



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Programa para la solución de  
armaduras bidimensionales**

**TESIS**

Que para obtener el título de

**Ingeniero Civil**

**P R E S E N T A**

José Reynaldo Moctezuma Espinosa

**DIRECTOR DE TESIS**

M. I. Miguel Ángel Rodríguez Vega



Ciudad Universitaria, Cd. Mx., 2018



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

DIVISIÓN DE INGENIERÍAS CIVIL Y GEOMÁTICA  
COMITÉ DE TITULACIÓN  
FING/DICyG/SEAC/UTIT/042/18

Señor  
JOSÉ REYNALDO MOCTEZUMA ESPINOSA  
Presente

En atención a su solicitud me es grato hacer de su conocimiento el tema que propuso el profesor M.I. MIGUEL ÁNGEL RODRÍGUEZ VEGA, que aprobó este Comité, para que lo desarrolle usted como tesis de su examen profesional de INGENIERO CIVIL.

**"PROGRAMA PARA LA SOLUCIÓN DE ARMADURAS BIDIMENSIONALES"**

- INTRODUCCIÓN
- I. ANTECEDENTES
- II. RESUMEN DEL PROGRAMA
- III. SOLUCIÓN DE ARMADURAS EMPLEANDO EL PROGRAMA
- IV. EJEMPLOS DE APLICACIÓN
- V. CONCLUSIONES

Ruego a usted cumplir con la disposición de la Dirección General de la Administración Escolar en el sentido de que se imprima en lugar visible de cada ejemplar de la tesis el Título de ésta.

Asimismo le recuerdo que la Ley de Profesiones estipula que deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito para sustentar Examen Profesional.

Atentamente  
"POR MI RAZA HABLARÁ EL ESPÍRITU"  
Cd. Universitaria a 9 de mayo del 2018.  
EL PRESIDENTE

  
M.I. GERMÁN LÓPEZ RINCÓN

GLR/MTH\*gar.

# ÍNDICE

Resumen.....	5
Introducción .....	6
1. Antecedentes.....	7
1.1. Concepto de armadura.....	8
1.2. Leyes que rigen el método de rigideces en armaduras.....	8
1.2.1. Ley del paralelogramo .....	8
1.2.2. Primera ley de Newton .....	9
1.2.3. Tercera ley de Newton.....	10
1.3. Hipótesis empleadas en el método de rigideces en armaduras .....	10
1.3.1. Continuo.....	10
1.3.2. Elástico lineal.....	11
1.3.3. Homogéneo.....	15
1.3.4. Isotrópico .....	15
1.3.5. Principio de Saint Venant.....	15
1.4. Método de rigideces en armaduras .....	15
1.4.1. Solución de la ecuación diferencial del equilibrio en una barra .....	16
1.4.2. Matriz de rigidez (local) de una barra.....	19
1.4.3. Matriz de transformación .....	23
1.4.4. Matriz de rigidez global de una barra.....	23
1.4.5. Matriz de rigidez global de una armadura .....	23
1.5. Solución de sistemas de ecuaciones lineales.....	24
1.5.1. Sistemas de ecuaciones lineales.....	24
1.5.2. Eliminación de Gauss .....	25
1.6. Matriz de rigidez y su transformación a una matriz no singular.....	26
1.6.1. Enfoque de eliminación .....	26
1.6.2. Enfoque de penalización.....	27
2. Resumen del programa .....	28
2.1. Entrada .....	30
2.1.1. Obtención de la matriz de rigidez de la armadura y elaboración de gráfica descriptiva de la armadura .....	30
2.2. Solución .....	36
2.2.1. Obtención del vector de desplazamiento y vector de fuerza de la armadura .....	36
2.2.2. Obtención de las fuerzas y desplazamientos de cada barra .....	43
2.3. Análisis de resultados .....	45

2.3.1. Fuerza máxima .....	45
2.3.2. Desplazamiento máximo.....	46
2.3.3. Gráfica de la armadura deformada .....	47
2.3.4. Gráfica de la armadura mostrando fuerzas .....	48
2.3.5. Gráfica de la armadura mostrando desplazamientos.....	49
3. Solución de una armadura empleando el programa .....	51
3.1. Ingreso de los datos y generación de variables .....	52
3.2. Empleo de las funciones .....	54
4. Ejemplos de aplicación del programa .....	58
4.1. Primer ejemplo.....	59
4.2. Segundo ejemplo.....	64
4.3. Tercer ejemplo .....	70
4.4. Cuarto ejemplo.....	77
5. Conclusiones .....	86
Glosario .....	88
Referencias.....	91
Anexos.....	92
Código del programa.....	92

## Resumen

La formación del Ingeniero Civil permite “participar en las etapas de planeación, diseño, organización, construcción, operación y conservación de obras civiles y de infraestructura” [1].

El diseño consiste en emplear los conocimientos técnicos y científicos adquiridos por generaciones de ingenieros para conocer el comportamiento de la futura obra de infraestructura según los fenómenos que actúan sobre esta y así poder determinar la mejor alternativa de solución considerando también cuestiones económicas.

Para el diseño de estructuras es necesario conocer su comportamiento ante cargas, estas cargas pueden ser debidas a: sismos, pesos, viento, etcétera. El comportamiento ante cargas (fuerzas externas) se conoce por medio de los esfuerzos y las deformaciones (desplazamientos relativos) de la estructura.

El objetivo del programa es conocer el comportamiento de una armadura ante una carga cualquiera, con el fin de diseñar la armadura posteriormente.

A diferencia de los programas comerciales que permiten calcular las fuerzas y desplazamientos en armaduras también, el programa tiene la ventaja de que el código es libre y simple, donde simple denota que fue creado con los conocimientos adquiridos en la carrera de ingeniería.

La ventaja de que el código sea libre, es que esto permite saber cómo funciona el programa, a diferencia de la mayoría de los programas comerciales. Ya que en la actualidad el ingeniero peca de confianza en los programas comerciales, no demeritando la funcionalidad de estos pero al no conocer el funcionamiento del código en estos programas, emplearlos resulta más un acto de fe que de rigurosa ciencia, propia de la naturaleza del diseño.

# Introducción

El presente trabajo permite calcular las fuerzas en las barras y los desplazamientos en los nudos de una armadura a través del método de rigideces. Para lo anterior se creó un programa con código en Wolfram Mathematica 11.0 que permite calcular dichos elementos por medio del método.

Al ejecutar el programa, este muestra las fuerzas en las barras y los desplazamientos en los nudos, adicionalmente muestra dos diagramas de la armadura: Configuración original y deformada. También permite determinar la fuerza máxima en un grupo de barras y el desplazamiento máximo en un grupo de nudos.

El capítulo 1 muestra en forma breve la justificación teórica del método de rigideces en armaduras, al abordar las leyes y las hipótesis que defienden el método. También incluye temas referentes a la solución de sistemas de ecuaciones lineales, necesarios para la solución con código.

El capítulo 2 realiza una breve descripción de qué es Wolfram Mathematica, además describe el funcionamiento del código creado. Describe cómo funciona cada sección del código y el propósito de dicha sección al relacionarlo con el método de rigideces.

El capítulo 3 muestra cómo se usa el programa. Para ello explica cómo se ingresan los datos; las propiedades y dimensiones de la armadura y las condiciones de frontera: Fuerzas y deformaciones. Además muestra cómo usar los datos anteriores para su solución.

El capítulo 4 muestra la confiabilidad del programa mediante dos ejemplos de la literatura y muestra la utilidad del programa mediante dos ejemplos propuestos con analogías a armaduras reales.

## **1. Antecedentes**

## **1.1. Concepto de armadura**

El concepto de armadura se concibe como una estructura formada por barras dispuestas en forma de triángulos. Teóricamente se consideran las siguientes hipótesis:

a. Las barras están unidas entre sí mediante pasadores sin fricción que impiden el desplazamiento relativo pero permiten el giro.

b. Las barras son rectas.

c. Los desplazamientos son ocasionados por las deformaciones en las barras, aunque dichas deformaciones son muy pequeñas en comparación con las dimensiones de la armadura, lo que conduce a que la forma y dimensiones de la armadura queden inalteradas durante la carga.

d. Las fuerzas se aplican sólo en los nudos, por ende se desprecia el peso propio de las barras al considerarse de mucho menor magnitud que las fuerzas que actúan sobre la estructura o se considera repartido en los dos nudos de cada barra.

Todas las hipótesis anteriores conducen a que cada barra esté sometida únicamente a dos fuerzas colineales con el eje longitudinal de estas.

## **1.2. Leyes que rigen el método de rigideces en armaduras**

Las leyes que rigen el método de rigideces son:

### **1.2.1. Ley del paralelogramo**

“La resultante de dos fuerzas que concurren en un punto se encuentra en la diagonal del paralelogramo construido con dichas fuerzas, y pasa por dicho punto” [2].

Ilustrativamente se muestra como:

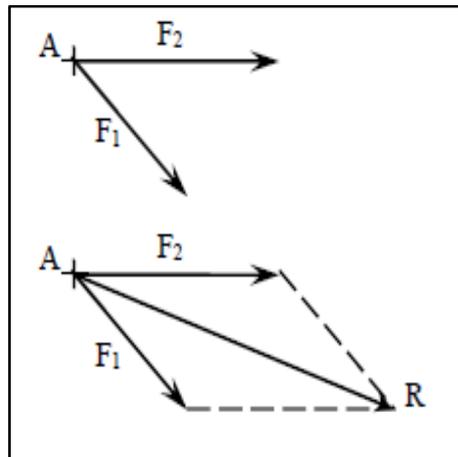


Figura 1. Ilustración de la ley del paralelogramo.

Fuente: Ocariz Castelazo, Juan, [2].

### 1.2.2. Primera ley de Newton

“Todo cuerpo persevera en su estado de reposo o de movimiento rectilíneo uniforme, a menos que sea obligado por fuerzas externas a cambiar ese estado” [2].

Al ser ley, resulta evidente que es aplicable a partículas de materia, pero no sobra decirlo. La generalización a cuerpos rígidos matemáticamente, y considerando un sistema cartesiano, es:

$$\sum F_x = 0$$

$$\sum F_y = 0$$

$$\sum F_z = 0$$

$$\sum M_x = 0$$

$$\sum M_y = 0$$

$$\sum M_z = 0$$

Donde:

$\sum F_x$ , suma de las componentes de las fuerzas externas en la dirección del eje x.

$\sum F_y$ , suma de las componentes de las fuerzas externas en la dirección del eje y.

$\sum F_z$ , suma de las componentes de las fuerzas externas en la dirección del eje z.

$\sum M_x$ , suma de los momentos alrededor la dirección del eje x de las fuerzas externas.

$\sum M_y$ , suma de los momentos alrededor la dirección del eje y de las fuerzas externas.

$\sum M_z$ , suma de los momentos alrededor la dirección del eje z de las fuerzas externas.

### 1.2.3. Tercera ley de Newton

“Para toda acción hay siempre una reacción igual y contraria: o bien, las acciones mutuas de dos cuerpos son siempre iguales y de sentido contrario” [2].

Dicha ley implica que para un cuerpo que se divide, formando dos cuerpos, con el fin de encontrar las fuerzas internas existe una continuidad en el esfuerzo. Al existir la continuidad se pueden emplear las matemáticas de variable continua para su representación.

## 1.3. Hipótesis empleadas en el método de rigideces en armaduras

Además de dichas leyes, en el estudio del método de rigideces se consideran las siguientes hipótesis:

### 1.3.1. Continuo

El material de las barras es un continuo, un continuo es un cuerpo que carece de huecos o vacíos, algo ideal al conocer la estructura molecular de la materia. Esto implica que existe una continuidad en la deformación. Al existir la continuidad se pueden emplear las matemáticas de variable continua para su representación.

### 1.3.2. Elástico lineal

El material de las barras es elástico lineal, para poder entender el término elástico lineal, es necesario ahondar en los términos esfuerzo y deformación unitaria.

Esfuerzo es la intensidad de las fuerzas internas que se producen debido a la acción de fuerzas externas, la intensidad se mide como la fuerza que actúa por unidad de superficie.

Para poder medir dicha intensidad es necesario hacer la superficie muy pequeña, de forma que se analice una partícula de área infinitesimal. La razón de las fuerzas internas que actúan sobre dicha área infinitesimal es llamada esfuerzo.

Los esfuerzos en una partícula, y considerando un sistema cartesiano, se muestran en la siguiente figura.

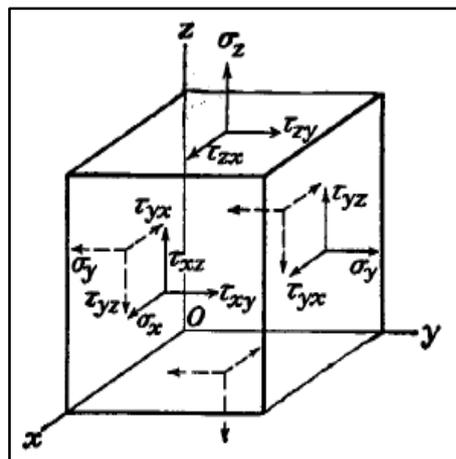


Figura 2. Ilustración de los esfuerzos en una partícula.

Fuente: Timoshenko, S., [3].

Para que la suma de momentos sea cero y se cumpla el equilibrio es necesario que los esfuerzos cortantes en dirección del eje y, en la cara con dirección del eje x y en dirección del eje x, en la cara con dirección del eje y sean

iguales. Y así mismo con los dos restantes esfuerzos cortantes. Esto queda resumido de la siguiente forma:

$$\tau_{xy} = \tau_{yx}$$

$$\tau_{zx} = \tau_{xz}$$

$$\tau_{yz} = \tau_{zy}$$

Deformación unitaria es la deformación por unidad de longitud; y deformación es el desplazamiento relativo entre partículas de un mismo cuerpo.

Matemáticamente, y considerando un sistema cartesiano, la deformación se define como:

$$\varepsilon_x = \frac{\partial u}{\partial x}$$

$$\varepsilon_y = \frac{\partial v}{\partial y}$$

$$\varepsilon_z = \frac{\partial w}{\partial z}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

$$\gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}$$

Donde:

$u$ , desplazamiento en la dirección del eje  $x$ .

$v$ , desplazamiento en la dirección del eje  $y$ .

$w$ , desplazamiento en la dirección del eje  $z$ .

$\varepsilon_x$ , deformación unitaria longitudinal en la dirección del eje  $x$ .

$\varepsilon_y$ , deformación unitaria longitudinal en la dirección del eje  $y$ .

$\varepsilon_z$ , deformación unitaria longitudinal en la dirección del eje  $z$ .

$\gamma_{xy}$ , deformación unitaria angular del ángulo asociada al desplazamiento al eje  $x$  en dirección del eje  $y$ .

$\gamma_{zx}$ , deformación unitaria angular del ángulo asociada al desplazamiento al eje z en dirección del eje x.

$\gamma_{yz}$ , deformación unitaria angular del ángulo asociada al desplazamiento al eje y en dirección del eje z.

Para que no existe rotación como cuerpo rígido es necesario que la deformación angular sea la misma a la asociada al desplazamiento al eje x en dirección del eje y y a la asociada al desplazamiento al eje y en dirección del eje x. Y así mismo con las dos restantes deformaciones unitarias angulares. Lo anterior queda resumido de la siguiente forma:

$$\gamma_{xy} = \gamma_{yx}$$

$$\gamma_{zx} = \gamma_{xz}$$

$$\gamma_{yz} = \gamma_{zy}$$

De aquí en adelante, por practicidad, el término deformación unitaria se contraerá a deformación.

Un material elástico lineal refiere a aquel material que se comporta según la ley de Hooke.

Ley de Hooke

Considerando un sistema cartesiano, matemáticamente dicha ley se expresa como:

$$\varepsilon_x = \frac{1}{E} (\sigma_x - \nu(\sigma_y + \sigma_z))$$

$$\varepsilon_y = \frac{1}{E} (\sigma_y - \nu(\sigma_x + \sigma_z))$$

$$\varepsilon_z = \frac{1}{E} (\sigma_z - \nu(\sigma_x + \sigma_y))$$

$$\gamma_{xy} = \frac{1}{G} \tau_{xy}$$

$$\gamma_{yz} = \frac{1}{G} \tau_{yz}$$

$$\gamma_{zx} = \frac{1}{G} \tau_{zx}$$

Donde:

$\varepsilon_x$ , deformación longitudinal en la dirección del eje x.

$\varepsilon_y$ , deformación longitudinal en la dirección del eje y.

$\varepsilon_z$ , deformación longitudinal en la dirección del eje z.

$\gamma_{xy}$ , deformación angular del ángulo que forman el eje x y el eje y.

$\gamma_{zx}$ , deformación angular del ángulo que forman el eje z y el eje x.

$\gamma_{yz}$ , deformación angular del ángulo que forman el eje y y el eje z.

$\sigma_x$ , esfuerzo normal en la dirección del eje x.

$\sigma_y$ , esfuerzo normal en la dirección del eje y.

$\sigma_z$ , esfuerzo normal en la dirección del eje z.

$\tau_{xy}$ , esfuerzo cortante en dirección del eje y, en la cara con dirección del eje x.

$\tau_{zx}$ , esfuerzo cortante en dirección del eje x, en la cara con dirección del eje z.

$\tau_{yz}$ , esfuerzo cortante en dirección del eje z, en la cara con dirección del eje y.

$E$ , módulo de rigidez.

$G$ , módulo de rigidez al cortante.

$\nu$ , razón de Poisson.

El esfuerzo normal provoca deformación longitudinal en la dirección donde es aplicado, pero al cambiar la dimensión en dicha dirección, necesariamente se presentan cambios en las dimensiones ortogonales a esta. Es decir, un esfuerzo normal provoca deformación longitudinal en la dirección donde es aplicado y deformaciones longitudinales en las direcciones ortogonales a esta dirección. Existe una relación entre estas deformaciones, y está dada por la razón de Poisson.

El esfuerzo tangencial, al no provocar cambios en el volumen, no produce más efectos que los propios a la deformación angular correspondiente.

El módulo de rigidez al cortante es dependiente del módulo de rigidez y la razón de Poisson.

$$G = \frac{E}{2(1 + \nu)}$$

Donde:

$E$ , módulo de rigidez.

$G$ , módulo de rigidez al cortante.

$\nu$ , razón de Poisson.

Debido a que las deformaciones son proporcionales a los esfuerzos y pequeñas en comparación, es posible la superposición de causas y efectos.

### 1.3.3. Homogéneo

Un material homogéneo “debe tener la misma composición (y en consecuencia las mismas propiedades elásticas) en cada punto” [4].

### 1.3.4. Isotrópico

Un material isotrópico es aquel que “tiene las mismas propiedades en todas las direcciones (ya sea axial, lateral o cualquier otra dirección)” [4].

### 1.3.5. Principio de Saint Venant

En la carga de las barras es aplicable el principio de Saint Venant, el cual dicta que el estado de esfuerzos no resulta alterado alejado de los extremos cuando se cambia la distribución de esfuerzos en los extremos siempre y cuando dicho cambio sea por un sistema equivalente.

## 1.4. Método de rigideces en armaduras

El método de rigideces es un método que emplea sistema de ecuaciones lineales donde las incógnitas están dada por los desplazamientos lineales y angulares de los nudos y los coeficientes de las incógnitas por las características de la armadura: dimensiones y material de las barras y forma de la armadura.

#### 1.4.1. Solución de la ecuación diferencial del equilibrio en una barra

Para conocer los desplazamientos lineales y angulares en una barra es necesario encontrar primeramente el estado de esfuerzos de una barra en cada punto de la barra, y el estado de esfuerzo implica conocer los esfuerzos a los que está sometido una partícula en el infinito de direcciones posibles.

Para conocer el estado de esfuerzos basta con conocer los esfuerzos normales y tangenciales en tres direcciones ortogonales. Para un sistema de coordenadas cartesiano se necesita conocer los siguientes esfuerzos:

$\sigma_x$ , esfuerzo normal en la dirección del eje x.

$\sigma_y$ , esfuerzo normal en la dirección del eje y.

$\sigma_z$ , esfuerzo normal en la dirección del eje z.

$\tau_{xy}$ , esfuerzo cortante en dirección del eje y, en la cara con dirección del eje x.

$\tau_{zx}$ , esfuerzo cortante en dirección del eje x, en la cara con dirección del eje z.

$\tau_{yz}$ , esfuerzo cortante en dirección del eje z, en la cara con dirección del eje y.

Para conocer los seis esfuerzos anteriores es necesario resolver las ecuaciones diferenciales del equilibrio, considerando las ecuaciones de compatibilidad de deformación y las condiciones de frontera.

Las ecuaciones diferenciales del equilibrio (sistema cartesiano) sin considerar peso propio son:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} = 0$$

$$\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + \frac{\partial \sigma_z}{\partial z} = 0$$

Donde:

$\sigma_x$ , esfuerzo normal en la dirección del eje x.

$\sigma_y$ , esfuerzo normal en la dirección del eje y.

$\sigma_z$ , esfuerzo normal en la dirección del eje z.

$\tau_{xy}$ , esfuerzo cortante en dirección del eje y, en la cara con dirección del eje x.

$\tau_{zx}$ , esfuerzo cortante en dirección del eje x, en la cara con dirección del eje z.

$\tau_{yz}$ , esfuerzo cortante en dirección del eje z, en la cara con dirección del eje y.

