x}$  y en  $\mathbf{y}$ . La matriz de rigidez del elemento en coordenadas locales está dada por la suma de la rigidez de la barra articulada y de la rigidez del cable inextensible.

$$[k] = [k_b] + [k_c] \quad (2.02)$$

donde  $[k_b]$  es la matriz de rigidez elástica y está dada por:

$$[k_b] = \frac{A_0 E_{tan}}{l} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.03)$$

y  $[k_c]$  es la matriz de rigidez geométrica y está dado por:

$$[k_c] = \frac{H}{l} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (2.04)$$

$E_{tan}$  es el módulo elástico tangencial,  $A_0$  es el área de la sección transversal,  $l$  es la longitud del elemento y  $H$  es la fuerza axial en los extremos. La matriz  $[k]$  es la matriz de rigidez tangente del elemento.

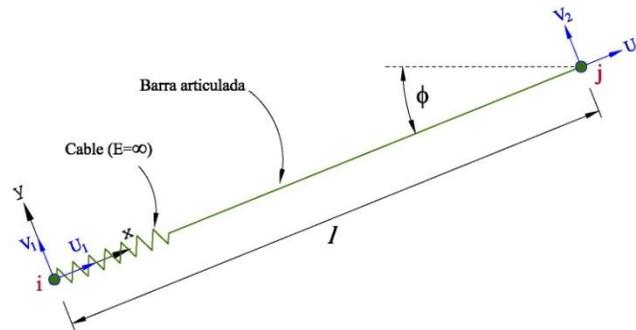


Figura 2.03.- Elemento tirante y sus grados de libertad locales.

### 2.2.3 Presfuerzo

Para los tendones de presfuerzo se usa la formulación propuesta por Ketchum [14]. Un tendón de presfuerzo que tiene un perfil arbitrario es modelado como una secuencia de segmentos lineales los cuales se considera que están en un estado de esfuerzo axial uniforme. Los nodos que forman los segmentos del tendón se conectan a los nodos de un elemento barra por medio de restricciones rígidas. Las coordenadas globales  $\mathbf{X}$  y  $\mathbf{Y}$  de cada uno de los puntos del tendón se pueden introducir directamente o pueden ser generadas usando un esquema de generación paramétrica. La entrada o la generación pueden estar alternativamente en un sistema de coordenadas locales  $\mathbf{r}$ - $\mathbf{s}$  especificados por el usuario. Las coordenadas  $\mathbf{r}$  y  $\mathbf{s}$  se convierten internamente al sistema global  $\mathbf{X}$ - $\mathbf{Y}$ . El programa puede determinar las pérdidas instantáneas de fricción y de anclaje. La carga y la rigidez que aporta cada segmento del tendón son agregadas al elemento barra asociado.

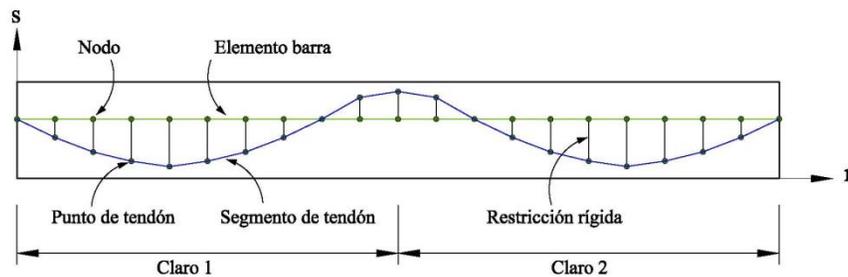


Figura 2.04.- Discretización del tendón.

## 2.3 ANÁLISIS SECUENCIAL POR ETAPAS

Para poder predecir el comportamiento de una estructura en el tiempo, se requiere una serie de etapas constructivas o de análisis. Las etapas quedan delimitadas por aquellos instantes en los que se producen variaciones en las cargas exteriores, se efectúan cambios en las características resistentes o se modifican las condiciones de frontera en la estructura.

El programa que se presenta en esta tesis es capaz de considerar los diferentes cambios que puedan ocurrir en la estructura durante el proceso de construcción y a lo largo de su vida útil. En cualquier etapa de construcción, los cambios en la geometría pueden suceder debido a la adición o eliminación de elementos barra o tirantes; los tendones de presfuerzo y los tirantes se pueden instalar, retensar o eliminar en cualquier momento y las condiciones de frontera o apoyos pueden modificarse también en cualquier etapa de construcción.

Para incorporar las no linealidades que dependen del tiempo en el análisis, el tiempo transcurrido de una etapa a la siguiente es dividido en intervalos de tiempo, separados por pasos de tiempo. En donde para un paso siguiente la integración es realizada, y en la que los incrementos de desplazamiento, de deformaciones y de esfuerzos son agregados sucesivamente a los totales anteriores. Para cada paso de tiempo, un análisis de rigidez directa basado en el método de desplazamientos se lleva a cabo en el dominio del espacio. Dependiendo del tipo de integración que se seleccione para el flujo plástico, las ecuaciones de equilibrio pueden ser no lineales y una solución por iteraciones es requerida.

El esquema general del programa está representado en la Fig. 2.05, en donde la entrada general de datos (1) incluye la geometría de la estructura, condiciones de apoyo, propiedades de los materiales, refuerzo, presfuerzo, tirantes, condiciones ambientales y tolerancias de convergencia. Para cada etapa de construcción, los datos de entrada (2) incluyen variaciones en la geometría, variaciones en las condiciones de frontera o apoyos, cargas, presfuerzo, así como el número de pasos de tiempo entre etapas constructivas y factores de carga para los pasos de carga. A continuación se describe el proceso de análisis para un intervalo de tiempo.

### 2.3.1 Procedimiento de análisis no lineal en el tiempo

La entrada al proceso de análisis consiste en proveer la geometría de la estructura, las características del presfuerzo, la historia de cargas en los nodos, las relaciones de esfuerzo deformación, las propiedades no lineales que dependen del tiempo y las condiciones de frontera.

Los resultados del análisis consisten en los desplazamientos nodales  $r$ , las deformaciones  $\varepsilon$  y esfuerzos  $\sigma$  internos tanto en el concreto, como en el acero de refuerzo, en el acero de presfuerzo y en los tirantes para algún instante en el tiempo.

Considerando un intervalo típico  $t_{n-1}$  a  $t_n$ , el procedimiento de análisis consiste en:

1. El estado de la estructura para el tiempo  $t_{n-1}$  es conocido; esto consiste en los desplazamientos en los nodos  $r$ , deformaciones totales  $\varepsilon$ , deformaciones no mecánicas totales  $\varepsilon^{nm}$  y los esfuerzos  $\sigma$  para todos los puntos de la estructura.

2. Ensamblar el vector de incremento de carga  $\Delta R_n$  para el tiempo  $t_n$ .

$$\Delta R_n = \Delta R_n^j + \Delta R_n^{nm} + R_{n-1}^u$$

Donde  $\Delta R_n^j$  es el incremento de carga en los nodos,  $R_{n-1}^u$  es la carga desbalanceada obtenida para el tiempo  $t_{n-1}$  y  $\Delta R_n^{nm}$  es el vector de carga equivalente en los nodos debido a las deformaciones no mecánicas acumuladas durante el paso de tiempo. Las deformaciones no mecánicas son debidas al flujo plástico y contracción en el concreto, a la relajación en los aceros de presfuerzo y tirantes y al cambio de temperatura que ocurren durante el paso de tiempo.

3. Formar la rigidez tangente en coordenadas locales para cada elemento barra basado en la geometría actual y propiedades de los materiales.
4. Ensamblar la matriz de rigidez tangente de la estructura  $K_t$  en coordenadas globales utilizando la matriz de transformación actual de desplazamientos para cada elemento.
5. Resolver para el incremento de desplazamiento  $\Delta r$ .

$$K_t \Delta r = \Delta R_n$$

6. Transformar el incremento de desplazamiento global  $\Delta r$  a incrementos locales de deformación,  $\Delta \varepsilon$ , en el elemento utilizando relaciones no lineales de desplazamiento-deformación.
7. Agregar  $\Delta r$  y  $\Delta \varepsilon$  a los totales anteriores para obtener los desplazamientos totales en los nudos  $r$  y deformaciones totales  $\varepsilon$ , respectivamente. Actualizar la geometría y las matrices de transformación de los elementos.
8. Restar las deformaciones no mecánicas actuales totales  $\varepsilon^{nm}$  de las deformaciones actuales totales  $\varepsilon$  para obtener la deformación mecánica total actual  $\varepsilon^m$ . Calcular los esfuerzos en el concreto, acero de refuerzo, acero de presfuerzo y tirantes utilizando relaciones no lineales de esfuerzo-deformación.
9. Calcular las fuerzas internas en el extremo del elemento en coordenadas globales mediante la integración de los esfuerzos internos en coordenadas locales y transformándolos a coordenadas globales utilizando las matrices actualizadas de transformación de desplazamientos. Éstas se incorporan dentro del vector de cargas resistentes internas del nodo.
10. Obtener la carga desbalanceada restando las cargas resistentes internas del nodo de las cargas externas totales actuales del nodo.

$$R^u = R^j - R^i$$

11. Establecer  $\Delta R = R^u$ . Repetir del paso 3 al paso 11 hasta que las cargas desbalanceadas estén dentro de las tolerancias permisibles.

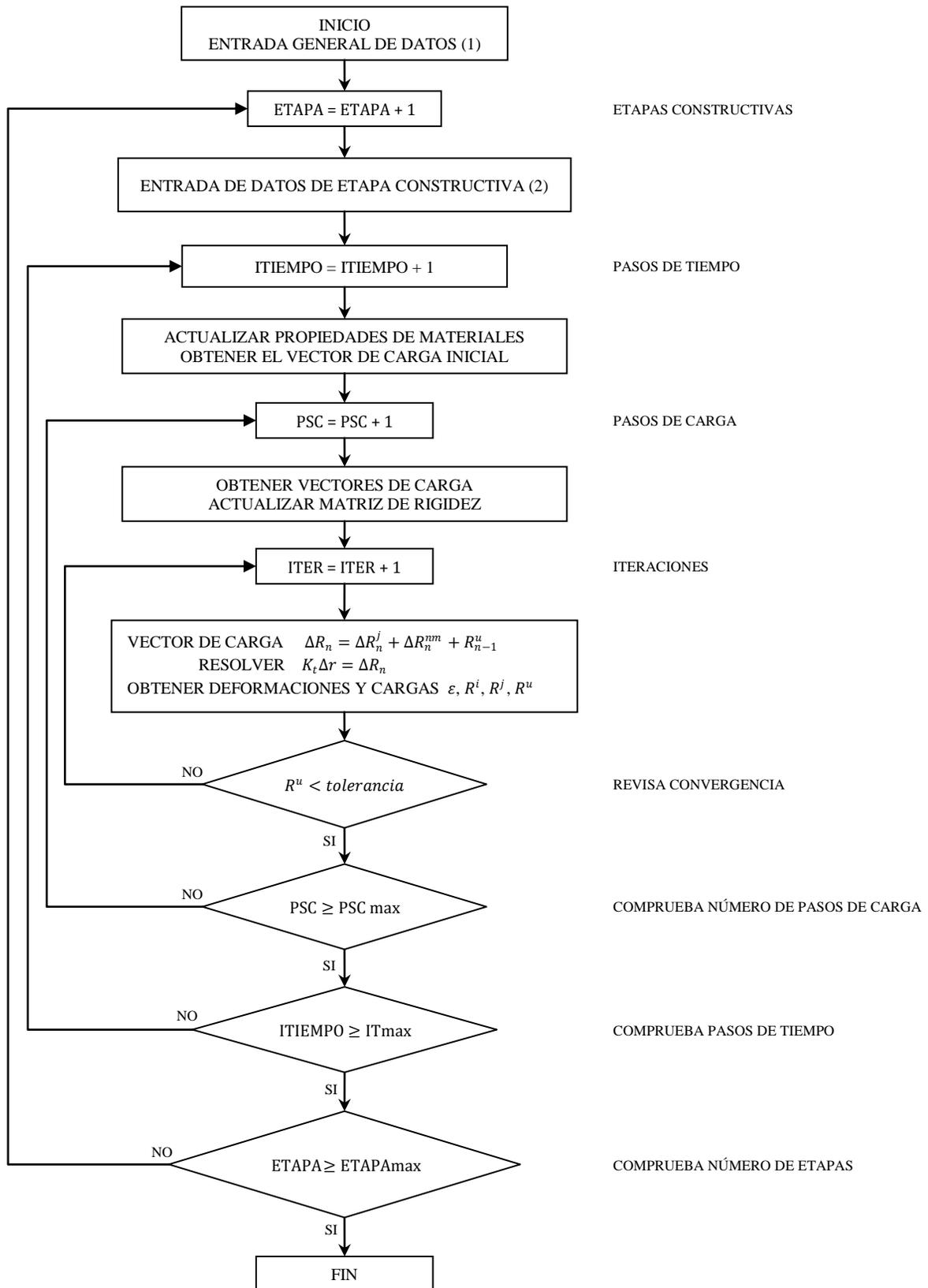


Figura 2.05.- Diagrama de flujo del módulo de análisis.

### 2.3.2 Ecuaciones fundamentales

Las ecuaciones fundamentales en que se basa el análisis de una estructura son las siguientes:

1. Compatibilidad de deformaciones. Sirve para revisar la compatibilidad entre los desplazamientos de los nodos y deformaciones en el elemento.
2. Equilibrio. En un elemento, que está sujeto a cargas externas, deben generarse esfuerzos en los materiales cuya resultante debe estar en equilibrio con las fuerzas externas aplicadas.
3. Ecuaciones constitutivas. Sirven para establecer las relaciones entre las deformaciones y los esfuerzos aplicados en los materiales de un elemento. Establecen el comportamiento del material ante diferentes niveles de esfuerzo, esta relación puede ser no lineal. Más adelante se mencionan los tipos de relaciones esfuerzo-deformación para los diferentes materiales.

### 2.3.3 Inclusión de las no linealidades geométricas

La inclusión del efecto de las no linealidades de origen geométrico o de segundo orden se ha realizado estableciendo el equilibrio sobre la geometría deformada. Se ha mantenido la hipótesis de pequeñas deformaciones, lo cual permite mantener la condición de compatibilidad similar a la de primer orden. Así, en un planteamiento general, la matriz de rigidez se va actualizando de acuerdo con la deformación de la estructura y se tiene en cuenta la influencia de los desplazamientos sobre los esfuerzos en las secciones.

## 2.4 COMPORTAMIENTO DE LOS MATERIALES

Los modelos no lineales de los materiales usados en el programa, son los incluidos en el programa SPCFRAME [12] y en el programa NASCAB [1]. Las propiedades de los materiales para alguna etapa de análisis dependen del estado de deformación del material. A continuación se presentan los modelos para cada material.

### 2.4.1 Concreto

Para el concreto que está sometido a un estado de esfuerzo axial, las siguientes relaciones básicas son necesarias para modelar su comportamiento:

1. Relación esfuerzo-deformación a corto plazo, incluyendo cargas reversibles.
2. Relaciones en función del tiempo para flujo plástico, contracción, envejecimiento e historia de temperatura; éstos son discutidos más adelante.

Las deformaciones diferidas se han englobado dentro del concepto de deformaciones no mecánicas  $\varepsilon^{nm}$ , considerando como deformaciones mecánicas  $\varepsilon^m$  las producidas de manera instantánea por un esfuerzo aplicado.

$$\varepsilon_c(t) = \varepsilon_c^m(t) + \varepsilon_c^{nm}(t) \quad (2.05)$$

$$\varepsilon_c^{nm}(t) = \varepsilon_c^c(t) + \varepsilon_c^s(t) + \varepsilon_c^a(t) + \varepsilon_c^T(t) \quad (2.06)$$

donde  $\varepsilon_c^c(t)$  es la deformación de flujo plástico,  $\varepsilon_c^s(t)$  es la debida a la contracción,  $\varepsilon_c^a(t)$  es la del envejecimiento y  $\varepsilon_c^T(t)$  corresponde a los efectos de temperatura. El subíndice  $c$  hace referencia al concreto.

### Comportamiento instantáneo

La curva esfuerzo-deformación a corto plazo utilizada para las capas de concreto está basada en las ecuaciones de Kent-Park [13]. Se puede modelar tanto concreto confinado como no confinado. El efecto *tension stiffening* puede incluirse; la componente de descarga de la curva esfuerzo-deformación está basada en la ecuación cúbica propuesta por Lin [15]. Considerando cargas reversibles adicionales, en total dieciséis estados posibles son obtenidos para el concreto, Fig. 2.06.

La parte ascendente de la curva (estado 2) es descrita por la ecuación:

$$\sigma = f_c' \frac{\varepsilon^m}{\varepsilon_0} \left( 2 - \frac{\varepsilon^m}{\varepsilon_0} \right) \quad (2.07)$$

donde  $\varepsilon_0$ , que corresponde al esfuerzo de compresión máximo  $f_c'$ , está dado por:

$$\varepsilon_0 = \frac{2f_c'}{E_i} \quad (2.08)$$

La parte descendente de la curva es una línea recta. El módulo tangente de ésta se considera como cero para propósito de rigidez; sin embargo, la descarga correspondiente al incremento del esfuerzo mecánico es calculada con:

$$\sigma = -0.15 f_c' \frac{\varepsilon^m - \varepsilon_0}{\varepsilon_u - \varepsilon_0} \quad (2.09)$$

El módulo tangente inicial  $E_i$  es el recomendado por ACI-209:

$$E_i = 33\omega^{1.5}\sqrt{f_c'} \quad (2.10)$$

donde  $f_c'$  es el esfuerzo de compresión en psi y  $\omega$  es el peso unitario del concreto en pcf.

También el esfuerzo de tensión del concreto,  $f_t'$  es calculado de acuerdo al ACI-209:

$$f_t' = r_t\sqrt{\omega f_c'} \quad (2.11)$$

donde  $r_t$  es un parámetro entre 0.6 y 1.0

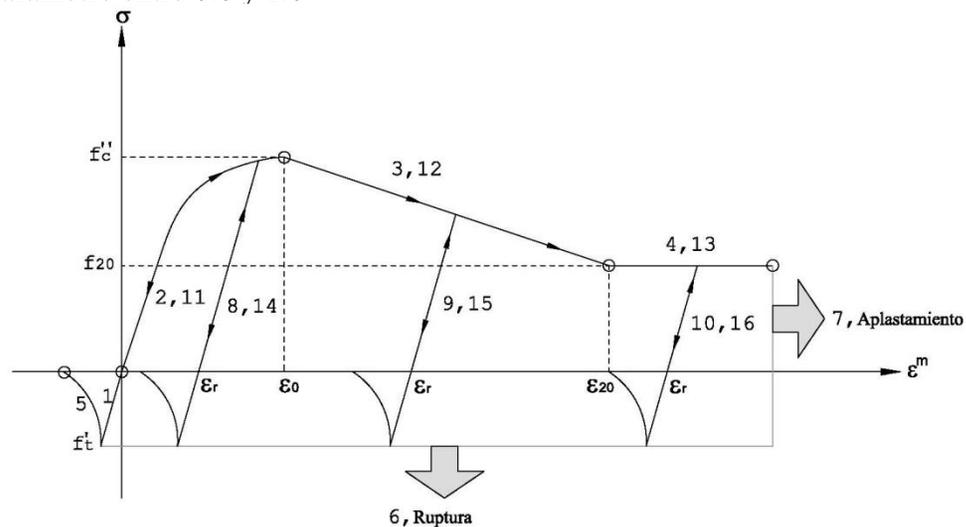


Figura 2.06.- Curva esfuerzo-deformación para el concreto.

El agrietamiento en el concreto es más complicado de modelar debido a su naturaleza intrínseca. Cuando las tensiones en el concreto superan el valor de la resistencia a tensión se produce la fisuración o agrietamiento. Debido a su carácter discreto, entre dos grietas consecutivas de un elemento tensionado de concreto armado, existe una parte de concreto que colabora en la resistencia de la pieza gracias a las tensiones de adherencia con el acero. Esta colaboración del concreto entre fisuras ocasiona un efecto de rigidización al cual se le llama “*tension stiffening*”. El efecto de *tension stiffening* tiene gran influencia en el comportamiento deformacional de las estructuras de concreto armado, especialmente en las situaciones de servicio. Los modelos propuestos por Lin y Scordelis [15] se mantienen en el programa presentado.

### Comportamiento a largo plazo

El comportamiento a largo plazo del concreto está influenciado básicamente por los fenómenos de flujo plástico o fluencia, la contracción, el envejecimiento y las variaciones de temperatura. El flujo plástico origina un aumento en la deformación en el concreto sometido a un esfuerzo sostenido; el efecto de contracción es eso, una contracción progresiva sin importar que el concreto esté sometido a esfuerzos; y el envejecimiento toma en cuenta la variación de las propiedades mecánicas del concreto a lo largo del tiempo. Estos fenómenos vienen influenciados por una multitud de factores tales como las condiciones ambientales y su uniformidad en la pieza, las características de los materiales usados en la construcción del concreto o la naturaleza de las cargas aplicadas.

### Flujo plástico del concreto

La deformación por flujo plástico se define como el incremento en la deformación bajo esfuerzo sostenido. Puede ser que el concreto esté sujeto a esfuerzo constante o bien que el concreto esté sujeto a esfuerzos variables.

### Flujo plástico bajo esfuerzo constante

La deformación del concreto en un tiempo  $t$ , bajo esfuerzo constante aplicado en un tiempo  $t_0$ , es expresada como:

$$\varepsilon(t, t_0) = \varepsilon_i(t_0) + \varepsilon_c(t, t_0) \quad (2.12)$$

donde:

$$\varepsilon_i(t_0) = \frac{\sigma(t_0)}{E_c(t_0)} \quad (2.13)$$

es la deformación instantánea y

$$\varepsilon_c(t, t_0) = \phi(t, t_0) \frac{\sigma(t_0)}{E_{28}} \quad (2.14)$$

es la deformación por flujo plástico;  $E_c(t_0)$  es el módulo elástico del concreto en el tiempo  $t_0$ ,  $\phi(t, t_0)$  es la función de flujo plástico y  $E_{28}$  es el módulo de elasticidad a los 28 días de edad del concreto. Podemos hacer uso de la función:

$$J(t, t_0) = \frac{1}{E_c(t_0)} + \frac{\phi(t, t_0)}{E_{28}} \quad (2.15)$$

y entonces podemos reescribir 2.12 como:

$$\varepsilon(t, t_0) = J(t, t_0) \sigma(t_0) \quad (2.16)$$

### Flujo plástico bajo esfuerzos variables

La deformación de flujo plástico en un tiempo  $t$ , bajo esfuerzos variables, está dada por la ecuación:

$$\varepsilon(t) = \int_0^t J(t, \tau) \frac{\partial \sigma(\tau)}{\partial \tau} d\tau \quad (2.17)$$

donde  $J(t, \tau)$  es la función de fluencia específica que depende de la edad de carga  $\tau$ , y  $\sigma(\tau)$  es el esfuerzo aplicado en el instante  $\tau$ . Esta fórmula es más general. Puede observarse que el principal inconveniente es la necesidad de conocer toda la historia de esfuerzos lo cual, en determinados casos, puede provocar problemas de capacidad de almacenamiento de datos.

El método numérico incluido en el programa para evaluar la deformación debida al flujo plástico es el procedimiento desarrollado por Kabir [10]. Este método asume que el principio de superposición es válido en el sentido de que la deformación total de flujo plástico para cualquier

tiempo  $t$  puede encontrarse como la suma de deformaciones producidas por cambios de esfuerzos en diferentes tiempos  $\tau$  con duración mayor a  $t$ . Tal formulación es la que sigue:

1. La deformación por flujo plástico  $\varepsilon(t)$  para cualquier tiempo  $t$  está dada por la ecuación para un material viscoelástico:

$$\varepsilon(t) = \int_0^t J(t, \tau) \frac{\partial \sigma(\tau)}{\partial \tau} d\tau \quad (2.18)$$

2. Valores discretos para la función  $J(t, \tau)$  para tiempos de observación  $t_1, t_2 \dots t_n$  debido a cambios de esfuerzos unitarios para el tiempo  $\tau$  pueden obtenerse por experimentos de laboratorio o por relaciones empíricas dadas por ACI-209 [2] o CEB-FIP [8].