Las ecuaciones de compatibilidad de deformación (sistema cartesiano) son:

$$\begin{aligned} \frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial y} \\ \frac{\partial^2 \varepsilon_y}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial y^2} &= \frac{\partial^2 \gamma_{yz}}{\partial y \partial z} \\ \frac{\partial^2 \varepsilon_z}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial z^2} &= \frac{\partial^2 \gamma_{zx}}{\partial x \partial z} \\ 2 \frac{\partial^2 \varepsilon_x}{\partial y \partial z} &= \frac{\partial}{\partial x} \left( -\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right) \\ 2 \frac{\partial^2 \varepsilon_y}{\partial z \partial x} &= \frac{\partial}{\partial y} \left( \frac{\partial \gamma_{yz}}{\partial x} - \frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right) \\ 2 \frac{\partial^2 \varepsilon_z}{\partial x \partial y} &= \frac{\partial}{\partial z} \left( \frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} - \frac{\partial \gamma_{xy}}{\partial z} \right) \end{aligned}$$

Donde:

$\varepsilon_x$ , deformación longitudinal en la dirección del eje x.

$\varepsilon_y$ , deformación longitudinal en la dirección del eje y.

$\varepsilon_z$ , deformación longitudinal en la dirección del eje z.

$\gamma_{xy}$ , deformación angular del ángulo que forman el eje x y el eje y.

$\gamma_{zx}$ , deformación angular del ángulo que forman el eje z y el eje x.

$\gamma_{yz}$ , deformación angular del ángulo que forman el eje y y el eje z.

El estado de esfuerzos que resulta en una barra cargada en sus extremos por dos fuerzas en equilibrio colineales con su eje longitudinal, figura 3, resulta en un esfuerzo normal uniforme es:

$$\sigma_x = \frac{F}{A}$$

Donde:

$\sigma_x$ , esfuerzo normal en la dirección del eje x.

$F$ , fuerza colineal con el eje longitudinal, el eje x.

$A$ , área de la sección transversal.

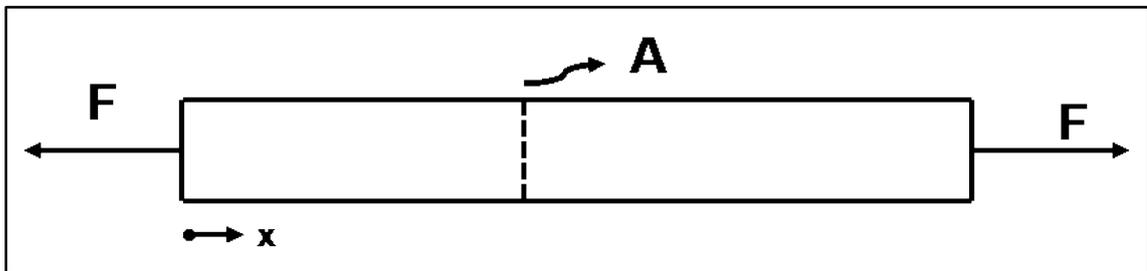


Figura 3. Ilustración de barra sometida a fuerzas axiales.

Fuente: Propio autor.

Para hallar las deformaciones se emplea ley de Hooke, por lo tanto las deformaciones son:

$$\varepsilon_x = \frac{F}{EA}$$

$$\varepsilon_y = -\nu \frac{F}{EA}$$

$$\varepsilon_z = -\nu \frac{F}{EA}$$

Donde:

$\varepsilon_x$ , deformación longitudinal en la dirección del eje x.

$\varepsilon_y$ , deformación longitudinal en la dirección del eje y.

$\varepsilon_z$ , deformación longitudinal en la dirección del eje z.

$F$ , fuerza colineal con el eje longitudinal, el eje x.

$A$ , área de la sección transversal.

$E$ , módulo de rigidez del material de la barra.

#### 1.4.2. Matriz de rigidez (local) de una barra

El desplazamiento de interés para el método es el desplazamiento longitudinal en la dirección  $x$  de la barra, ejes locales. Dicho desplazamiento depende de los apoyos en los extremos de las barra, pero en virtud del método, se considerará un extremo fijo (sin desplazamiento) y el otro extremo libre (permite desplazamiento). Además el inicio del eje  $x$  se sitúa en el extremo fijo. Lo anterior se muestra en la figura 4.

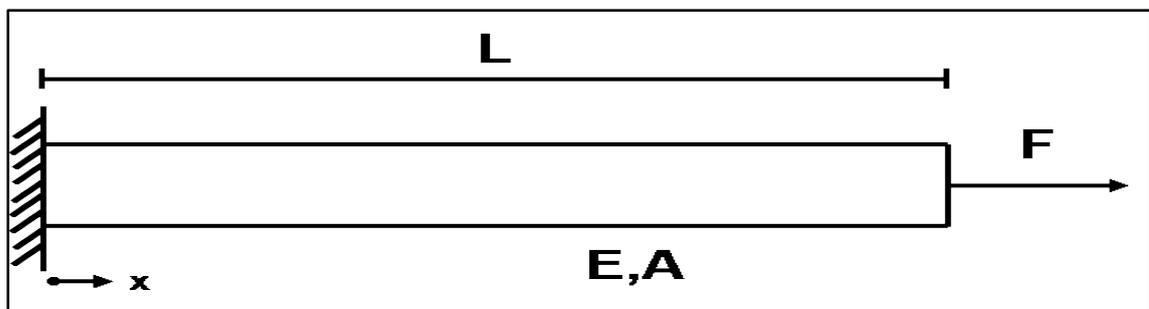


Figura 4. Ilustración de barra sin desplazamiento en un extremo.

Fuente: Propio autor.

Entonces el desplazamiento es:

$$u = \frac{F}{EA}x$$

Donde:

$u$ , desplazamiento en la dirección del eje  $x$ .

$F$ , fuerza colineal con el eje longitudinal, el eje  $x$ .

$A$ , área de la sección transversal.

$E$ , módulo de rigidez del material de la barra.

El desplazamiento en el extremo libre de la barra es:

$$u_{x=L} = \frac{FL}{EA}$$

Donde:

$L$ , longitud inicial de la barra.

Haciendo la analogía de la barra con un resorte, se define la rigidez de la barra como la fuerza necesaria para provocar un desplazamiento unitario en el extremo libre de la barra. Así la rigidez de la barra es:

$$k = \frac{EA}{L}$$

Y la relación de la fuerza con el desplazamiento en el extremo de la barra es ahora:

$$F = ku$$

Debido a que cualquier partícula de la sección transversal se desplaza la misma distancia, basta con conocer el valor del desplazamiento  $u$  para conocer la configuración de la barra.

El método de rigideces relaciona las fuerzas con los desplazamientos en los extremos de las barras (nudos). Para esto, se emplea el método de superposición, al sumar las fuerzas asociadas a los desplazamientos de los nudos de las barras individualmente. Para obtener un desplazamiento cualquiera en cierto nudo, es necesario considerar que los otros nudos se encuentran fijos.

Para una barra se obtiene la matriz de rigidez, que representa la rigidez de la barra al desplazamiento de sus dos nudos, como nudos libres de desplazamiento, figura 5.

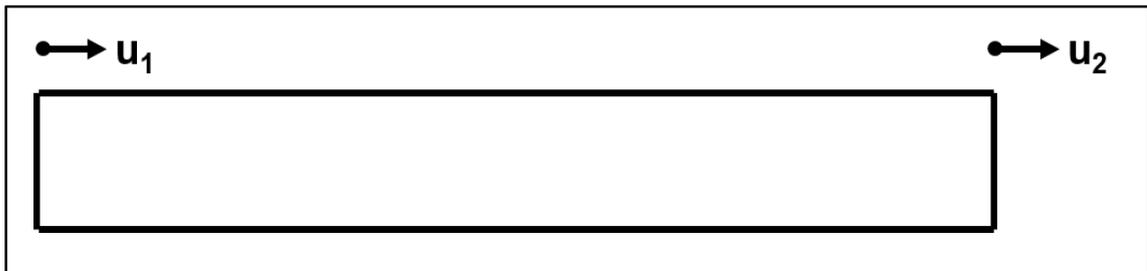


Figura 5. Ilustración de barra con sus dos grados de libertad.

Fuente: Propio autor.

Para generalizar, se supone las fuerzas y los desplazamientos positivos en los nudos, y por practicidad se divide la solución del problema en dos problemas más simples, donde cada problema tiene un solo desplazamiento permitido.

#### Problema 1

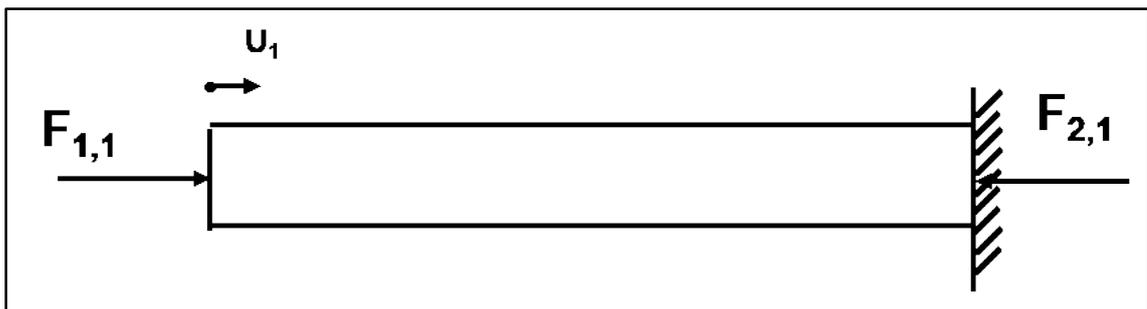


Figura 6. Ilustración de barra con nudo 1 libre.

Fuente: Propio autor.

Dejando libre el nudo 1 (izquierdo) y fijo el nudo 2 (derecho) se tiene la siguiente relación de fuerzas y desplazamientos:

$$\begin{bmatrix} F_{1,1} \\ F_{2,1} \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} \\ -\frac{EA}{L} \end{bmatrix} u_1$$

#### Problema 2

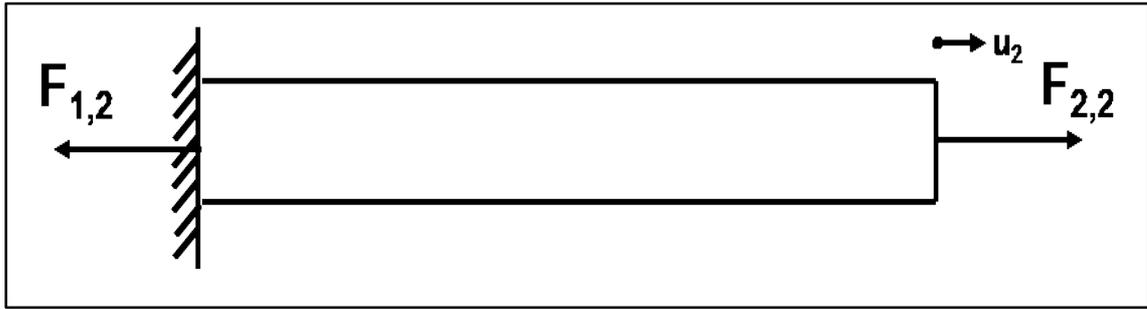


Figura 7. Ilustración de barra con nudo 2 libre.

Fuente: Propio autor.

Dejando libre el nudo 1 (izquierdo) y fijo el nudo 2 (derecho) se tiene la siguiente relación de fuerzas y desplazamientos:

$$\begin{bmatrix} F_{1,2} \\ F_{2,2} \end{bmatrix} = \begin{bmatrix} -\frac{EA}{L} \\ \frac{EA}{L} \end{bmatrix} u_2$$

Como el planteamiento que propone el método es dejando libres ambos nudos, se realiza la superposición de las fuerzas de cada problema, lo que resulta en:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

La matriz de rigidez de una barra es:

$$K = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix}$$

La matriz de rigidez anterior es la matriz de rigidez referida al sistema de coordenadas local, el de la barra, en adelante se le llamará matriz de rigidez local. Para obtener la matriz de rigidez referida al sistema de coordenadas global, el de la armadura, es necesario proyectar la matriz de rigidez local por medio de

una matriz de transformación. La matriz de rigidez referida al sistema de coordenadas global en adelante se le llamará matriz de rigidez global.

#### 1.4.3. Matriz de transformación

La matriz de transformación proyecta un vector en el sistema local a otro vector en el sistema global. El sistema local está referido a la barra, de forma que el eje x coincida con el eje longitudinal de la barra y el sistema global es de la armadura, y es elegido arbitrariamente.

La matriz de transformación es.

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix}$$

#### 1.4.4. Matriz de rigidez global de una barra

Debido a que la matriz de rigidez está asociada a un vector fuerza y a un vector desplazamientos, la matriz de rigidez en coordenadas locales debe ser premultiplicada por la matriz de transformación y posmultiplicada por la matriz de transformación transpuesta para obtener la matriz de rigidez global.

#### 1.4.5. Matriz de rigidez global de una armadura

Para obtener la matriz de rigidez global de la armadura basta realizar la superposición de las matrices de rigidez global de todas las barras según los grados de libertad que les corresponden. A lo anterior se le llama ensamble de la matriz de rigidez global de la armadura.

La superposición es válida debido a que los desplazamientos son pequeños, y estos no modifican sustancialmente las características geométricas de la barra, y por ende la rigidez se mantiene constante. Otra razón de la aplicación de la superposición es la naturaleza de la ley de Hooke, donde existe

una relación lineal entre el esfuerzo y la deformación. Y una última justificación de la superposición es la naturaleza de la ecuación diferencial del equilibrio.

Las características de la matriz de rigidez desde el punto de vista del álgebra de matrices, consisten en ser una matriz simétrica, positiva y singular. Además dicha matriz generalmente está plagada de ceros, lo que conlleva emplear técnicas de solución más rápidas.

## 1.5. Solución de sistemas de ecuaciones lineales

### 1.5.1. Sistemas de ecuaciones lineales

Un sistema de ecuaciones lineales es un conjunto de ecuaciones de la forma:

$$\begin{aligned}a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n &= b_1 \\a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n &= b_2 \\a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n &= b_m\end{aligned}$$

Donde:

$a_{i,j}$ , coeficientes de las incógnitas.

$x_i$ , incógnitas.

$b_i$ , términos independientes.

$n$ , número de incógnitas.

$m$ , número de ecuaciones.

Matricialmente se puede representar como:

$$\begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Se define la matriz del sistema (matriz de rigidez en el método) como:

$$\bar{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix}$$

Al vector de incógnitas (vector de desplazamiento en el método) como:

$$\bar{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Al vector de términos independientes (vector de condiciones de frontera en el método) como:

$$\bar{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Existen múltiples métodos de resolver sistemas de ecuaciones, las de interés en el texto son aquellas que sean metodológicas y sistemáticas. Aquí se plantea la forma más usual en la programación para la solución de sistemas de ingeniería estructural: Eliminación de Gauss.

### 1.5.2. Eliminación de Gauss

Este enfoque consiste en realizar operaciones entre renglones de la matriz ampliada, formada por la matriz del sistema y el vector de términos independientes, hasta que la matriz del sistema se convierta en una matriz triangular superior.

Dichas operaciones consisten en: elegir un elemento pivote en la diagonal principal, yendo de arriba abajo. El renglón de la matriz ampliada con el elemento pivote es dividida entre el mismo elemento pivote. Después se resta el renglón modificado que contiene al elemento pivote multiplicado por el elemento en la misma columna de cierto renglón distinto. Se repiten los pasos hasta que la matriz del sistema se convierte en una matriz triangular superior. Lo anterior se resume en lo siguiente:

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} a_{k,j}^{(k-1)} \quad i = k + 1, \dots, n; j = k + 1, \dots, n$$

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} b_k^{(k-1)} \quad i = k + 1, \dots, n$$

En la programación para la solución de sistemas de ingeniería estructural se emplea una forma reducida del método al considerar las propiedades de la matriz de rigidez.

### 1.6. Matriz de rigidez y su transformación a una matriz no singular

Es notable que la matriz de rigidez de la armadura es una matriz singular, es decir, no tienen inversa, y por ende el sistema de ecuaciones no tiene solución, sistema incompatible. Para que el sistema de ecuaciones tenga solución, es necesario introducir las condiciones de frontera que proporcionen estabilidad a la estructura y hagan a la matriz de rigidez una matriz no singular.

Existen dos enfoques para introducir las condiciones de frontera: de eliminación y de penalización.

#### 1.6.1. Enfoque de eliminación

Consiste en eliminar los renglones y columnas de la matriz de rigidez asociados a los nudos donde se conoce el desplazamiento, eliminar el término del vector de fuerzas de los renglones eliminados y modificar el vector de fuerzas anterior al restar la suma de los productos de los desplazamientos conocidos con la rigidez asociada al desplazamiento, esto para cada renglón.

Se resuelve dicho sistema de ecuaciones y se obtienen los desplazamientos, finalmente se calculan las reacciones (negativos de las fuerzas en los nudos de desplazamientos conocidos) al sustituir los desplazamientos obtenidos en los renglones eliminados.

### 1.6.2. Enfoque de penalización

Consiste en considerar un resorte de gran rigidez en los desplazamientos conocidos de los nudos. De forma que el desplazamiento debido a las cargas en los nudos donde están los resortes sea prácticamente cero. Lo anterior resulta más trabajoso, pero para una computadora, las propiedades de la matriz de rigidez original pueden ser aplicadas, y no así si se eliminan columnas y renglones.

Lo anterior se resume en sumar a la rigidez del nudo la rigidez del resorte, esto en la diagonal principal, obviamente. Y sumar la fuerza que produce el resorte dado el desplazamiento conocido, en otras palabras, sumar el producto de la rigidez del resorte con el desplazamiento conocido.

Se resuelve dicho sistema de ecuaciones y se obtienen los desplazamientos. Las fuerzas de reacción se calculan al restar la fuerza asociada al desplazamiento ficticio con el resorte menos la fuerza provocada por el resorte con el desplazamiento conocido.

## **2. Resumen del programa**

Wolfram Mathematica es un programa que se emplea para resolver problemas matemáticos, pero es también un lenguaje de programación. El programa creado usa la naturaleza de lenguaje de programación, aunque también se apoya en las funciones que por defecto tiene para resolver problemas matemáticos.

Wolfram Mathematica utiliza la interface denominada *Cuaderno*, donde el usuario interactúa con el programa, esta interface permite dividir el área de trabajo en diferentes secciones, que pueden relacionarse o no entre sí.

Las funciones empleadas en el código son:

**Module:** Permite crear funciones propias, la función creada se podrá ejecutar las veces que sean y se emplean las variables que se determinen según la función creada.

**Union:** Permite hacer la unión de dos conjuntos diferentes, como dos conjuntos de pares ordenados que representan puntos en el espacio.

**Length:** Calcula longitud de un vector, en el sentido de encontrar el número de elementos que contiene el vector.

**Position:** Encuentra una variable o vector dentro de un conjunto indicando la posición en que se encuentra según el orden en el conjunto.

**Graphics:** Permite graficar en dos dimensiones, entre los objetos que puede graficar se encuentran líneas y puntos.

**For:** Permite generar un ciclo donde las operaciones dentro el ciclo se repiten hasta que cierta variables alcanza un determinado valor.

**Max:** Encuentra el valor máximo en un conjunto de números.

**Abs:** Permite obtener el valor absoluto de un número.

If: Permite realizar una operación u otra dependiendo del valor de una variable.

El programa creado emplea la función Module, para poder realizar las operaciones un sinnúmero de veces y solamente, de ser el caso, cambiar el valor de las variables utilizadas. Se divide en tres secciones fundamentales, que se describen a continuación.

## 2.1. Entrada

Tiene una subsección llamada “Obtención de la matriz de rigidez de la armadura y elaboración de gráfica descriptiva de la armadura”.

### 2.1.1. Obtención de la matriz de rigidez de la armadura y elaboración de gráfica descriptiva de la armadura

Aquí se ingresan los datos: Número de barras, coordenadas de los nudos de las barras, el área y módulo de rigidez de las barras y las condiciones de frontera de fuerzas y desplazamientos.

Primeramente se almacenan las coordenadas de los a partir de las coordenadas ingresadas de las barras con la función Union, esta función evita repetir coordenadas de los nudos. También se obtiene el número de nudos con la función Length.

```
CooNod = CooEle[[1]];

For[i = 2, i ≤ NumEle, i++,
|para cada
  CooNod = Union[CooNod, CooEle[[i]]
|unión
];
|
NumNod = Length[CooNod];
|longitud
```

Figura 8. Fragmento del código para hallar las coordenadas de los nudos y el número de nudos.

Fuente: Propio autor.

Después se identifican para cada barra los números de los nudos extremos por medio de la función Position. Y a continuación también se identifican los números de los grados de libertad de los extremos de las barras. Dichos números están referenciados a una lista ordenada de todos los nudos y a una lista ordenada de todos los grados de libertad.

```
IdeEleNod = Table[0, {NumEle}, {2}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++,
|para cada |para cada
IdeEleNod[i, j] = Norm[Flatten[Position[CooNod, CooEle[i, j]]]]
|norma |aplana |posición
]
];

IdeEleGraLib = Table[0, {NumEle}, {4}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 4, j++,
|para cada |para cada
If[j ≤ 2,
|si
IdeEleGraLib[i, j] = ((2 * (IdeEleNod[i, 1] - 1)) + j),
IdeEleGraLib[i, j] = ((2 * (IdeEleNod[i, 2] - 1)) + (j - 2))
]
]
];
```

Figura 9. Fragmento del código para identificar los números de los grados de libertad y los nudos en los extremos de las barras.

Fuente: Propio autor.

Ahora se obtiene la longitud de las barras, por medio del teorema de Pitágoras. Y después se encuentra la matriz de transformación, para ello primero se hallan los cosenos directores de vectores ficticios que van a lo largo de la barra, y tiene por punto de origen la primer coordenada ingresada. También se obtiene la matriz inversa de transformación al transponer la matriz de transformación.

```

LonEle = Table[0, {NumEle}];
      |tabla
For[i = 1, i ≤ NumEle, i++,
  |para cada
  LonEle[[i]] = (Sqrt[(CooEle[[i, 2, 1]] - CooEle[[i, 1, 1]])2 + (CooEle[[i, 2, 2]] - CooEle[[i, 1, 2]])2])
      |raíz cuadrada
];

```

Figura 10. Fragmento del código para hallar la longitud de las barras.

Fuente: Propio autor.

```

CosAngEle = Table[0, {NumEle}, {2}];
      |tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++,
  |para cada |para cada
  CosAngEle[[i, j]] = ((CooNod[[IdeEleNod[[i, 2]], j]] - CooNod[[IdeEleNod[[i, 1]], j]]) / LonEle[[i]])
  ]
];

MatTra = Table[0, {NumEle}, {1}, {2}, {4}];
      |tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++, For[k = 1, k ≤ 4, k++,
  |para cada |para cada |para cada
  If[j == 1,
  |si
  If[k ≤ 2,
  |si
  MatTra[[i, 1, j, k]] = CosAngEle[[i, k]]
  ],
  If[k > 2,
  |si
  MatTra[[i, 1, j, k]] = CosAngEle[[i, k - 2]]
  ]
  ]
];

```

Figura 11. Fragmento del código para hallar la matriz de transformación de las barras.

Fuente: Propio autor.

```

MatInvTra = Table[0, {NumEle}, {1}, {4}, {2}];
      |
      |tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 4, j++, For[k = 1, k ≤ 2, k++,
|para cada          |para cada          |para cada
  If[j ≤ 2,
  |si
    If[k == 1,
    |si
      MatInvTra[[i, 1, j, k]] = CosAngEle[[i, j]]
    ],
    If[k == 2,
    |si
      MatInvTra[[i, 1, j, k]] = CosAngEle[[i, j - 2]]
    ]
  ]
]
]
];

```

Figura 12. Fragmento del código para hallar la matriz inversa de transformación de las barras.

Fuente: Propio autor.

Posteriormente se obtiene la matriz de rigidez local de las barras utilizando la longitud de los elementos y los módulos de rigidez y áreas de las secciones transversales ingresadas. Dichas matrices son de orden 2x2.

```

MatRigEleLoc = Table[0, {NumEle}, {1}, {2}, {2}];
      |
      |tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++, For[k = 1, k ≤ 2, k++,
|para cada          |para cada          |para cada
  If[j == k,
  |si
    MatRigEleLoc[[i, 1, j, k]] = (((AreEle[[i]] * ModRigEle[[i]]) / LonEle[[i]]),
    MatRigEleLoc[[i, 1, j, k]] = -(((AreEle[[i]] * ModRigEle[[i]]) / LonEle[[i]]))
  ]
]
]
];

```

Figura 13. Fragmento del código para obtener la matriz de rigidez local de las barras.

Fuente: Propio autor.

A continuación se obtiene la matriz de rigidez global de la barras al premultiplicar la matriz de rigidez local por la matriz de transformación y posmultiplicar por la matriz inversa de transformación.

```

MatApoRigEleGlo = Table[0, {NumEle}, {1}, {2}, {4}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++, For[k = 1, k ≤ 2, k++, For[h = 1, h ≤ 4, h++,
|para cada |para cada |para cada |para cada
    MatApoRigEleGlo[[i, 1, j, h]] += (MatRigEleLoc[[i, 1, j, k]] * MatTra[[i, 1, k, h]])

    ]
    ]
    ]
];

MatRigEleGlo = Table[0, {NumEle}, {1}, {4}, {4}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 4, j++, For[k = 1, k ≤ 2, k++, For[h = 1, h ≤ 4, h++,
|para cada |para cada |para cada |para cada
    MatRigEleGlo[[i, 1, j, h]] += (MatInvTra[[i, 1, j, k]] * MatApoRigEleGlo[[i, 1, k, h]])

    ]
    ]
    ]
];

```

Figura 14. Fragmento del código para obtener la matriz de rigidez global de las barras.

Fuente: Propio autor.

Después se obtiene la matriz de rigidez global de la armadura al ensamblar las matrices de rigidez globales de las barras, para ello se emplean los identificadores de los grados de libertad de los extremos de las barras.

```

MatRigEstGlo = Table[0, {NumDesPos}, {NumDesPos}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 4, j++, For[k = 1, k ≤ 4, k++,
|para cada |para cada |para cada
    MatRigEstGlo[[IdeEleGraLib[[i, j]], IdeEleGraLib[[i, k]]]] += MatRigEleGlo[[i, 1, j, k]]

    ]
    ]
];

```

Figura 15. Fragmento del código para obtener la matriz de rigidez global de la armadura.

Fuente: Propio autor.

Ahora se almacenan las condiciones de frontera de desplazamiento y de fuerza identificando el número del grado de libertad que les corresponde.

```
VecConFroFue = Table[0, {NumDesDes}];  
  |tabla  
For[i = 1, i ≤ Length[ConFroFue], i++, For[j = 1, j ≤ 2, j++,  
  |para cada |longitud |para cada  
  VecConFroFue[[Flatten[Position[DesDes, ((2 * (ConFroFue[[i, 2] - 1) + j)]]]] = ConFroFue[[i, 1, j]]  
    |aplana |posición  
  ]  
];  
  
VecConFroDes = Table[0, {NumDesCon}];  
  |tabla  
For[i = 1, i ≤ NumDesCon, i++,  
  |para cada  
  VecConFroDes[[i] = ConFroDes[[i, 3]]  
];
```

Figura 16. Fragmento del código identificar el número de los grados de libertad donde se encuentran las condiciones de frontera de desplazamiento y fuerza.

Fuente: Propio autor.

Finalmente se genera una gráfica de la armadura por medio de la función Graphics. En dicha gráfica se identifican el número de las barras y de los nudos.

```

GraEst = Graphics[{
  gráfico
  Style[Line[CooEle], GrayLevel[0.9], Thickness[0.0025]],
  |estilo |línea |nivel de gris |grosor
  Table[Style[Text[i, ((1/2) * CooEle[[i, 2]]) + ((1/2) * CooEle[[i, 1]]) + {(0.1 * Min[LonEle]), (0.1 * Min[LonEle])}], Black, FontWeight -> Bold,
  |tabla |estilo |texto |mínimo |mínimo |negro |peso de tipo de |negrita
    FontFamily -> "Arial", FontSize -> 10], {i, 1, NumEle}],
  |familia de tipo de letra |tamaño de tipo de letra

  Style[Point[CooNod], Blue, PointSize[0.005]],
  |estilo |punto |azul |tamaño de punto

  Table[Style[Text[i, CooNod[[i]] + {(0.1 * Min[LonEle]), (0.1 * Min[LonEle])}], Blue, FontWeight -> Bold, FontFamily -> "Arial", FontSize -> 10], {i, 1, NumNod}],
  |tabla |estilo |texto |mínimo |mínimo |azul |peso de tipo de |negrita |familia de tipo de letra |tamaño de tipo de letra

  Table[Style[Arrow[{CooNod[ConFroFue[[i, 2]]], ((ConFroFue[[i, 1]] / (3 * Max[Abs[ConFroFue[[i, 1]]]) * (3 * Min[LonEle]))) + CooNod[ConFroFue[[i, 2]]]}],
  |tabla |estilo |flecha |má- |valor absoluto |mínimo
    Red, Thickness[0.0025], Arrowheads[0.01], {i, 1, Length[ConFroFue]}],
  |rojo |grosor |cabeceras de flecha |longitud

  Table[Style[Text[ConFroFue[[i, 1]], ((1/2) * ((ConFroFue[[i, 1]] / (3 * Max[Abs[ConFroFue[[i, 1]]]) * (3 * Min[LonEle])))) + CooNod[ConFroFue[[i, 2]]] -
  |tabla |estilo |texto |má- |valor absoluto |mínimo
    {(0.3 * Min[LonEle]), (0.3 * Min[LonEle])}], Red, FontWeight -> Bold, FontFamily -> "Arial", FontSize -> 10], {i, 1, Length[ConFroFue]}
    |mínimo |mínimo |rojo |peso de tipo de |negrita |familia de tipo de letra |tamaño de tipo de letra |longitud
  }];

```

Figura 17. Fragmento del código para generar la gráfica de la armadura señalando el número de las barras y los nudos.

Fuente: Propio autor.

## 2.2. Solución

Esta sección se divide en subsecciones, la subsección “Obtención del vector de desplazamiento y vector de fuerza de la armadura” y la subsección “Obtención de las fuerzas y desplazamientos de cada barra”.

### 2.2.1 Obtención del vector de desplazamiento y vector de fuerza de la armadura

Esta subsección se divide en dos subsubsecciones diferenciadas por el enfoque aplicado para convertir la matriz de rigidez a una matriz singular.

#### 2.2.1.1 Enfoque de eliminación

En la subsubsección “Enfoque de eliminación” se emplea el enfoque del mismo nombre.

Primeramente se obtiene la matriz del sistema al eliminar las filas y columnas de los desplazamientos desconocidos según las condiciones de frontera de desplazamientos en la matriz de rigidez global de la armadura.

```
MatRigEst = Table[0, {NumDesDes}, {NumDesDes}];  
|tabla  
For[i = 1, i ≤ NumDesDes, i++, For[j = 1, j ≤ NumDesDes, j++],  
|para cada |para cada  
MatRigEst[[i, j]] = MatRigEstGlo[[DesDes[[i]], DesDes[[j]]]]  
]  
];
```

Figura 18. Fragmento del código para encontrar la matriz del sistema del sistema de ecuaciones lineales a resolver.

Fuente: Propio autor.

Después se obtiene el vector de términos independientes. Para ello se agregan las condiciones de frontera de fuerza restadas por las fuerzas provocadas por las condiciones de frontera de desplazamiento. Las fuerzas de las condiciones de frontera de desplazamientos se obtienen al multiplicar los desplazamientos conocidos por la rigidez correspondiente dentro de la matriz de rigidez global de la armadura.

```
VecFueEst = Table[0, {NumDesDes}];  
|tabla  
For[i = 1, i ≤ NumDesDes, i++, For[j = 1, j ≤ NumDesCon, j++],  
|para cada |para cada  
VecFueEst[[i]] += (MatRigEstGlo[[DesDes[[i]], DesCon[[j]]]] * VecConFroDes[[j]])  
]  
];  
  
For[i = 1, i ≤ NumDesDes, i++],  
|para cada  
VecFueEst[[i]] = (VecConFroFue[[i]] - VecFueEst[[i]])  
];
```

Figura 19. Fragmento del código para encontrar el vector de términos independientes del sistema de ecuaciones lineales a resolver.

Fuente: Propio autor.

A continuación se utiliza la eliminación de Gauss para hallar el vector de incógnitas. Al ser un proceso iterativo se utilizan múltiples ciclos For.

```
For[k = 1, k ≤ (NumDesDes - 1), k++, For[i = (k + 1), i ≤ NumDesDes, i++,
|para cada                               |para cada

    EscApoRigEst = (MatRigEst[[i, k]] / MatRigEst[[k, k]]);

    For[j = (k + 1), j ≤ NumDesDes, j++,
|para cada
        MatRigEst[[i, j]] = (MatRigEst[[i, j]] - (EscApoRigEst * MatRigEst[[k, j]]));
    ];
    VecFueEst[[i]] = (VecFueEst[[i]] - (EscApoRigEst * VecFueEst[[k]]))
];

VecDesEst = Table[0, {NumDesDes}];
|tabla
VecDesEst[[NumDesDes]] = (VecFueEst[[NumDesDes]] / MatRigEst[[NumDesDes, NumDesDes]]);

For[i = 1, i ≤ (NumDesDes - 1), i++,
|para cada
    EscApoDesEst = 0;

    For[j = ((NumDesDes - i) + 1), j ≤ NumDesDes, j++,
|para cada
        EscApoDesEst += (MatRigEst[[NumDesDes - i, j]] * VecDesEst[[j]])
    ];

    VecDesEst[[NumDesDes - i]] = ((1 / MatRigEst[[NumDesDes - i, NumDesDes - i]]) * (VecFueEst[[NumDesDes - i]] - EscApoDesEst))
];
```

Figura 20. Fragmento del código para encontrar el vector de incógnitas por medio de la eliminación de Gauss.

Fuente: Propio autor.

Posteriormente se obtiene el vector de desplazamientos global al combinar el vector de incógnitas y las condiciones de frontera de desplazamientos.

```

VecDesEstGlo = Table[0, {NumDesPos}];
|tabla
For[i = 1, i ≤ NumDesCon, i++,
|para cada
  VecDesEstGlo[DesCon[i]] = VecConFroDes[i]
];

For[i = 1, i ≤ NumDesDes, i++,
|para cada
  VecDesEstGlo[DesDes[i]] = VecDesEst[i]
];

```

Figura 21. Fragmento del código para encontrar el vector de desplazamientos global.

Fuente: Propio autor.

Finalmente se obtiene el vector de fuerzas global con las condiciones de frontera de fuerzas y al calcular las fuerzas actuantes a partir del vector de desplazamientos y la matriz de rigidez global. Para calcular las fuerzas debido a los desplazamientos se multiplica cada desplazamiento por su elemento correspondiente en la matriz de rigidez y se suma todas las multiplicaciones del renglón asociado a la fuerza.

```

VecFueEstGlo = Table[0, {NumDesPos}];
|tabla
For[i = 1, i ≤ NumDesDes, i++,
|para cada
  VecFueEstGlo[DesDes[i]] = VecConFroFue[i]
];

For[i = 1, i ≤ NumDesCon, i++, For[j = 1, j ≤ NumDesPos, j++,
|para cada |para cada
  VecFueEstGlo[DesCon[i]] += (MatRigEstGlo[DesCon[i], DesPos[j]] * VecDesEstGlo[j])
];
];

```

Figura 22. Fragmento del código para encontrar el vector de fuerzas global.

Fuente: Propio autor.

### 2.2.1.2 Enfoque de penalización

En la subsubsección “Enfoque de penalización” se emplea el enfoque del mismo nombre.

Primero se determina una rigidez muy grande, que será la de los resortes ficticios ubicados en las condiciones de frontera de desplazamientos. La referencia [6] recomienda que esta sea de cuatro órdenes de magnitud mayor que la rigidez máxima en la matriz de rigidez global.

```
EscApoRigFic = (Max[MatRigEstGlo] * 104);  
                [máximo]
```

Figura 23. Fragmento del código para encontrar la rigidez de apoyo del enfoque de penalización.

Fuente: Propio autor.

Después se obtiene la matriz del sistema. Para ello en la diagonal principal de los matriz de rigidez en los renglones de los desplazamientos conocidos se suma a la rigidez propia de la matriz la rigidez de los resortes ficticios.

```
MatRigEst = Table[0, {NumDesPos}, {NumDesPos}];  
                [tabla]  
|  
For[i = 1, i ≤ NumDesPos, i++,  
  [para cada  
    For[j = 1, j ≤ NumDesPos, j++,  
      [para cada  
        MatRigEst[[i, j]] = MatRigEstGlo[[i, j]]  
      ]  
    ]  
  ]  
];  
  
For[i = 1, i ≤ NumDesCon, i++,  
  [para cada  
    MatRigEst[[DesCon[[i]], DesCon[[i]]]] = MatRigEstGlo[[DesCon[[i]], DesCon[[i]]]] + EscApoRigFic  
  ]  
];
```

Figura 24. Fragmento del código para encontrar la matriz del sistema del sistema de ecuaciones lineales a resolver.

Fuente: Propio autor.

A continuación se obtiene el vector de términos independientes. Para ello se combinan las condiciones de frontera de fuerzas y, utilizando las condiciones de frontera de desplazamientos y reconociendo que existe un resorte de gran rigidez en dichas condiciones, se obtiene el término independiente correspondiente al multiplicar las condiciones de frontera de desplazamientos, magnitud del desplazamiento, por la rigidez del resorte.

```
VecFueEst = Table[0, {NumDesPos}];  
|tabla  
For[i = 1, i ≤ NumDesDes, i++,  
|para cada  
VecFueEst[[DesDes[[i]]] = VecConFroFue[[i]]  
];  
  
For[i = 1, i ≤ NumDesCon, i++,  
|para cada  
VecFueEst[[DesCon[[i]]] = (EscApoRigFic * VecConFroDes[[i]])  
];
```

Figura 25. Fragmento del código para encontrar el vector de términos independientes del sistema de ecuaciones lineales a resolver.

Fuente: Propio autor.

Posteriormente se utiliza la eliminación de Gauss para hallar el vector de incógnitas. Al ser un proceso iterativo se utilizan múltiples ciclos For.

```

For[k = 1, k ≤ (NumDesPos - 1), k++, For[i = (k + 1), i ≤ NumDesPos, i++,
|para cada                               |para cada

    EscApoRigEst = (MatRigEst[[i, k]] / MatRigEst[[k, k]]);

    For[j = (k + 1), j ≤ NumDesPos, j++,
|para cada
        MatRigEst[[i, j]] = (MatRigEst[[i, j]] - (EscApoRigEst * MatRigEst[[k, j]]));
    ];
    VecFueEst[[i]] = (VecFueEst[[i]] - (EscApoRigEst * VecFueEst[[k]]))
];

VecDesEst = Table[0, {NumDesPos}];
|tabla
VecDesEst[[NumDesPos]] = (VecFueEst[[NumDesPos]] / MatRigEst[[NumDesPos, NumDesPos]]);

For[i = 1, i ≤ (NumDesPos - 1), i++,
|para cada
    EscApoDesEst = 0;

    For[j = ((NumDesPos - i) + 1), j ≤ NumDesPos, j++,
|para cada
        EscApoDesEst += (MatRigEst[[NumDesPos - i, j]] * VecDesEst[[j]]);
    ];

    VecDesEst[[NumDesPos - i]] = ((1 / MatRigEst[[NumDesPos - i, NumDesPos - i]]) * (VecFueEst[[NumDesPos - i]] - EscApoDesEst))
];

```

Figura 26. Fragmento del código para encontrar el vector de incógnitas por medio de la eliminación de Gauss.

Fuente: Propio autor.

Después se obtiene el vector de desplazamientos global que resulta ser el mismo que el vector de incógnitas.

```

VecDesEstGlo = Table[0, {NumDesPos}];
|tabla
For[i = 1, i ≤ NumDesPos, i++,
|para cada
    VecDesEstGlo[[i]] = VecDesEst[[i]]
];

```

Figura 27. Fragmento del código para encontrar el vector de desplazamientos global.

Fuente: Propio autor.

Finalmente se obtiene el vector de fuerzas global con las condiciones de frontera de fuerzas. Para ello se calculan las fuerzas actuantes a partir del vector de desplazamientos, las condiciones de frontera de desplazamientos y la rigidez de los resortes ficticios. Para calcular dichas fuerzas se resta el desplazamiento debido al vector de desplazamientos al desplazamiento de las condiciones de frontera de desplazamientos, y el resultado se multiplica por el negativo de la rigidez de los resortes ficticios.

```
VecFueEstGlo = Table[0, {NumDesPos}];  
|tabla  
For[i = 1, i ≤ NumDesDes, i++,  
|para cada  
VecFueEstGlo[[DesDes[i]]] = VecConFroFue[i]  
];  
  
For[i = 1, i ≤ NumDesCon, i++,  
|para cada  
VecFueEstGlo[[DesCon[i]]] = (- (EscApoRigFic * (VecDesEstGlo[[DesCon[i]]] - VecConFroDes[i])))  
];
```

Figura 28. Fragmento del código para encontrar el vector de fuerzas global.

Fuente: Propio autor.

### 2.2.2. Obtención de las fuerzas y desplazamientos de cada barra

En la subsección “Obtención de las fuerzas y desplazamientos de cada barra” se obtienen las fuerzas y los desplazamientos de cada barra.

Para lo anterior, primero se obtiene el vector de desplazamientos global de cada barra al extraer los desplazamientos del vector de desplazamientos global de la armadura usando los identificadores de los grados de libertad de las barras, que indican los grados de libertad de las barras en el sistema global.

```

For [i = 1, i ≤ NumEle, i++, For [j = 1, j ≤ 4, j++,
|para cada |para cada
    VecDesEleGlo[i, 1, j] = VecDesEstGlo[IdeEleGralib[i, j]]
]
];

```

Figura 29. Fragmento del código para encontrar el vector de desplazamientos global de cada barra.

Fuente: Propio autor.

Después se obtiene el vector de fuerzas global de cada barra, para encontrar dicho vector se premultiplica la matriz de rigidez global de cada barra por el vector de desplazamientos global de cada barra.

```

VecFueEleGlo = Table[0, {NumEle}, {1}, {4}];
|tabla
For [i = 1, i ≤ NumEle, i++, For [j = 1, j ≤ 4, j++, For [k = 1, k ≤ 4, k++,
|para cada |para cada |para cada
    VecFueEleGlo[i, 1, j] += (MatRigEleGlo[i, 1, j, k] * VecDesEleGlo[i, 1, k])
]
]
];

```

Figura 30. Fragmento del código para encontrar el vector de fuerzas global de cada barra.

Fuente: Propio autor.