De acuerdo con ACI-209 la deformación por flujo plástico puede ser calculada usando la siguiente relación:

$$C(t) = K_s K_H K_h K_\tau \frac{(t - \tau)^{0.6}}{10 + (t - \tau)^{0.6}} C_u \quad (2.19)$$

donde  $C(t)$  es el coeficiente de flujo plástico, el cual es la relación de la deformación de flujo plástico para un tiempo  $t$  y la deformación inicial instantánea,  $C_u$  es el coeficiente último de flujo plástico,  $K_s, K_H, K_h$  y  $K_\tau$  son factores de corrección para revenimiento, humedad, tamaño del elemento y edad de carga,  $t$  es el tiempo de observación en días y  $\tau$  es la edad de carga en días.

3. La función de flujo plástico  $J(t, \tau)$  puede aproximarse con series de Dirichlet:

$$J(t, \tau) = \sum_{i=1}^m a_i(\tau) \left[ 1 - e^{-\frac{(t-\tau)}{\Gamma_i}} \right] \quad (2.20)$$

donde  $a_i(\tau)$  son coeficientes de envejecimiento y  $\Gamma_i$  son tiempos de retardo. La ecuación de series de Dirichlet es ajustada al conjunto de valores discretos de flujo plástico obtenidos en el paso 2 mediante mínimos cuadrados. Esto lleva a tener coeficientes  $a_i$  para varias edades de carga  $\tau$ .

4. La deformación total de flujo plástico para cualquier tiempo  $t$  está dado por:

$$\varepsilon^c(t) = \int_0^t \sum_{i=1}^m a_i(\tau) \left[ 1 - e^{-\frac{(t-\tau)}{\Gamma_i}} \right] d\sigma(\tau) \quad (2.21)$$

Usando un esquema de integración numérica, la ecuación anterior es desarrollada en una relación recursiva de la forma:

$$\varepsilon^c(t_j) = \varepsilon^c(t_{j-1}) + \Delta\varepsilon^c(t_j) \quad (2.22)$$

El incremento de flujo plástico  $\Delta\varepsilon^c(t_j)$  del tiempo  $t_{j-1}$  a  $t_j$  puede ser evaluado asumiendo tanto una variación lineal o una variación constante en los parámetros del material o de esfuerzos sobre el intervalo  $\Delta t_j$ , como es implementado por Ketchum en el programa SFRAME.

### Contracción del concreto

Sucede la contracción del concreto, en el tiempo, cuando éste no está sometido a cargas, restricciones o variaciones de temperatura. La contracción está causada básicamente por un efecto de secado y se ve principalmente afectada por las condiciones ambientales, la geometría de la pieza, los componentes del concreto y sus proporciones y el tiempo transcurrido desde la finalización del curado.

Al igual que para el flujo plástico existen diversos modelos para determinar la deformación debida a la contracción del concreto, El programa presentado en esta tesis tiene como opciones las formulaciones del ACI-209 y del CEB-FIP para la contracción.

De acuerdo al ACI-209 la deformación por contracción es calculada usando la siguiente expresión:

$$\varepsilon^s(t) = \varepsilon_u^s k_s k_h k_H \frac{(t-t_0)^e}{f+(t-t_0)^e} \quad (2.23)$$

donde  $\varepsilon^s(t)$  es la deformación por contracción para el tiempo  $t$ ,  $t_0$  es la edad de curado,  $f$  y  $e$  son constantes determinadas de experimentos,  $k_s$  es un factor de corrección por revenimiento o asentamiento,  $k_h$  es el factor de corrección por tamaño del elemento y  $k_H$  es el factor por corrección por humedad relativa.

El CEB-FIP propone la siguiente ecuación:

$$\varepsilon^s(t, \tau) = \varepsilon_{su} [\beta_s(t) - \beta_s(\tau)] \quad (2.24)$$

donde  $\varepsilon_{su}$  es la deformación por contracción última,  $\beta_s$  representa el efecto de contracción en el espesor teórico,  $t$  es la edad del concreto en el tiempo de observación y  $\tau$  es la edad del concreto al inicio del intervalo de tiempo.

### Envejecimiento del concreto

El efecto del envejecimiento tiene en cuenta la variación de las propiedades mecánicas del concreto en el tiempo. La deformación por envejecimiento  $\delta\varepsilon^\alpha$ , que ocurre en un elemento de concreto sujeto a esfuerzo constante en el intervalo de tiempo  $t_{n-1}$  a  $t_n$  está dado por:

$$\delta\varepsilon^\alpha = \sigma \left[ \frac{1}{E(t_{n-1})} - \frac{1}{E(t_n)} \right] \quad (2.25)$$

Considerando un elemento de concreto que está sujeto a esfuerzo constante en un intervalo de tiempo  $t_1$  a  $t_2$ , Fig. 2.07, se observa que el módulo de elasticidad cambia con el tiempo. Asignamos los módulos  $E_1$  y  $E_2$  a los tiempos  $t_1$  y  $t_2$ , respectivamente. Designamos a la deformación mecánica total por  $\varepsilon$ . Si se desea determinar el esfuerzo en el concreto para el tiempo  $t_2$  usando las relaciones instantáneas esfuerzo-deformación, podríamos utilizar la siguiente ecuación:

$$\sigma = E_2 \varepsilon \tag{2.26}$$

Pero esta ecuación no predice tal esfuerzo, por lo que requiere que sea corregida como sigue:

$$\sigma = E_2(\varepsilon - \varepsilon^\alpha) \tag{2.27}$$

De esto se concluye que el envejecimiento es una deformación ficticia y no representa una deformación física del concreto; puede considerársele como una corrección a la deformación mecánica total usada en las relaciones esfuerzo-deformación instantánea para permitir el cambio del módulo elástico.

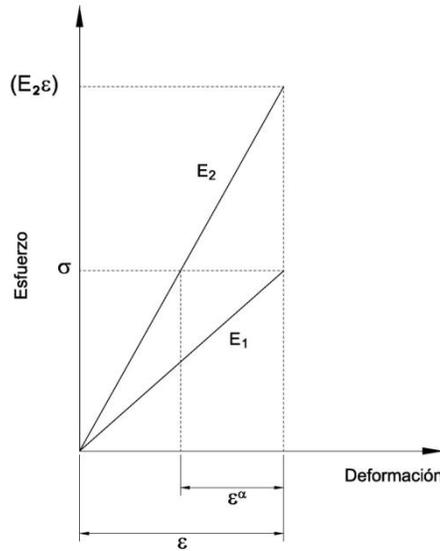


Figura 2.07.- Deformación por envejecimiento.

### Deformaciones térmicas

Las deformaciones uniaxiales debidas al gradiente de temperatura en cualquiera de los materiales que componen la sección del elemento pueden expresarse por:

$$\varepsilon^T = \alpha \Delta T \tag{2.28}$$

donde  $\Delta T$  es la variación de temperatura en el período considerado y  $\alpha$  el coeficiente de dilatación lineal del material. El perfil de temperatura en la sección puede influir en los esfuerzos y deformaciones que se originen.

### 2.4.2 Acero de refuerzo

Para el acero de refuerzo se usa una relación esfuerzo-deformación bi-lineal con inversión de carga teniendo así cuatro estados posibles del material, Fig. 2.08. Para la ruta de la carga reversible se considera que lo hace a un ángulo igual al módulo inicial. La falla ocurre cuando la deformación mecánica  $\epsilon^m$  excede la deformación última  $\epsilon_u$ .

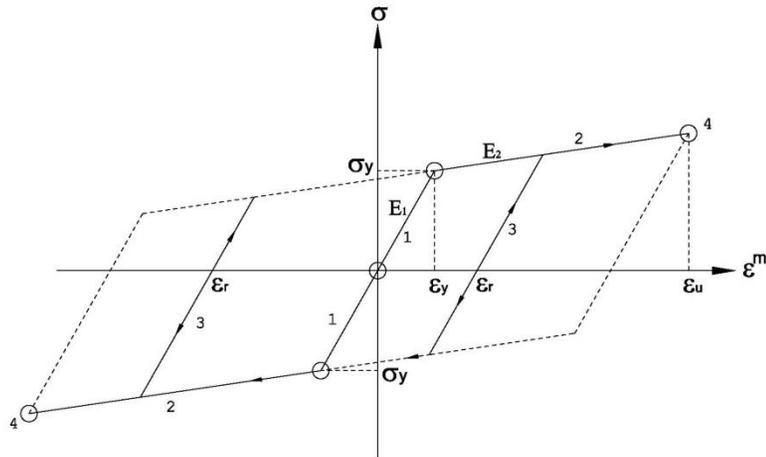


Figura 2.08.- Curva esfuerzo-deformación para el acero de refuerzo.

### 2.4.3 Acero de presfuerzo

La curva esfuerzo-deformación para el acero de presfuerzo es diferente a la del acero de refuerzo en dos aspectos:

1. Existe una gran diferencia en la magnitud de la resistencia a tensión.
2. En el caso del acero de presfuerzo no existe una meseta de fluencia bien definida.

Para poder modelar adecuadamente este comportamiento se utiliza una curva esfuerzo-deformación multilínea, Fig. 2.09.

### 2.4.4 Acero de tirantes

También se usa una curva multilínea, Fig. 2.09. El ángulo para la descarga y recarga es el mismo que para el módulo inicial.

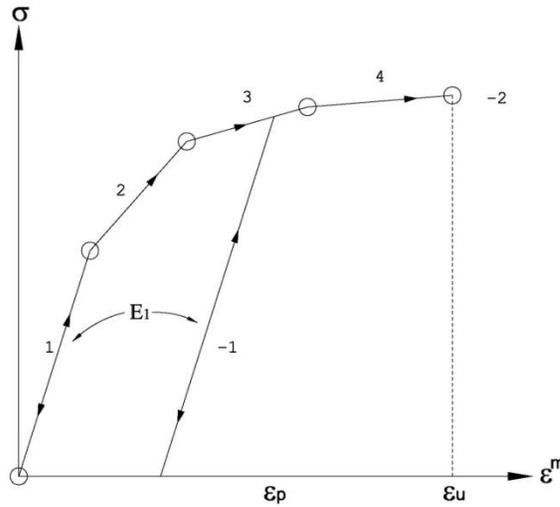


Figura 2.09.- Curva esfuerzo-deformación para el acero de presfuerzo y tirantes.

### Relajación en los aceros de presfuerzo y tirantes

Los aceros de presfuerzo y de los tirantes, se consideran como materiales elásticos lineales sujetos al fenómeno de relajación. La relajación del acero se produce cuando el acero de presfuerzo se mantiene en tensión, experimentando un reacomodo y rompimiento interno de partículas. Esta relajación debe tomarse en cuenta, ya que produce una pérdida significativa de la fuerza presforzante. La siguiente ecuación se usa para calcular el esfuerzo de relajación tanto en tendones de presfuerzo como en tirantes.

$$\frac{f_s}{f_{si}} = 1.0 - \frac{\log t}{c} \left( \frac{f_{si}}{f_{sy}} - 0.55 \right) \tag{2.29}$$

donde  $f_s$  es el esfuerzo en el acero para el tiempo  $t$ ,  $f_{si}$  es el esfuerzo inicial,  $f_{sy}$  es el esfuerzo de fluencia o resistencia del acero,  $c$  es una constante y  $t$  es el tiempo en horas después del tensado. Esta ecuación es válida solo si la deformación permanece constante durante el intervalo de tiempo considerado. Sin embargo en la realidad no permanece constante. Para considerar las variaciones de las deformaciones, Hernandez y Gamble [9] sugieren un procedimiento ilustrado en la Fig. 2.10. Este procedimiento fue implementado en NASCAB y es también usado en el UNASCAB. Se basa en el supuesto de que no todos los cambios en la relajación de la fuerza del tendón se producen al final de un paso de tiempo.

La fuerza inicial en el tendón  $f_{i0}$  aplicada en el tiempo  $t_0$  gradualmente se relaja una cantidad  $\delta f_{r1}$  hasta  $f'_{s1}$  durante el intervalo  $\delta t_1$ :

$$\delta f_{r1} = f_{i0} - f'_{s1} \tag{2.30}$$

En el tiempo  $t_1$ , debido a causas externas cambia la deformación lo que causa que la fuerza en el tendón disminuya hasta  $f_{s1}$ . Para calcular el esfuerzo de relajación  $\delta f_{r2}$  durante el intervalo  $\delta t_2$ , la ecuación 2.29 es usada para calcular la fuerza inicial ficticia  $f_{i1}$  que se ha relajado por  $\delta f_{r1}$

hasta  $f_{s1}$  durante  $\delta t_1$ . El esfuerzo  $f'_{s2}$  puede entonces determinarse, considerando  $f_{i1}$  como la fuerza inicial y aplicando la ecuación 2.29:

$$\delta f_{r2} = f_{s1} - f'_{s2} \quad (2.31)$$

El procedimiento es aplicado para cada segmento de tendón durante cada paso de tiempo hasta llegar al esfuerzo de relajación total en el tiempo  $t_n$ .

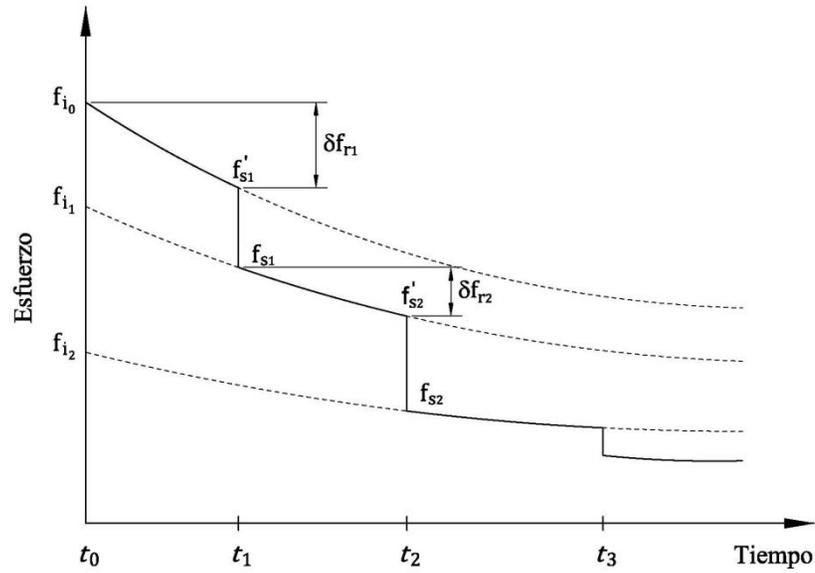


Figura 2.10.- Esfuerzo de relajación.

## Capítulo 3

# PROGRAMA DE CÓMPUTO

### 3.1 ESTRATEGIA DE DESARROLLO

En la actualidad hay varias metodologías de desarrollo de software. Dependiendo de lo que se quiera programar, puede ser más adecuada una que otra. En este trabajo se utilizaron las ideas propuestas en la metodología de programación por capas, debido a que por un lado requerimos un módulo de cálculo y por otro el de interfaz de usuario.

La programación por capas es un estilo de programación en el que el objetivo es la separación de la lógica del cálculo de la lógica del diseño. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que ocurra algún cambio, sólo se ataca el nivel requerido sin tener que revisar entre código mezclado. Se identifican dos capas en este trabajo: Capa de presentación y Capa de cálculo.

En la capa de presentación se provee de una interfaz de usuario y métodos de acceso a base de datos. Aquí es donde el programa presenta información al usuario y acepta entradas o respuestas del usuario para manipular las opciones del programa.

En la capa de cálculo se encuentran todas las subrutinas de análisis estructural y de generación de archivos de la base de datos.

### 3.2 HERRAMIENTAS DE DESARROLLO

En este trabajo se utilizaron diversas herramientas de programación, a continuación se describen las más importantes.

#### 3.2.1 FORTRAN

Debido a que FORTRAN continúa siendo el lenguaje de desarrollo de elección para las aplicaciones científicas y de ingeniería de alto rendimiento por la velocidad de proceso y las potentes librerías numéricas que posee, y a que el código original de las subrutinas de análisis estructural están escritas en este lenguaje de programación, se mantuvo este lenguaje para el módulo de análisis estructural. Se utilizó el sistema de desarrollo *Compaq Visual Fortran 6.6* para compilar y generar el módulo.

#### 3.2.2 Visual BASIC 6.0

Existen múltiples herramientas de desarrollo de programación visual para el entorno Windows que proporcionan estructuras de interfaz para el desarrollo de aplicaciones, lo que permite la creación de una interfaz de usuario (GUI, *Graphical User Interface*) de manera rápida. En este caso se eligió la herramienta de programación *Visual Basic 6.0* por su sencillez para programar aplicaciones de cierta complejidad para Windows, por las librerías de funciones con que cuenta, por su potencia en el manejo de base de datos y por la amplia documentación que existe del sistema.

Visual Basic 6.0 está orientado a la realización de programas para Windows, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc. Prácticamente todos los elementos de interacción con el usuario de los que dispone Windows pueden ser programados en Visual Basic 6.0. La esencia de las interfaces gráficas de usuario y de la programación orientada a eventos, reside en que el usuario realiza acciones y la aplicación responde. Con este sistema se desarrolló la interfaz de usuario y el acceso a la base de datos de los resultados.

### 3.2.3 API de Windows

Las API (Interfaz de Programación de Aplicaciones) de Windows son funciones o procedimientos que constituyen el corazón del sistema operativo Microsoft Windows y son usadas en la programación para controlar el aspecto y el comportamiento de cada elemento de Windows. Cada línea de código que se programa en Visual Basic se transforma a llamadas de estas funciones, pero además, podemos hacer llamadas a estas funciones directamente desde *Visual Basic*.

Casi todas las APIs de Windows se unen formando librerías de enlace dinámico (*Dynamic Link Libraries*, abreviadamente DLL). Una librería dinámica permite que las aplicaciones en Windows compartan código y recursos. Una DLL es un archivo ejecutable que puede ser utilizado por cualquier aplicación, la mayoría de estos archivos están almacenados en la carpeta *System* de Windows (como *User32.dll*, *GDI32.dll*, *Shell32.dll*...).

Las dos principales ventajas que obtenemos con la utilización de APIs es la gran funcionalidad que podemos darle a la aplicación debido a que no están dentro de nuestro código, sino que en el momento que realizamos la llamada a la función, la DLL se carga en memoria, se utiliza la API, y a continuación se descarga, y en segundo lugar la gran velocidad de proceso, ya que a menudo es mucho más rápido realizar una función a través de la API adecuada que por medio del lenguaje en sí mismo.

### 3.2.4 Controles ActiveX

Para desarrollar una aplicación en Visual Basic, se parte de un objeto base denominado formulario o ventana, se introduce en ella diversos objetos o controles que simbolizan datos o acciones que hay que llevar a cabo, se establecen propiedades de esos objetos y se añade código según el comportamiento de cada control. Estos controles pueden ser ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc.

Visual Basic tiene un conjunto fijo de controles estándar llamados controles intrínsecos y son controles base a cualquier aplicación. Un control *ActiveX* es una extensión al conjunto de los controles intrínsecos. Los controles *ActiveX* se usan como cualquiera de los controles estándar.

Cuando se agrega un control *ActiveX* a un programa, pasa a formar parte del entorno de desarrollo y de tiempo de ejecución y proporciona nuevas funcionalidades a la aplicación en desarrollo. Los controles *ActiveX* incrementan la capacidad en la programación debido a que incorporan métodos (funciones) y propiedades que aumentan enormemente la flexibilidad y capacidad del programador de Visual Basic.

Los controles *ActiveX* se encuentran dentro de archivos con extensión *ocx* y normalmente pueden ser utilizados por cualquier entorno de programación como Visual Basic.

Aparte de los controles de Visual Basic, en este trabajo se utilizaron dos controles externos: el CodeMax (*cmax20.ocx*) y el ChartSpace (*owc11.dll*); más adelante se menciona su uso.

### 3.2.5 ActiveX Data Objects (ADO)

Varios son los mecanismos que usan los programas de computadora para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas. Uno de estos mecanismos es el uso de *ActiveX Data Objects* (ADO).

Con ADO, un programa puede leer, insertar, editar, o borrar, la información contenida en diferentes áreas de almacenamiento dentro de la base de datos llamadas tablas. Además, se puede manipular la propia base de datos para crear nuevas áreas para el almacenamiento de información (tablas), como también alterar o eliminar las ya existentes, entre otras cosas.

ADO fue desarrollado por Microsoft y es usado en ambientes Windows por lenguajes de programación como *Visual Basic*, *C++*, *Delphi* entre otros, como también en la *Web* mediante el uso de *Active Server Pages* (ASP) y el lenguaje *VBScript*.

Los principales componentes de ADO son.

1. *Connection*. Permite establecer una conexión con la base de datos
2. *Recordset*. Maneja un conjunto de records de la base de datos
3. *Command*. Permite enviar órdenes SQL para ser ejecutados por la base de datos

Con este método se conectó la base de datos de resultados con la interfaz de usuario en este trabajo.

## 3.3 MODELO CONCEPTUAL

El programa *UNASCAB* consta de dos módulos básicos enlazados, *UNASCAB.exe* y *NasGo.exe* que procesan tareas complementarias. En la Fig. 3.01 se puede observar el diagrama conceptual de los módulos, siendo el módulo *UNASCAB.exe* la aplicación central y la que controla al otro módulo.

*UNASCAB.exe*, programado en Visual Basic 6.0, es el núcleo que soporta las tareas de comunicación con el usuario, gestiona la captura de datos y la presentación de resultados en un entorno gráfico; transfiere parámetros, recoge los resultados de los cálculos y controla el flujo general de la aplicación. En pocas palabras, es la interfaz de usuario.

*NasGo.exe* compilado en Fortran, es el núcleo del análisis estructural: es donde residen las técnicas numéricas matriciales para el análisis de los puentes atirantados.

El programa funciona de la siguiente manera: el archivo de comandos se genera desde el editor interno del programa (*UNASCAB.exe*). Desde este editor se pueden seguir dos caminos: por medio de un módulo de análisis de sintaxis (en inglés *parser*) interpretar los diferentes comandos y visualizar la geometría de la estructura en el graficador 2D para alguna etapa constructiva; por otro lado, se puede ejecutar todo el análisis haciendo una llamada al módulo *NasGo.exe* a través de una conexión tipo tubería (en inglés *pipe*). El módulo *NasGo.exe*, a la vez que analiza la estructura, genera los archivos de base de datos con los resultados para todas las etapas.

Una vez ejecutado el proceso de cálculo, desde el visualizador gráfico 2D se puede conectar la base de datos con la interfaz y desplegar los resultados de los elementos estructurales en forma de tabla y en forma de gráfico.

En los subcapítulos siguientes se explica el desarrollo de cada módulo y sus partes que la conforman.

### 3.4 MÓDULO DE ANÁLISIS

Las subrutinas de cálculo originales están desarrolladas en el lenguaje de programación FORTRAN. Debido a la potencia de este lenguaje en el manejo numérico y a su velocidad de proceso convino dejar las subrutinas en este lenguaje.

Se modificaron todas las subrutinas para actualizarlas de FORTRAN 77 a FORTRAN 90, debido a que la segunda es una actualización de FORTRAN 77 y por lo tanto incorpora más funciones intrínsecas del lenguaje que son de mucha ayuda en el desarrollo de la programación.

Se modificaron las subrutinas de manejo de archivos que permitiera también el uso de las rutas de cada archivo. Se implementó código que pudiera generar los archivos de resultados de cada etapa de forma independiente, con la ventaja de hacer consultas con mayor rapidez desde el módulo principal. Estos archivos son de formato CSV (Valores Separados por Comas) y conforman la base de datos de resultados para los elementos barra, los elementos tirantes y los nodos.

La subrutina principal de este módulo se modificó de tal manera que pudiera recibir parámetros desde el programa principal, de tal forma que todas las subrutinas pueden empaquetarse en un módulo independiente. El módulo generado es un archivo ejecutable llamado *NasGo.exe*.

Las modificaciones, la depuración y la compilación de las subrutinas se hicieron desde la plataforma de desarrollo *Compaq Visual Fortran 6.6*.