Finalmente se obtiene el vector de desplazamientos local y el vector de fuerzas local de cada barra. Para esto simplemente se utiliza la matriz de transformación. Se premultiplica la matriz de transformación por cualquier vector en coordenadas globales para obtener el vector en coordenadas locales.

```

VecDesEleLoc = Table[0, {NumEle}, {1}, {2}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++, For[k = 1, k ≤ 4, k++,|
|para cada |para cada |para cada
VecDesEleLoc[[i, 1, j]] += (MatTra[[i, 1, j, k]] * VecDesEleGlo[[i, 1, k]])
]
]
];

VecFueEleLoc = Table[0, {NumEle}, {1}, {2}];
|tabla
For[i = 1, i ≤ NumEle, i++, For[j = 1, j ≤ 2, j++, For[k = 1, k ≤ 4, k++,|
|para cada |para cada |para cada
VecFueEleLoc[[i, 1, j]] += (MatTra[[i, 1, j, k]] * VecFueEleGlo[[i, 1, k]])
]
]
];

```

Figura 31. Fragmento del código para encontrar el vector de desplazamientos local y el vector de fuerzas local de cada barra.

Fuente: Propio autor.

## 2.3. Análisis de resultados

Esta sección se divide en cinco subsecciones: “Fuerza máxima”, “Desplazamiento máximo”, “Gráfica de la armadura deformada”, “Gráfica de la armadura mostrando fuerzas” y “Gráfica de la armadura mostrando desplazamientos”.

### 2.3.1. Fuerza máxima

La subsección “Fuerza máxima” permite obtener la fuerza máxima en un grupo de barras elegido.

Aquí se ingresan los datos: Barras de comparación para la fuerza máxima.

Primeramente se obtiene la magnitud de la fuerza máxima por medio de la función Max y la función Abs, para que el signo matemático sea indiferente. Se comparan las barras de comparación para la fuerza máxima ingresadas anteriormente.

```

FueMax = 0;

For [i = 1, i ≤ Length[EleComFueMax], i++,
|para cada      |longitud
FueMax = Max[FueMax, Abs[VecFueEleLoc[[EleComFueMax[[i], 1, 1]]]
|máximo      |valor absoluto
];

```

Figura 32. Fragmento del código para encontrar la magnitud de la fuerza máxima.

Fuente: Propio autor.

Después se identifica el número de la barra donde ocurre dicha fuerza máxima por medio de la función If.

```

For [i = 1, i ≤ NumEle, i++,
|para cada
If [(0.999 * FueMax) ≤ Abs[VecFueEleLoc[[i, 1, 1]]] ≤ (1.001 * FueMax),
|si      |valor absoluto
EleFueMax = i
]
];

```

Figura 33. Fragmento del código para encontrar la barra donde se presenta la fuerza máxima.

Fuente: Propio autor.

### 2.3.2. Desplazamiento máximo

La subsección “Desplazamiento máximo” permite obtener el desplazamiento máximo según cierta dirección seleccionada para un grupo de nudos elegidos.

Aquí se ingresan los datos: Nudos de comparación para el desplazamiento máximo y dirección del desplazamiento donde se quiere determinar el máximo.

Primeramente se obtiene la magnitud del desplazamiento máximo por medio de la función Max, y para que el signo matemático sea indiferente se emplea la función Abs, Se comparan los nudos de comparación para el desplazamiento máximo según la dirección donde se desea conocer el desplazamiento máximo.

```
DesMax = 0;

For [i = 1, i ≤ Length[NodComDesMax], i++,
|para cada |longitud
DesMax = Max[DesMax, Abs[VecDesEstGlo[(2 * (NodComDesMax[i] - 1)) + DirDesMax]]]
|máximo |valor absoluto
];
```

Figura 34. Fragmento del código para encontrar la magnitud del desplazamiento máximo.

Fuente: Propio autor.

Finalmente se identifica el número del nudo donde ocurre el desplazamiento máximo por medio de la función If.

```
For [i = 1, i ≤ NumNod, i++,
|para cada
If [(0.999 * DesMax) ≤ Abs[VecDesEstGlo[(2 * (i - 1)) + DirDesMax]]] ≤ (1.001 * DesMax),
|si |valor absoluto
NodDesMax = i
]
];
```

Figura 35. Fragmento del código para encontrar el nudo donde se presenta el desplazamiento máximo.

Fuente: Propio autor.

### 2.3.3. Gráfica de la armadura deformada

La subsección “Gráfica de la armadura deformada” se grafica la armadura en su configuración original y en su configuración deformada, exagerando los desplazamientos nodales con el propósito de ser observados mejor. También se

muestran las reacciones en los grados de libertad donde el desplazamiento está predeterminado.

```

GraEstyGraEstDes = Graphics[
  |grafico
  Style[Line[CooEle], GrayLevel[0.9], Thickness[0.0025]],
  |estilo |linea |nivel de gris |grosor
  Table[Style[Line[{CooEle[[i, 1]] + ((Min[LonEle] / (10 * Max[Abs[VecDesEstLo]])) * {VecDesEstLo[[i, 1, 1], VecDesEstLo[[i, 1, 2]]}],
  |tabla |estilo |linea |mínimo |má- |valor absoluto
  CooEle[[i, 2]] + ((Min[LonEle] / (10 * Max[Abs[VecDesEstLo]])) * {VecDesEstLo[[i, 1, 3], VecDesEstLo[[i, 1, 4]]}], Pink, Thickness[0.0025], Dashed], {i, 1, NumEle}],
  |mínimo |má- |valor absoluto |rosa |grosor |rayado
  Style[Point[CooMod], Blue, PointSize[0.005]],
  |estilo |punto |azul |tamaño de punto
  Table[Style[Point[CooMod[[i]] + ((Min[LonEle] / (10 * Max[Abs[VecDesEstLo]])) * {VecDesEstLo[[2 * (i - 1) + 1], VecDesEstLo[[2 * (i - 1) + 2]]}], Gray, PointSize[0.005]], {i, 1, NumMod}],
  |tabla |estilo |punto |mínimo |má- |valor absoluto |gris |tamaño de punto
  Table[
  |tabla
  Style[Arrow[{CooMod[ConFroDes[[i, 1]]], CooMod[ConFroDes[[i, 1]]] + ((VecFueEstLo[[2 * (ConFroDes[[i, 1] - 1) + ConFroDes[[i, 2]]] / (3 * Max[Abs[VecFueEstLo]] * (3 * Min[LonEle])) *
  |estilo |flecha
  {If[ConFroDes[[i, 2]] == 1, 1, 0], If[ConFroDes[[i, 2]] == 2, 1, 0}], Red, Thickness[0.0025], Arrowheads[0.01], {i, 1, Length[ConFroDes]}],
  |sí |sí |rojo |grosor |cabeceras de flecha |longitud
  Table[Style[Text[NumberForm[VecFueEstLo[[2 * (ConFroDes[[i, 1] - 1) + ConFroDes[[i, 2]]], {10, 1}], CooMod[ConFroDes[[i, 1]]] - ((0.3 * Min[LonEle]) +
  |tabla |estilo |texto |forma de número |mínimo |mínimo
  ((1 / 2) * ((VecFueEstLo[[2 * (ConFroDes[[i, 1] - 1) + ConFroDes[[i, 2]]] / (3 * Max[Abs[VecFueEstLo]] * (3 * Min[LonEle])) * {If[ConFroDes[[i, 2]] == 1, 1, 0], If[ConFroDes[[i, 2]] == 2, 1, 0}],
  |má- |valor absoluto |mínimo |sí |sí
  Red, FontWeight -> Bold, FontFamily -> "Arial", FontSize -> 10], {i, 1, Length[ConFroDes]}]
  |rojo |peso de tipo de - |negrita |familia de tipo de letra |tamaño de tipo de letra |longitud
  ]];
  
```

Figura 36. Fragmento del código para generar la gráfica de la armadura en su configuración original y deformada.

Fuente: Propio autor.

### 2.3.4. Gráfica de la armadura mostrando fuerzas

La subsección “Gráfica de la armadura mostrando fuerzas” se grafica la armadura mostrando la magnitud de las fuerzas presentes en cada barra, y se indica la naturaleza de las fuerzas, compresión o tensión.

```

VecApoVecFueEleLoc = Table[0, {NumEle}, {2}];
|tabla

For[i = 1, i <= NumEle, i++,
|para cada
VecApoVecFueEleLoc[i] = Flatten[VecFueEleLoc[i]]
|aplana
];

EsfAxiEleLoc = Table[0, {NumEle}];
|tabla

For[i = 1, i <= NumEle, i++,
|para cada
If[Sign[VecApoVecFueEleLoc[i, 1]] == -1,
|el [función signo
EsfAxiEleLoc[i] = "Ten.",
If[Sign[VecApoVecFueEleLoc[i, 1]] == 1,
|el [función signo
EsfAxiEleLoc[i] = "Com.",
If[Sign[VecApoVecFueEleLoc[i, 1]] == 0,
|el [función signo
EsfAxiEleLoc[i] = "Nada"
]]];
];

GraEstFue = Graphics[{
|gráfico
Style[Line[CooEle], Black, Thickness[0.0025]],
|estilo [línea negro [grosor
Style[Point[CooLoc], Blue, PointSize[0.005]],
|estilo [punto azul [tamaño de punto
Table[Style[Text[NumberForm[Abs[VecApoVecFueEleLoc[i, 1]], {3, 0}], (1/2) * CooEle[i, 2]] + ((1/2) * CooEle[i, 1]] + ((0.1 * Min[LonEle]), (0.1 * Min[LonEle]))], Pink, FontWeight -> Bold, FontFamily -> "Arial", FontSize -> 10],
|tabla [estilo [texto [forma de número [valor absoluto [mínimo [mínimo [rosa [peso de tipo de [negrita [familia de tipo de letra [tamaño de tipo de letra
{i, 1, NumEle}],
Table[Style[Text[EsfAxiEleLoc[i], (1/2) * CooEle[i, 2]] + ((1/2) * CooEle[i, 1]] - ((0.1 * Min[LonEle]), (0.1 * Min[LonEle]))], Gray, FontWeight -> Plain, FontFamily -> "Arial", FontSize -> 8], {i, 1, NumEle}]]
|tabla [estilo [texto [mínimo [mínimo [gris [peso de tipo de [tipo p [familia de tipo de letra [tamaño de tipo de letra
}}];

```

Figura 37. Fragmento del código para generar la gráfica de la armadura mostrando las fuerzas en la barras.

Fuente: Propio autor.

### 2.3.5. Gráfica de la armadura mostrando desplazamientos

La subsección “Gráfica de la armadura mostrando desplazamientos” se grafica la armadura mostrando la magnitud de los desplazamientos presentes en cada nudo, donde el signo indica el sentido, los desplazamientos son una milésima parte de la unidad elegida como longitud.

```

GraEstDes = Graphics[{
  gráfico

  Style[Line[CooEle], GrayLevel[0.9], Thickness[0.0025]],
  [estilo línea [nivel de gris grosor]

  Style[Point[CooNod], Blue, PointSize[0.005]],
  [estilo punto azul [tamaño de punto]

  Table[Style[Text[NumberForm[VecDesEstGlo[(2*i) - 1] * 10^3, {10, 3}], CooNod[[i]] - {(0.3 * Min[LonEle]), 0}], Pink, FontWeight -> Bold,
  [tabla [estilo texto [forma de número [mínimo [rosa [peso de tipo de [negrita]

    FontFamily -> "Arial", FontSize -> 10], {i, 1, NumNod}],
  [familia de tipo de letra [tamaño de tipo de letra]

  Table[Style[Text[NumberForm[VecDesEstGlo[(2*i)] * 10^3, {10, 3}], CooNod[[i]] - {0, (0.3 * Min[LonEle])}], Pink, FontWeight -> Bold,
  [tabla [estilo texto [forma de número [mínimo [rosa [peso de tipo de [negrita]

    FontFamily -> "Arial", FontSize -> 10], {i, 1, NumNod}]
  [familia de tipo de letra [tamaño de tipo de letra]

  }];

```

Figura 38. Fragmento del código para generar la gráfica de la armadura mostrando los desplazamientos en los nudos.

Fuente: Propio autor.

### **3. Solución de armaduras empleando el programa**

### 3.1. Ingreso de los datos y generación de variables

Los datos a ingresar son: El número de barras, las coordenadas de cada barra, el módulo de rigidez y el área de la sección transversal de cada barra, las condiciones de frontera de fuerza y desplazamiento, las barras seleccionadas para calcular cuál es la fuerza máxima y los nudos seleccionados así como la dirección para calcular cuál es el desplazamiento máximo. Por simplicidad se crean variables que contengan los datos a ingresar, en dichas variables es irrelevante el texto que las representa, aunque se propone uno que sea familiar con lo que representan.

El número de barras se ingresa como un escalar. Ejemplificando.

$$NumBar = n$$

Donde:

$n$ , número de barras.

Las coordenadas de las barras se ingresan por medio de corchetes de la siguiente forma.

$$CooBar = \left\{ \left\{ \{x_{1,1}, y_{1,1}\}, \{x_{1,2}, y_{1,2}\} \right\}, \left\{ \{x_{2,1}, y_{2,1}\}, \{x_{2,2}, y_{2,2}\} \right\}, \dots, \left\{ \{x_{n,1}, y_{n,1}\}, \{x_{n,2}, y_{n,2}\} \right\} \right\}$$

Donde:

$\left\{ \{x_{1,1}, y_{1,1}\}, \{x_{1,2}, y_{1,2}\} \right\}$ , coordenadas de los dos nudos de la barra 1.

$\left\{ \{x_{2,1}, y_{2,1}\}, \{x_{2,2}, y_{2,2}\} \right\}$ , coordenadas de los dos nudos de la barra 2.

$\left\{ \{x_{n,1}, y_{n,1}\}, \{x_{n,2}, y_{n,2}\} \right\}$ , coordenadas de los dos nudos de la barra  $n$ .

El módulo de rigidez y el área de la sección transversal se ingresan de la siguiente forma.

$$ModRigBar = \{E_1, E_2, \dots, E_n\}$$

$$AreBar = \{A_1, A_2, \dots, A_n\}$$

Donde:

$E_1$ , módulo de rigidez de la barra 1.

$E_2$ , módulo de rigidez de la barra 2.

$E_n$ , módulo de rigidez de la barra n.

$A_1$ , área de la sección transversal de la barra 1.

$A_2$ , área de la sección transversal de la barra 2.

$A_n$ , área de la sección transversal de la barra n.

Las condiciones de frontera de fuerzas se ingresan de la siguiente forma.

$$ConFroFue = \left\{ \left\{ \{F_{x,1}, F_{y,1}\}, 1 \right\}, \dots, \left\{ \{F_{x,k}, F_{y,k}\}, k \right\} \right\}$$

Donde:

$\left\{ \{F_{x,1}, F_{y,1}\}, 1 \right\}$ , componentes de la fuerza presente en el nudo 1

$\left\{ \{F_{x,k}, F_{y,k}\}, k \right\}$ , componentes de la fuerza presente en el nudo k.

Las condiciones de frontera de desplazamientos se ingresan de la siguiente forma.

$$ConFroDes = \left\{ \{1,1, u_1\}, \{1,2, v_1\}, \dots, \{k, 1, u_k\} \right\}$$

Donde:

$\{1,1, u_1\}$ , desplazamientos de magnitud  $u_1$  en el nudo 1 en la dirección 1 (eje x).

$\{1,2, v_1\}$ , desplazamientos de magnitud  $v_1$  en el nudo 1 en la dirección 2 (eje y).

$\{k, 1, u_k\}$ , desplazamientos de magnitud  $u_k$  en el nudo k en la dirección 1 (eje x).

Para conocer las fuerzas máximas en un grupo de barras se ingresa lo siguiente.

$$BarComFueMax = \{j, \dots, k\}$$

Donde:

$j$ , barra  $j$ .

$k$ , barra  $k$ .

La barra  $j$  y  $k$  se encuentran dentro del conjunto de barras a comparar.

Para conocer los desplazamientos máximos en un grupo de nudos se ingresa lo siguiente.

$$DirDesMax = 1$$

$$NudComDesMax = \{j, \dots, k\}$$

Donde:

1, dirección 1 (eje  $x$ )

$j$ , nudo  $j$ .

$k$ , nudo  $k$ .

Los nudos  $j$  y  $k$  se encuentran dentro del conjunto de nudos a comparar y la dirección 1 es aquella donde se desea conocer el desplazamiento máximo.

Teniendo almacenada la información en variables, se utiliza en las funciones que tiene el código del programa. Y se almacena la información de los cálculos en nuevas variables.

### **3.2. Empleo de las funciones**

Al utilizar las funciones generadas por medio de la función `Module`, se recomienda almacenar en variables los resultados arrojados por cada función generada empleada.

Primeramente se utiliza la función (creada por medio de la función `Module`) `MatRigEstGlo` de la siguiente forma.

$\{GraArm, NumNud, Coonud, IdeBarNud, IdeBarGraLib, NumDesPos, NumDesCon, NumDesDes, DesPos, DesCon, DesDes, LonBar, CosAngBar, MatTra, MatInvTra, MatRigBarGlo, MatRigArmGlo, VecConFroFue, VecConFroDes\} = MatRigEstGlo[NumBar, Coobar, ModRigBar, AreBar, ConFroFue, ConFroDes]$

Donde (sólo se indican las variables de interés):

*GraArm*, gráfica descriptiva de la armadura.

*NumNud*, número de nudos.

*LonBar*, longitud de las barras.

*MatTra*, matriz de transformación de las barras.

*MatRigBarGlo*, matriz de rigidez global de las barras

*MatRigArmGlo*, matriz de rigidez global de la armadura.

Después es posible emplear una función u otra en relación al enfoque a emplear, penalización o eliminación.

Para el enfoque de eliminación se utiliza la función *VecDesEstGloyVecFueEstGloEnfEli* de la siguiente manera.

$\{VecDesArmGlo, VecFueArmGlo\} = VecDesEstGloyVecFueEstGloEnfEli[NumDesPos, NumDesCon, NumDesDes, DesPos, DesCon, DesDes, MatRigArmGlo, VecConFroFue, VecConFroDes]$

Donde:

*VecDesArmGlo*, vector de desplazamiento global de la armadura.

*VecFueArmGlo*, vector de fuerza global de la armadura.

Para el enfoque de penalización se utiliza la función *VecDesEstGloyVecFueEstGloEnfPen* de la siguiente manera.

$\{VecDesArmGlo, VecFueArmGlo\} = VecDesEstGloyVecFueEstGloEnfPen[NumDesPos, NumDesCon, NumDesDes, DesPos, DesCon, DesDes, MatRigArmGlo, VecConFroFue, VecConFroDes]$

Donde:

*VecDesArmGlo*, vector de desplazamiento global de la armadura.

*VecFueArmGlo*, vector de fuerza global de la armadura.

Después se utiliza la función *VecFueEleLocyVecDesEleLoc* de la siguiente manera.

$$\{VecDesBarGlo, VecFueBarGlo, VecDesBarLoc, VecFueBarLoc\} \\ = VecFueEleLocyVecDesEleLoc[NumBar, IdeBarGraLib, MatTra, VecDesArmGlo, MatRigBarGlo]$$

Donde:

*VecDesBarGlo*, vector de desplazamiento global de las barras.

*VecFueBarGlo*, vector de fuerza global de las barras.

*VecDesBarLoc*, vector de desplazamiento local de las barras.

*VecFueBarLoc*, vector de fuerza local de las barras.

Ejecutando todas las funciones anteriores, se tiene la solución de la armadura de forma numérica, almacenada en variables. Para el análisis de la solución, se recomienda emplear las siguientes funciones

La función *FueMax* se utiliza de la siguiente forma.

$$\{FueMax, BarFueMax\} = FueMax[NumBar, NumNud, BarComFueMax, VecFueBarLoc]$$

Donde:

*FueMax*, magnitud de la fuerza máxima en las barras seleccionadas.

*BarFueMax*, número de la barra donde se presenta la fuerza máxima.

La función *DesMax* se utiliza de la siguiente forma.

$$\{DesMax, NudDesMax\} = \\ DesMax[NumBar, NumNud, DirDesMax, NudComDesMax, VecDesArmGlo]$$

Donde:

*DesMax*, magnitud del desplazamiento máximo de los nudos seleccionados.

*NudDesMax*, número del nudo de desplazamiento máximo.

La función  $GraEstyGrafEstDes$  de la siguiente forma.

$$\{GraArmyGraArmDes\} = GraEstyGraEstDes[NumBar, NumNud, Coobar, Coonud, LonBar, VecDesBarGlo, VecDesArmGlo, ConFroDes, VecFueArmGlo]$$

Donde:

$GraArmyGraArmDes$ , gráfica de la armadura en su configuración original y deformada.

La función  $GraEstFue$  de la siguiente forma.

$$\{GraArmFue\} = GraEstFue[NumBar, NumNud, Coobar, Coonud, LonBar, VecFueBarLoc]$$

Donde:

$GraArmFue$ , gráfica de la armadura mostrando la magnitud de las fuerzas en las barras y su naturaleza, compresión o tensión.

La función  $GraEstDes$  de la siguiente forma.

$$\{GraArmDes\} = GraEstDes[NumBar, NumNud, Coobar, Coonud, LonBar, VecDesArmGlo]$$

Donde:

$GraArmDes$ , gráfica de la armadura mostrando los desplazamientos de los nudos, y el sentido de los mismos según su signo.

## **4. Ejemplos de aplicación del programa**

Se mostrarán dos ejemplos, con el fin de mostrar la confiabilidad del programa. Y dos ejemplos más con el fin de mostrar la utilidad y capacidad del programa.