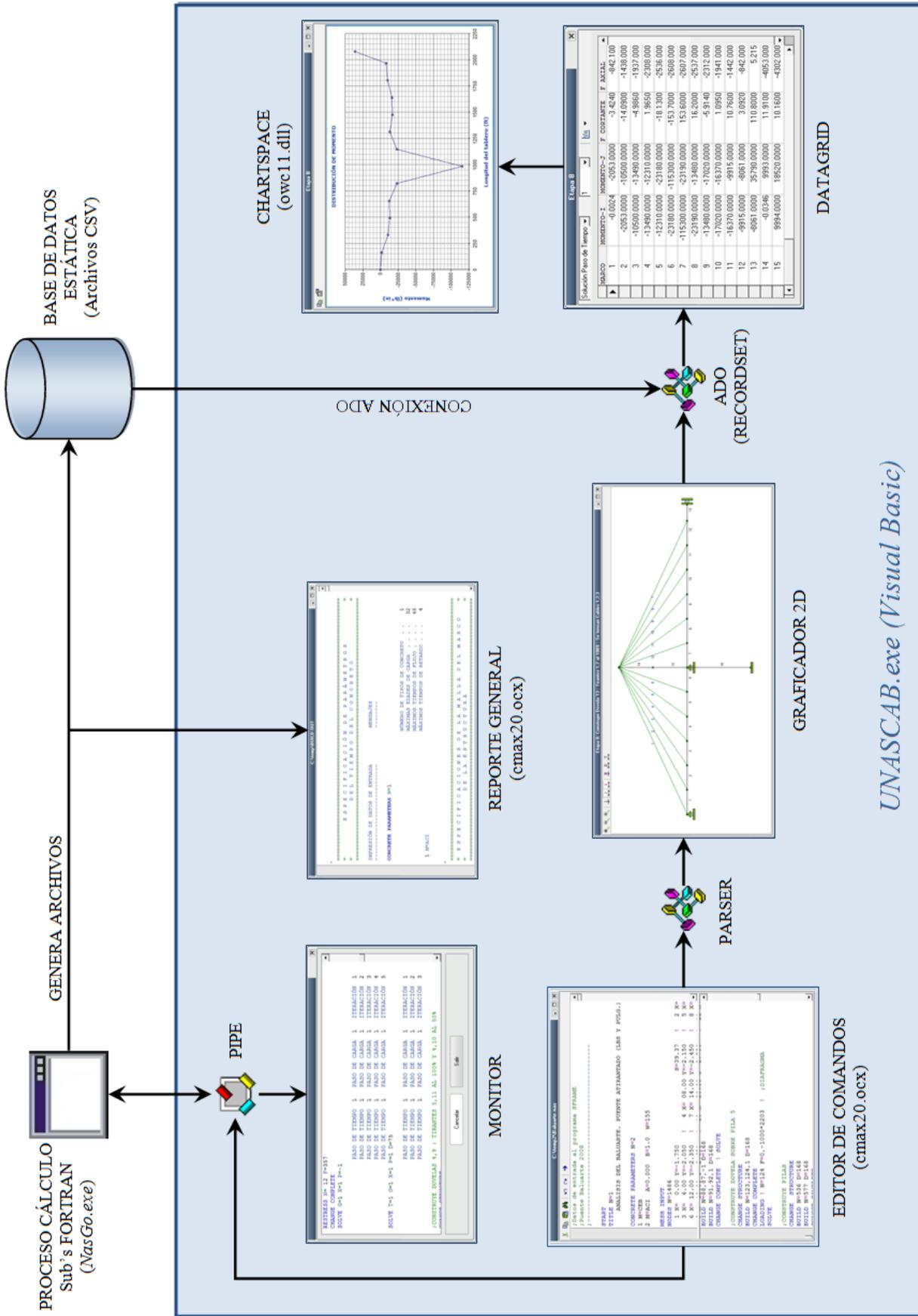


Figura 3.01.- Esquema conceptual

## 3.5 MÓDULO DE PRESENTACIÓN

El módulo de presentación es propiamente la interfaz de usuario (*UNASCAB.exe*), y se desarrolló en la plataforma Visual Basic 6.0; consta de varios componentes que ejecutan tareas específicas. A continuación se hace una descripción de cada uno de ellos.

### 3.5.1 Editor de comandos

Definir la estructura del puente no es una tarea sencilla, los datos a ingresar a veces son muchos y de características muy diferentes: tipos de elementos, geometría, restricciones, cargas, etc. Por ello se ha mantenido el ingreso de datos a través de un archivo de entrada y el uso de comandos originales. Esta metodología, si bien inicialmente puede requerir de un tiempo mayor de entrenamiento, le confiere al programa la capacidad de trabajar con datos de estructuras grandes y complejas que de otra forma sería difícil administrar.

Toda la información requerida para definir en forma completa la estructura se encuentra en este archivo de comandos. El tamaño reducido de este archivo facilita su transporte y copia de seguridad. Los comandos consisten en un grupo de palabras claves, seguido por datos numéricos o de carácter requeridos para la ejecución de esos comandos. El programa parte siempre de este archivo, ya sea para iniciar el proceso de análisis o para graficar la geometría de la estructura.

En la interfaz de usuario se ha integrado un editor para el manejo del archivo de comandos. Este editor se basó en el control *CodeMax* (cmax20.ocx). Se trata de un control ActiveX de edición de código y tiene mayores capacidades que los controles editores de texto propios de Visual Basic (*TextBox* y *RichTextBox*).

La ventaja principal del control *CodeMax* está en su potencia para manejar archivos de gran tamaño. Aunque el archivo de entrada normalmente no es grande, dependiendo del tamaño de la estructura, el archivo de salida (reporte general) sí puede llegar a serlo, así el visualizador del archivo de salida (Reporte general de resultados) también se programó con este control.

Otra funcionalidad del control es el resaltado de sintaxis; es decir, se le puede programar un lenguaje de comandos propio para que distinga y resalte las palabras claves, los comentarios y los datos de comando.

Por lo tanto el editor del programa se hizo con la característica de resalte de la sintaxis del lenguaje de comandos utilizado en la definición del puente. También se programó al editor con las opciones clásicas de un editor de texto: apertura y guardado de archivos, opciones de edición (cortar, copiar, pegar) y opciones de reemplazo y búsqueda de texto. En la Fig. 3.02 se muestra la apariencia del editor.

La forma del archivo de entrada no es única, dependiendo de la estructura a analizar, el uso de los comandos y sus parámetros va a cambiar, ya que hay parámetros cuyos valores se pueden introducir de manera directa o mediante una generación paramétrica. Sumado a esto, el lenguaje de comandos permite el uso de caracteres especiales a lo largo del archivo de entrada; por ejemplo, el signo de exclamación (!) sirve para usar varias líneas de comandos en una sola línea física, la barra

invertida (\) se utiliza para continuar un comando en la siguiente línea física y el punto y coma (;) se usa para colocar comentarios los cuales el programa ignora en cuestión de análisis.

Por lo anterior se requirió de la programación de un intérprete (*parser*) que pudiera reconocer los comandos; así como, obtener la estructura de datos para cada comando.

```

C:\temp\baluarte.nas
;Datos de entrada al programa UNASCAB
;Puente Baluarte 2008-2009
;-----
START
TITLE N=1
ANALISIS DEL PUENTE ESPECIAL BALUARTE (LBS Y PULG
CONCRETE PARAMETERS N=2
1 M=CEB
2 M=ACI A=0.000 B=1.0 W=155

MESH INPUT
NODES N=1486
1 X= 0.00 Y=-1.750 S=39.37 !
3 X= 6.00 Y=-2.050 ! 4 X= 08.00 Y=-2.150 !
6 X= 12.00 Y=-2.350 ! 7 X= 14.00 Y=-2.450 !

SOLVE

;11 CONSTRUYE ZAPATA DE PILA 8
CHANGE STRUCTURE
RESTRAINTS
602 R=1,1,1 !
BUILD N=601 D=110
CHANGE COMPLETE
SOLVE

;12 CONSTRUYE DOVELA DE PILA 4
CHANGE STRUCTURE

```

Figura 3.02.- Editor de archivo de comandos.

### 3.5.2 Módulo de Análisis de Sintaxis (*parser*)

Un *parser* es un programa o la parte de un programa que analiza una cadena alfanumérica para poder reconocer unidades gramaticales (como los comandos y parámetros de comando dentro del archivo de entrada) y transformar una entrada de texto en una estructura de datos que es apropiada para ser procesada.

Visual Basic incorpora funciones para el tratamiento de cadenas. Ejemplo de estas funciones son: *Len*, permite saber la longitud de cadena, *InStr*, se utiliza para buscar una cadena o parte de una cadena dentro de otra cadena, *Left*, *Right* y *Mid*, son utilizadas para extraer partes de una cadena, *Split*, permite separar una cadena utilizando un caracter como separador, etc. Con estas funciones se crearon las subrutinas de parseo de texto. El módulo reconoce los caracteres especiales (;,\) y ;) y reconoce e interpreta la generación paramétrica utilizada por algunos comandos. Por ejemplo, el parser obtiene del archivo de comandos las coordenadas de los nodos y conectividades de los elementos barra y de los elementos tirantes, ya sea que se hallan introducidos de manera directa o por medio de la generación paramétrica. También es capaz de saber los elementos estructurales que están activos en cada etapa y por lo tanto el número de etapas consideradas en el análisis.

### 3.5.3 Graficador 2D

El graficador 2D (Fig. 3.03) se encarga de mostrar la geometría actual para alguna etapa en particular. A través del parser, el visualizador obtiene las estructuras de datos de las coordenadas nodales y conectividades de las barras y tirantes activos de la etapa que está mostrando. También,

desde el graficador se activa la tabla con los resultados de los elementos mecánicos de las barras, tirantes y los desplazamientos de los nodos, para la etapa actual. Para acceder a los resultados, el visualizador utiliza la función de conexión a la base de datos (ADO) que más adelante se explica.

El graficador 2D se programó utilizando los controles gráficos de Visual Basic junto con operaciones de algebra matricial para el dibujado de la estructura. Los controles que se usaron son los controles *PictureBox*, *Line*, *Shape* e *Image*.

Con el control *Line* se modelaron tanto los elementos tipo barra como los elementos tipo tirantes. Como estos controles solo son meramente decorativos, es decir, no reconocen ningún evento, se programó una función para que se pudiera simular la selección de elementos y que de manera automática ubicara sus resultados en la tabla de resultados. El control *Shape* se utilizó para dibujar los nodos. Con el control *Image* se muestran los diferentes tipos de apoyos que se encuentran en la estructura, ya que es un contenedor de imágenes. El control *PictureBox* es el control gráfico más potente y general de Visual Basic. Se trata de una especie de formulario reducido, pues puede contener imágenes y otros tipos de controles tales como el *Line*, el *Shape* y el *Image*. Precisamente con esa función se utilizó el *PictureBox* en este visualizador, para servir de contenedor de los controles anteriores.

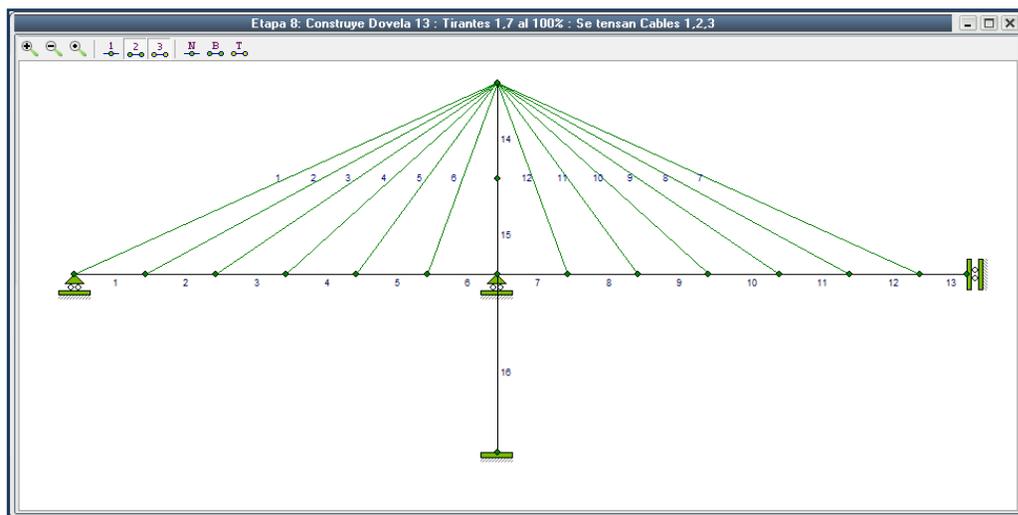


Figura 3.03.- Graficador 2D.

### 3.5.4 Conexión entre procesos

Como ya se mencionó, las subrutinas de cálculo quedaron empaquetadas dentro del archivo ejecutable *NasGo.exe*, independiente del programa principal (*UNASCAB.exe*). Así, para realizar el cálculo de la estructura es necesario que el programa principal cargue, ejecute y se comunique con el archivo externo *NasGo.exe*. Para ello se programó una clase de Visual Basic que es capaz de establecer esa comunicación, además de mostrar las tareas de cálculo que va realizando *NasGo.exe* en un objeto de la interfaz (Monitor, Fig. 3.04).

La arquitectura Windows proporciona un conjunto de mecanismos que permiten la comunicación entre diferentes procesos, uno de ellos son las llamadas tuberías (en inglés *pipe*). Una tubería (*pipe*) es un conducto o vía de comunicación entre dos terminales. Se puede usar una tubería para intercambiar datos entre dos procesos diferentes, o dentro del mismo proceso.

Hay dos tipos de tuberías: anónima y con nombre. Una tubería anónima se puede usar sin saber su nombre. Una tubería con nombre es lo opuesto: se tiene que conocer su nombre antes de poder usarla dentro del programa. Otra manera de clasificar las tuberías es de acuerdo a su propiedad: de un sentido o de dos sentidos. En una tubería de un sentido, los datos pueden fluir sólo en una dirección: de un proceso a otro. Mientras que en una tubería de dos sentidos, los datos pueden ser intercambiados entre dos terminales. Una tubería anónima siempre es de un sentido mientras que una tubería con nombre puede ser de un sentido o de dos sentidos.

La tubería anónima es realmente útil cuando tratamos con una aplicación de consola. Una aplicación de consola es un tipo de programa que usa una línea de comando de texto para pedir datos al usuario, también se le conoce como interfaz de solo texto, tal es el caso del módulo *NasGo.exe*.

Además de crear la tubería, es necesario crear el proceso con el que se desea conectar. Cuando Windows recibe la orden de crear un proceso, crea un espacio de direcciones privado para el proceso y luego proyecta el archivo ejecutable en la memoria de ese proceso (*NasGo.exe*).

La clase que se incluyó para crear la tubería utiliza la API *CreatePipe*, con la cual se crea una tubería anónima y regresa sus manejadores (*handles*) para lectura y escritura.

Para crear el proceso hijo, la clase utiliza la API *CreateProcessA*, la cual crea un nuevo proceso y su hilo primario (*Using Api in VB [4]*). El nuevo proceso ejecuta el archivo ejecutable especificado (en nuestro caso el *NasGo.exe*). Con la API *ReadFile* se leen los datos desde la terminal de lectura de la tubería.

Este módulo de clase a su vez está conectado a un objeto de la interfaz (monitor) donde se despliegan los mensajes que manda el archivo ejecutable externo (*NasGo.exe*).

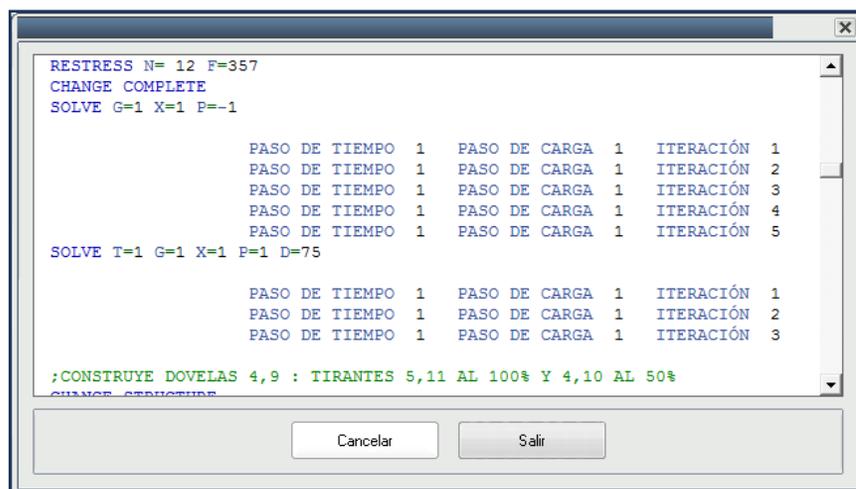


Figura 3.04.- Monitor.

### 3.5.5 Manejo de base de datos

El módulo *NasGo.exe* a la vez que analiza la estructura, genera los archivos que conforman la base de datos de resultados para cada etapa y para cada iteración considerada. Estos archivos son independientes del reporte general que también el módulo crea, lo que facilita en gran manera las operaciones de consulta y almacenamiento de los resultados.

Una base de datos consiste en un conjunto de datos relacionados y organizados lógicamente en un dispositivo de almacenamiento (Ej. disco duro, cintas magnéticas, discos ópticos, etc) (Rodríguez [25]). La base de datos del programa la constituyen 3 archivos tipo *CSV*, uno para los resultados de elementos tipo barra, otro de tirantes y otro para los de nodos. Los archivos *CSV* son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.

Los resultados para los tipo barra son momento en los extremos de cada barra, cortante y fuerza axial. Para los tipo tirante, fuerza axial y módulo tangente. Y para los nodos, restricciones y desplazamientos en sus grados de libertad; desplazamientos en **X** y en **Y** y rotaciones en **Z**.

Existen varias formas para conectar una base de datos a una aplicación de Visual Basic, entre las cuales cabe mencionar la utilización de tecnologías ODBC, DAO, RDO, ODBCdirect, OLE DB y ADO. Sólo se explican las utilizadas en este trabajo.

OLE DB es una tecnología de acceso a datos de bajo nivel y está basada en COM, una arquitectura que ha demostrado ser lo suficientemente robusta como para mover grandes cantidades de datos. OLE DB puede conectar a cualquier tipo de origen de datos (Balena [5]).

ADO (Objetos de Datos ActiveX) es una tecnología que permite acceder a cualquier base de datos u origen de datos. ADO es una interfaz de alto nivel con OLE DB, y está construida sobre un proveedor OLE DB para proporcionar funciones que no se encuentran disponibles directamente en OLE DB [5].

Las principales ventajas de ADO son su facilidad de uso, su gran velocidad, su escasa utilización de memoria y el poco espacio que ocupa en disco. Debido a esto, se utilizó ADO para conectar la base de datos de los resultados al programa principal (*UNASCAB.exe*).

ADO contiene tres objetos principales: *Connection*, *Recordset* y *Command*. El objeto *Connection* permite establecer la conexión a un origen de datos y especificar todos los parámetros necesarios antes de abrir una base de datos. El *Recordset*, contiene todos los datos que se leerán de una base de datos o que se enviarán a la misma. Un *Recordset* puede incluir varias filas y columnas de datos. Cada fila es un registro y cada columna es un campo del registro. El objeto *Command* permite hacer solicitudes o dar órdenes a la base de datos mediante una sentencia *SQL* (Lenguaje de Consulta Estructurado).

Para mostrar en forma tabular los resultados en la interfaz, se enlazó el *Recordset* a un control *DataGrid* de Visual Basic. El *DataGrid* es un control que muestra un conjunto de datos de manera tabular. Este control se actualiza automáticamente conforme el *Recordset* se actualiza. En la Fig. 3.05 se muestra en un *Datagrid* los resultados para una etapa (para nodos, para barras y para

tirantes). También esta ventana incluye las opciones de filtrado de datos para una solución instantánea o para una solución paso de tiempo, así como, para cada iteración en la solución.

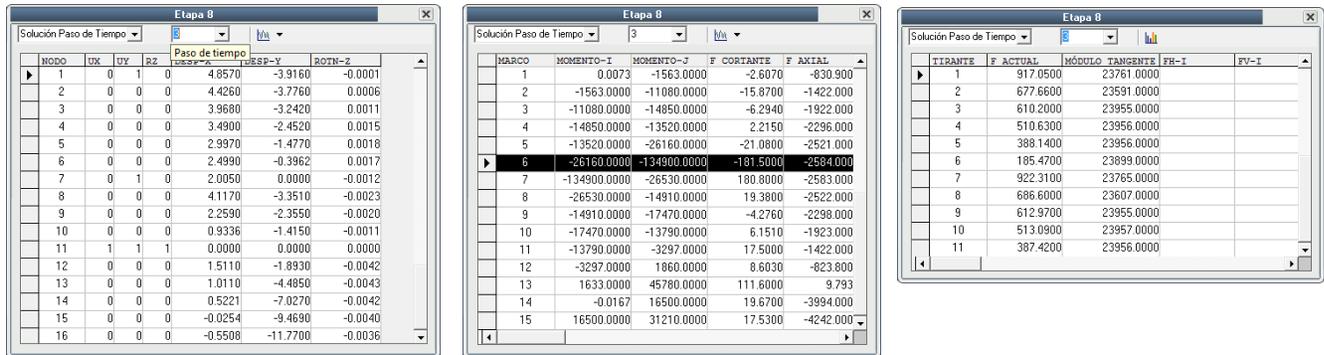


Figura 3.05.- Resultados en forma tabular (DataGrid).

Para mostrar los resultados en forma gráfica se utilizó el control *ChartSpace*, el cual es un componente del *Office Web Components* (OWC). El OWC es una colección de controles para la publicación en el Web de hojas de cálculo, gráficos y bases de datos. El *ChartSpace* permite hacer todo tipo de gráficas (de líneas, de barras, de pastel, etc), pero además tiene las opciones de edición del gráfico en tiempo de ejecución. Los datos a graficar se extraen directamente del *Recordset* formando las series a graficar correspondientes. A manera de ejemplo, en la Fig. 3.06, se muestran gráficas de momento y de tensiones en tirantes para un ejemplo dado.

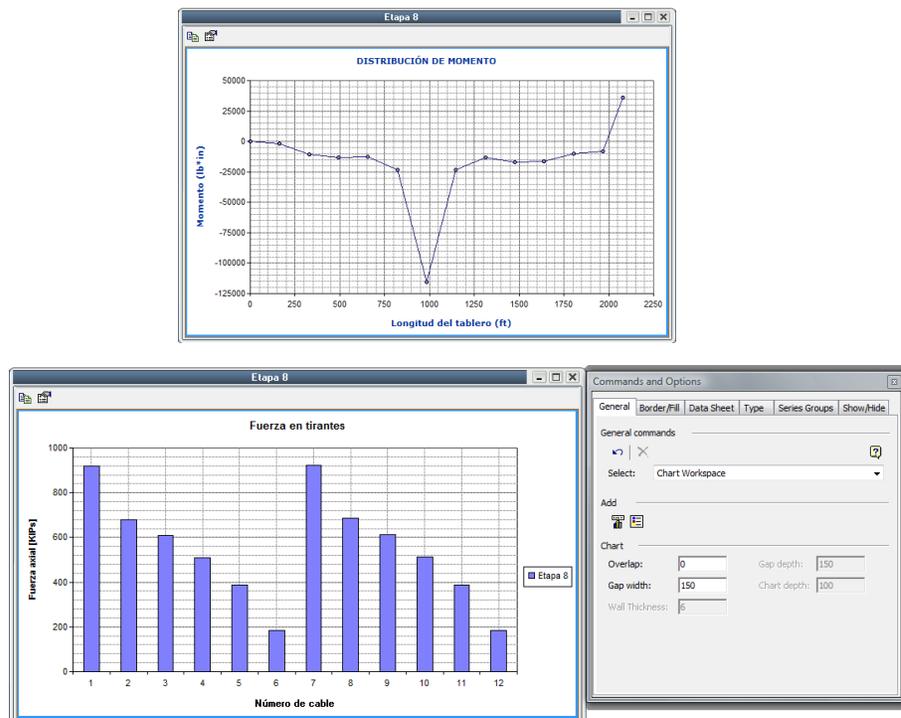


Figura 3.06.- Resultados en forma gráfica (ChartSpace).

## Capítulo 4

### EJEMPLO DE APLICACIÓN

Con este ejemplo se muestra el empleo del programa para realizar un análisis en el tiempo de un puente atirantado, en las diferentes etapas de construcción y para 10,000 días (27 años) después de completada su construcción. Se pretende mostrar primordialmente el uso y las características del programa más que el hecho de llegar a conclusiones determinadas acerca de la estructura del ejemplo, que requerirían un estudio más amplio y sistemático. Sin embargo, se hacen comentarios acerca de los resultados del análisis del ejemplo. La geometría del puente es propuesta por Seif et al., [26] y luego adaptada por Abbas para el programa NASCAB [1].

#### 4.1 DESCRIPCIÓN DEL MODELO

Las dimensiones generales del puente de este ejemplo se muestran en la Fig. 4.01. El puente consta de un claro central de 666 pies y dos claros laterales de 300 pies cada uno. Las torres tienen una altura de 270 pies, aproximadamente a la mitad de dicha altura se encuentra el nivel del tablero. Cada dovela o segmento del tablero es de 50 pies de largo por 69 pulgadas de ancho, de concreto presforzado. La dovela de cierre en medio del claro central tiene 66 pies de longitud. Cada dovela está soportada por un tirante, el cual a su vez está anclado a la parte superior de la torre. La configuración del sistema de atirantado es en forma de abanico simétrico.

Cada torre tiene forma de “H” y cada pierna tiene una sección rectangular hueca de concreto reforzado. El tablero está soportado en la torre por medio de un apoyo tipo rodillo que permite el movimiento solo en la dirección longitudinal, solamente restringida por los tirantes.

Debido a la simetría del puente, se ha modelado solo la mitad del mismo, con el fin de reducir el tiempo de análisis. En la Fig. 4.02 se muestra la geometría del modelo de análisis. Se observa el uso de un apoyo especial en el nodo 18 simulando la conexión con la otra mitad del puente. A continuación se muestran las propiedades de los elementos estructurales del modelo. Para mayor detalle ver archivo de entrada en el apéndice A.

#### Tablero

El tablero es de concreto con una resistencia a la compresión  $f'_c$  de 6000 psi y tiene una deformación última al aplastamiento de 0.0035. El acero de refuerzo utilizado en el tablero tiene una resistencia  $f_y$  de 60,000 psi y una deformación última  $\epsilon_{su}$  de 0.03.

La sección transversal del tablero se muestra en la Fig. 4.03 (a). Para el análisis, la sección transversal se puede considerar como una sección “I” equivalente y se discretiza en ocho capas de concreto y tres capas de acero de refuerzo como se muestra en la Fig. 4.03 (b y c). El número de capas de concreto seleccionadas son las suficientes para calcular las propiedades de sección con diferencias despreciables respecto de la sección sin discretizar (en área menor del 1% y en inercia alrededor del 1.15%) y la cantidad de capas de acero se seleccionó de acuerdo con la presencia de concentraciones de acero importantes. Los detalles de las capas de concreto y acero de refuerzo, así como las distancias de dichas capas al centroide de la sección se muestran en la Fig. 4.03 (d).

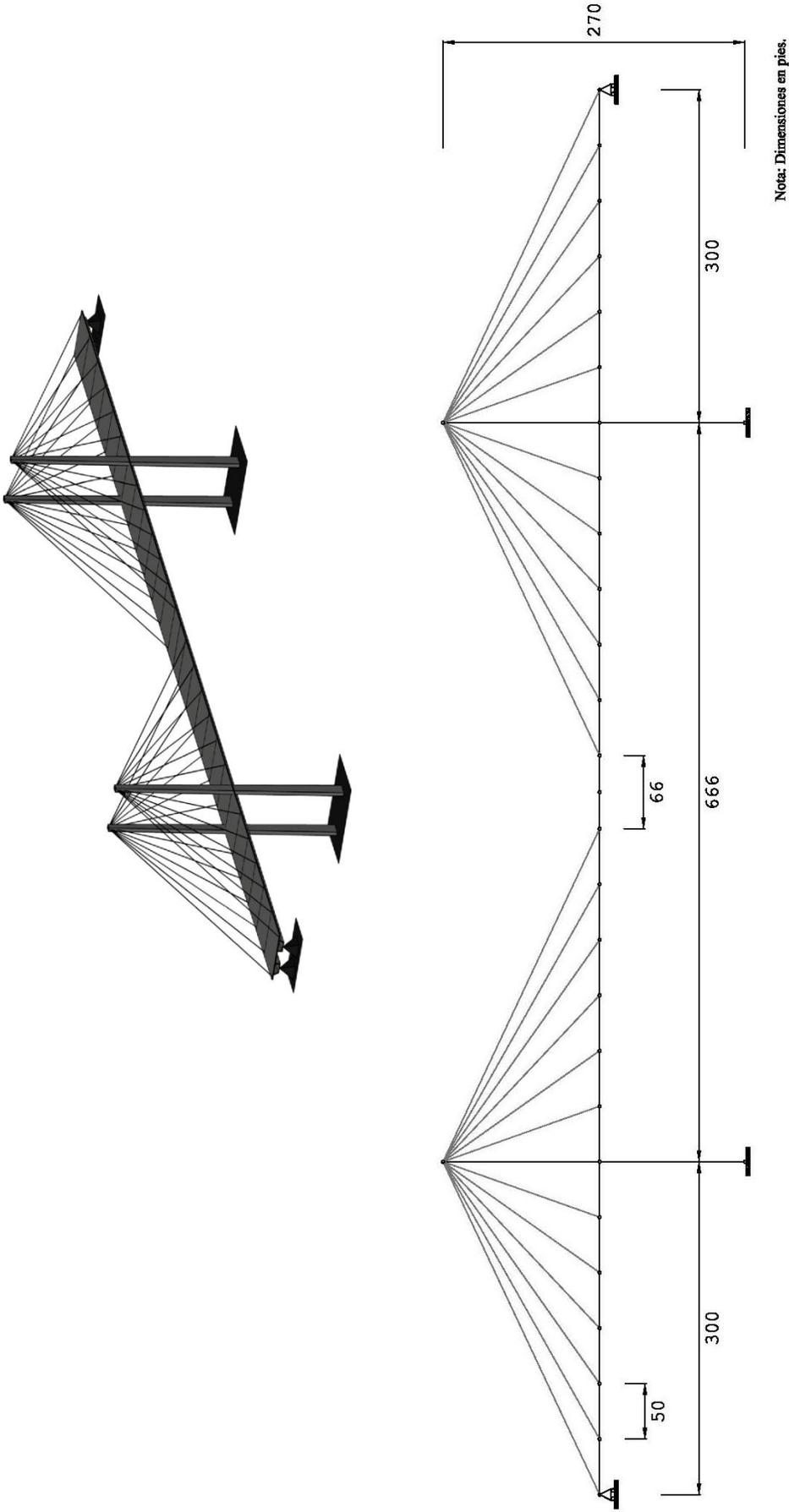
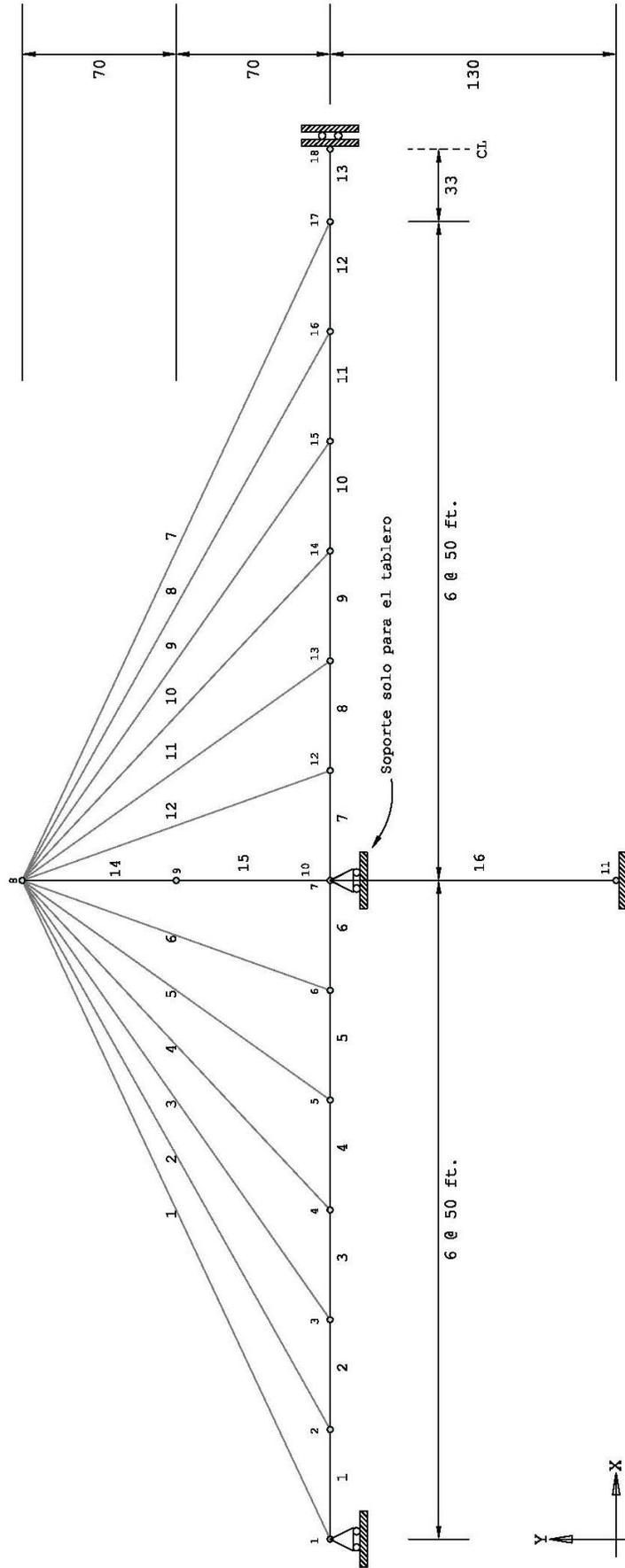
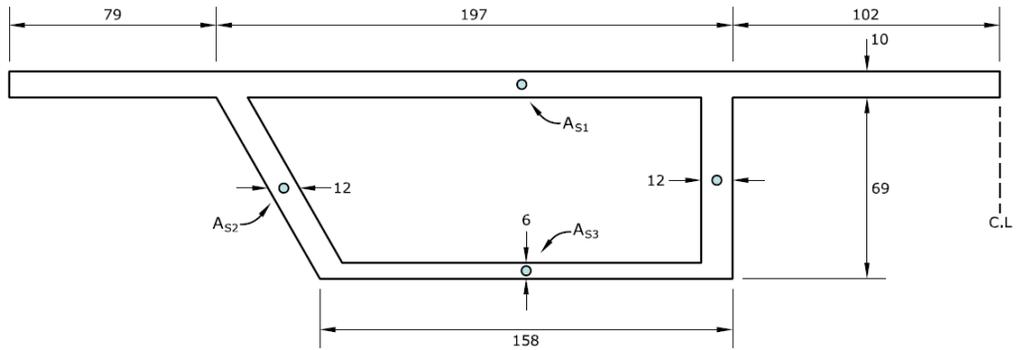


Figura 4.01. - Dimensiones generales del puente.



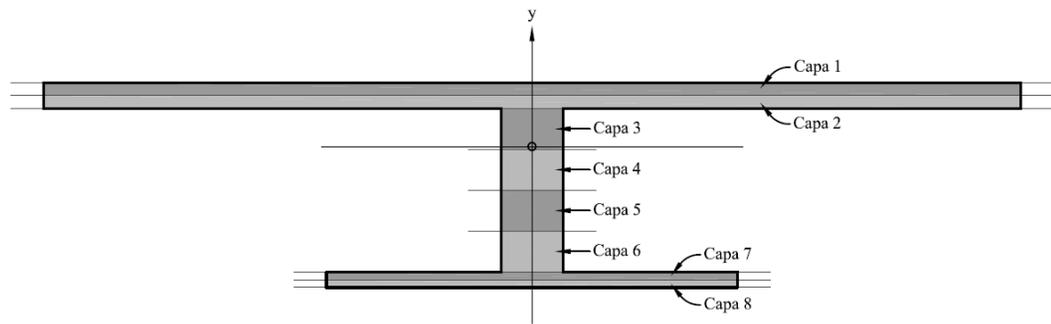
Nota: Dimensiones en pies.

Figura 4.02.- Modelo de análisis.

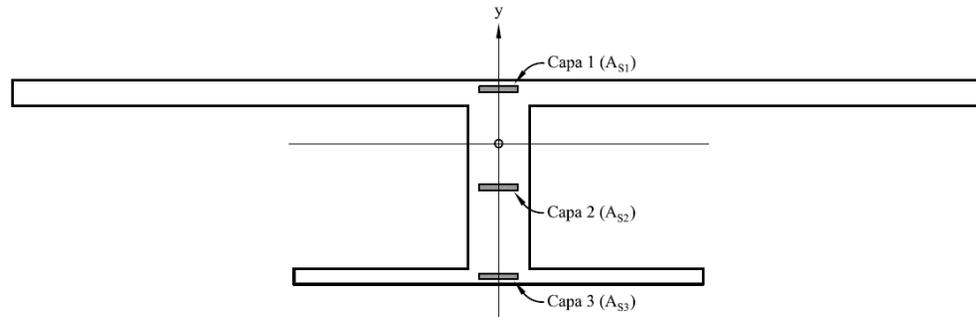


Nota: Dimensiones en pulgadas.

(a) Media Sección transversal del tablero.



(b) Capas de concreto.



(c) Capas de acero.

Capas de concreto		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	1890	22.131
2	1890	17.131
3	378	6.756
4	378	-8.994
5	378	-24.744
6	378	-40.494
7	1890	-49.869
8	1890	-52.869

Capas de acero de refuerzo		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	23.25	21.631
2	7.75	-16.869
3	23.25	-51.369

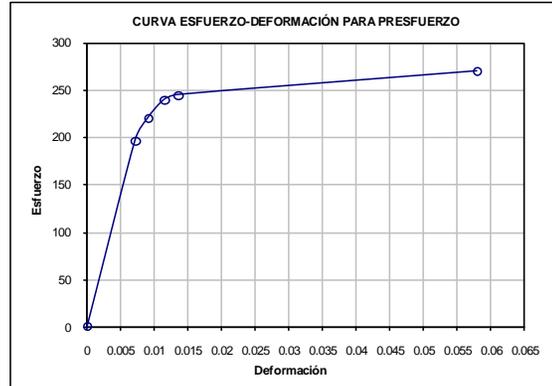
(d) Detalle de capas.

Figura 4.03.- Sección transversal del tablero.

### Tendones de presfuerzo

El acero de presfuerzo utilizado en el tablero tiene una curva esfuerzo-deformación multilínea como sigue:

Punto	Esfuerzo (kip/pulg <sup>2</sup> )	Deformación (%)
1	196.6	0.00715
2	220.0	0.00900
3	240.0	0.01150
4	245.0	0.01350
5	270.0	0.05800



Los tendones de presfuerzo son postensados y tienen un perfil lineal que coincide con el centroide de la sección de concreto, Fig. 4.04. En la siguiente tabla se muestra la geometría y los valores de tensado inicial para cada tendón.

Tendón No.	Nodos	Área (pulg <sup>2</sup> )	Fuerza (kips)
1	1-18	12.5	1500
2	14-18	12.5	1500
3	16-18	12.5	1500

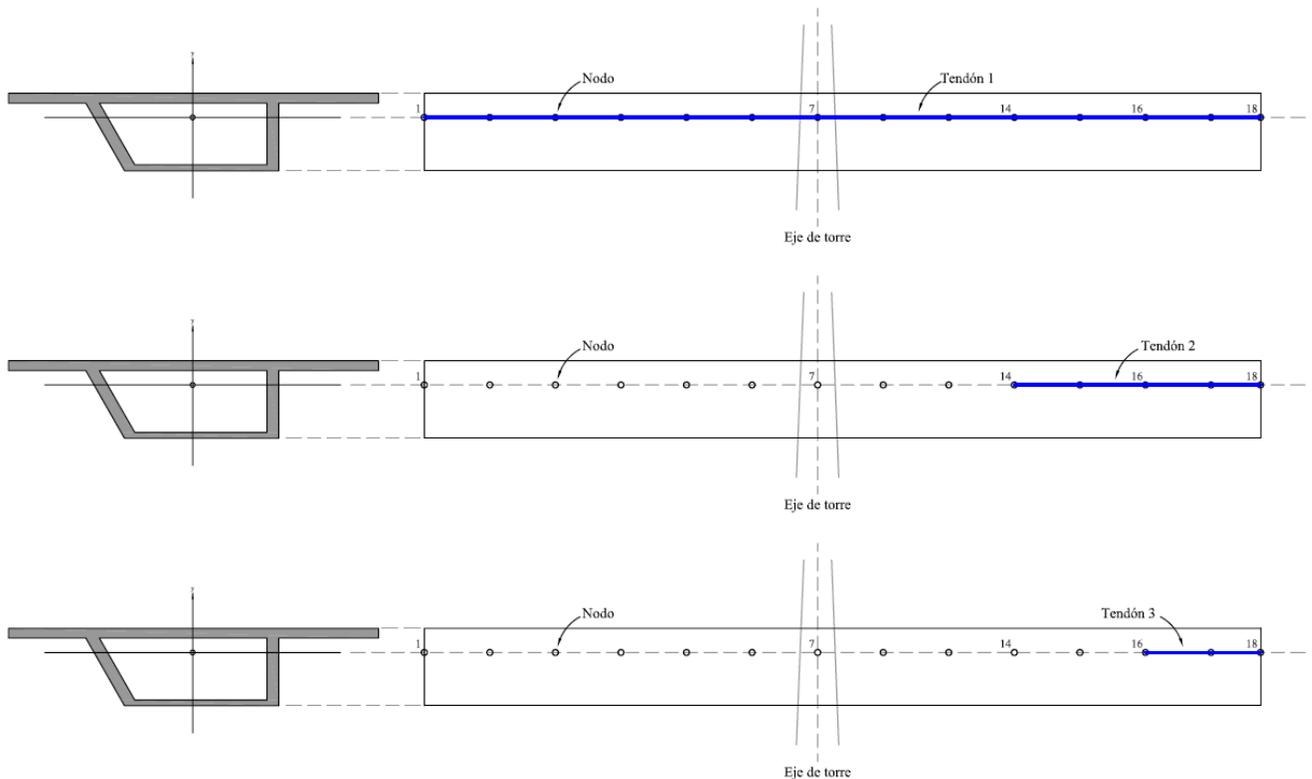
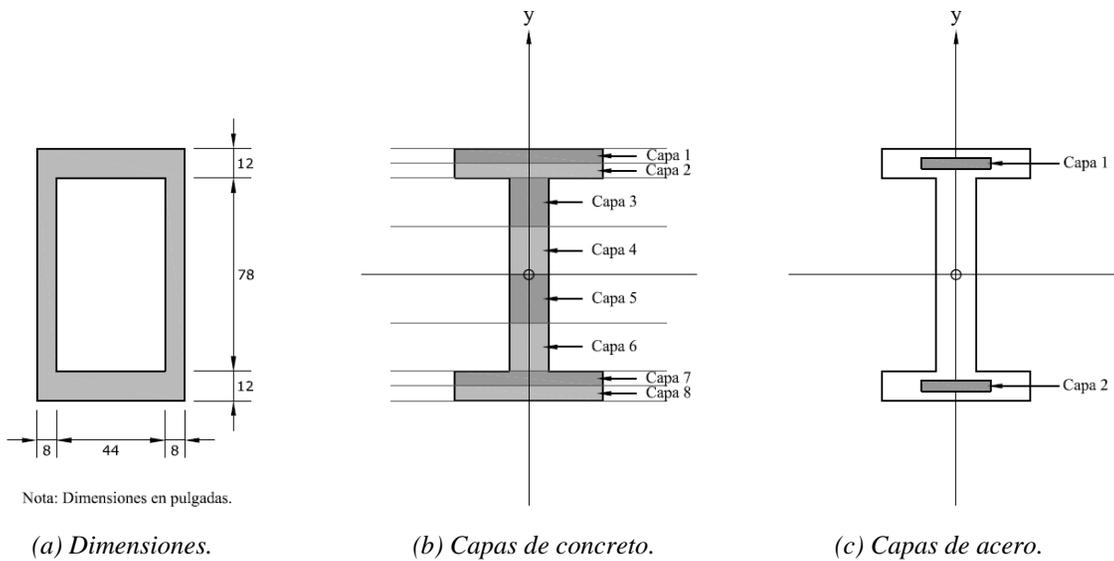


Figura 4.04.- Perfil de los tendones de presfuerzo.

## Torre

El concreto y el acero de refuerzo para las torres son del mismo material utilizados en el tablero. Para modelar la torre se utilizan tres secciones de tipo cajón. Las secciones transversales, al igual que la sección del tablero, se pueden considerar como secciones tipo “I” y también se discretizan en ocho capas de concreto y dos capas de acero de refuerzo. Al igual que la sección del tablero, el número de capas de concreto utilizadas son las suficientes para determinar el área de la sección y su inercia sin diferencias importantes respecto de la sección original. En la Fig. 4.05 se muestran las dimensiones de la sección, las capas de concreto, las capas de acero de refuerzo y detalles de las capas para la parte superior de la torre (elemento 14). En la Fig. 4.06 se muestra lo anterior pero para la parte media de la torre (elemento 15) y en la Fig. 4.07 para la parte inferior de la torre (elemento 16).



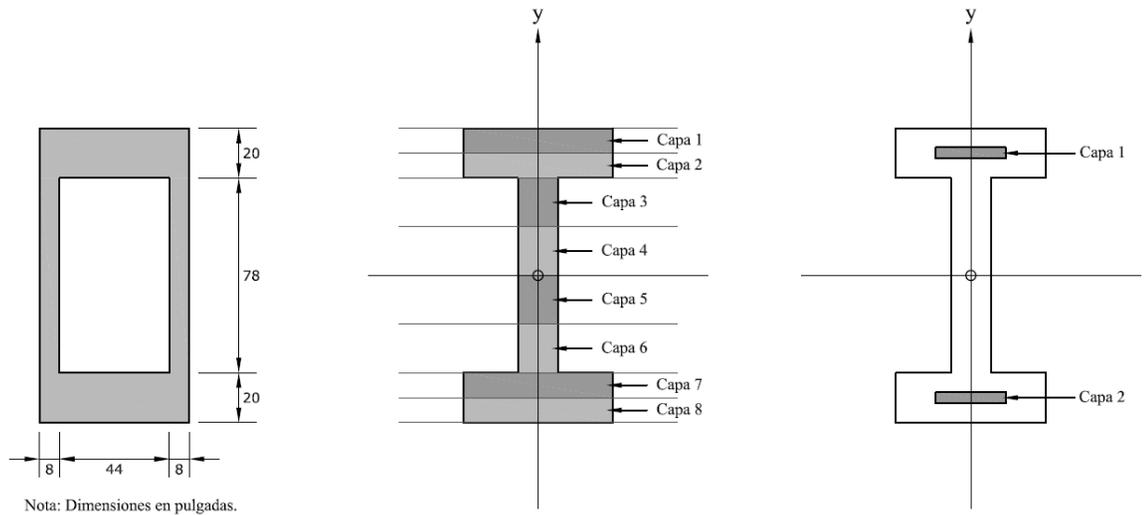
Nota: Dimensiones en pulgadas.

Capas de concreto		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	360	48
2	360	42
3	312	29.25
4	312	9.75
5	312	-9.75
6	312	-29.25
7	360	-42
8	360	-48

Capas de acero de refuerzo		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	20	45
2	20	-45

(d) Detalle de capas.

Figura 4.05.- Sección transversal del elemento 14.



(a) Dimensiones.

(b) Capas de concreto.