#### 4.1. Primer ejemplo

De la referencia [5] en la página 499, el ejemplo 24.1 será resuelto. El ejemplo es el siguiente.

“La armadura mostrada en la figura está sometida a una fuerza horizontal  $P$  aplicada en el nudo 3. Determine las componentes de desplazamiento resultante en el nudo libre (nudo 3), y las fuerzas resultantes en la barra” [5].

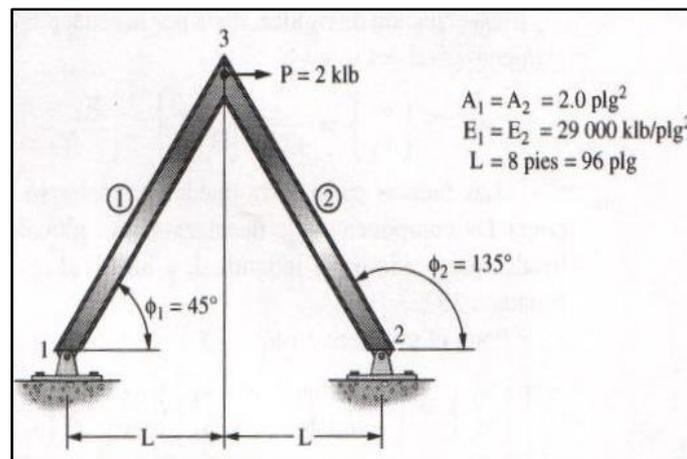


Figura 39. Imagen y características de la armadura.

Fuente: McCormac, J. C., [5].

Para ingresar la armadura se hace lo siguiente.

Primero se indican el número de barras así como las coordenadas de los nudos de cada barra según un sistema de coordenadas elegido al azar, pero cartesiano.

$$NumBar1 = 2$$

$$CooBar1 = \{ \{ \{0,0\}, \{96,96\} \}, \{ \{192,0\}, \{96,96\} \} \}$$

Es importante aclarar que las unidades empleadas en el programa deben pertenecer a un solo sistema de unidades. Por ello la unidad de longitud se encuentra dada en pulgadas.

Después se proporcionan el área de las secciones transversales y los módulos de rigidez de las barras, asignando dichas propiedades en el mismo orden que se dieron las coordenadas.

$$ModRigBar1 = \{29 * 10^6, 29 * 10^6\}$$

$$AreBar1 = \{2,2\}$$

Ahora se ingresan las condiciones de frontera de fuerza y desplazamiento.

$$ConFroFue1 = \{\{2000,0\}, 2\}$$

$$ConFroDes1 = \{\{1,1,0\}, \{1,2,0\}, \{3,1,0\}, \{3,2,0\}\}$$

Lamentablemente, seleccionar los números que corresponden a los nudos donde se aplican las condiciones de frontera debe ser, primeramente, aleatorio, para después emplear la imagen que genera el programa y seleccionar el número real donde se aplican.

Todo lo anterior se muestra en la siguiente imagen, así como la forma en que se emplean las variables generadas en el programa.

```

NumEle1 = 2;
CooEle1 = {{{0, 0}, {96, 96}}, {{192, 0}, {96, 96}}};
ModRigEle1 = {29 * 106, 29 * 106, 29 * 106, 29 * 106};
AreEle1 = {2, 2};

ConFroFue1 = {{{2000, 0}, 2}};
ConFroDes1 = {{1, 1, 0}, {1, 2, 0}, {3, 1, 0}, {3, 2, 0}};

{GraEst1, NumNod1, CooNod1, IdeEleNod1, IdeEleGralLib1, NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, LonEle1, CosAngEle1, MatTra1,
  MatRigEleGlo1, MatRigEstGlo1, VecConFroFue1, VecConFroDes1} = MatRigEstGlo[NumEle1, CooEle1, ModRigEle1, AreEle1, ConFroFue1, ConFroDes1];

{VecDesEstGlo1, VecFueEstGlo1} = VecDesEstGloyVecFueEstGloEnfEli[NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, MatRigEstGlo1,
  VecConFroFue1, VecConFroDes1];

{VecDesEstGlo2, VecFueEstGlo2} = VecDesEstGloyVecFueEstGloEnfPen[NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, MatRigEstGlo1,
  VecConFroFue1, VecConFroDes1];

{VecDesEleGlo1, VecFueEleGlo1, VecDesEleLoc1, VecFueEleLoc1} = VecFueEleLocyVecDesEleLoc[NumEle1, IdeEleGralLib1, MatTra1, VecDesEstGlo1, MatRigEleGlo1];

{GraEstyGraEstDes1} = GraEstyGraEstDes[NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecDesEleGlo1, VecDesEstGlo1, ConFroDes1, VecFueEstGlo1];

GraEst1

GraEstyGraEstDes1

```

Figura 40. Ingreso de las variables y forma en que se emplea el programa.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen descriptiva de la armadura, indicando el número de los nudos y de las barras. Y como resultado arroja la imagen de la armadura original y deformada.

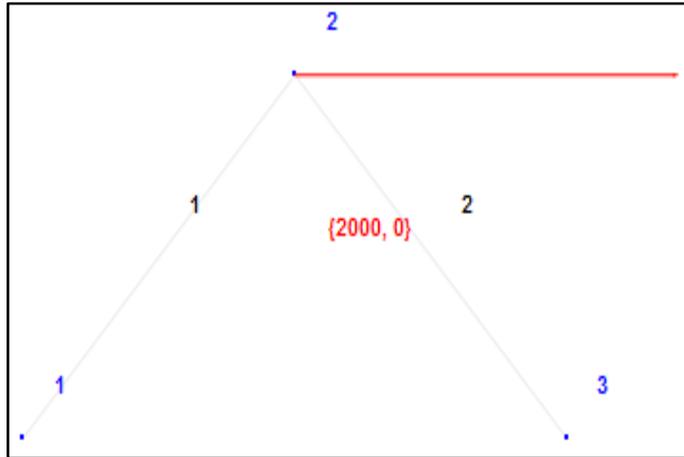


Figura 41. Gráfica de la armadura generada por el programa, indicando el número que corresponde a los nudos y las barras.

Fuente: Propio autor.

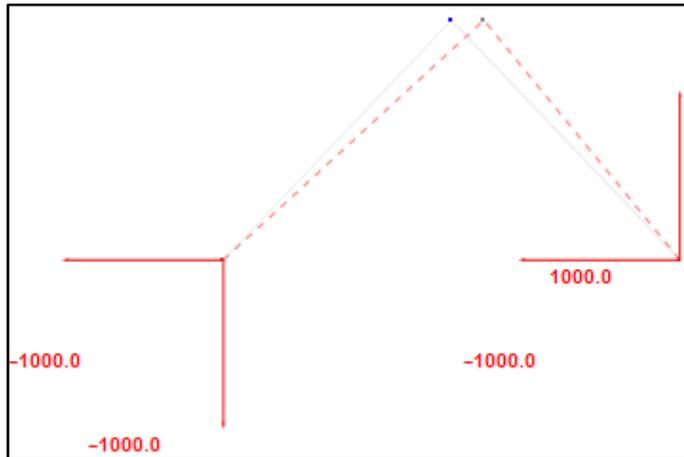


Figura 42. Gráfica de la armadura original y la armadura deformada generada por el programa, mostrando el desplazamiento (exagerado) de los nudos.

Fuente: Propio autor.

Para conocer los desplazamientos de la armadura y los desplazamientos de las barras, se muestran sus respectivas variables. Esto se muestra en la siguiente figura.

```
In[36]:= VecDesEstGlo1 // MatrixForm
[forma de matriz]

Out[36]//MatrixForm=

$$\begin{pmatrix} 0 \\ 0 \\ 0.00468153 \\ 0. \\ 0 \\ 0 \end{pmatrix}$$

```

Figura 43. Vector de desplazamientos global, enfoque de eliminación  
Fuente: Propio autor.

```
In[37]:= VecDesEstGlo2 // MatrixForm
[forma de matriz]

Out[37]//MatrixForm=

$$\begin{pmatrix} 2.34077 \times 10^{-7} \\ 2.34077 \times 10^{-7} \\ 0.004682 \\ -6.49726 \times 10^{-23} \\ 2.34077 \times 10^{-7} \\ -2.34077 \times 10^{-7} \end{pmatrix}$$

```

Figura 44. Vector de desplazamientos global, enfoque de penalización  
Fuente: Propio autor

Cabe decir que las variables de finalización 1 están asociadas al enfoque de eliminación y las de finalización 2 al enfoque de penalización.

El desplazamiento en la armadura del nudo superior (2 en el programa) en la dirección del eje x (enfoque de eliminación) es:

$$u_2 = 0.004\ 681\ 53 \text{ [in]}$$

$$v_2 = 0 \text{ [in]}$$

El desplazamiento en la armadura del nudo superior (2 en el programa) en la dirección del eje x (enfoque de penalización) es:

$$u_2 = 0.004\ 682 \text{ [in]}$$

$$v_2 = 0 \text{ [in]}$$

La relación que existe entre los números empleados en la referencia y el programa para los nudos es el siguiente:

Número del nudo en la referencia	Número del nudo en el programa
1	1
2	3
3	2

Tabla 1. Relación entre los números que representan a los nudos en el programa y la referencia.

Fuente: Propio autor

Los resultados del ejemplo son.

$$\begin{Bmatrix} u_3 \\ v_3 \end{Bmatrix} = \frac{1}{427.2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} X_3 = P = 2 \\ Y_3 = 0 \end{Bmatrix} = \begin{Bmatrix} 0.0047 \\ 0 \end{Bmatrix} \text{ plg}$$

Figura 45. Resultados del ejemplo 24.1, desplazamientos del nudo superior.

Fuente: McCormac, J. C., [5].

Al comparar los resultados se observa que la diferencia entre ellos está asociada al número de decimales empleados. Y se comprueba la confiabilidad del programa.

#### 4.2. Segundo ejemplo

De la referencia [6] en las páginas 104 y 105 el ejemplo 4.1 será resuelto. El ejemplo es el siguiente.

“Considere la armadura de cuatro barras mostrado en la figura. Para todas las barras el módulo de elasticidad es igual a  $29.5 \times 10^6$  psi y el área de la sección transversal es  $1 \text{ in}^2$ . Realice lo siguiente: Determine la matriz de rigidez estructural

para toda la armadura y usando el método de eliminación, encuentre los desplazamientos nodales” [6].

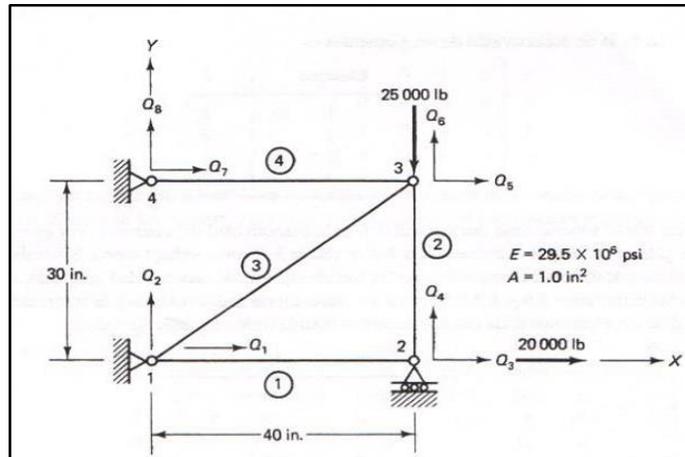


Figura 46. Imagen y características de la armadura.

Fuente: Chandrupatla, T. R., [6].

Para ingresar la armadura se hace lo siguiente.

Primero se indican el número de barras así como las coordenadas de los nudos de cada barra según un sistema de coordenadas elegido al azar.

$$NumBar1 = 4$$

$$CooBar1 = \{ \{ \{0,0\}, \{40,0\} \}, \{ \{40,30\}, \{40,0\} \}, \{ \{0,0\}, \{40,30\} \}, \{ \{0,30\}, \{40,30\} \} \}$$

Después se proporcionan el área de las secciones transversales y los módulos de rigidez de las barras.

$$ModRigBar1 = \{ 29.5 * 10^6, 29.5 * 10^6, 29.5 * 10^6, 29.5 * 10^6 \}$$

$$AreBar1 = \{ 1, 1, 1, 1 \}$$

Ahora se ingresan las condiciones de frontera de fuerza y desplazamiento.

$$ConFroFue1 = \{\{20000,0\}, 3\}, \{0, -25000\}, 4\}$$

$$ConFroDes1 = \{1,1,0\}, \{1,2,0\}, \{2,1,0\}, \{2,2,0\}, \{3,2,0\}$$

Todo lo anterior se muestra en la siguiente imagen, así como la forma en que se emplean las variables generadas en el programa.

```

NumEle1 = 4;
CooEle1 = {{0, 0}, {40, 0}}, {{40, 30}, {40, 0}}, {{0, 0}, {40, 30}}, {{0, 30}, {40, 30}};
ModRigEle1 = {29.5*106, 29.5*106, 29.5*106, 29.5*106};
AreEle1 = {1, 1, 1, 1};

ConFroFue1 = {{20000, 0}, 3}, {{0, -25000}, 4};
ConFroDes1 = {{1, 1, 0}, {1, 2, 0}, {2, 1, 0}, {2, 2, 0}, {3, 2, 0}};

{GraEst1, NumNod1, CooNod1, IdeEleNod1, IdeEleGralib1, NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, LonEle1, CosAngEle1, MatTra1,
  MatRigEleGlo1, MatRigEstGlo1, VecConFroFue1, VecConFroDes1} = MatRigEstGlo(NumEle1, CooEle1, ModRigEle1, AreEle1, ConFroFue1, ConFroDes1);

{VecDesEstGlo1, VecFueEstGlo1} = VecDesEstGloVecFueEstGloEnfEli(NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, MatRigEstGlo1,
  VecConFroFue1, VecConFroDes1);

{VecDesEleGlo1, VecFueEleGlo1, VecDesEleLoc1, VecFueEleLoc1} = VecFueEleLocVecDesEleLoc(NumEle1, IdeEleGralib1, MatTra1, VecDesEstGlo1, MatRigEleGlo1);

{GraEstyGraEstDes1} = GraEstyGraEstDes(NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecDesEleGlo1, VecDesEstGlo1, ConFroDes1, VecFueEstGlo1);

GraEst1

GraEstyGraEstDes1

```

Figura 47. Ingreso de las variables y forma en que se emplea el programa.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen descriptiva de la armadura, indicando el número de los nudos y de las barras. Y como resultado arroja la imagen de la armadura original y deformada.

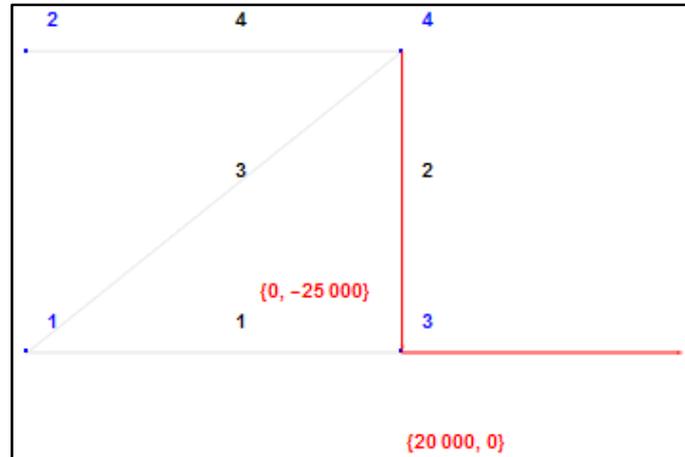


Figura 48. Gráfica de la armadura generada por el programa, indicando el número que corresponde a los nudos y las barras.

Fuente: Propio autor.

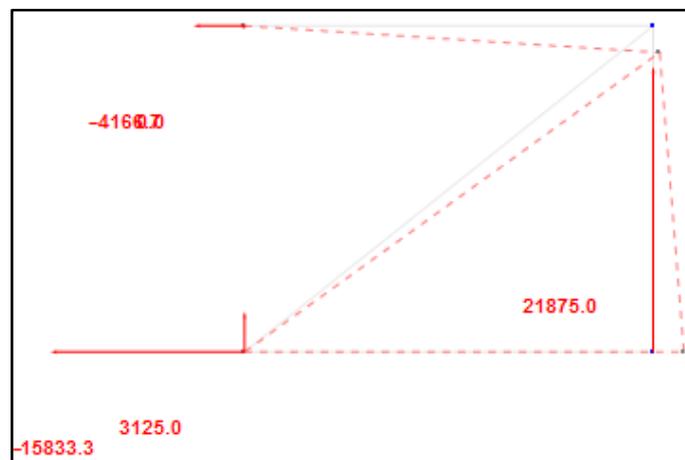


Figura 49. Gráfica de la armadura original y la armadura deformada generada por el programa, mostrando el desplazamiento (exagerado) de los nudos.

Fuente: Propio autor.

Para conocer los desplazamientos de la armadura y la matriz de rigidez de la armadura, se muestran sus respectivas variables. Esto se muestra en la siguiente figura.

```
In[107]:= VecDesEstGlo1 // MatrixForm
          |forma de matriz

Out[107]/MatrixForm=

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.0271186 \\ 0 \\ 0.00564972 \\ -0.0222458 \end{pmatrix}$$

```

Figura 50. Vector de desplazamientos global.

Fuente: Propio autor.

```
In[130]:= MatRigEstGlo1 // MatrixForm
          |forma de matriz

Out[130]/MatrixForm=

$$\begin{pmatrix} 1.1151 \times 10^6 & 283\,200. & 0 & 0 & -737\,500. & 0. & -377\,600. & -283\,200. \\ 283\,200. & 212\,400. & 0 & 0 & 0. & 0. & -283\,200. & -212\,400. \\ 0 & 0 & 737\,500. & 0. & 0 & 0 & -737\,500. & 0. \\ 0 & 0 & 0. & 0. & 0 & 0 & 0. & 0. \\ -737\,500. & 0. & 0 & 0 & 737\,500. & 0. & 0. & 0. \\ 0. & 0. & 0 & 0 & 0. & 983\,333. & 0. & -983\,333. \\ -377\,600. & -283\,200. & -737\,500. & 0. & 0. & 0. & 1.1151 \times 10^6 & 283\,200. \\ -283\,200. & -212\,400. & 0. & 0. & 0. & -983\,333. & 283\,200. & 1.19573 \times 10^6 \end{pmatrix}$$

```

Figura 51. Matriz de rigidez global.

Fuente: Propio autor.

La relación que existe entre los números empleados en la referencia y el programa para los grados de libertad es el siguiente:

Número del grado de libertad en la referencia	Número del grado de libertad en el programa
1	1
2	2
3	5
4	6
5	7

6	8
7	3
8	4

Tabla 2. Relación entre los números que representan a los grados de libertad en el programa y la referencia.

Fuente: Propio autor

Los resultados del ejemplo son.

$$\mathbf{K} = \frac{29.5 \times 10^6}{600} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 22.68 & 5.76 & -15.0 & 0 & -7.68 & -5.76 & 0 & 0 \\ 5.76 & 4.32 & 0 & 0 & -5.76 & -4.32 & 0 & 0 \\ -15.0 & 0 & 15.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.0 & 0 & -20.0 & 0 & 0 \\ -7.68 & -5.76 & 0 & 0 & 22.68 & 5.76 & -15.0 & 0 \\ -5.76 & -4.32 & 0 & -20.0 & 5.76 & 24.32 & 0 & 0 \\ 0 & 0 & 0 & 0 & -15.0 & 0 & 15.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix}$$

Figura 52. Resultado del ejemplo 4.1, matriz de rigidez.

Fuente: Chandrupatla, T. R., [6].

$$\begin{Bmatrix} Q_3 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 27.12 \times 10^{-3} \\ 5.65 \times 10^{-3} \\ -22.25 \times 10^{-3} \end{Bmatrix} \text{ in.}$$

Figura 53. Resultado del ejemplo 4.1, parte del vector de desplazamientos global.

Fuente: Chandrupatla, T. R., [6].

Al comparar los resultados se observa que la diferencia entre ellos está asociada al número de decimales empleados. Con esto se comprueba la confiabilidad del programa.

### 4.3. Tercer ejemplo

De la idea de un puente, se propone el tercer ejemplo, figura 54. Se calcularán los desplazamientos de los nudos y las fuerzas en las barras.