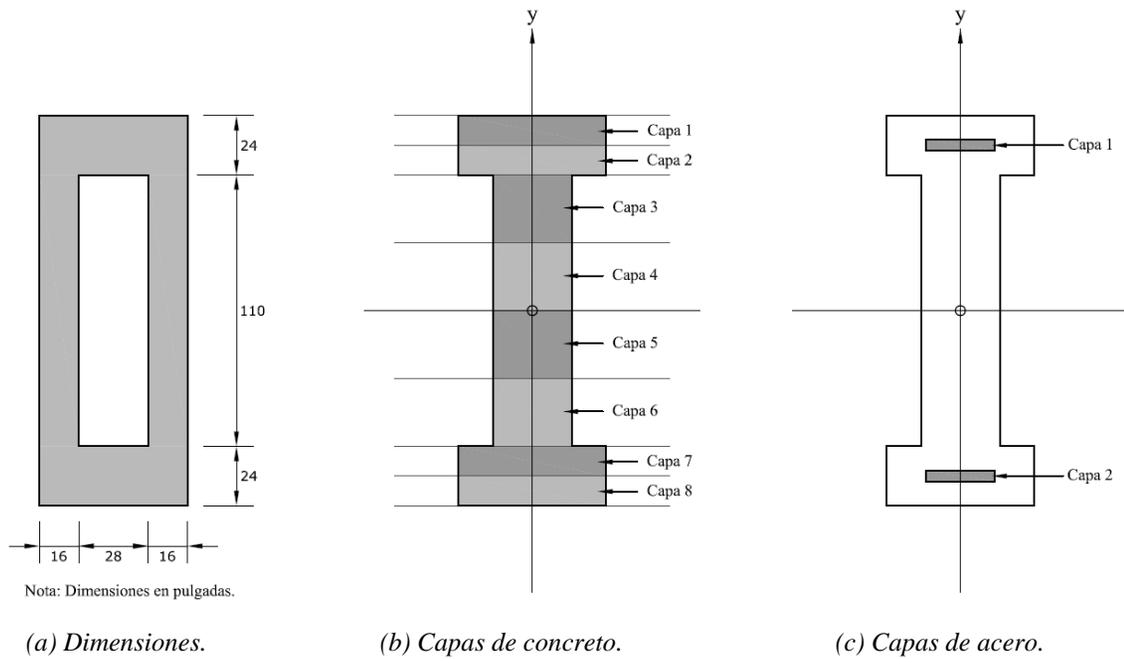
(c) Capas de acero.

Capas de concreto		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	600	54
2	600	44
3	312	29.25
4	312	9.75
5	312	-9.75
6	312	-29.25
7	600	-44
8	600	-54

Capas de acero de refuerzo		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	20	49
2	20	-49

(d) Detalle de capas.

Figura 4.06.- Sección transversal del elemento 15.



Capas de concreto		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	720	73
2	720	61
3	880	41.25
4	880	13.75
5	880	-13.75
6	880	-41.25
7	720	-61
8	720	-73

Capas de acero de refuerzo		
Capa	Área (pulg <sup>2</sup> )	Distancia al centroide (pulg)
1	32	67
2	32	-67

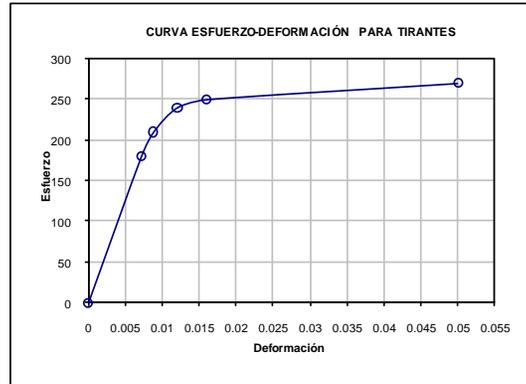
(d) Detalle de capas.

Figura 4.07.- Sección transversal del elemento 16.

### Sistema de atirantado

La configuración del sistema de atirantado es en forma de abanico simétrico tal y como se muestra en la Fig. 4.02. Los tirantes 1, 2, 7 y 8 tienen un área transversal de 15.5 in<sup>2</sup> y los tirantes 3, 4, 5, 6, 9, 10, 11 y 12 tienen un área de 7.75 in<sup>2</sup>. El acero empleado en los tirantes se modela mediante una curva esfuerzo-deformación multilínea con los siguientes valores:

Punto	Esfuerzo (kip/in <sup>2</sup> )	Deformación (%)
1	180	0.00715
2	210	0.00880
3	240	0.01200
4	250	0.01600
5	270	0.05000



### 4.2 ANÁLISIS DE LA SECUENCIA CONSTRUCTIVA EN CANTILIVER

La construcción del puente usando el método de construcción en cantiliver es completada en 135 días. Se supone que todos los elementos son prefabricados. En la Fig. 4.08 se muestran las configuraciones para las etapas constructivas consideradas. En la siguiente tabla se muestra el programa constructivo, los valores entre paréntesis son los valores de tensado inicial en kips.

Día Núm.	Elemento colado	Elemento construido	Cable instalado	Cable retensado	Tendón tensado
0	14,15,16				
15	6,7				
30	5,8	14,15,16			
45	4,9	6,7	6,12 (179)		
60	3,10	5,8	5,11 (207)	6,12 (357)	
75	2,11	4,9	4,10 (246)	5,11 (414)	
90	1,12	3,10	3,9 (293)	4,10 (492)	
105	13	2,11	2,8 (344)	3,9 (586)	
120		1,12	1,7 (397)	2,8 (687)	
135		13		1,7 (975)	1,2,3 (1500)

Cada configuración de la estructura es analizada instantáneamente en el día en que los nuevos elementos son agregados y luego dos semanas después, justo antes que nuevos elementos sean agregados.

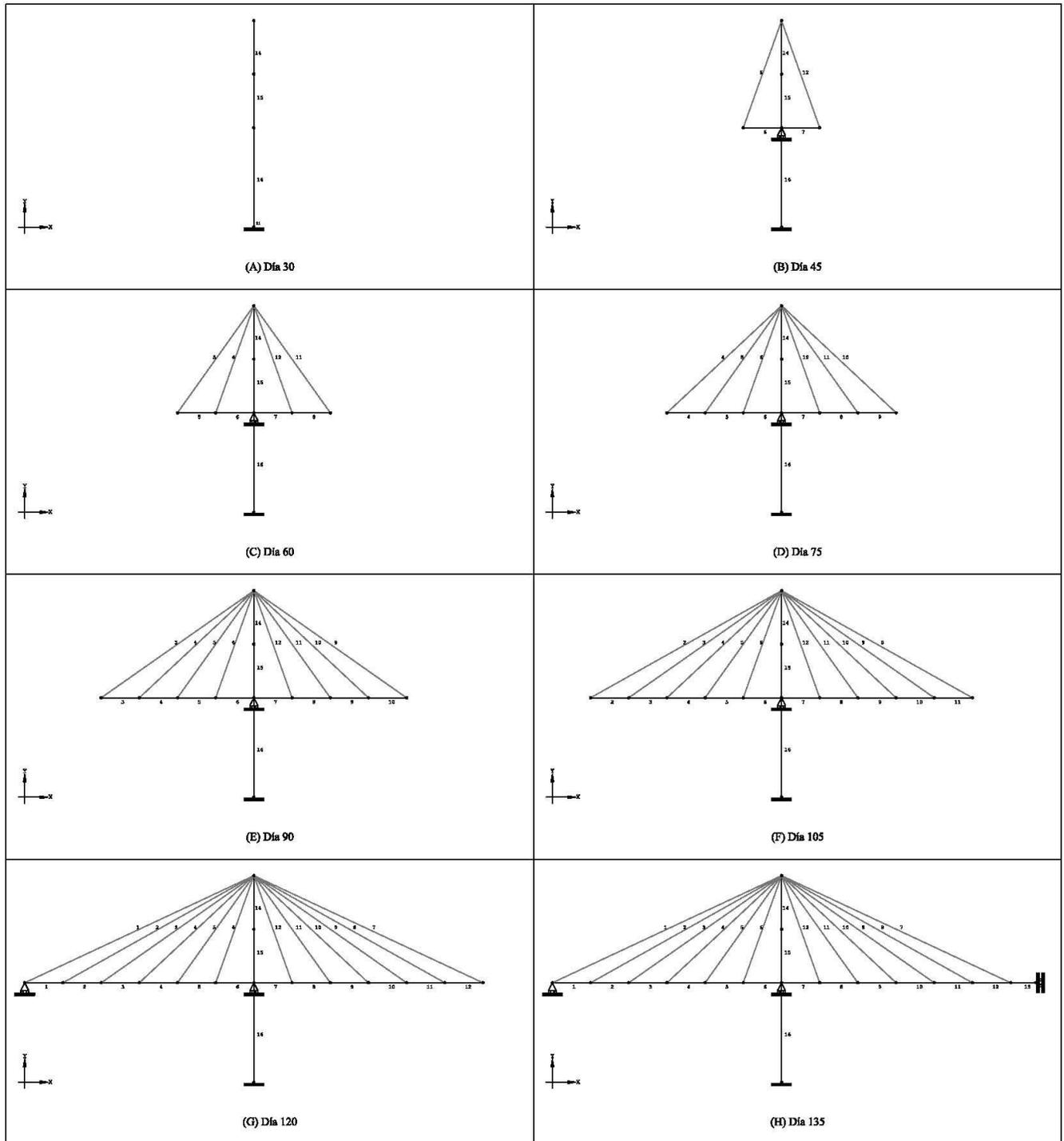


Figura 4.08.- Configuración de la estructura para cada etapa constructiva.

### 4.3 USO DEL PROGRAMA

Al abrir el programa, se mostrará la ventana principal desde la cual se ejecutan todas las tareas del programa. Fig. 4.09. Se puede maximizar la ventana principal al tamaño de la pantalla para tener un área de trabajo mayor.

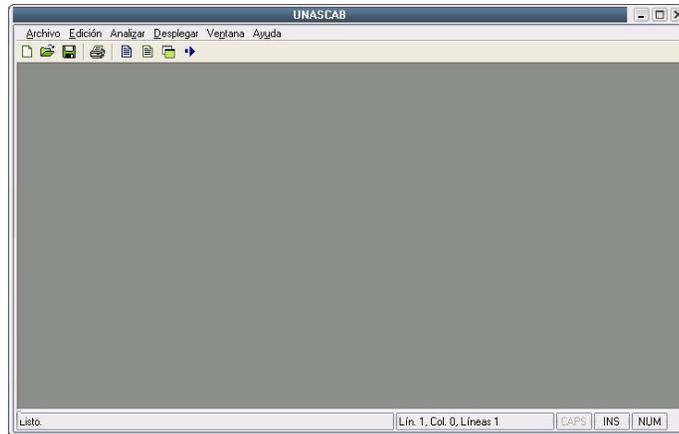


Figura 4.09.-Ventana principal.

### Creación del archivo de entrada

Para analizar un puente, primero se tiene que crear el archivo de definición de comandos o archivo de entrada, en el cual se definen todas las características de la estructura y sus etapas de análisis. En el programa se selecciona el menú “Archivo\Nuevo” para abrir el editor interno con el que se puede generar el archivo, Fig. 4.10. El editor cuenta con las funciones básicas de edición de texto (copiar, cortar, pegar y búsqueda) y de inserción de comandos. En el apéndice A se muestra el archivo de entrada del ejemplo.

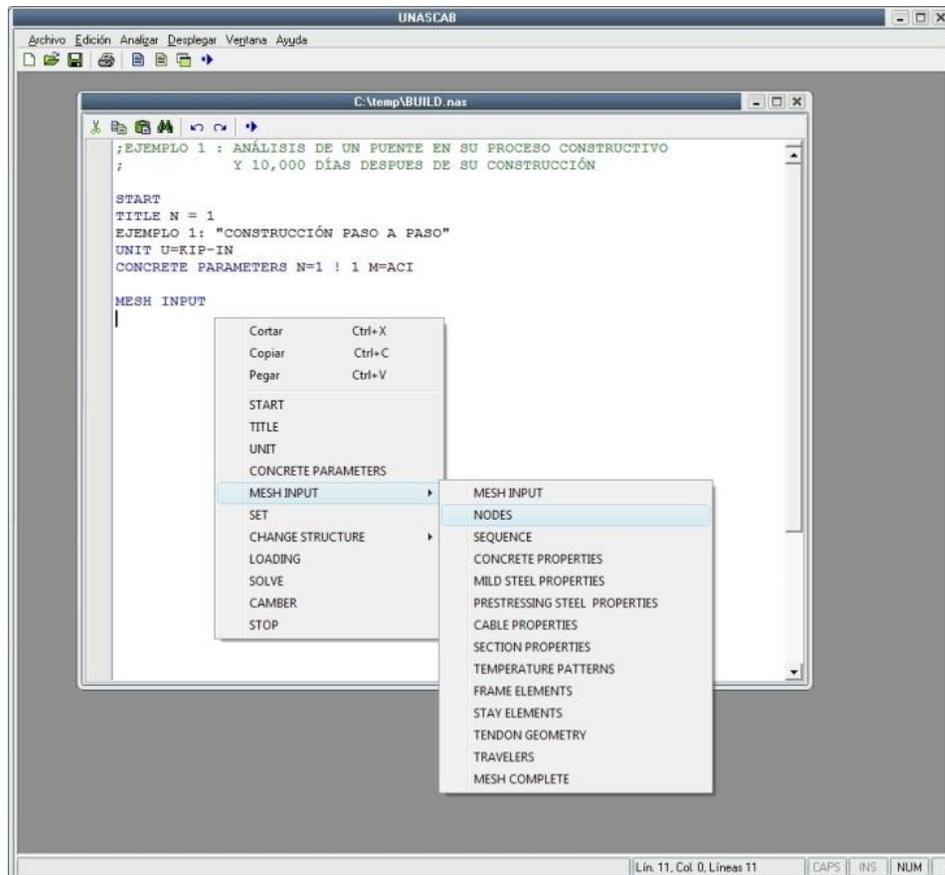
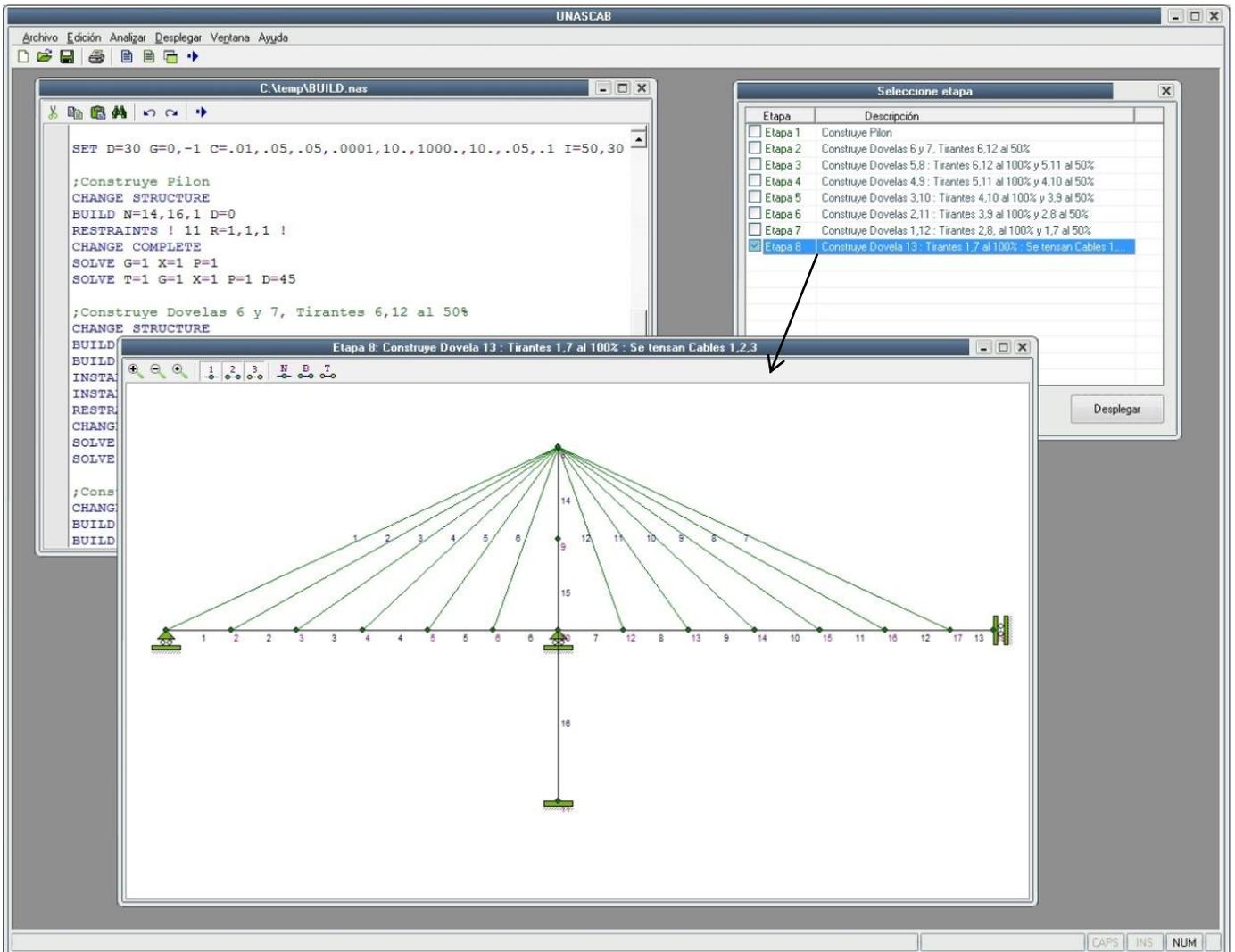


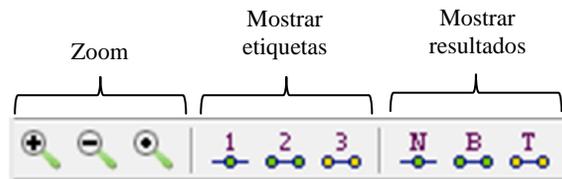
Figura 4.10.- Editor de archivo de entrada.

## Graficado de la configuración de la estructura

Una vez generado el archivo de entrada, se puede graficar la configuración geométrica de la estructura para alguna etapa deseada. Se considera que cada comando *Change Structure* que se utilice en el archivo de entrada define una etapa de análisis. Mediante el menú “Desplegar/Etapas” se muestra una ventana que muestra todas las etapas consideradas en el archivo de entrada, también permite seleccionar la etapa a visualizar, Fig. 4.11 (a). El graficador 2D posee herramientas de *zoom*, activación de etiquetas de identificación para los nodos, para las barras y para los tirantes y activación de ventanas de resultados para nodos, para barras y para tirantes, Fig. 4.11 (b).



(a) Graficador 2D y ventana de selección de etapas.



(b) Barra de herramientas del graficador 2D.

Figura 4.11.- Visualización de la estructura.

## Ejecución del análisis

Mediante el menú “Analizar/Iniciar análisis” o con el botón  de la barra de herramientas del editor se puede ejecutar el análisis de la estructura. Al seleccionar esta opción se activa una ventana (ventana monitor) que muestra el avance del análisis (Fig. 4.12).

Debido a la gran cantidad de datos que podría generar un análisis de este tipo, el programa utiliza una serie de archivos adicionales al de entrada (base de datos) para guardar los resultados; por esta razón, si se ejecuta el análisis de una estructura previamente analizada, el programa pregunta si se desea re-calcular la estructura, al elegir la opción *Aceptar*, el programa borra la base de datos de resultados del análisis anterior (Fig. 4.13).

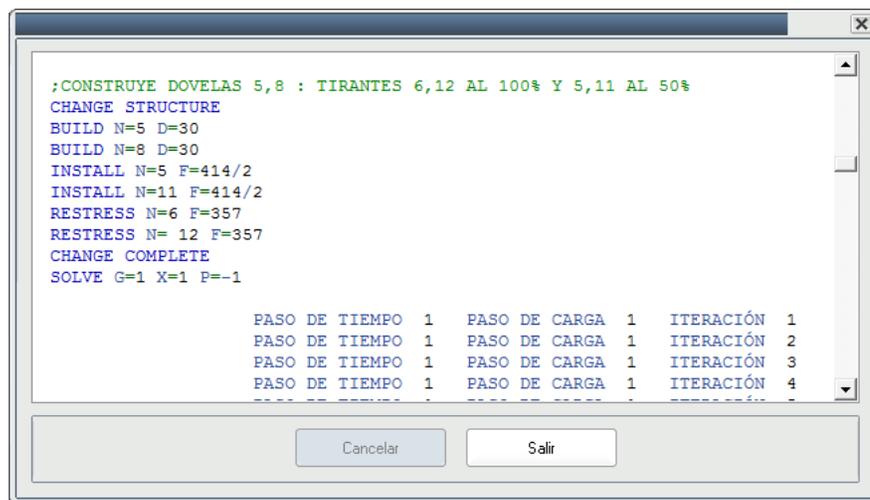


Figura 4.12.- Proceso de cálculo (ventana monitor).

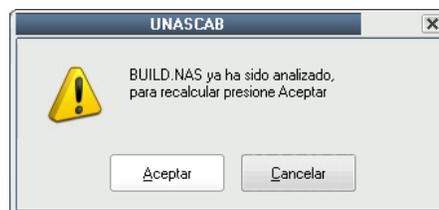


Figura 4.13.- Alerta para reiniciar el cálculo.

## Reporte general de resultados

Conforme el programa resuelve la estructura, también genera un reporte general del análisis. Este reporte se guarda en la misma dirección que el archivo de entrada y tiene la extensión *OUT*. Al terminar el análisis, el programa muestra este reporte en una ventana secundaria de manera automática, aunque también con el menú “Desplegar/Archivo de salida” se puede mostrar el reporte (Fig. 4.14).

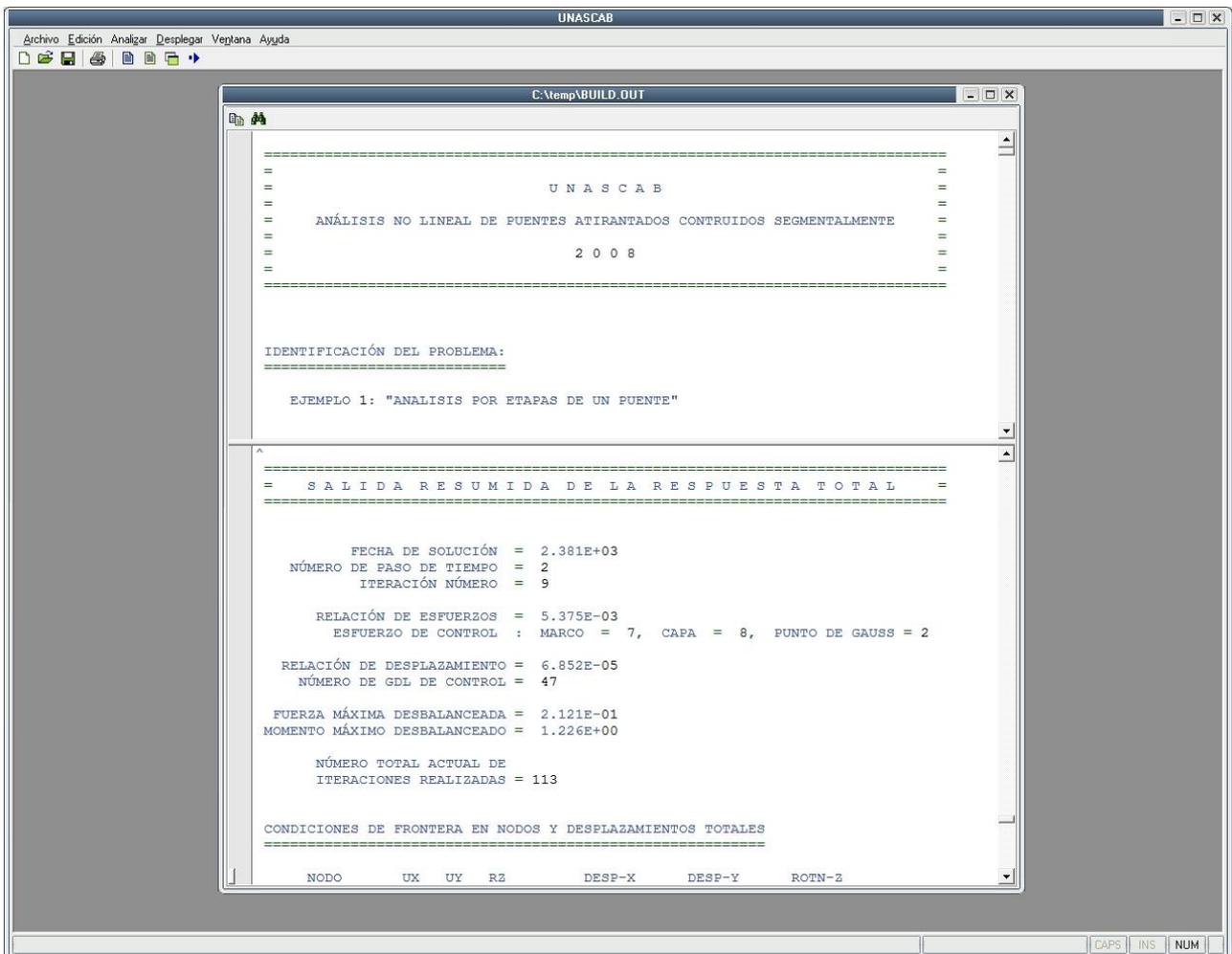


Figura 4.14.- Reporte general del análisis.

## Despliegue de resultados

El graficador 2D muestra la configuración de la estructura para una etapa en particular, pero también desde éste se puede desplegar los resultados de esa etapa en forma tabular. La figura 4.15 muestra los resultados de los nodos para la etapa 5 del ejemplo. La figura 4.16 muestra resultados para las barras de la etapa 5 y la Fig. 4.17 las tensiones en los tirantes para la misma etapa.

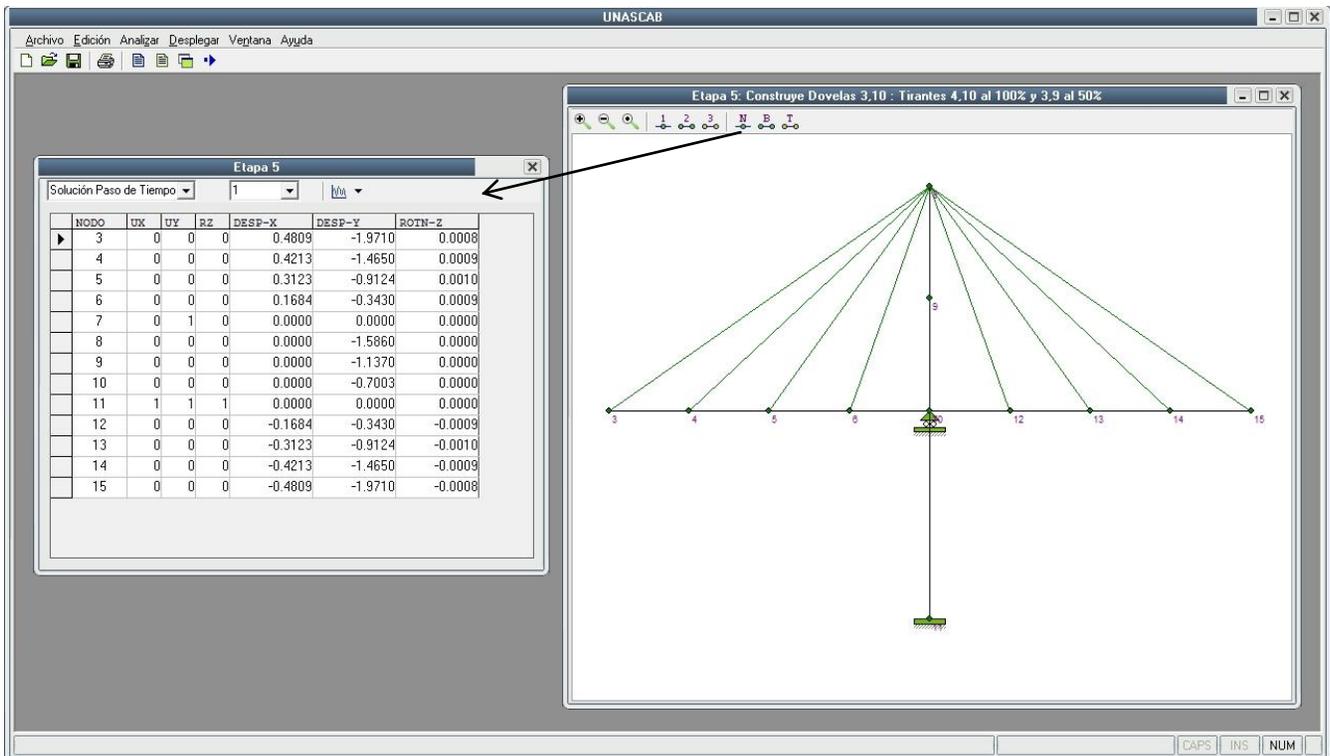


Figura 4.15.- Resultados de nodos (etapa 5).

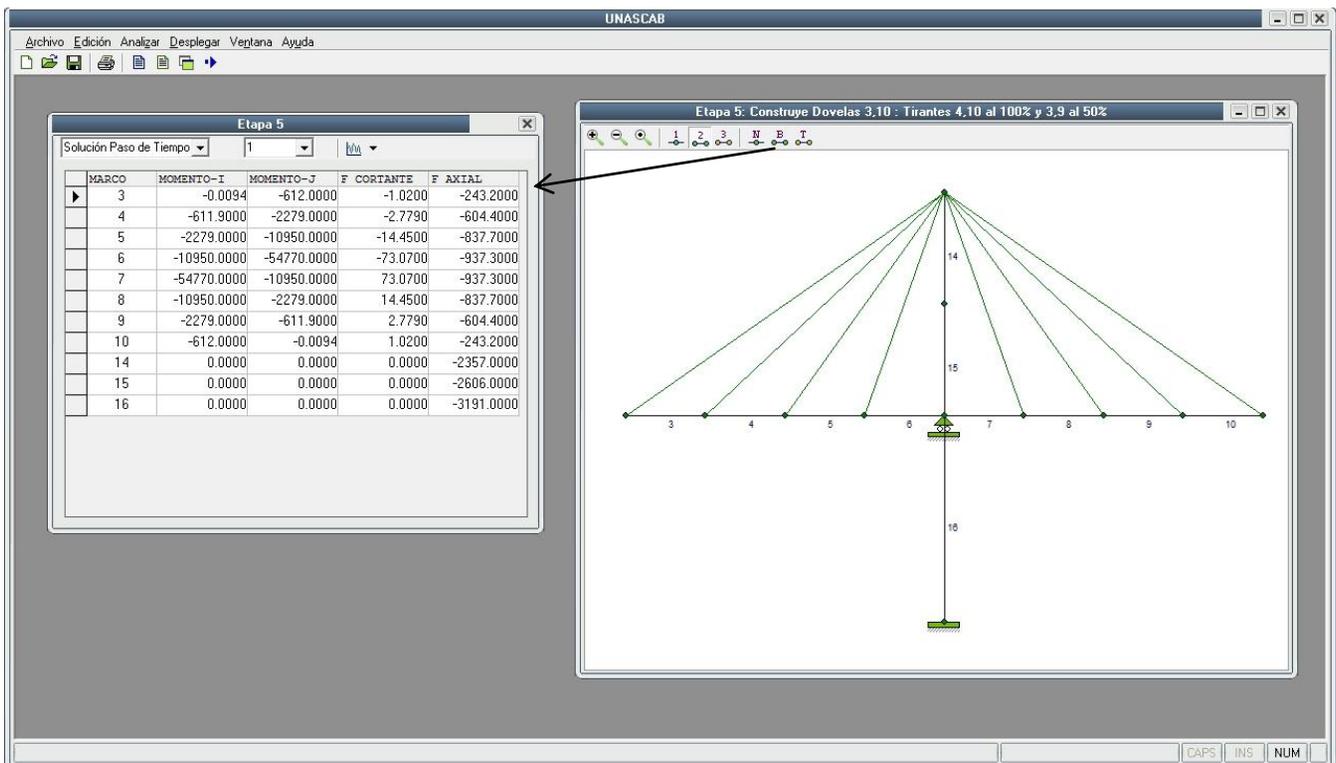


Figura 4.16.- Resultados de barras (etapa 5).

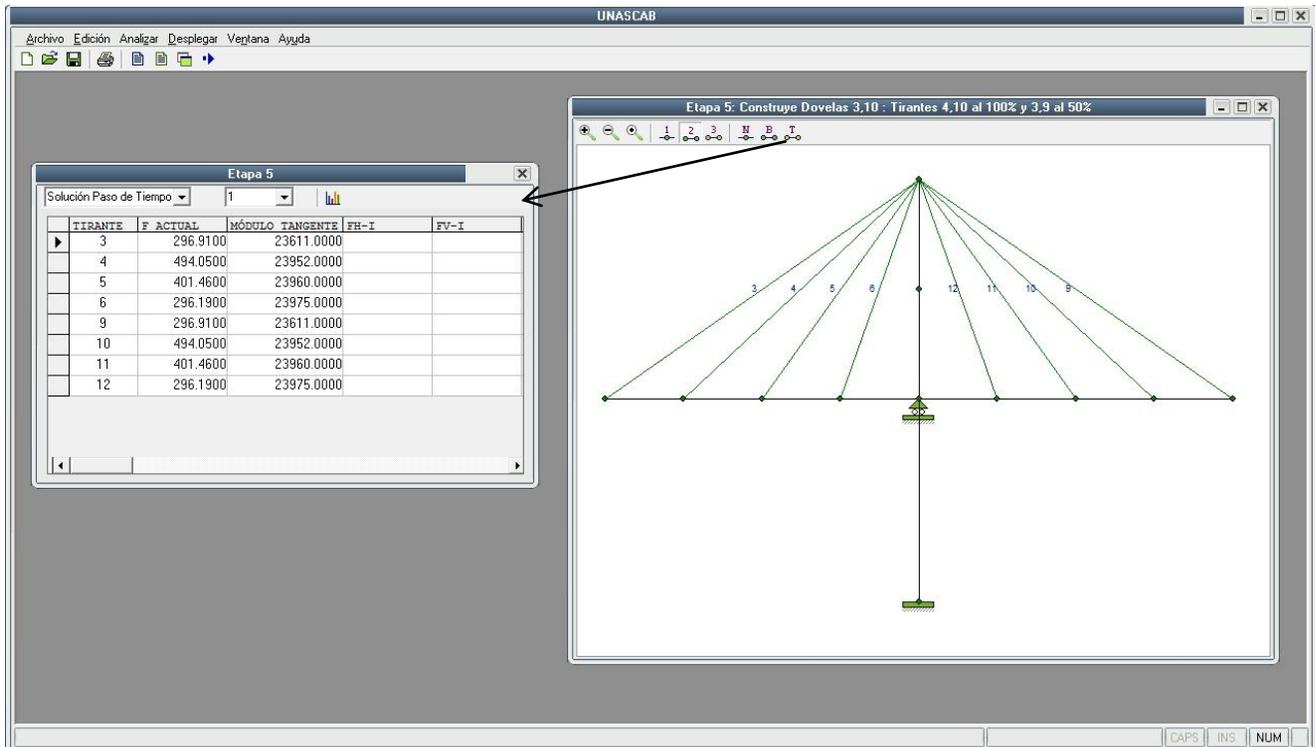


Figura 4.17.- Resultados de tirantes (etapa 5).

En el caso de los elementos barra, el programa tiene la capacidad de desplegar los resultados por grupos de elementos barra. Con el menú “Desplegar/Grupos” o mediante la opción “Seleccionar grupo” de la ventana de resultados se muestra un cuadro de dialogo donde se puede seleccionar, editar, eliminar o crear nuevos grupos (Fig. 4.18). Para definir un grupo, solo se asigna un nombre de grupo y las etiquetas de los elementos barra que conforman el grupo. Se puede utilizar la siguiente nomenclatura para definir las etiquetas de los elementos.

Ejemplo de elementos a incluir en grupo	Nomenclatura
1, 2, 3, 4, y 5	1-5
1, 3 y 5	1,3,5
1, 2, 3, 4, 5, 8, 9 y 10	1-5,8-10
1, 2, 3, 4, 5, 8, 11 y 15	1-5,8,11,15

En la Fig. 4.18 se muestra la definición de un grupo llamado “Tablero” y las barras que lo conforman son: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 y 13 (o 1-13, de acuerdo a la nomenclatura). Estas barras son las que conforman el tablero completo del modelo de análisis (Fig. 4.02). Cabe mencionar que en la definición del grupo se pueden incluir barras que en la etapa no han sido construidas, esto tiene la ventaja de definir grupos que sirvan para cualquier etapa. El programa detecta automáticamente los elementos que están activos en la etapa actual. En la Fig. 4.18 se muestra el grupo “Tablero” con las barras antes mencionadas, sin embargo para la etapa mostrada (etapa 5) solo las barras 3, 4, 5, 6, 7, 8, 9 y 10, de ese grupo, están activas; por lo que el diagrama de momento de la Fig. 4.18 corresponde solo a esas barras.

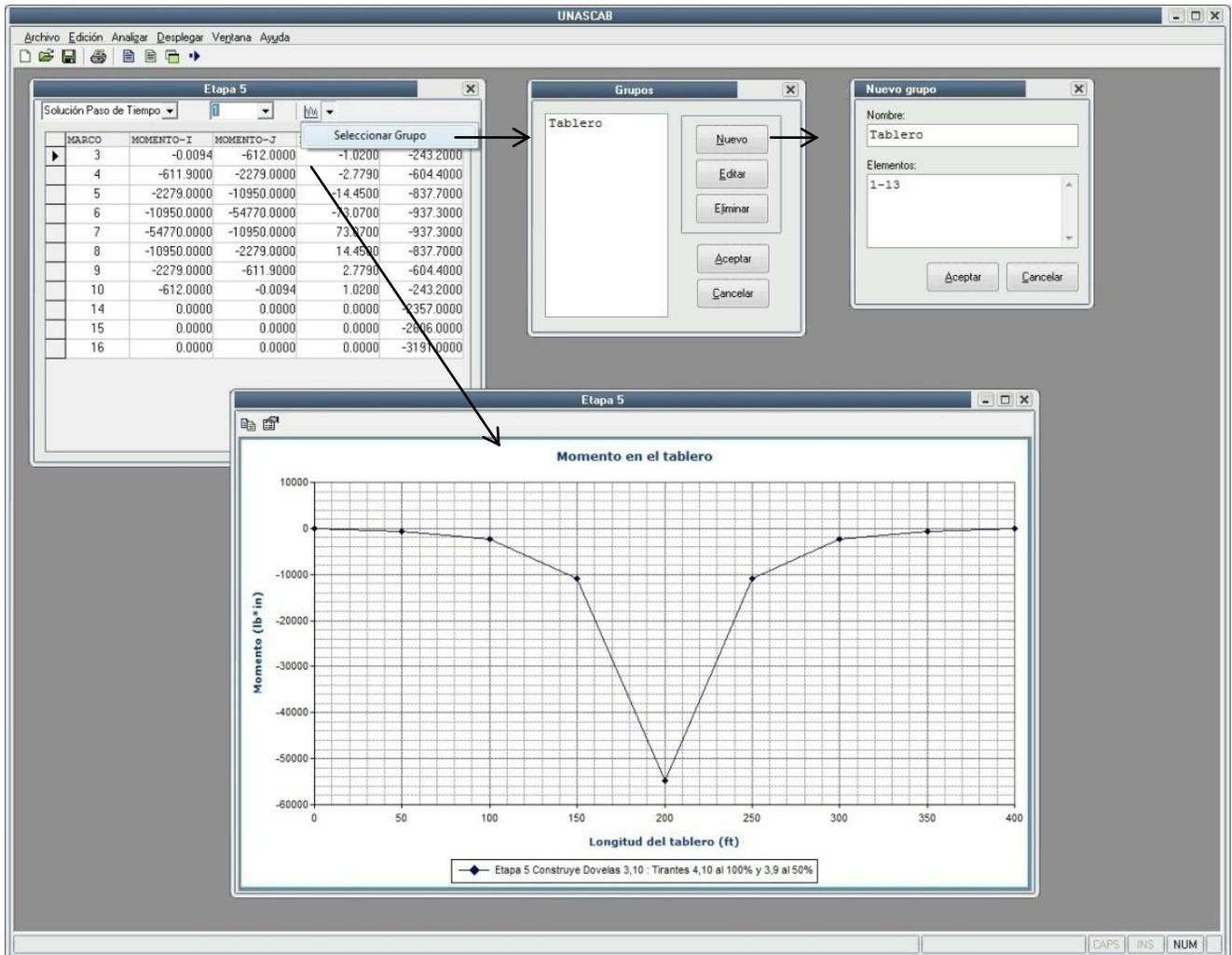


Figura 4.18.- Diagrama de momento por grupo.

Los grupos definidos para barras también sirven para mostrar gráficas de desplazamiento de los nodos, en la Fig. 4.19 se muestra una gráfica de desplazamiento vertical de los nodos del tablero para la etapa 5 del ejemplo.

En el caso de los tirantes, el programa detecta automáticamente los tirantes instalados en la etapa activa. Directamente de la ventana de resultados se despliega un gráfico de barras que muestra la tensión de los tirantes (Fig. 4.20).

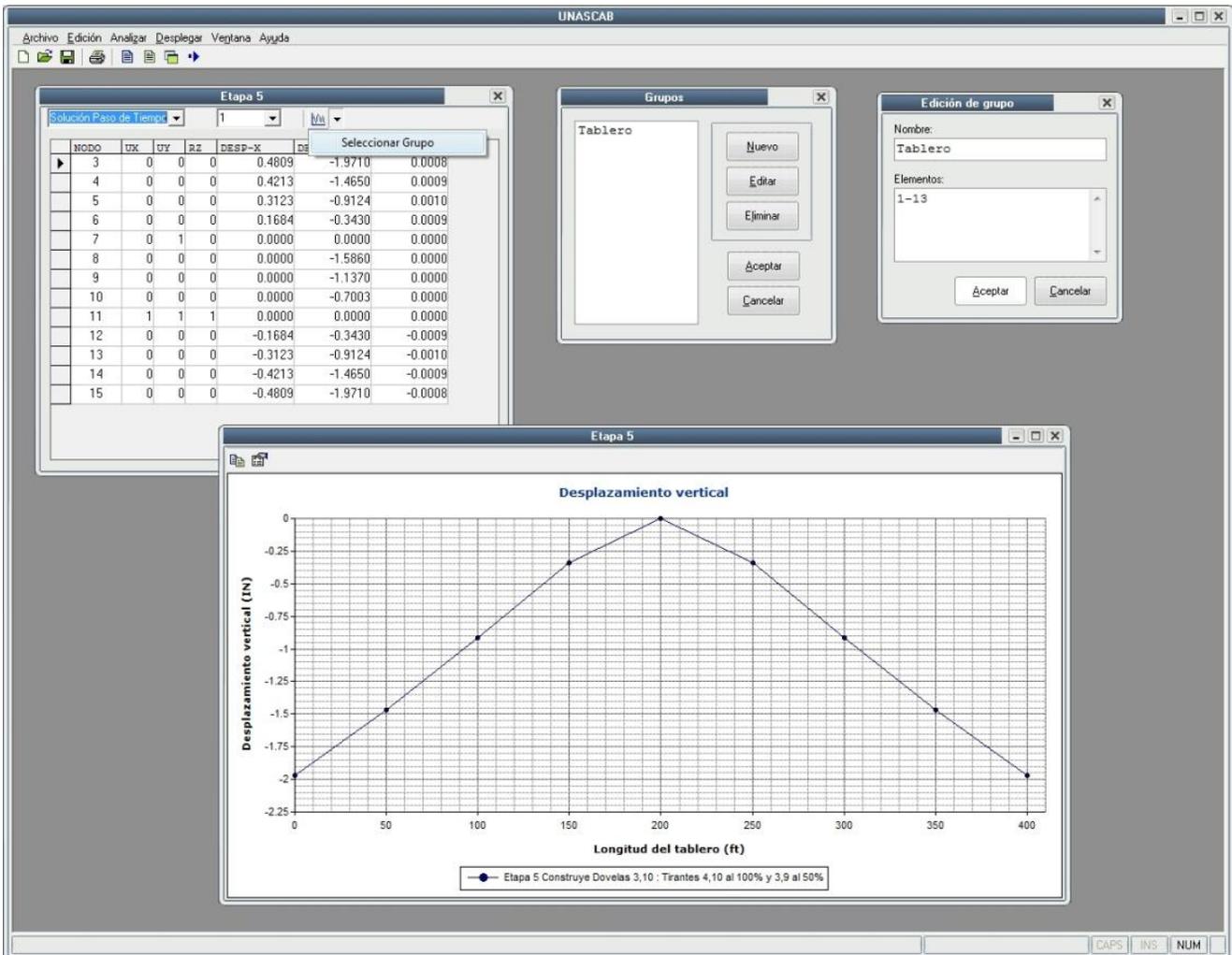


Figura 4.19.- Diagrama de desplazamiento vertical de nodos por grupo.

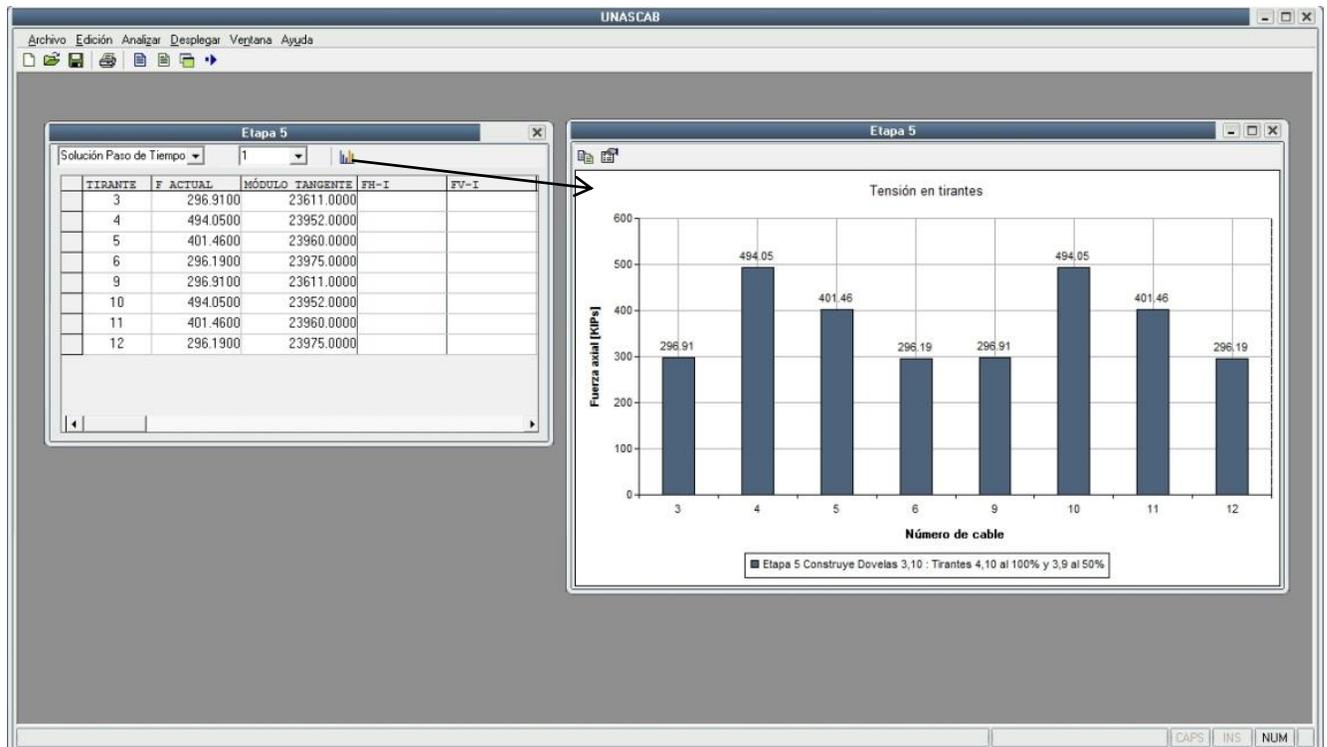


Figura 4.20.- Tensión en tirantes.

Todas las ventanas de resultados (para nodos, para barras y para tirantes) tienen las opciones de selección de tipo de solución y número de paso de tiempo o número de iteración para la etapa que se está visualizando, los tipos de solución así como la cantidad de pasos de tiempo corresponden a las definidas en el archivo de entrada con el uso de los parámetros del comando *Solve*.

Para la etapa 8 del ejemplo, se utilizan dos tipos de solución: una solución instantánea y una solución de paso de tiempo con 3 pasos de tiempo (ver archivo de entrada en apéndice A). En la Fig. 4.21 se muestra la ventana de resultados de los tirantes para esta etapa. En la Fig. 4.21 (a) se muestra la sintaxis de los comandos de solución en el archivo de entrada, en la Fig. 4.21 (b) se muestra la selección del tipo de solución que se desea desplegar y la Fig. 4.21 (c) muestra la selección del paso de tiempo para la solución paso de tiempo.

```
SOLVE G=1 X=1 P=1
SOLVE T=1 G=1 X=1 S=3 P=1 D=10000
```

(a) Comandos de solución para la etapa 8 del ejemplo.