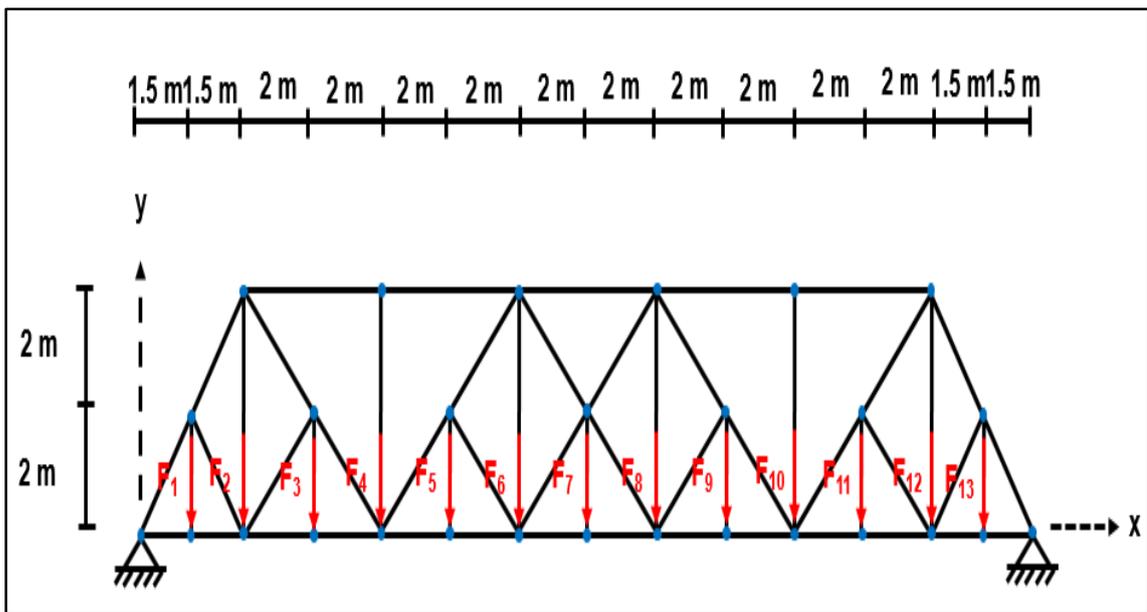


Figura 54. Imagen y características de la armadura.

Fuente: Propio autor.

Las fuerzas se muestran en la tabla 3.

Fuerza	Magnitud [kN]
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10

Tabla 3. Magnitud de las fuerzas del tercer ejemplo.

Fuente: Propio autor

Las propiedades de las barras se muestran en la tabla 4.

Barras	Área [m <sup>2</sup> ]	Módulo de rigidez [kPa]
Todas	50×10 <sup>-4</sup>	200×10 <sup>6</sup>

Tabla 4. Propiedades de las barras del tercer ejemplo.

Fuente: Propio autor

Para ingresar la armadura se hace lo siguiente. Se emplean variables donde se almacenará la información de la armadura.

Primero se indican el número de barras así como las coordenadas de los nudos de cada barra según un sistema de coordenadas elegido al azar.

Después se proporcionan el área de las secciones transversales y los módulos de rigidez de las barras.

Ahora se ingresan las condiciones de frontera de fuerza y desplazamiento.

Todo lo anterior se muestra en la siguiente imagen.

```

NumEle1 = 54;

CooEle1 = {{(0, 0), (1.5, 0)}, {(0, 0), (1.5, 2)}, {(1.5, 0), (3, 0)}, {(1.5, 0), (1.5, 2)}, {(3, 0), (5, 0)}, {(3, 0), (5, 2)}, {(3, 0), (3, 4)}, {(3, 0), (1.5, 2)},
{(5, 0), (7, 0)}, {(5, 0), (5, 2)}, {(7, 0), (9, 0)}, {(7, 0), (9, 2)}, {(7, 0), (7, 4)}, {(7, 0), (5, 2)}, {(9, 0), (11, 0)}, {(9, 0), (9, 2)}, {(11, 0), (13, 0)},
{(11, 0), (13, 2)}, {(11, 0), (11, 4)}, {(11, 0), (9, 2)}, {(13, 0), (15, 0)}, {(13, 0), (13, 2)}, {(15, 0), (17, 0)}, {(15, 0), (17, 2)}, {(15, 0), (15, 4)},
{(15, 0), (13, 2)}, {(17, 0), (19, 0)}, {(17, 0), (17, 2)}, {(19, 0), (21, 0)}, {(19, 0), (21, 2)}, {(19, 0), (19, 4)}, {(19, 0), (17, 2)}, {(21, 0), (23, 0)},
{(21, 0), (21, 2)}, {(23, 0), (24.5, 0)}, {(23, 0), (24.5, 2)}, {(23, 0), (23, 4)}, {(23, 0), (21, 2)}, {(24.5, 0), (26, 0)}, {(24.5, 0), (24.5, 2)},
{(26, 0), (24.5, 2)}, {(1.5, 2), (3, 4)}, {(5, 2), (3, 4)}, {(9, 2), (11, 4)}, {(13, 2), (15, 4)}, {(13, 2), (11, 4)}, {(17, 2), (15, 4)}, {(21, 2), (23, 4)},
{(24.5, 2), (23, 4)}, {(3, 4), (7, 4)}, {(7, 4), (11, 4)}, {(11, 4), (15, 4)}, {(15, 4), (19, 4)}, {(19, 4), (23, 4)}};

ModRigEle1 = {210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106,
210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106,
210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106,
210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106, 210*106,
210*106, 210*106, 210*106, 210*106};

AreEle1 = {50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4,
50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4,
50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4,
50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4, 50*10-4,
50*10-4, 50*10-4, 50*10-4};

ConFroFue1 = {{(0, -10), 2}, {(0, -10), 4}, {(0, -10), 6}, {(0, -10), 8}, {(0, -10), 10}, {(0, -10), 12}, {(0, -10), 14}, {(0, -10), 16}, {(0, -10), 18},
{(0, -10), 20}, {(0, -10), 22}, {(0, -10), 24}, {(0, -10), 26}};

ConFroDes1 = {(1, 1, 0), (1, 2, 0), (28, 1, 0), (28, 2, 0)};

```

Figura 55. Ingreso de las variables.

Fuente: Propio autor.

Para ejecutar el programa, se utilizan las funciones del mismo con las variables que contienen la información de la armadura.

Todo lo anterior se muestra en la siguiente imagen.

```
{GraEst1, NumNod1, CooNod1, IdeEleGralib1, IdeEleGralib1, NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, LonEle1, CosAngEle1, MatTra1, MatRigEleGlo1, MatRigEstGlo1, VecConFroFue1, VecConFroDes1} = MatRigEstGlo[NumEle1, CooEle1, ModRigEle1, AreEle1, ConFroFue1, ConFroDes1];  
  
GraEst1  
  
{VecDesEstGlo1, VecFueEstGlo1} = VecDesEstGloyVecFueEstGloEnfPen[NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, MatRigEstGlo1, VecConFroFue1, VecConFroDes1];  
  
{VecDesEleGlo1, VecFueEleGlo1, VecDesEleLoc1, VecFueEleLoc1} = VecFueEleLocyVecDesEleLoc[NumEle1, IdeEleGralib1, MatTra1, VecDesEstGlo1, MatRigEleGlo1];  
  
{GraEstyGraEstDes1} = GraEstyGraEstDes[NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecDesEleGlo1, VecDesEstGlo1];  
  
GraEstyGraEstDes1
```

Figura 56. Forma en que se emplea el programa.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen descriptiva de la armadura, indicando el número de los nudos y de las barras. Y como resultado arroja la imagen de la armadura original y deformada.

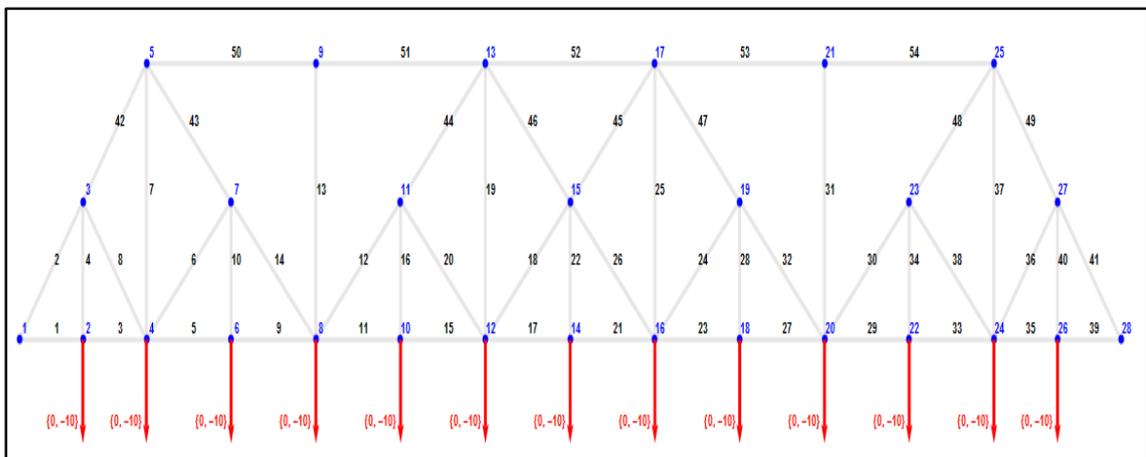


Figura 57. Gráfica de la armadura generada por el programa, indicando el número que corresponde a los nudos y las barras.

Fuente: Propio autor.

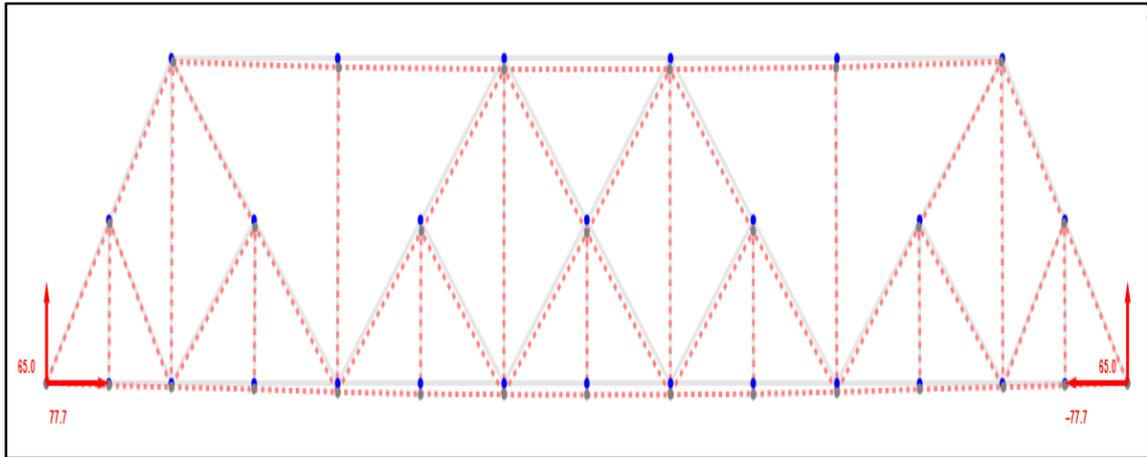


Figura 58. Gráfica de la armadura original y la armadura deformada generada por el programa, mostrando el desplazamiento (exagerado) de los nudos.

Fuente: Propio autor.

De forma adicional, se encontrará la fuerza máxima en las barras y los desplazamientos máximos en los nudos en la dirección x y la dirección y.

Para ello se ingresan las barras donde se desea comparar la fuerza máxima y los nudos donde se desean conocer los desplazamientos máximos así como las direcciones donde se quieren conocer dichos desplazamientos máximos.

Todo lo anterior se muestra en la siguiente imagen.

```

EleComFueMax1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54};

DirDesMax1 = 2;

NodComDesMax1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28};

```

Figura 59. Ingreso de las variables para obtener la fuerza y desplazamiento máximos.

Fuente: Propio autor.

Para ejecutar el programa, se utilizan las funciones del mismo con las variables que contienen la información de comparación.

Todo lo anterior se muestra en las siguientes imágenes.

```
{FueMax1, EleFueMax1} = FueMax[NumEle1, NumNod1, EleComFueMax1, VecFueEleLoc1];  
{DesMax1, NodDesMax1} = DesMax[NumEle1, NumNod1, DirDesMax1, NodComDesMax1, VecDesEstGlo1];
```

Figura 60. Forma en que se emplea el programa para obtener la fuerza y desplazamiento máximos.

Fuente: Propio autor.

Para conocer la magnitud de la fuerza máxima y la barra donde actúa y la magnitud del desplazamiento máximo y el nudo desplazado, se muestran sus respectivas variables. Esto se muestra en la siguiente figura.

```
FueMax1  
EleFueMax1  
  
103.078  
  
52
```

Figura 61. Variables de fuerza máxima del ejemplo.

Fuente: Propio autor.

```
DesMax1  
NodDesMax1  
  
0.0033663  
  
14
```

Figura 62. Variables de desplazamiento máximo en la dirección x del ejemplo.

Fuente: Propio autor.

De forma adicional, también se mostrará las gráficas de la armadura mostrando la magnitud de las fuerzas en las barras y los desplazamientos en los nudos.

La forma en que se emplean las funciones necesarias, se muestra en la siguiente imagen.

```
{GraEstFue1} = GraEstFue[NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecFueEleLoc1];
|
{GraEstDes1} = GraEstDes[NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecDesEstGlo1];

GraEstFue1
GraEstDes1
```

Figura 63. Forma en que se emplea el programa para obtener la gráfica de la armadura mostrando las fuerzas en las barras y los desplazamientos en los nudos.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen de la armadura mostrando las fuerzas en las barras y la imagen de la armadura mostrando los desplazamientos de los nudos.

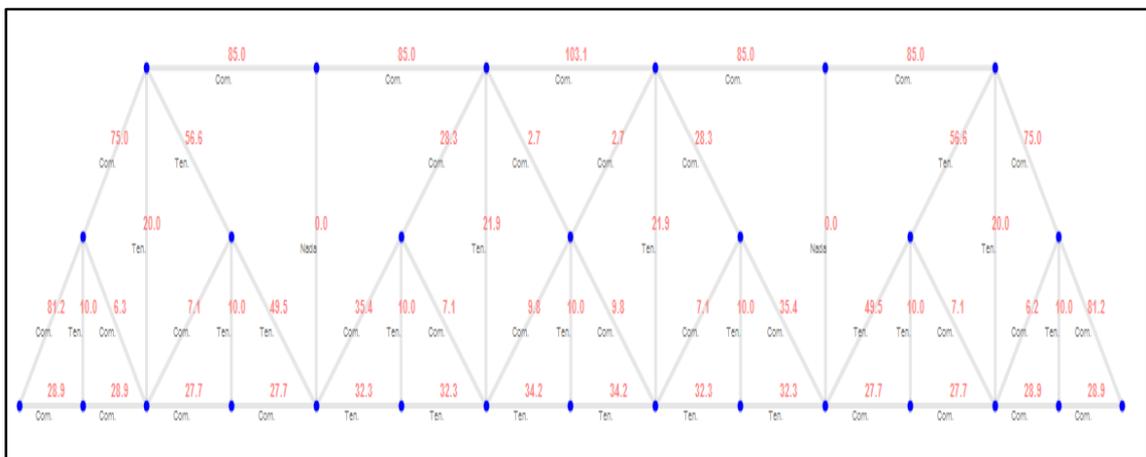


Figura 64. Gráfica de la armadura mostrando las fuerzas en las barras, generada por el programa.

Fuente: Propio autor.

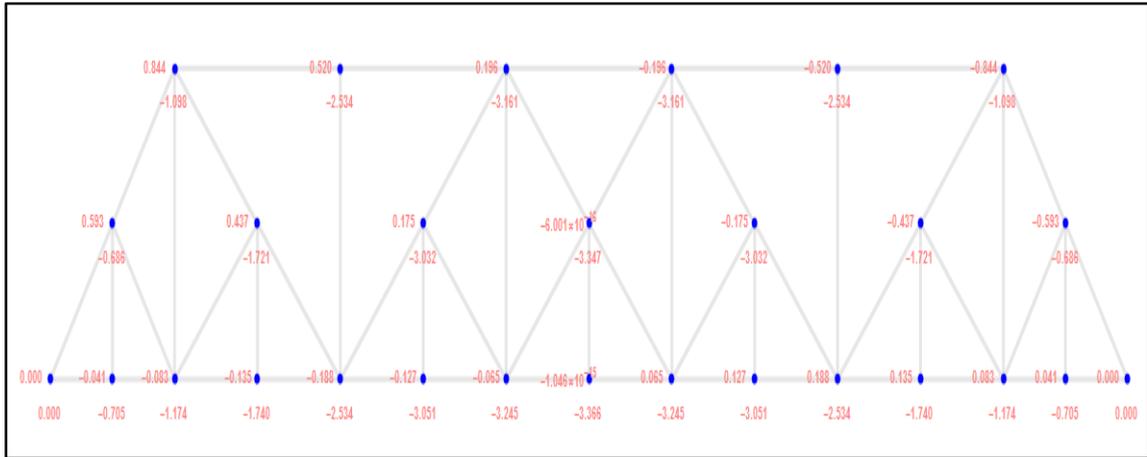


Figura 65. Gráfica de la armadura mostrando los desplazamientos de los nudos, generada por el programa.

Fuente: Propio autor.

#### 4.4. Cuarto ejemplo

De la idea de una torre de transmisión, se propone el cuarto ejemplo, figura 66. Se calcularán los desplazamientos de los nudos y las fuerzas en las barras.

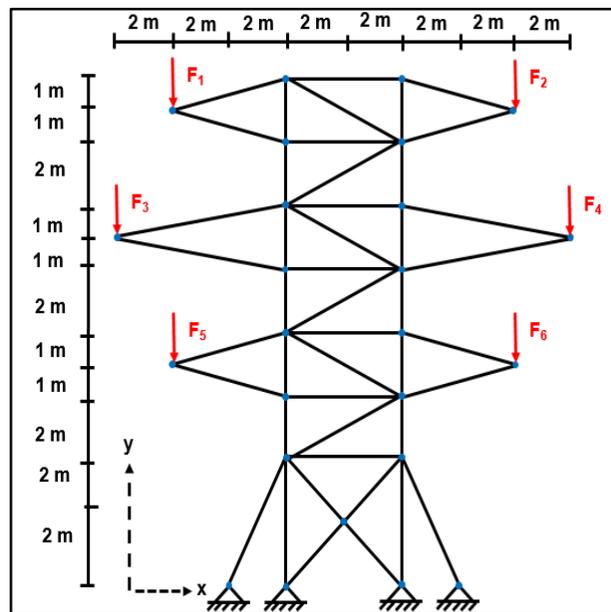


Figura 66. Imagen y características de la armadura.

Fuente: Propio autor.

Las fuerzas se muestran en la tabla 5.

Fuerza	Magnitud [kN]
1	20
2	20
3	20
4	20
5	20
6	20

Tabla 5. Magnitud de las fuerzas del tercer ejemplo.

Fuente: Propio autor

Las propiedades de las barras se muestran en la tabla 6.

Barras	Área [m <sup>2</sup> ]	Módulo de rigidez [kPa]
Todas	$50 \times 10^{-4}$	$200 \times 10^6$

Tabla 4. Propiedades de las barras del tercer ejemplo.

Fuente: Propio autor

Para ingresar la armadura se hace lo siguiente. Se emplean variables donde se almacenará la información de la armadura.

Primero se indican el número de barras así como las coordenadas de los nudos de cada barra según un sistema de coordenadas elegido al azar.

Después se proporcionan el área de las secciones transversales y los módulos de rigidez de las barras.

Ahora se ingresan las condiciones de frontera de fuerza y desplazamiento.

Todo lo anterior se muestra en la siguiente imagen.

```

NumEle1 = 45;

CooEle1 = {{(4, 0), (6, 4)}, {(6, 0), (6, 4)}, {(6, 0), (8, 2)}, {(10, 0), (8, 2)}, {(10, 0), (10, 4)}, {(12, 0), (10, 4)}, {(8, 2), (6, 4)},
{(8, 2), (10, 4)}, {(6, 4), (10, 4)}, {(6, 4), (10, 6)}, {(6, 4), (6, 6)}, {(10, 4), (10, 6)}, {(6, 6), (10, 6)}, {(6, 6), (6, 8)}, {(10, 6), (10, 8)},
{(10, 6), (6, 8)}, {(6, 8), (10, 8)}, {(6, 8), (10, 10)}, {(6, 8), (6, 10)}, {(10, 8), (10, 10)}, {(6, 10), (10, 10)}, {(6, 10), (6, 12)},
{(10, 10), (10, 12)}, {(10, 10), (6, 12)}, {(6, 12), (10, 12)}, {(6, 12), (10, 14)}, {(6, 12), (6, 14)}, {(10, 12), (10, 14)}, {(6, 14), (10, 14)},
{(6, 14), (6, 16)}, {(10, 14), (10, 16)}, {(10, 14), (6, 16)}, {(6, 16), (10, 16)}, {(2, 7), (6, 6)}, {(2, 7), (6, 8)}, {(0, 11), (6, 12)},
{(0, 11), (6, 10)}, {(2, 15), (6, 14)}, {(2, 15), (6, 16)}, {(14, 7), (10, 6)}, {(14, 7), (10, 8)}, {(16, 11), (10, 10)}, {(16, 11), (10, 12)},
{(14, 15), (10, 14)}, {(14, 15), (10, 16)}};

ModRigEle1 = {210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106,
210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106,
210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106, 210 * 106};

AreEle1 = {50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4,
50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4,
50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4, 50 * 10-4};

ConFroFue1 = {{(0, -20), 1}, {(0, -20), 2}, {(0, -20), 3}, {(0, -20), 23}, {(0, -20), 24}, {(0, -20), 25}};

ConFroDes1 = {(4, 1, 0), (4, 2, 0), (5, 2, 0), (5, 1, 0), (14, 2, 0), (14, 1, 0), (22, 2, 0), (22, 1, 0)};

```

Figura 67. Ingreso de las variables.

Fuente: Propio autor.

Para ejecutar el programa, se utilizan las funciones del mismo con las variables que contienen la información de la armadura.