	L	MÓDULO	TANGENTE	FH-I	FV-I
▶	1	957.1300	23789.0000		
	2	682.5100	23600.0000		
	3	600.3200	23952.0000		
	4	496.9300	23953.0000		
	5	393.7800	23958.0000		
	6	253.9700	23961.0000		
	7	962.3200	23793.0000		
	8	685.0100	23604.0000		
	9	599.5000	23952.0000		
	10	495.1200	23952.0000		
	11	390.8600	23957.0000		

(b) Selección del tipo de solución.

	TIRANTE	F ACTUAL	MÓDULO	TANGENTE	FH-I	FV-I
▶	1	929.4300	929.4300	23601.0000		
	2	682.9400	682.9400	23954.0000		
	3	609.0200	609.0200	23956.0000		
	4	507.8600	507.8600	23957.0000		
	5	391.7700	391.7700	23933.0000		
	6	212.2900	212.2900	23775.0000		
	7	935.7100	935.7100	23611.0000		
	8	688.9900	688.9900	23954.0000		
	9	609.5900	609.5900	23956.0000		
	10	507.5800	507.5800	23956.0000		
	11	389.0500	389.0500	23956.0000		

(c) Selección del paso de tiempo para la solución paso de tiempo.

Figura 4.21.- Opciones de despliegado de la ventana de resultados.

### Otras opciones

Al seleccionar con el puntero un elemento barra o un tirante en el graficador 2D, éste cambia a modo de resalte pero también se resaltarán de manera automática su ubicación dentro de la ventana de resultados (Fig. 4.22).

El gráfico donde se muestra los resultados por grupo de elementos o las tensiones en los tirantes en forma de barras, cuenta con opciones intrínsecas del control de graficación que permite dar formato a las distintas componentes del gráfico (área de trazado, ejes, formatos de números, etc), Fig. 4.23.

El programa cuenta con una ayuda de tipo *chm* (compilado de archivos *HTML* o páginas *web*) relacionada con el uso de los comandos. Con el menú “Ayuda/Contenido” se activa la ventana de la ayuda (Fig. 4.24).

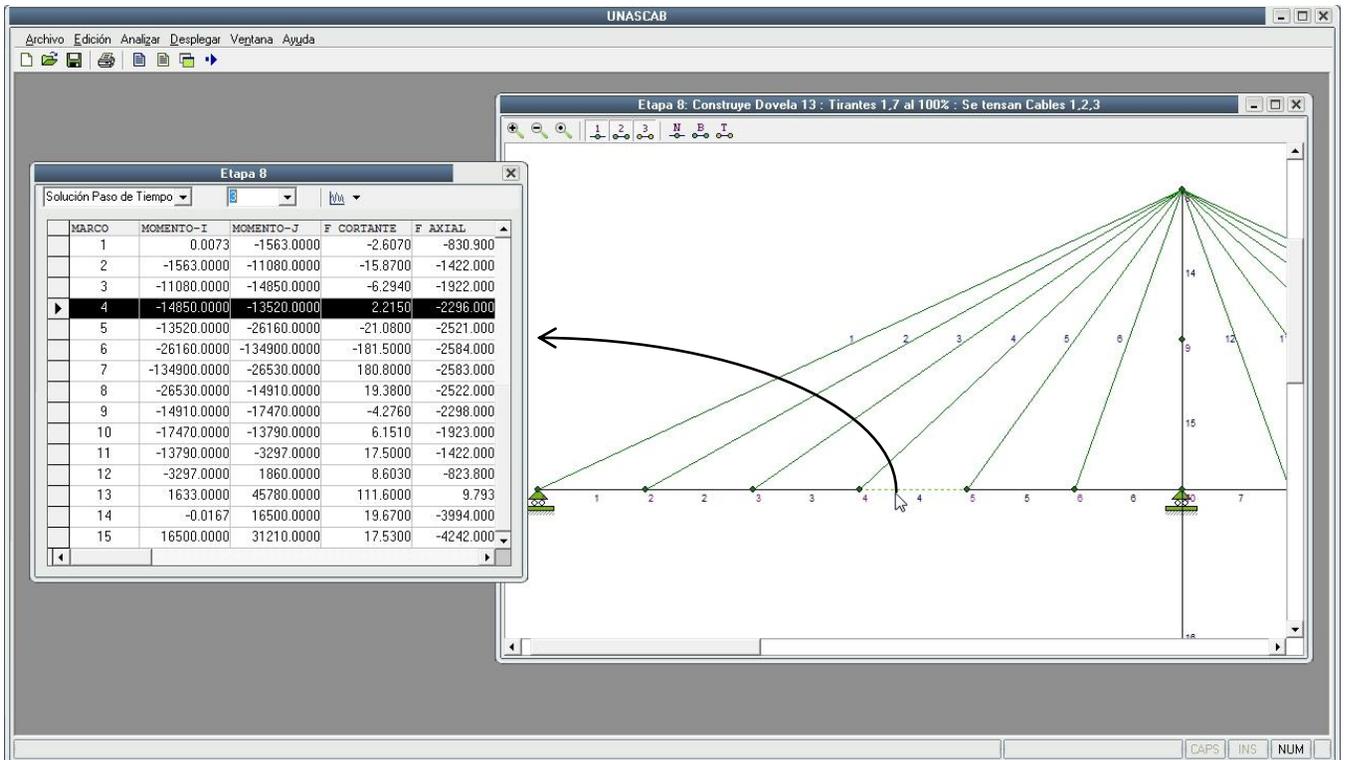


Figura 4.22.- Interacción del graficador 2D con ventana de resultados.

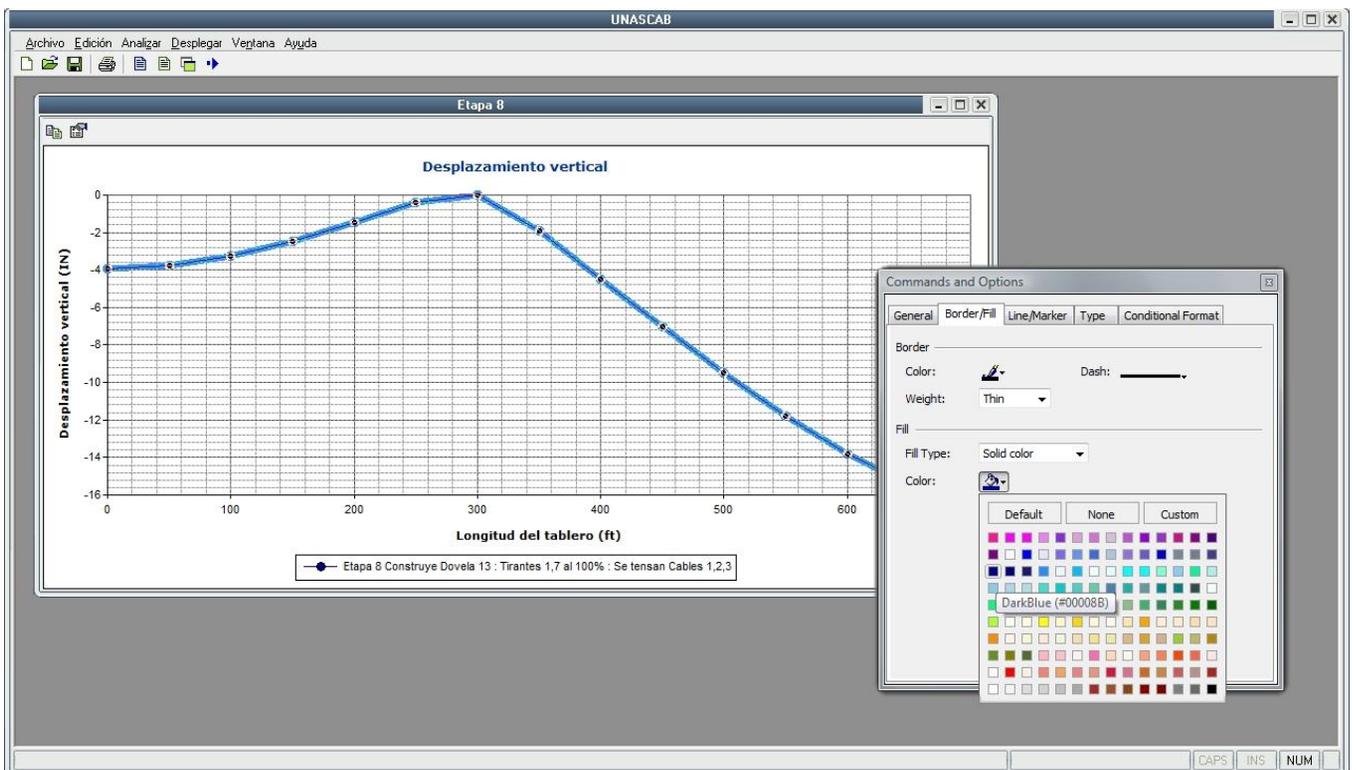


Figura 4.23.- Opciones de formato del control para graficar resultados.

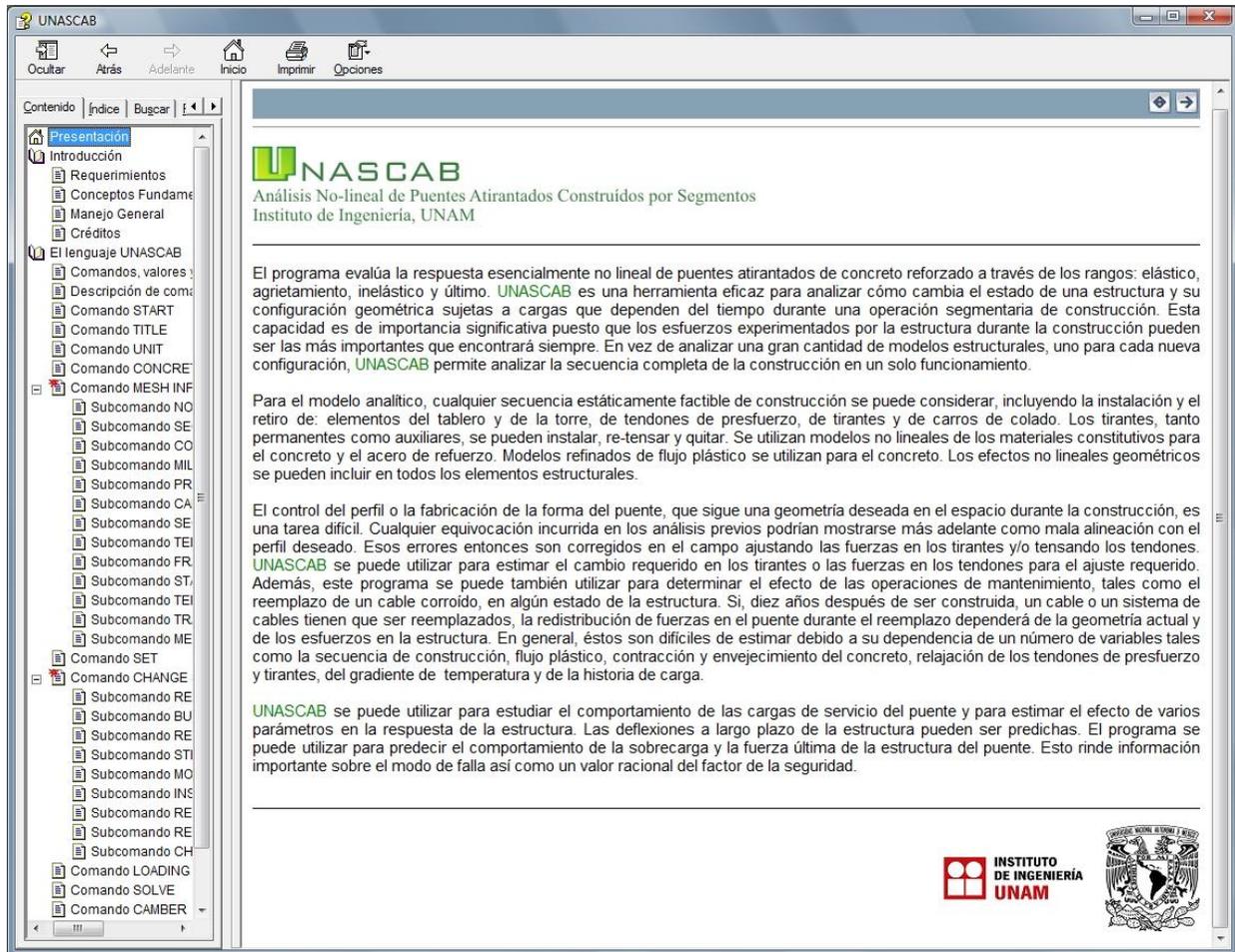


Figura 4.24.- Ayuda del programa.

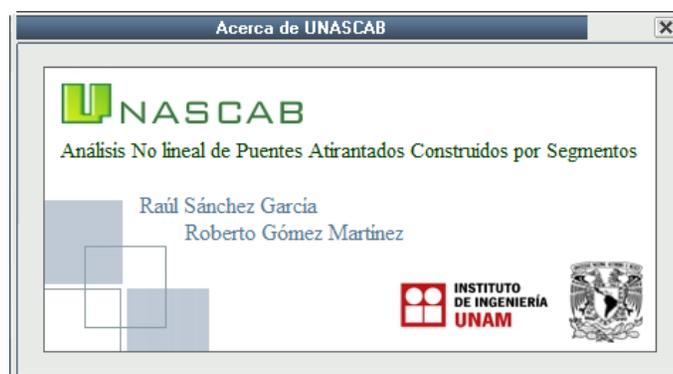


Figura 4.25.- Ventana Acerca de...

## 4.4 DISCUSIÓN DE RESULTADOS

Se presentan sólo los resultados más importantes con el fin de comprender el comportamiento de la estructura del ejemplo.

### Desplazamientos verticales

Los desplazamientos verticales que se muestran en las siguientes gráficas son los desplazamientos reportados por el programa *UNASCAB* utilizando el parámetro “P” del comando *SOLVE*, y representan los desplazamientos del tablero suponiendo que no se usa contraflecha, por lo que cada segmento nuevo del tablero se adjunta al segmento anterior de modo que es tangente al perfil del desplazamiento ya existente.

En la Fig. 4.25 se muestran desplazamientos verticales del tablero para diferentes tiempos de observación. Los tiempos de observación corresponden desde la etapa 2 (día 45) hasta la etapa 5 (día 105) y conforman la construcción en doble voladizo del tablero a partir de la torre. Los desplazamientos verticales del tablero en estas etapas son simétricos con respecto a la torre, como cabría esperar para una estructura simétrica con cargas también simétricas. En una secuencia de construcción este tipo de gráfico son de utilidad para ajustar las fuerzas en los cables y obtener el perfil del tablero deseado. La magnitud de las fuerzas iniciales en los cables, en este ejemplo, son solo las necesarias para equilibrar la carga muerta de los segmentos del tablero.

Después de que el segmento de cierre para el claro lateral es colado, los desplazamientos ya no son simétricos alrededor de la torre. En la Fig. 4.26 se muestran los desplazamientos del tablero después del cierre del claro lateral y para la etapa antes del cierre. Se observa que los desplazamientos para el claro lateral no cambian mucho de una etapa a otra debido a la restricción impuesta por el estribo, mientras que para el claro central aumentan debido a que permanece aún en voladizo. Esto muestra la influencia del proceso constructivo en el comportamiento de estructuras construidas por segmentos. Cabe mencionar que el programa *UNASCAB* también es útil para estudiar diferentes procesos constructivos para una misma estructura.

En la Fig. 4.27 se muestra el perfil del tablero al término de su construcción y después de 10000 días de terminada su construcción. La flecha en los claros laterales no cambian mucho, pero la flecha en el claro central aumenta a casi al doble de la que se presenta al finalizar la construcción de la estructura. A la mitad del claro central la flecha es de 7 pulgadas en la conclusión del puente y aumenta a 14.9 pulgadas después de 10000 días. Esto muestra la gran importancia de realizar estudios que consideren los efectos que dependen del tiempo para determinar el comportamiento de este tipo de estructuras.

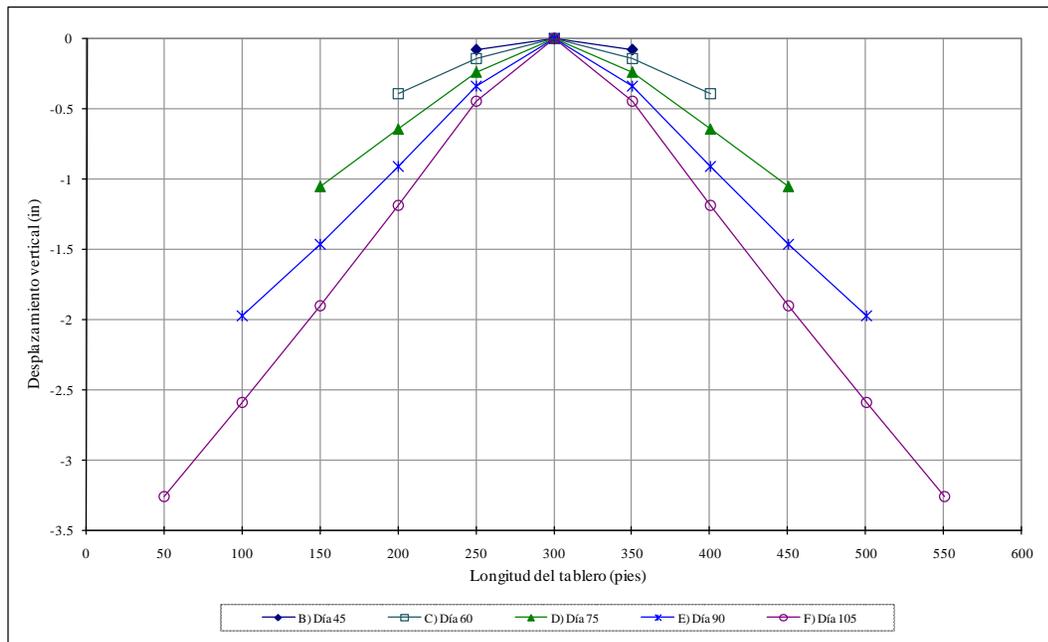


Figura 4.25.- Desplazamiento vertical del tablero en doble voladizo en etapas iniciales.

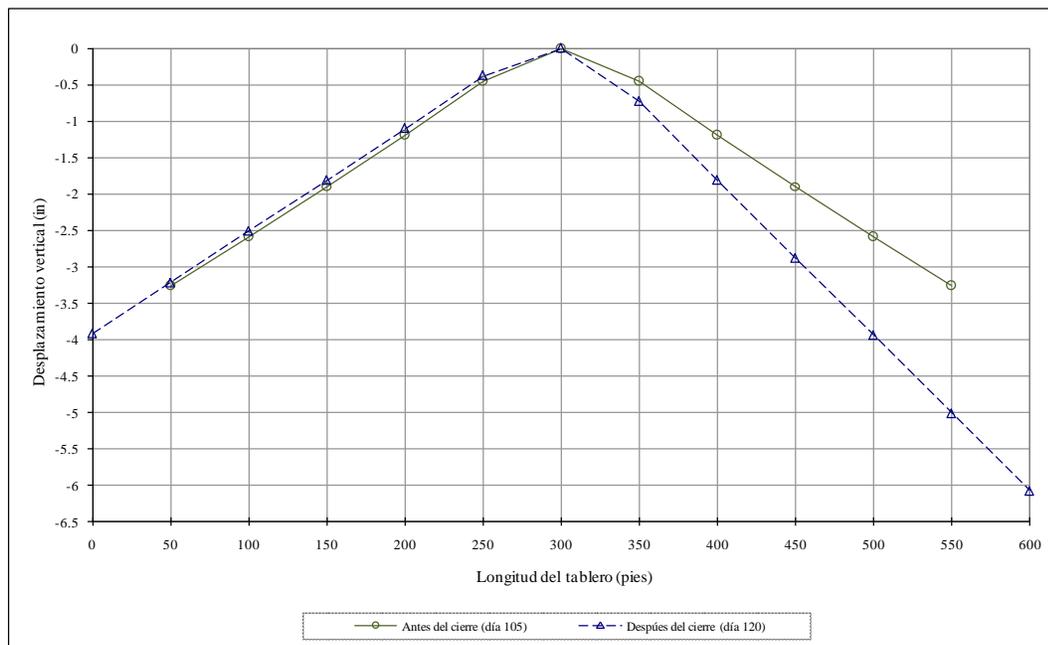


Figura 4.26.- Perfil del tablero antes y después del cierre del claro lateral.

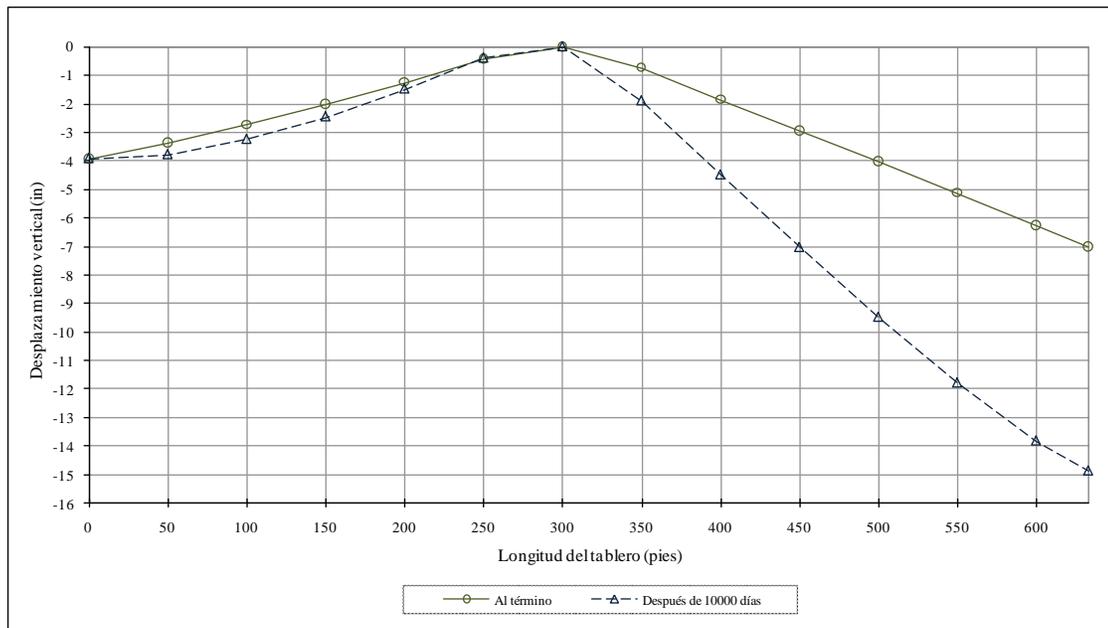


Figura 4.27.- Perfil del tablero al término de la construcción y después de 10,000 días.

### Redistribución de los momentos de flexión

Los momentos resistidos por la sección transversal de concreto son calculados por el programa UNASCAB mediante la integración de los esfuerzos sobre la sección transversal del tablero, e incluyen los efectos de presfuerzo. Los tiempos de observación mostrados en la Fig. 4.28 corresponden al término de construcción del puente y después de 10000 días de terminada la construcción.

La redistribución de momentos flexionantes debidos al flujo plástico y contracción del concreto después de 10000 días se muestra en la Fig. 4.28. La gráfica muestra, comparados con los valores al término de construcción, que los momentos negativos se incrementan a lo largo del tablero excepto para el segmento de cierre (elemento 13). El incremento mayor en el momento negativo ocurre en el punto donde el tablero es soportado por la torre. En el segmento de cierre, el momento cambia de momento negativo a momento positivo, esto es debido a la restricción rotacional que se impone al conectar los voladizos centrales.

### Tensión en tirantes

La Fig. 4.29 muestra el cambio de las fuerzas de los tirantes al término de construcción y después de 10000 días. Se observa que para los tirantes centrales de los claros la tensión resulta igual o ligeramente mayor después de 10000 días, los tirantes 1, 6, 7 y 12 tienen pérdidas significativas de tensión. Este gráfico es útil para observar cómo trabaja en conjunto el sistema de tirantes propuesto y así estudiar diferentes configuraciones de sistemas de atirantado.

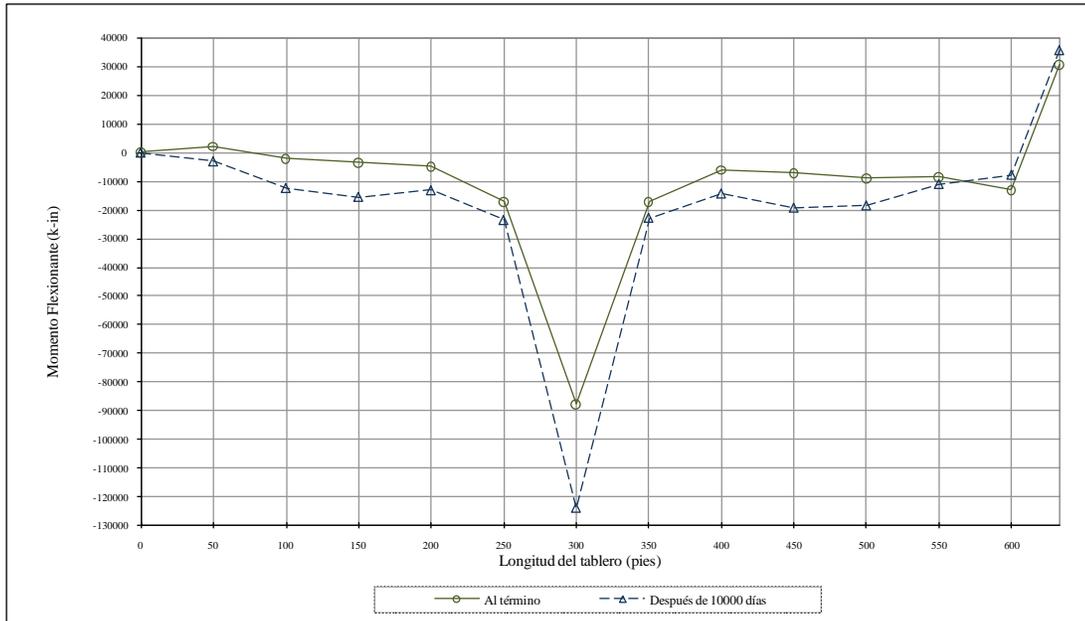


Figura 4.28.- Redistribución de momento al fin de construcción y después de 10,000 días.

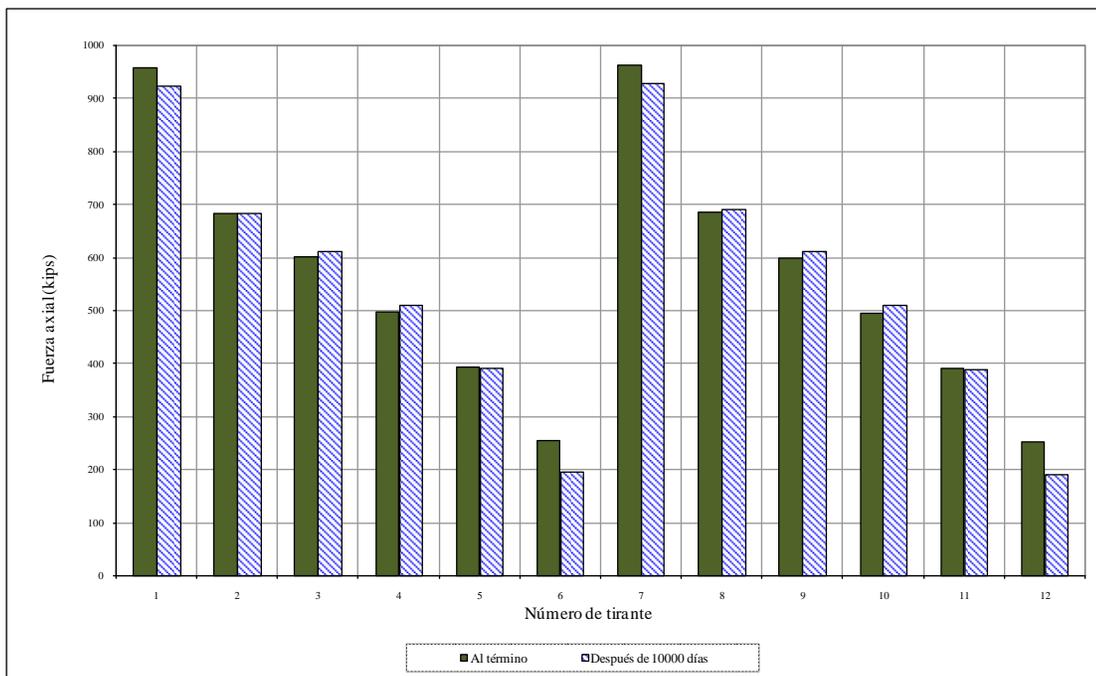


Figura 4.29.- Tensión en los tirantes al fin de construcción y después de 10,000 días.

## Capítulo 5

# CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

El objetivo principal de este trabajo fue la creación de un programa de análisis no lineal de puentes atirantados (UNASCAB) en ambiente Windows. Se modificaron y usaron las subrutinas de análisis estructural del programa NASCAB. Previo al desarrollo se hizo un repaso acerca del comportamiento de puentes atirantados de concreto presforzado y de la formulación matemática utilizada en el programa para realizar análisis estructural en el tiempo o por etapas constructivas. De las dificultades que se presentan en esta formulación, debidas a la intervención de multitud de factores, se concluye la importancia y utilidad del uso de herramientas computacionales (como la que se presenta en este trabajo) que permitan analizar cómo cambia el estado de un puente atirantado a lo largo de su vida útil, tanto en sus etapas constructivas como después de terminada su construcción. El uso de programas de computadora, en particular el programa UNASCAB, en estos tipos de análisis permite analizar los puentes de manera rápida, permite el análisis de nuevas y diferentes técnicas de construcción, la utilización de nuevos materiales o la combinación de ambos aspectos.

Respecto al desarrollo del programa, se puede concluir que la metodología empleada de programación en capas mostró tener varias ventajas en el mismo desarrollo, en su depuración y en el mantenimiento del programa, así mismo es susceptible de adición de nuevas características y mejoras al programa.

Debido a que las subrutinas de análisis están programadas en el lenguaje Fortran y el programa principal en Visual Basic 6.0, se logró hacer uso de la metodología de tuberías anónimas para comunicar ambos procesos. Lo anterior tiene la ventaja de separar la parte de interfaz de usuario con la parte de análisis estructural, de tal forma que si se requiere alguna actualización en un futuro solo se modifica y compila la parte correspondiente sin afectar a la otra parte. Además, presenta el aprovechamiento de la potencia de FORTRAN en el manejo numérico y las capacidades de Visual BASIC 6.0 en el desarrollo de aplicaciones tipo Windows. Son muchos los ingenieros que han desarrollado y siguen desarrollando programas en el lenguaje FORTRAN por el manejo numérico que posee y por su velocidad de procesamiento, sin embargo utilizan una interfaz de usuario de solo texto. La metodología utilizada en el desarrollo de este trabajo representa una alternativa para la creación de programas con interfaces gráficas en ambiente Windows de manera rápida y ágil utilizando las subrutinas de cálculo programadas en el lenguaje FORTRAN.

El *parser* incorporado en el programa haciendo uso de funciones de manejo de cadenas de texto propias del Visual BASIC 6.0 ha resultado ser eficiente en la extracción de datos del archivo de entrada el cual está en formato libre, por lo que se concluye la buena funcionalidad de Visual BASIC en el tratamiento de cadenas de texto.

La creación de la base de datos de los resultados desde FORTRAN en archivos tipo CSV y el manejo de esta base de datos mediante ADO ha mostrado también grandes ventajas, ya que de esta manera al acceso a los resultados es de manera directa y ágil. Otra ventaja es que el control ADO tiene funciones intrínsecas para el manejo de los registros tales como filtrado y búsqueda y por lo

tanto su programación es relativamente sencilla. Además, por la naturaleza de los archivos tipo CSV la exportación de los resultados a otros programas, como Excel de Microsoft, es de manera directa.

Para visualizar la estructura en alguna etapa, se programó un graficador utilizando los controles gráficos de Visual BASIC 6.0 (*Line, Shape, Label y Picture*). Aunque es útil para representar la geometría de la estructura, hacer zoom y selección de elementos, este método muestra deficiencias con geometrías con muchos detalles que requieren de buena calidad de resolución en el dibujo, por lo que conviene explorar otras tecnologías como OpenGL o las librerías gráficas GDI (*Graphics Device Interface*) de Microsoft.

El uso de los controles ActiveX *cmax20.ocx* y *owc11.dll* han mostrado buen funcionamiento y rendimiento, el primero en el manejo de archivos de texto grandes y el segundo en la cantidad de datos que puede graficar aunado a los métodos que posee para cambiar el formato del gráfico.

El ejemplo de aplicación presentado sirvió para probar y depurar el programa UNASCAB, también permitió mostrar las capacidades del programa y hacer una breve discusión de su comportamiento en el tiempo.

Cabe mencionar que el programa UNASCAB tiene aún desventajas con respecto a programas comerciales de análisis estructural en cuanto a calidad gráfica, sin embargo representa una herramienta alternativa de apoyo en el análisis no lineal de puentes atirantados en la práctica profesional del análisis de puentes atirantados. Por la metodología utilizada en su desarrollo y las herramientas auxiliares utilizadas, el proyecto desarrollado sirve como base para continuas mejoras del mismo programa pero también sirve de base para otros programas similares que cuentan con una interfaz de solo texto y se busca implementar interfaces de usuario más amigables en un entorno Windows.

En resumen, de tener un programa con interfaz de solo texto, ahora se tiene un programa integrado con interfaz de usuario gráfica tipo Windows que permite en la misma ventana:

- Generar y editar el archivo de comandos (archivo de definición de la estructura y etapas de análisis) mediante un potente editor de texto integrado al programa.
- Analizar el archivo de comandos con la potencia numérica proporcionada por el lenguaje FORTRAN.
- Desplegar en la misma ventana principal el reporte general de análisis.
- Graficar la configuración de la geometría de un puente conforme se va creando las etapas de análisis en el archivo de comandos.
- Una vez analizada la estructura, acceder directamente a los resultados para alguna etapa en particular y desplegarlos en forma tabular o en forma grafica (a la cual se puede aplicar diferentes formatos de presentación).

- Consultar la ayuda acerca del funcionamiento y descripción de los parámetros utilizados para cada comando.

## **Recomendaciones**

Debido a la utilidad de este programa en el análisis de puentes, se recomienda continuar con el mejoramiento del programa, utilizando nuevas tecnologías de desarrollo de software y como base el programa presentado en esta tesis.

En el desarrollo de aplicaciones en un entorno Windows existen en la actualidad nuevas herramientas de desarrollo con más funcionalidades, tal es el caso de la plataforma .NET de Microsoft. Esta plataforma incluye la nueva versión del lenguaje BASIC denominado Visual BASIC .NET por lo que conviene explorar esta nueva tecnología para nuevas versiones del programa UNASCAB.

Otro aspecto que tal vez convenga explorar es el manejo de la base de datos de resultados en forma binaria y no con archivos tipo CSV, aunque el uso de ADO ha resultado ser eficiente.

Para dotar de mayor capacidad al graficador 2D se recomienda explorar tecnologías de graficación como por ejemplo el uso de las librerías OpenGL o las librerías GDI+ de la plataforma .NET.

---

## Referencias

1. Abbas S. (1990). *Nonlinear Analisis of Segmentally Erected Reinforced and Prestressed Concrete Cable Stayed Bridges*. Research Report, UCB/SEMM-90/24, University of California, Berkeley.
2. ACI Committee 209, "Prediction of Creep, Shrinkage and Temperature Effects in Concrete Structures", *Reported by ACI Committee 209, 1992, 47pp.*
3. AllApi Network. *Introduction to the Win32API*. [En línea]. 1998-2007. [Citado el 15-06-2009]. Disponible en: <http://allapi.mentalis.org/vbtutor/iapi1.shtml>
4. AllApi Network. *Using API in VB*. [En línea]. 1998-2007. [Citado el 15-06-2009]. Disponible en: <http://allapi.mentalis.org/vbtutor/api1.shtml>
5. Balena Francesco. *Programación Avanzada con Microsoft Visual Basic 6.0*. 1ª ed. Madrid: McGraw-Hill; 2000.
6. Clive G. Page. *Professional Programmer's Guide to Fortran77*. [En línea]. University of Leicester, UK. 7-06-2005. [Citado el 15-06-2009]. Disponible en: <http://triton.fcaglp.unlp.edu.ar/fortran/libros/prof77.pdf>
7. Comité Euro International du Béton, CEB-FIP Model Code 1990 for Concrete and International Federation for Prestressing, Ed. Thomas Telford Services, Great Britain, 1991, 462 pp.
8. Comité Euro International du Béton, CEB-FIP Model Code for Concrete Structures, Cement and Concrete Association, Wexham Springs, Slough, England, 1978.
9. Hernandez, H. D., Gamble, W. L., (1975). *Time Dependent Prestress Losses in Pretensioned Concrete Construction*. Structural Research Series No. 417, Civil Engineering Studies, University of Illinois, Urbana, May 1975.
10. Kabir A. F., (1976). *Nonlinear Analysis of Reinforced Concrete Panels, Slabs and Shells for Time Dependent Effects*. Ph. D. Thesis, UC-SEMM-76-6, University of California, Berkeley.
11. Kang, Y. J. (1977). *Nonlinear Geometric, Material and Time-Dependent Analysis of Prestressed Concrete Frames*. Ph. D. Thesis, UC-SESM 77-1, University of California, Berkeley.
12. Kang, Y. J. (1989). *SPCFRAME – Computer Program for Nonlinear Segmental Analisis of planar Prestressed Concrete Frames*. Research Report, UCB/SEMM-89/07, University of California, Berkeley.

13. Kent, D. C., Park, R., (1971). "Flexural Members with Confined Concrete". *Journal of Structural División*, ASCE, Vol.97, No.ST7.
14. Ketchum, M. A. (1984). *Redistribution of Stresses in Segmentally Erected Prestressed Concrete Bridges*. Ph. D. Thesis, UC-SESM 84-10, University of California, Berkeley.
15. Lin, C. S. (1973). *Nonlinear Analysis of Reinforced Concrete Slabs and Shells*. UCB-SEMM 73-7, University of California, Berkeley.
16. Lohmann Daniel. *EditLog - fast logging into an edit control with cout*. [En línea]. 01-11-2003. [Citado el 15-06-2009]. Disponible en:  
<http://www.codeproject.com/KB/edit/editlog.aspx>
17. Luciano L.E. *Código fuente de un editor de código (Cedit - Open source)*. [En línea]. [Citado el 15-06-2009]. Disponible en:  
<http://www.recursosvisualbasic.com.ar/htm/utilidades-codigo-fuente/editor-de-codigo-avanzado-credit-opensource.htm>
18. Luciano L.E. *Recursos Visual Basic*. [En línea]. 2005 – 2009. [Citado el 15-06-2009]. Disponible en: <http://www.recursosvisualbasic.com.ar/>
19. Marshall A. C. *Fortran90 Course Notes*. [En línea]. The University of liverpool. 1997. [Citado el 15-06-2009]. Disponible en:  
<http://www.liv.ac.uk/HPC/HTMLFrontPageF90.html>
20. Microsoft Ayuda y Soporte. *Cómo utilizar el componente de gráfico Web con Visual Basic*. [En línea]. [Citado el 15-06-2009]. Disponible en:  
<http://support.microsoft.com/kb/235885/es>
21. Microsoft Ayuda y Soporte. *Complemento de Office 2003: Office Web Components*. [En línea]. [Citado el 15-06-2009]. Disponible en:  
<http://www.microsoft.com/downloads/details.aspx?FamilyId=7287252C-402E-4F72-97A5-E0FD290D4B76&displaylang=es>
22. Microsoft Ayuda y Soporte. *How To Use Named Pipes in a Visual Basic 32-bit Program*. [En línea]. 12-02-2007. [Citado el 15-06-2009]. Disponible en:  
<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/KB/ARTICLES/Q177/6/96.asp&NoWebContent=1>
23. n u M I T\_o r's Programming Page. *Tutoriales de Iczelion sobre Programación en Ensamblador para Win32, Tutorial 21: Tubería*. [En línea]. [Citado el 15-06-2009]. Disponible en: [http://mipagina.cantv.net/numetorl869/tut21\\_es.html](http://mipagina.cantv.net/numetorl869/tut21_es.html)
24. Regalado David. *Uso del control Chart Space de Office en vb*. [En línea]. [Citado el 15-06-2009]. Disponible en: <http://www.recursosvisualbasic.com.ar/htm/tutoriales/chart-office.htm>

- 
25. Rodríguez Carlos, Rodríguez Pablo. *VISUAL BASIC 6.0 Orientado a Bases de Datos*. 2ª Edición. 2008.
  26. Seif, S. P., Dilger, W. H. (1990). *Nonlinear Analysis and Collapse Load of P/C Cable-Stayed Bridges*. *Journal of Structures Engineering*, ASCE, March 1990.
  27. Som Guillermo. *Mundo Programación. Visual Basic 6.0 y anterior*. [En línea]. 1996-2009. [Citado el 15-06-2009]. Disponible en:  
<http://www.mundoprogramacion.com/vb/default.aspx>
  28. The Code Project. *The code Project, Your Development Resource*. [En línea]. 1999-2009. [Citado el 15-06-2009]. Disponible en: <http://www.codeproject.com/>
  29. Torres José Miguel. *Redirigir el resultado de un proceso en Visual Basic .NET*. [En línea]. 10/Feb/2005. [Citado el 15-06-2009]. Disponible en:  
[http://www.mundoprogramacion.com/colabora/NET2005/jtorres\\_redirigirVB.htm](http://www.mundoprogramacion.com/colabora/NET2005/jtorres_redirigirVB.htm)

## Apéndice

### Ejemplo de aplicación

#### Archivo de entrada (Ejemplo1.nas)

```
START
TITLE N = 1
      EJEMPLO 1: "ANALISIS POR ETAPAS"

UNIT U=KIP-IN
CONCRETE PARAMETERS N=1
1 M=ACI

MESH INPUT
NODES N=18
1 X=0 Y=130 S=12
7 X=300 Y=130 G=1,7,1
12 X=350 Y=130
17 X=600 Y=130 G=12,17,1
18 X=633 Y=130
8 X=300 Y=270
9 X=300 Y=200
10 X=300 Y=130
11 X=300 Y=0

CONCRETE PROPERTIES N=1
1 F=6.0 C=2.35 S=.0008 W=0.155/1728 T=2
  U=0.0035 E=0.003

MILD STEEL PROPERTIES N=1
1 E=29000 Y=60. U=0.03

PRESTRESSING STEEL N=1 P=5
1 C=.25 W=3.333E-5 F=240 R=10.
  F=196.6,220.,240.,245.,270.
  E=.00715,.009,.0115,.0135,.058

CABLE PROPERTIES N=1 P=5
1 F=240 D=.49/1728 A=0 R=0
  S=180,210,240,250,270
  E=.0075,.0088,.012,.016,.05

SECTION PROPERTIES N=4 C=8 S=3
1 C=8 S=2 K=1
  A=360,360,376,376,376,376,360,360
  Y=56,50,35.25,11.75,-11.75,-35.25,-50,-56
  B=20,20
  Z=53,-53
  M=1,1
2 C=8 S=2 K=1
  A=600,600,312,312,312,312,600,600
  Y=54,44,29.25,9.75,-9.75,-29.25,-44,-54
  B=20,20
  Z=49,-49
  M=1,1
3 C=8 S=2 K=1
  A=720,720,880,880,880,880,720,720
  Y=73,61,41.25,13.75,-13.75,-41.25,-61,-73
  B=32,32
  Z=67,-67
  M=1,1
4 C=8 S=3 K=1
  A=1890,1890,378,378,378,378,474,474
  Y=22.131,17.131,6.756,-8.994,-24.744,-40.494,-49.869,-52.869
  B=23.25,7.75,23.25
  Z=21.631,-16.869,-51.369
  M=1,1,1
```

```
FRAME ELEMENTS N=16
1,1,2    X=4 G=1,6,1,1,1,0,0
7,7,12   X=4
8,12,13  X=4 G=8,13,1,1,1,0,0
14,8,9   X=1
15,9,10  X=2
16,10,11 X=3

TENDON GEOMETRY N=3 P=2
1 S=1 M=1 A=12.5
  1 N=14 B=0,1560 E=600,1560
  L=1,2,3,4,5,6,7,12,13,14,15,16,17,18
  1 S=0 ! 2 S=0 ! 3 S=0 ! 4 S=0 ! 5 S=0 ! 6 S=0 ! 7 S=0
  8 S=0 ! 9 S=0 ! 10 S=0 ! 11 S=0 ! 12 S=0 ! 13 S=0 ! 14 S=0
2 S=1 M=1 A=12.5
  1 N=5 B=5400,1560 E=6000,1560
  L=14,15,16,17,18
  1 S=0 ! 2 S=0 ! 3 S=0 ! 4 S=0 ! 5 S=0
3 S=1 M=1 A=12.5
  1 N=3 B=6600,1560 E=7200,1560
  L=16,17,18
  1 S=0 ! 2 S=0 ! 3 S=0

STAY ELEMENTS N=12
1,1,8    P=1 A=15.5
2,2,8    P=1 A=15.5
3,3,8    P=1 A=7.75 G=3,6,1,1,0
7,8,17   P=1 A=15.5
8,8,16   P=1 A=15.5
9,8,15   P=1 A=7.75 G=9,12,1,0,-1

MESH COMPLETE

SET D=30 G=0,-1 C=.01,.05,.05,.0001,10.,1000.,10.,.05,.1 I=50,30 N=100

;Construye Pilon
CHANGE STRUCTURE
BUILD N=14,16,1 D=0
RESTRAINTS ! 11 R=1,1,1 !
CHANGE COMPLETE
SOLVE G=1 X=1 P=1
SOLVE T=1 G=1 X=1 P=1 D=45

;Construye Dovelas 6,7 : Tirantes 6,12 al 50%
CHANGE STRUCTURE
BUILD N=6 D=15
BUILD N=7 D=15
INSTALL N=6 F=357/2
INSTALL N=12 F=357/2
RESTRAINTS ! 7 R=0,2,0 !
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=60

;Construye Dovelas 5,8 : Tirantes 6,12 al 100% y 5,11 al 50%
CHANGE STRUCTURE
BUILD N=5 D=30
BUILD N=8 D=30
INSTALL N=5 F=414/2
INSTALL N=11 F=414/2
RESTRESS N=6 F=357
RESTRESS N= 12 F=357
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=75

;Construye Dovelas 4,9 : Tirantes 5,11 al 100% y 4,10 al 50%
CHANGE STRUCTURE
BUILD N=4 D=45
BUILD N=9 D=45
INSTALL N=4 F=492/2
INSTALL N=10 F=492/2
RESTRESS N=5 F=414
```

```
RESTRESS N=11 F=414
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=90

;Construye Dovelas 3,10 : Tirantes 4,10 al 100% y 3,9 al 50%
CHANGE STRUCTURE
BUILD N=3 D=60
BUILD N=10 D=60
INSTALL N=3 F=586/2
INSTALL N=9 F=586/2
RESTRESS N=4 F=492
RESTRESS N=10 F=492
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=105

;Construye Dovelas 2,11 : Tirantes 3,9 al 100% y 2,8 al 50%
CHANGE STRUCTURE
BUILD N=2 D=75
BUILD N=11 D=75
INSTALL N=2 F=687/2
INSTALL N=8 F=687/2
RESTRESS N=3 F=586
RESTRESS N=9 F=586
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=120

;Construye Dovelas 1,12 : Tirantes 2,8, al 100% y 1,7 al 50%
CHANGE STRUCTURE
BUILD N=1 D=90
BUILD N=12 D=90
INSTALL N=1 F=397
INSTALL N=7 F=397
RESTRESS N=2 F=687
RESTRESS N=8 F=687
RESTRAINTS ! 1 R=0,1,0 !
CHANGE COMPLETE
SOLVE G=1 X=1 P=-1
SOLVE T=1 G=1 X=1 P=1 D=135

;Construye Dovela de cierre 13 : Tirantes 1,7 al 100% : Se tensan Cables 1,2,3
CHANGE STRUCTURE
BUILD N=13 D=105
STRESS N=1,3,1 S=120,120
RESTRESS N=1 F=975
RESTRESS N=7 F=975
RESTRAINTS ! 18 R=1,0,1 !
CHANGE COMPLETE
SOLVE G=1 X=1 P=1
SOLVE T=1 G=1 X=1 S=3 P=1 D=10000

STOP
```