Todo lo anterior se muestra en la siguiente imagen.

```

{GraEst1, NumNod1, CooNod1, IdeEleNod1, IdeEleGralib1, NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, LonEle1, CosAngEle1, MatTra1,
  MatRigEleGlo1, MatRigEstGlo1, VecConFroFue1, VecConFroDes1} = MatRigEstGlo[NumEle1, CooEle1, ModRigEle1, AreEle1, ConFroFue1, ConFroDes1];

GraEst1

{VecDesEstGlo1, VecFueEstGlo1} = VecDesEstGloyVecFueEstGloEnfPen[NumDesPos1, NumDesCon1, NumDesDes1, DesPos1, DesCon1, DesDes1, MatRigEstGlo1,
  VecConFroFue1, VecConFroDes1];

{VecDesEleGlo1, VecFueEleGlo1, VecDesEleLoc1, VecFueEleLoc1} = VecFueEleLocyVecDesEleLoc[NumEle1, IdeEleGralib1, MatTra1, VecDesEstGlo1, MatRigEleGlo1];

{GraEstyGraEstDes1} = GraEstyGraEstDes[NumEle1, NumNod1, CooEle1, CooNod1, LonEle1, VecDesEleGlo1, VecDesEstGlo1];

GraEstyGraEstDes1

```

Figura 68. Forma en que se emplea el programa.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen descriptiva de la armadura, indicando el número de los nudos y de las barras. Y como resultado arroja la imagen de la armadura original y deformada.

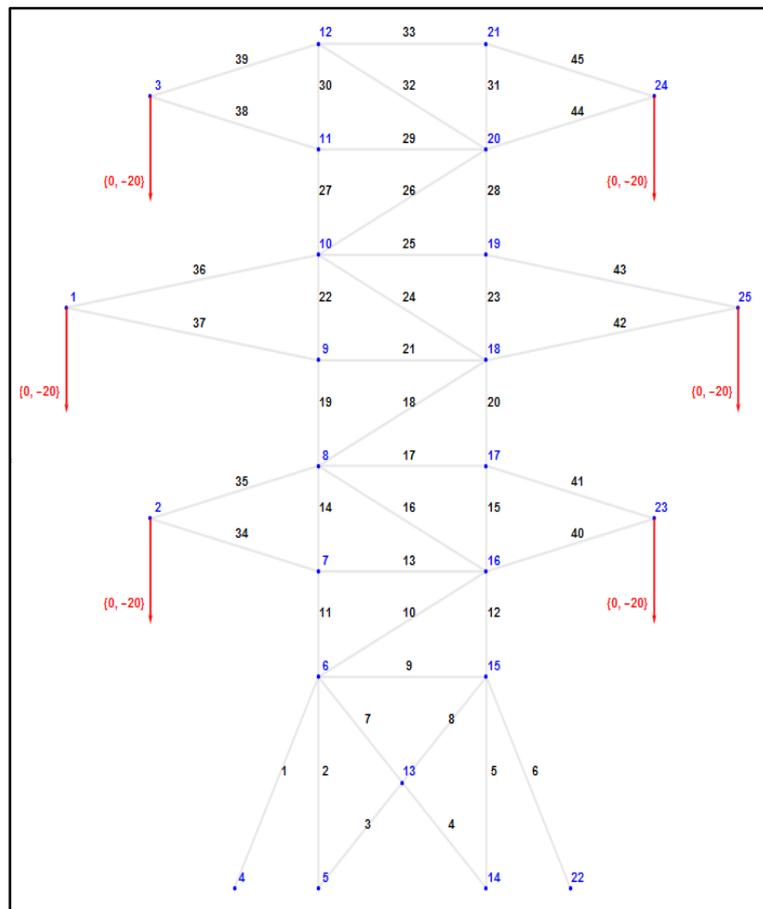


Figura 69. Gráfica de la armadura generada por el programa, indicando el número que corresponde a los nudos y las barras.

Fuente: Propio autor.

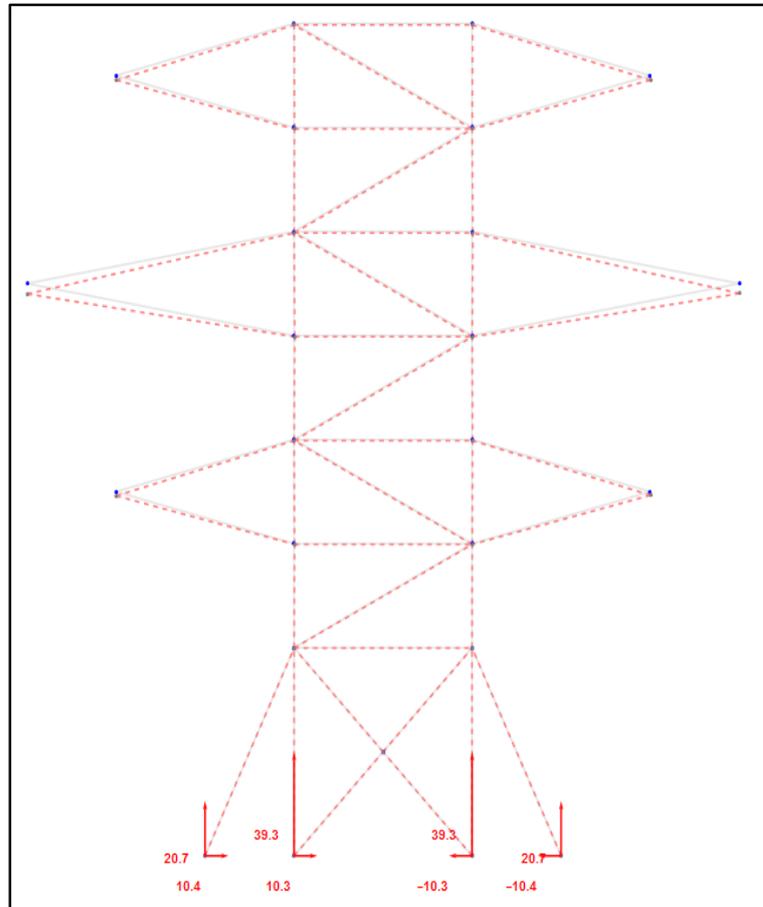


Figura 70. Gráfica de la armadura original y la armadura deformada generada por el programa, mostrando el desplazamiento (exagerado) de los nudos.

Fuente: Propio autor.

De forma adicional, se encontrará la fuerza máxima en las barras y los desplazamientos máximos en los nudos en la dirección x y la dirección y.

Para ello se ingresan las barras donde se desea comparar la fuerza máxima y los nudos donde se desean conocer los desplazamientos máximos así como las direcciones donde se quieren conocer dichos desplazamientos máximos.

Todo lo anterior se muestra en la siguiente imagen.

```
EleComFueMax1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,  
39, 40, 41, 42, 43, 44, 45};  
  
DirDesMax1 = 2;  
  
NodComDesMax1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25};
```

Figura 71. Ingreso de las variables para obtener la fuerza y desplazamiento máximos.

Fuente: Propio autor.

Para ejecutar el programa, se utilizan las funciones del mismo con las variables que contienen la información de comparación.

Todo lo anterior se muestra en la siguiente imagen.

```
{FueMax1, EleFueMax1} = FueMax[NumEle1, NumNod1, EleComFueMax1, VecFueEleLoc1];  
{DesMax1, NodDesMax1} = DesMax[NumEle1, NumNod1, DirDesMax1, NodComDesMax1, VecDesEstGlo1];
```

Figura 72. Forma en que se emplea el programa para obtener la fuerza y desplazamiento máximos.

Fuente: Propio autor.

Para conocer la magnitud de la fuerza máxima y la barra donde actúa y la magnitud del desplazamiento máximo y el nudo desplazado, se muestran sus respectivas variables. Esto se muestra en la siguiente figura.

```
FueMax1  
EleFueMax1  
60.8276  
43
```

Figura 73. Variables de fuerza máxima del ejemplo.

Fuente: Propio autor.

```
DesMax1  
NodDesMax1  
0.003333964  
1
```

Figura 74. Variables de desplazamiento máximo en la dirección x del ejemplo.

Fuente: Propio autor.

De forma adicional, también se mostrará las gráficas de la armadura mostrando la magnitud de las fuerzas en las barras y los desplazamientos en los nudos.

La forma en que se emplean las funciones necesarias, se muestra en la siguiente imagen.

```
{GraEstFue1} = GraEstFue[NumEle1, NumNod1, CoeEle1, CoeNod1, LonEle1, VecFueEleLoc1];  
|  
{GraEstDes1} = GraEstDes[NumEle1, NumNod1, CoeEle1, CoeNod1, LonEle1, VecDesEstGlo1];  
  
GraEstFue1  
GraEstDes1
```

Figura 75. Forma en que se emplea el programa para obtener la gráfica de la armadura mostrando las fuerzas en las barras y los desplazamientos en los nudos.

Fuente: Propio autor.

El programa arroja las siguientes imágenes; la imagen de la armadura mostrando las fuerzas en las barras y la imagen de la armadura mostrando los desplazamientos de los nudos.

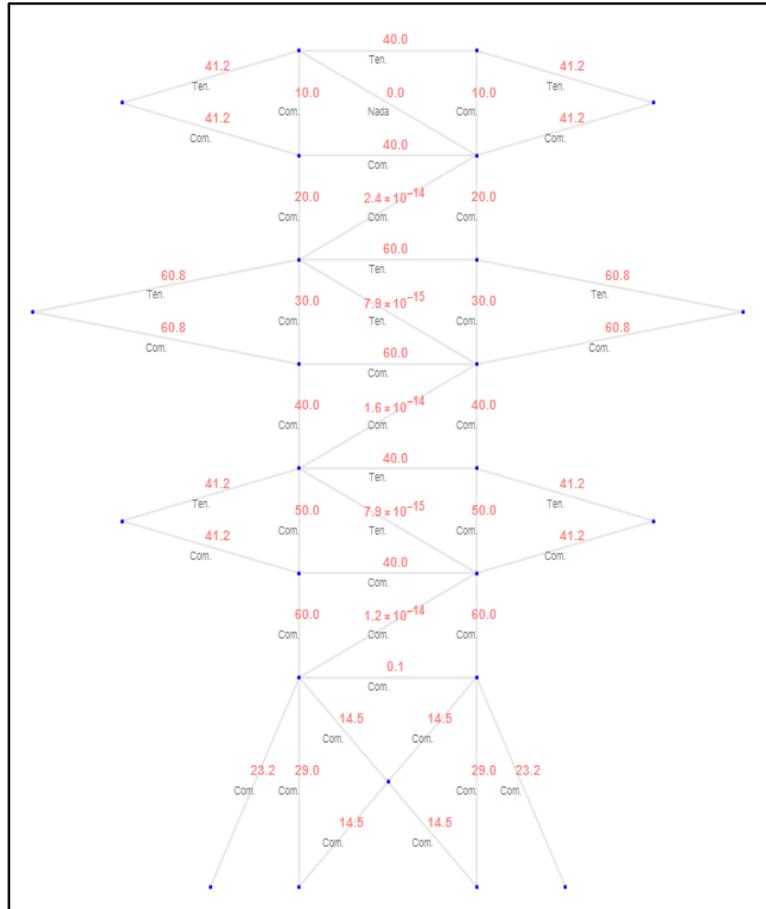


Figura 76. Gráfica de la armadura mostrando las fuerzas en las barras, generada por el programa.

Fuente: Propio autor.

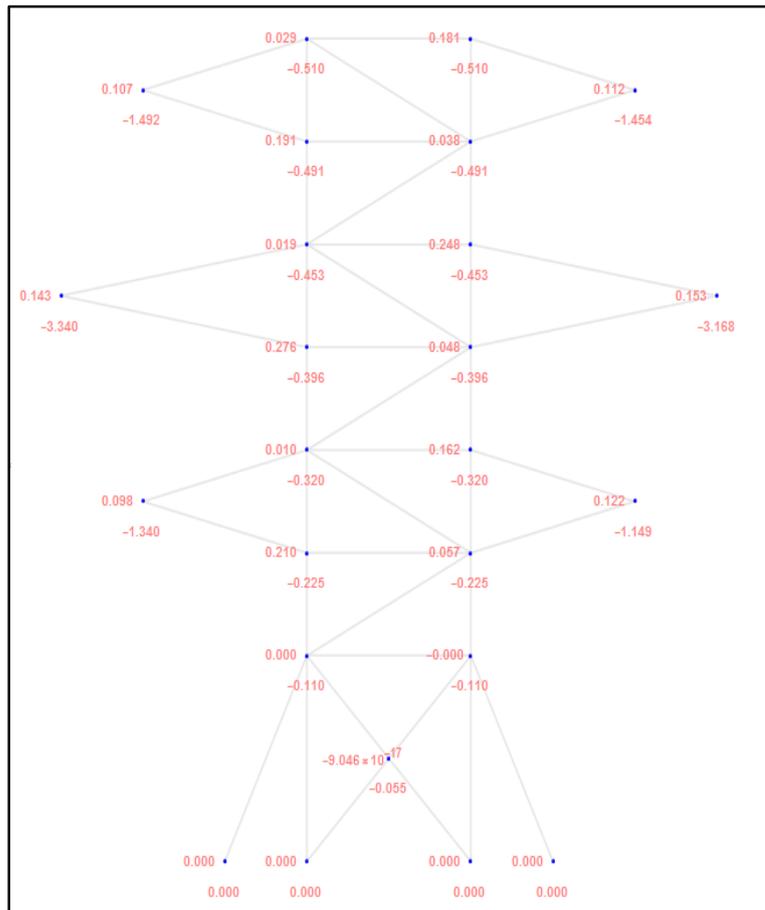


Figura 77. Gráfica de la armadura mostrando los desplazamientos de los nudos, generada por el programa.

Fuente: Propio autor.

## **5. Conclusiones**

El código del programa es pesado, ya que no utiliza las ventajas de la matriz de rigidez que implican menos cálculos internos y menos memoria empleada en la solución. Los más relevantes son la simetría de la matriz de rigidez y la banda angosta de la matriz, debido a que dicha matriz está plagada de ceros. En versiones posteriores del programa se recomienda utilizar dichas ventajas. Hay que decir que al no tener gran cantidad de nudos y elementos en una armadura, en el método de rigideces no existe gran problema en no utilizar las ventajas de la matriz de rigidez, a diferencia de que si se hiciera empleando el método del elemento finito.

La confiabilidad del código quedó plasmada con los ejemplos resueltos y comparados, aunque insuficientes quizás, resulta complicado mostrar la solución de gran cantidad de ejemplos. Debe decirse que no son los únicos ejemplos resueltos satisfactoriamente con el programa.

Se puede decir, en cuestión de confiabilidad, que existe el riesgo de que el programa arroje resultados erróneos si la armadura es inestable en el caso de que la solución emplee el enfoque de penalización, ya que en el enfoque de eliminación el programa no puede arrojar ningún resultado.

Este es un problema frecuente en los programas comerciales, ya que la mayoría de estos emplean el enfoque de penalización y algunos de ellos no ponen interés en la estabilidad de la armadura.

En versiones posteriores del programa se recomienda poner atención en la estabilidad de la armadura, y avisar al usuario de posibles inestabilidades.

Una ventaja del código es su naturaleza, en el sentido de que fácilmente puede ser modificado para utilizar el método de rigideces considerando la flexión en las barras.

## Glosario

Armadura: Estructura formada por barras dispuestas en forma de triángulos unidas mediante pasadores sin fricción.

Barra: Cuerpo de longitud mucho más grande comparado con sus otras dimensiones.

Cargar un cuerpo: Someter un cuerpo a la acción de fuerzas externas.

Continuo: Cuerpo que carece de huecos o vacíos.

Coordenadas globales: Sistema coordenadas elegidas arbitrariamente en una estructura.

Coordenadas locales: Sistema coordenadas más prácticas para conocer la deformación de una barra.

Cuerpo: Porción limitada de materia [2].

Deformación: Desplazamiento relativo entre partículas de un mismo cuerpo.

Deformación unitaria: Intensidad de la deformación.

Desplazamiento: Cambio de posición.

Equilibrio: Estado de un cuerpo en el que unas fuerzas compensan a otras [2].

Esfuerzo: Intensidad de las fuerzas internas.

Esfuerzo cortante: Esfuerzo tangencial al área donde actúa.

Esfuerzo normal: Esfuerzo ortogonal al área donde actúa.

Estabilidad: Que mantiene o recupera el equilibrio [7].

Estado de esfuerzos: Esfuerzos alrededor de una partícula.

Fuerza: Acción de un cuerpo sobre otro capaz de alterar su movimiento [2].

Fuerzas externas: Fuerzas provocadas por otros cuerpos o fenómenos que actúan sobre un cuerpo en estudio.

Fuerzas internas: Fuerzas presentes en el interior de un cuerpo provocadas por fuerzas externas.

Grados de libertad: Variables necesarias para definir la deformación de una barra.

Homogéneo: Cuerpo con la misma composición en cada partícula.

Isotrópico: Cuerpo que tiene las mismas propiedades en todas las direcciones.

Materia: Lo que ocupa un lugar en el espacio [2].

Matriz de rigidez global: Matriz de rigidez de una barra referida a las coordenadas globales.

Matriz de rigidez local: Matriz de rigidez de una barra referida a las coordenadas locales.

Método de las rigideces: Método para encontrar desplazamientos y reacciones en estructuras. Para ello emplea sistemas de ecuaciones lineales donde las incógnitas están dadas por los desplazamientos, los términos independientes por las condiciones de frontera de fuerzas y desplazamientos conocidos y los coeficientes de las incógnitas por las características de la armadura: dimensiones y material de las barras y forma de la armadura.

Nudo: Punto en el extremo longitudinal de una barra situado en el centro de área de la sección transversal.

Partícula: Punto dotado de materia [2].

Par de fuerzas o par: Sistema formado por dos fuerzas paralelas, de la misma magnitud, pero de sentido contrario [2].

Sistemas equivalentes de fuerzas: Aquellos que producen los mismos efectos externos [2].

## Referencias

### Páginas web

[1] UNAM. 20 de febrero de 2017. Facultad de Ingeniería. Ciudad de México, México. [www.ingeniería.unam.mx/programas\\_academicos/licenciatura/civil.php](http://www.ingeniería.unam.mx/programas_academicos/licenciatura/civil.php).

### Documentos (informes)

[2] Ocariz Castelazo, Juan. Estática. Fundamentos de la Mecánica Clásica. [www.dcb.unam.mx/users/juanoc/archivos/curso/curso.htm](http://www.dcb.unam.mx/users/juanoc/archivos/curso/curso.htm).

### Libros

[3] Timoshenko, S., (1951), Theory of Elasticity, EE. UU., Editorial: McGraw-Hill

[4] Gere, J., (2009), Mecánica de Materiales, Distrito Federal, México, Editorial: CENGAGE Learning.

[5] McCormac, J. C., (2010), Análisis de Estructuras. Métodos Clásico y Matricial, Distrito Federal, México, Editorial: Alfaomega.

[6] Chandrupatla, T. R., (1999), Introducción al Estudio del Elemento Finito en Ingeniería, Distrito Federal, México, Editorial: Pearson.

[7] Real Academia Español, (2009), Diccionario de la Lengua Española, México, Editorial: Espasa

## Anexos

### Código del programa

Sección: Entrada.

Subsección: Obtención de la matriz de rigidez de la armadura y elaboración de gráfica descriptiva de la armadura.

```
MatRigEstGlo[NumEle_, CooEle_, ModRigEle_, AreEle_, ConFroFue_,  
ConFroDes_] := Module[{CooNod, NumNod, IdeEleNod, IdeEleGraLib,  
NumDesPos, DesPos, NumDesCon, DesCon, DesDes, NumDesDes, LonEle,  
CosAngEle, MatTra, MatInvTra, MatRigEleLoc, MatApoRigEleGlo,  
MatRigEleGlo, MatRigEstGlo, VecConFroFue, VecConFroDes, GraEst},
```

```
CooNod = CoeEle[[1]];
```

```
For[i = 2, i <= NumEle, i++,  
CooNod = Union[CooNod, CoeEle[[i]]]  
];
```

```
NumNod = Length[CooNod];
```

```
IdeEleNod = Table[0, {NumEle}, {2}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++,  
IdeEleNod[[i, j]] = Norm[Flatten[Position[CooNod, CoeEle[[i, j]]]]]  
];
```

```
IdeEleGraLib = Table[0, {NumEle}, {4}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++,  
If[j <= 2,
```

```

IdeEleGraLib[[i, j]] = ((2*(IdeEleNod[[i, 1]] - 1)) + j),
IdeEleGraLib[[i, j]] = ((2*(IdeEleNod[[i, 2]] - 1)) + (j - 2))
]
]];
NumDesPos = (2*NumNod);

DesPos = Range[NumDesPos];

NumDesCon = Length[ConFroDes];

DesCon = Table[0, {NumDesCon}];

For[i = 1, i <= NumDesCon, i++,
DesCon[[i]] = ((2*(ConFroDes[[i, 1]] - 1)) + ConFroDes[[i, 2]])
];

DesDes = Complement[DesPos, DesCon];

NumDesDes = Length[DesDes];

LonEle = Table[0, {NumEle}];

For[i = 1, i <= NumEle, i++,
LonEle[[i]] = (Sqrt[(CooEle[[i, 2, 1]] - CooEle[[i, 1, 1]])^2 + (CooEle[[i, 2, 2]] -
CooEle[[i, 1, 2]])^2)
];

CosAngEle = Table[0, {NumEle}, {2}];

For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++,
CosAngEle[[i,j]] = ((CooNod[[IdeEleNod[[i, 2]], j]] - CooNod[[IdeEleNod[[i, 1]],
j]])/LonEle[[i]]
)];

```

```
MatTra = Table[0, {NumEle}, {1}, {2}, {4}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++, For[k = 1, k <= 4, k++,  
If[j == 1,  
If[k <= 2,  
MatTra[[i, 1, j, k]] = CosAngEle[[i, k]]  
],  
If[k > 2,  
MatTra[[i, 1, j, k]] = CosAngEle[[i, k - 2]]  
]]  
]]];
```

```
MatInvTra = Table[0, {NumEle}, {1}, {4}, {2}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++, For[k = 1, k <= 2, k++,  
If[j <= 2,  
If[k == 1 ,  
MatInvTra[[i, 1, j, k]] = CosAngEle[[i, j]]  
],  
If[k == 2,  
MatInvTra[[i, 1, j, k]] = CosAngEle[[i, j - 2]]  
]]  
]]];
```

```
MatRigEleLoc = Table[0, {NumEle}, {1}, {2}, {2}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++, For[k = 1, k <= 2, k++,  
If[j == k,  
MatRigEleLoc[[i, 1, j, k]] = (((AreEle[[i]]*ModRigEle[[i]])/LonEle[[i]])),  
MatRigEleLoc[[i, 1, j, k]] = -(((AreEle[[i]]*ModRigEle[[i]])/LonEle[[i]]))  
]  
]]];
```

```
MatApoRigEleGlo = Table[0, {NumEle}, {1}, {2}, {4}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++, For[k = 1, k <= 2, k++, For[h = 1, h <= 4, h++,  
MatApoRigEleGlo[[i, 1, j, h]] += (MatRigEleLoc[[i, 1, j, k]]*MatTra[[i, 1, k, h]])  
]]]]];
```

```
MatRigEleGlo = Table[0, {NumEle}, {1}, {4}, {4}];  
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++, For[k = 1, k <= 2, k++, For[h = 1, h <= 4, h++,  
MatRigEleGlo[[i, 1, j, h]] += (MatInvTra[[i, 1, j, k]]*MatApoRigEleGlo[[i, 1, k, h]])  
]]]]];
```

```
MatRigEstGlo = Table[0, {NumDesPos}, {NumDesPos}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++, For[k = 1, k <= 4, k++,  
MatRigEstGlo[[IdeEleGraLib[[i, j]], IdeEleGraLib[[i, k]]]] += MatRigEleGlo[[i, 1, j,  
k]]  
]]];
```

```
VecConFroFue = Table[0, {NumDesDes}];
```

```
For[i = 1, i <= Length[ConFroFue], i++, For[j = 1, j <= 2, j++,  
VecConFroFue[[Flatten[Position[DesDes, ((2*(ConFroFue[[i, 2]] - 1)) + j)]]]] =  
ConFroFue[[i, 1, j]]  
]]];
```

```
VecConFroDes = Table[0, {NumDesCon}];
```

```
For[i = 1, i <= NumDesCon, i++,  
VecConFroDes[[i]] = ConFroDes[[i, 3]]  
];
```

```
GraEst=Graphics[{  
Style[Line[CooEle],Black,Thickness[0.0025]],
```

```
Table[Style[Text[i,((1/2)*CooEle[[i,2]])+((1/2)*CooEle[[i,1]])+{(0.1*Min[LonEle]),(0.1*Min[LonEle])}],Black,FontWeight->Bold,FontFamily->"Arial",FontSize->10],{i,1,NumEle}],
```

```
Style[Point[CooNod],Blue,PointSize[0.005]],
```

```
Table[Style[Text[i,CooNod[[i]]+{(0.1*Min[LonEle]),(0.1*Min[LonEle])}],Blue,FontWeight->Bold,FontFamily->"Arial",FontSize->10],{i,1,NumNod}],
```

```
Table[Style[Arrow[{CooNod[[ConFroFue[[i,2]]]],((ConFroFue[[i,1]]/(3*Max[Abs[ConFroFue[[i,1]]])*(3*Min[LonEle])))+CooNod[[ConFroFue[[i,2]]]]},Red,Thickness[0.0025],Arrowheads[0.025]],{i,1,Length[ConFroFue]}],
```

```
Table[Style[Text[ConFroFue[[i,1]],((1/2)*((ConFroFue[[i,1]]/(3*Max[Abs[ConFroFue[[i,1]]])*(3*Min[LonEle])))+CooNod[[ConFroFue[[i,2]]]]-{(0.3*Min[LonEle]),(0.3*Min[LonEle])}],Red,FontWeight->Bold,FontFamily->"Arial",FontSize->10],{i,1,Length[ConFroFue]}];
```

```
{GraEst, NumNod, CooNod, IdeEleNod, IdeEleGraLib, NumDesPos, NumDesCon, NumDesDes, DesPos, DesCon, DesDes, LonEle, CosAngEle, MatTra, MatRigEleGlo, MatRigEstGlo, VecConFroFue, VecConFroDes}
]
```

Sección: Solución.

Subsección: Obtención del vector de desplazamiento y vector fuerza de la armadura.

Subsección: Enfoque de eliminación.

```
VecDesEstGloyVecFueEstGloEnfEli[NumDesPos_, NumDesCon_, NumDesDes_, DesPos_, DesCon_, DesDes_, MatRigEstGlo_, VecConFroFue_, VecConFroDes_] := Module[{MatRigEst, VecFueEst, EscApoRigEst, VecDesEst, EscApoDesEst, VecDesEstGlo, VecFueEstGlo},
```

```
MatRigEst = Table[0, {NumDesDes}, {NumDesDes}];
```

```
For[i = 1, i <= NumDesDes, i++, For[j = 1, j <= NumDesDes, j++,
```

```
MatRigEst[[i, j]] = MatRigEstGlo[[DesDes[[i]], DesDes[[j]]]]];
```

```
VecFueEst = Table[0, {NumDesDes}];
```

```
For[i = 1, i <= NumDesDes, i++, For[j = 1, j <= NumDesCon, j++,  
VecFueEst[[i]] += (MatRigEstGlo[[DesDes[[i]], DesCon[[j]]]]*VecConFroDes[[j]]  
)];
```

```
For[i = 1, i <= NumDesDes, i++,  
VecFueEst[[i]] = (VecConFroFue[[i]] - VecFueEst[[i]])  
];
```

```
For[k = 1, k <= (NumDesDes - 1), k++, For[i = (k + 1), i <= NumDesDes, i++,  
EscApoRigEst = (MatRigEst[[i, k]]/MatRigEst[[k, k]]);  
For[j = (k + 1), j <= NumDesDes, j++,  
MatRigEst[[i, j]] = (MatRigEst[[i, j]] - (EscApoRigEst*MatRigEst[[k, j]]));  
];  
VecFueEst[[i]] = (VecFueEst[[i]] - (EscApoRigEst*VecFueEst[[k]]))  
];
```

```
VecDesEst = Table[0, {NumDesDes}];
```

```
VecDesEst[[NumDesDes]] =  
(VecFueEst[[NumDesDes]]/MatRigEst[[NumDesDes, NumDesDes]]);
```

```
For[i = 1, i <= (NumDesDes - 1), i++,  
EscApoDesEst = 0;  
For[j = ((NumDesDes - i) + 1), j <= NumDesDes, j++,  
EscApoDesEst += (MatRigEst[[NumDesDes - i, j]]*VecDesEst[[j]]  
];  
VecDesEst[[NumDesDes - i]] = ((1/MatRigEst[[NumDesDes - i, NumDesDes - i]])*(VecFueEst[[NumDesDes - i]] - EscApoDesEst))  
];
```

```
VecDesEstGlo = Table[0, {NumDesPos}];
```

```
For[i = 1, i <= NumDesCon, i++,  
VecDesEstGlo[[DesCon[[i]]]] = VecConFroDes[[i]]  
];
```

```
For[i = 1, i <= NumDesDes, i++,  
VecDesEstGlo[[DesDes[[i]]]] = VecDesEst[[i]]  
];
```

```
VecFueEstGlo = Table[0, {NumDesPos}];
```

```
For[i = 1, i <= NumDesDes, i++,  
VecFueEstGlo[[DesDes[[i]]]] = VecConFroFue[[i]]  
];
```

```
For[i = 1, i <= NumDesCon, i++, For[j = 1, j <= NumDesPos, j++,  
VecFueEstGlo[[DesCon[[i]]], DesPos[[j]]] += (MatRigEstGlo[[DesCon[[i]],  
DesPos[[j]]]]*VecDesEstGlo[[j]]  
)];
```

```
{VecDesEstGlo, VecFueEstGlo}  
]
```

Subsección: Enfoque de penalización.

```
VecDesEstGloyVecFueEstGloEnfPen[NumDesPos_, NumDesCon_,  
NumDesDes_, DesPos_, DesCon_, DesDes_, MatRigEstGlo_, VecConFroFue_,  
VecConFroDes_] := Module[{EscApoRigFic, MatRigEst, VecFueEst,  
EscApoRigEst, VecDesEst, EscApoDesEst, VecDesEstGlo, VecFueEstGlo},
```

```
EscApoRigFic = (Max[MatRigEstGlo]*10^4);
```

MatRigEst = Table[0, {NumDesPos}, {NumDesPos}];

For[i = 1, i <= NumDesPos, i++, For[j = 1, j <= NumDesPos, j++,  
MatRigEst[[i, j]] = MatRigEstGlo[[i, j]]  
]];

For[i = 1, i <= NumDesCon, i++,  
MatRigEst[[DesCon[[i]], DesCon[[i]]]] = MatRigEstGlo[[DesCon[[i]], DesCon[[i]]]]  
+ EscApoRigFic  
];

VecFueEst = Table[0, {NumDesPos}];

For[i = 1, i <= NumDesDes, i++,  
VecFueEst[[DesDes[[i]]]] = VecConFroFue[[i]]  
];

For[i = 1, i <= NumDesCon, i++,  
VecFueEst[[DesCon[[i]]]] = (EscApoRigFic\*VecConFroDes[[i]])  
];

For[k = 1, k <= (NumDesPos - 1), k++, For[i = (k + 1), i <= NumDesPos, i++,  
EscApoRigEst = (MatRigEst[[i, k]]/MatRigEst[[k, k]]);  
For[j = (k + 1), j <= NumDesPos, j++,  
MatRigEst[[i, j]] = (MatRigEst[[i, j]] - (EscApoRigEst\*MatRigEst[[k, j]]));  
];  
VecFueEst[[i]] = (VecFueEst[[i]] - (EscApoRigEst\*VecFueEst[[k]]))  
]];

VecDesEst = Table[0, {NumDesPos}];

VecDesEst[[NumDesPos]] =  
(VecFueEst[[NumDesPos]]/MatRigEst[[NumDesPos, NumDesPos]]);

```

For[i = 1, i <= (NumDesPos - 1), i++,
EscApoDesEst = 0;
For[j = ((NumDesPos - i) + 1), j <= NumDesPos, j++,
EscApoDesEst += (MatRigEst[[{NumDesPos - i, j}]]*VecDesEst[[j]])
];
VecDesEst[[{NumDesPos - i}]] = ((1/MatRigEst[[{NumDesPos - i, NumDesPos - i}]])*(VecFueEst[[{NumDesPos - i}]] - EscApoDesEst))
];

VecDesEstGlo = Table[0, {NumDesPos}];

For[i = 1, i <= NumDesPos, i++,
VecDesEstGlo[[i]] = VecDesEst[[i]]
];

VecFueEstGlo = Table[0, {NumDesPos}];

For[i = 1, i <= NumDesDes, i++,
VecFueEstGlo[[DesDes[[i]]]] = VecConFroFue[[i]]
];

For[i = 1, i <= NumDesCon, i++,
VecFueEstGlo[[DesCon[[i]]]] = (-(EscApoRigFic*(VecDesEstGlo[[DesCon[[i]]]] - VecConFroDes[[i]])))
];

{VecDesEstGlo, VecFueEstGlo}
]

```

Subsección: Obtención de las fuerzas y desplazamientos de cada barra

```

VecFueEleLocyVecDesEleLoc[NumEle_, IdeEleGraLib_, MatTra_,
VecDesEstGlo_, MatRigEleGlo_] := Module[{VecDesEleGlo, VecFueEleGlo,
VecDesEleLoc, VecFueEleLoc},

```

```
VecDesEleGlo = Table[0, {NumEle}, {1}, {4}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++,  
VecDesEleGlo[[i, 1, j]] = VecDesEstGlo[[IdeEleGraLib[[i, j]]]  
]]];
```

```
VecFueEleGlo = Table[0, {NumEle}, {1}, {4}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 4, j++, For[k = 1, k <= 4, k++,  
VecFueEleGlo[[i, 1, j]] += (MatRigEleGlo[[i, 1, j, k]]*VecDesEleGlo[[i, 1, k]]  
]]];
```

```
VecDesEleLoc = Table[0, {NumEle}, {1}, {2}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++, For[k = 1, k <= 4, k++,  
VecDesEleLoc[[i, 1, j]] += (MatTra[[i, 1, j, k]]*VecDesEleGlo[[i, 1, k]]  
]]];
```

```
VecFueEleLoc = Table[0, {NumEle}, {1}, {2}];
```

```
For[i = 1, i <= NumEle, i++, For[j = 1, j <= 2, j++, For[k = 1, k <= 4, k++,  
VecFueEleLoc[[i, 1, j]] += (MatTra[[i, 1, j, k]]*VecFueEleGlo[[i, 1, k]]  
]]];
```

```
{VecDesEleGlo, VecFueEleGlo, VecDesEleLoc, VecFueEleLoc}  
]
```

Sección: Análisis de resultados.

Subsección: Fuerza máxima.

```
FueMax[NumEle_, NumNod_, EleComFueMax_, VecFueEleLoc_] :=  
Module[{FueMax, DesMax, EleFueMax, NodDesMax},
```

```
FueMax = 0;
```

```
For[i = 1, i <= Length[EleComFueMax], i++,  
FueMax = Max[FueMax, Abs[VecFueEleLoc[[EleComFueMax[[i]], 1, 1]]]]  
];
```

```
For[i = 1, i <= NumEle, i++,  
If[(0.999* FueMax) <= Abs[VecFueEleLoc[[i, 1, 1]]] <= (1.001* FueMax),  
EleFueMax = i  
]  
];
```

```
{FueMax, EleFueMax}  
]
```

Subsección: Desplazamiento máximo.

```
DesMax[NumEle_, NumNod_, DirDesMax_, NodComDesMax_,  
VecDesEstGlo_] := Module[{DesMax, NodDesMax},
```

```
DesMax = 0;
```

```
For[i = 1, i <= Length[NodComDesMax], i++,  
DesMax = Max[DesMax, Abs[VecDesEstGlo[[((2*(NodComDesMax[[i]] - 1)) +  
DirDesMax)]]]]  
];
```

```
For[i = 1, i <= NumNod, i++,  
If[(0.999* DesMax) <= Abs[VecDesEstGlo[[((2*(i - 1)) + DirDesMax)]]] <= (1.001*  
DesMax),  
NodDesMax = i  
]  
];
```

```
{ DesMax, NodDesMax}
```

```
]
```

Subsección: Gráfica de la armadura deformada.

```
GraEstyGraEstDes[NumEle_, NumNod_, CooEle_, CooNod_, LonEle_,  
VecDesEleGlo_, VecDesEstGlo_, ConFroDes_, VecFueEstGlo_] :=  
Module[{GraEstyGraEstDes},
```

```
GraEstyGraEstDes = Graphics[{  
Style[Line[CooEle], Black, Thickness[0.0025]],
```

```
Table[Style[Line[{CooEle[[i, 1]], CooEle[[i, 2]] +  
((Min[LonEle]/(10*Max[Abs[VecDesEstGlo]]))*{VecDesEleGlo[[i, 1], 1]],  
VecDesEleGlo[[i, 1], 2]]}], CooEle[[i, 2]] +  
((Min[LonEle]/(10*Max[Abs[VecDesEstGlo]]))*{VecDesEleGlo[[i, 1], 3]],  
VecDesEleGlo[[i, 1], 4]]}], Gray, Thickness[0.0025], Dashed], {i, 1, NumEle}],
```

```
Table[Style[Text[i, ((1/2)*CooEle[[i, 2]] + ((1/2)*CooEle[[i, 1]] +  
{(0.1*Min[LonEle]), (0.1*Min[LonEle])}], Black, FontWeight -> Bold, FontFamily -  
> "Arial", FontSize -> 10], {i, 1, NumEle}],
```

```
Style[Point[CooNod], Blue, PointSize[0.005]], Table[Style[Point[CooNod[[i]] +  
((Min[LonEle]/(10*Max[Abs[VecDesEstGlo]]))*{VecDesEstGlo[[(2*(i - 1)) + 1]],  
VecDesEstGlo[[(2*(i - 1)) + 2]]}], Gray, PointSize[0.005]], {i, 1, NumNod}],
```

```
Table[Style[Text[i, CooNod[[i]] - {(0.1*Min[LonEle]), (0.1*Min[LonEle])}], Blue,  
FontWeight -> Bold, FontFamily -> "Arial", FontSize -> 10], {i, 1, NumNod}],
```

```
Table[Style[Arrow[{CooNod[[ConFroDes[[i, 1]]], CooNod[[ConFroDes[[i, 1]]] +  
((VecFueEstGlo[[(2*(ConFroDes[[i, 1]] - 1)) + ConFroDes[[i,  
2]]])]/(3*Max[Abs[VecFueEstGlo]]*(3*Min[LonEle])))*{If[ConFroDes[[i, 2]] == 1, 1,  
0], If[ConFroDes[[i, 2]] == 2, 1, 0]}}, Pink, Thickness[0.0025],  
Arrowheads[0.025]], {i, 1, Length[ConFroDes]}],
```

```
Table[Style[Text[NumberForm[VecFueEstGlo[[(2*(ConFroDes[[i, 1]] - 1)) +
ConFroDes[[i, 2]]], {10, 0}], Coonod[[ConFroDes[[i, 1]]] - {(0.1*Min[LonEle]),
(0.1*Min[LonEle])} + ((1/2)*((VecFueEstGlo[[(2*(ConFroDes[[i, 1]] - 1)) +
ConFroDes[[i,
2]]])/(3*Max[Abs[VecFueEstGlo]]*(3*Min[LonEle])))]*{If[ConFroDes[[i, 2]] == 1,
1, 0], If[ConFroDes[[i, 2]] == 2, 1, 0]}], Pink, FontWeight -> Bold, FontFamily ->
"Arial", FontSize -> 10], {i, 1, Length[ConFroDes]}]];
```

```
{GraEstyGraEstDes}
]
```

Subsección: Gráfica de la armadura mostrando fuerzas.

```
GraEstFue[NumEle_, NumNod_, CoeEle_, Coonod_, LonEle_,
VecFueEleLoc_] := Module[{GraEstFue, VecApoVecFueEleLoc, EsfAxiEleLoc},
```

```
VecApoVecFueEleLoc = Table[0, {NumEle}, {2}];
```

```
For[i = 1, i <= NumEle, i++,
VecApoVecFueEleLoc[[i]] = Flatten[VecFueEleLoc[[i]]]
];
```

```
EsfAxiEleLoc = Table[0, {NumEle}];
```

```
For[i = 1, i <= NumEle, i++, If[Sign[VecApoVecFueEleLoc[[i, 1]]] == -1,
EsfAxiEleLoc[[i]] = "Ten.", If[Sign[VecApoVecFueEleLoc[[i, 1]]] == 1,
EsfAxiEleLoc[[i]] = "Com.", If[Sign[VecApoVecFueEleLoc[[i, 1]]] == 0,
EsfAxiEleLoc[[i]] = "Nada"]]];
];
```

```
GraEstFue = Graphics[{
Style[Line[CoeEle], Black, Thickness[0.0025]],
```

```
Style[Point[CooNod], Blue, PointSize[0.005]],
```

```
Table[Style[Text[NumberForm[Abs[VecApoVecFueEleLoc[[i, 1]]], {10, 0}],  
((1/2)*CooEle[[i, 2]]) + ((1/2)*CooEle[[i, 1]]) + {(0.1*Min[LonEle]),  
(0.1*Min[LonEle])}], Pink, FontWeight -> Bold, FontFamily -> "Arial", FontSize ->  
10], {i, 1, NumEle}],
```

```
Table[Style[Text[EsfAxiEleLoc[[i]], ((1/2)*CooEle[[i, 2]]) + ((1/2)*CooEle[[i, 1]]) -  
{(0.1*Min[LonEle]), (0.1*Min[LonEle])}], Gray, FontWeight -> Plain, FontFamily ->  
"Arial", FontSize -> 8], {i, 1, NumEle}]  
};
```

```
{GraEstFue}  
]
```

Subsección: Gráfica de la armadura mostrando desplazamientos.

```
GraEstDes[NumEle_, NumNod_, CooEle_, CooNod_, LonEle_, VecDesEstGlo_] := Module[{GraEstDes},
```

```
GraEstDes = Graphics[{  
Style[Line[CooEle], Black, Thickness[0.0025]],
```

```
Style[Point[CooNod], Blue, PointSize[0.005]],
```

```
Table[Style[Text[NumberForm[VecDesEstGlo[[ $(2^i) - 1$ ]]*103, {10, 1}],  
CooNod[[i]] - {(0.3*Min[LonEle]), 0}], Pink, FontWeight -> Bold, FontFamily ->  
"Arial", FontSize -> 10], {i, 1, NumNod}],
```

```
Table[Style[Text[NumberForm[VecDesEstGlo[[ $(2^i)$ ]]*103, {10, 1}], CooNod[[i]] -  
{0, (0.3*Min[LonEle])}], Pink, FontWeight -> Bold, FontFamily -> "Arial", FontSize  
-> 10], {i, 1, NumNod}]  
};
```

{GraEstDes}

]