

**U**NIVERSIDAD **A**UTÓNOMA  
DE **N**UEVO **L**EÓN

**F**ACULTAD DE **I**NGENIERÍA  
**M**ECÁNICA Y **E**LÉCTRICA

**LAB** de **B**iomecánica

**Brigada 202**

**Práctica 3.** *Diseño de la estructura de un Panorámico.*

### **Equipo 3**

<u>Nombre</u>	<u>Matricula</u>	<u>Carrera</u>
<b>Fátima</b> <b>M</b> ontserrat <b>Z</b> arazúa <b>U</b> ribe	1811014	IMTC
<b>Jair</b> <b>A</b> lejandro <b>T</b> amayo <b>I</b> barra	1815498	IMTC
<b>Jorge</b> <b>L</b> uis <b>Á</b> vila <b>H</b> ernández	1905338	IMTC
<b>Karen</b> <b>A</b> lexa <b>P</b> érez <b>O</b> rtiz	1904708	IMTC
<b>Ana</b> <b>C</b> ristina <b>L</b> ucio <b>I</b> racheta	1907905	IMTC
<b>Ana</b> <b>B</b> elén <b>B</b> olaños <b>C</b> arbajal	1908896	IMTC

**Docente:** **Dra.** **Yadira** **M**oreno **V**era

Semestre **A**gosto – **D**iciembre **2022**

**Día y Hora:** **Martes** **N3**

San Nicolás de los Garza, N.L. a 18 de octubre de 2022

## Objetivo

Desarrollar en el estudiante la capacidad de análisis, implementación y solución de un problema propuesto.

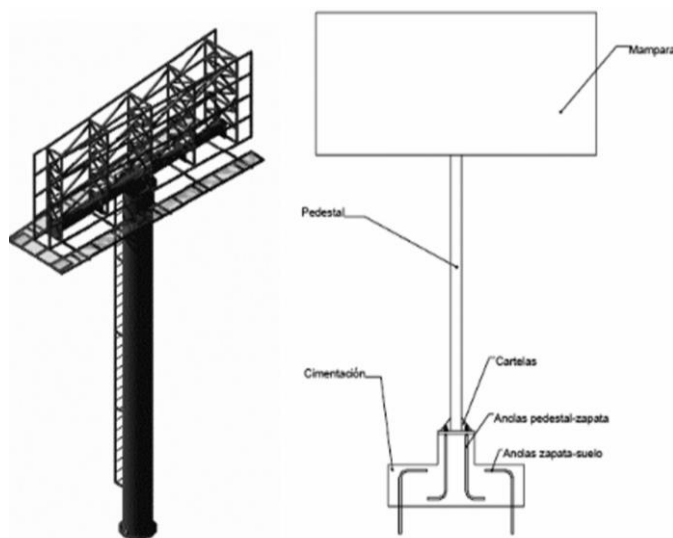
El estudiante deberá realizar el estudio de análisis paramétrico de la estructura de un panorámico, para el cual se desarrollará una propuesta de implementación y solución para su respectiva optimización topológica.

## 1. Nombre y definición de la forma Geométrica.

### Estructura de un Panorámico

Las estructuras tipo anuncios espectaculares son susceptibles a los efectos de las acciones de empuje de viento de acuerdo con su configuración geométrica, tomando como punto principal de estudio el pedestal de soporte de este.

Este tipo de estructura aparentemente es muy sencilla porque cuenta solamente con *tres partes principales* que son: la *mampara*, el *pedestal* y la *cimentación*; en la **Figura 1** se muestran esquemáticamente estas tres partes.



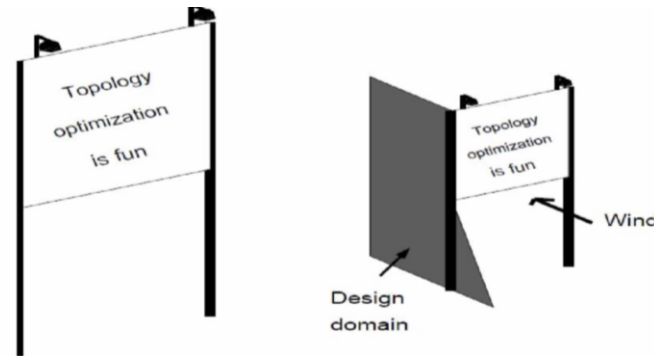
**Figura 1.-Estructura de un Panorámico**

Sin embargo, vista en forma minuciosa, una mampara consta de varios componentes y accesorios que hacen que esta estructura sea realmente muy compleja tanto en su diseño estructural, como en su construcción y también en su comportamiento sobre todo ante viento como el producido por huracán o tromba.

Tanto la cimentación como el pedestal y la mampara elevada pueden constar de diversos elementos tales como: anclas suelo–zapata, vigas estabilizadoras, anclas pedestal–zapata, lastres, placas-base, acartelamientos, el tubo del pedestal, escaleras externas e internas, andamios, placas de conexión pedestal-mampara, travesaño principal de la mampara, placas verticales del travesaño, armaduras, pernos de sujeción, láminas de la mampara, accesorios de iluminación, ganchos o argollas de sujeción, travesaños secundarios; además, se tienen diversos elementos de sujeción o conexión tales como pernos, tornillos, remaches, soldaduras, etc.

## 2. Estado del arte.

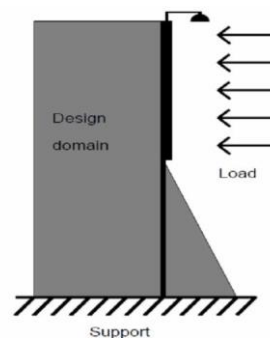
Los panorámicos se exponen a altas ráfagas de viento, por lo que su estructura ocupa ser muy rígida para soportar estas fuerzas. El espacio de diseño a evaluar será de 2 dimensiones, las cargas y los apoyos de observan en la **Figura 3.1**.



**Figura 2.-**Imagen de un Panorámico

En la **Figura 3.2** se puede ver el espacio de diseño para esta práctica. Se espera una fracción volumétrica aproximada de 0.20% del espacio de diseño. Supongamos que el panorámico es muy rígido 1, y sus patas son del mismo material que el marco.

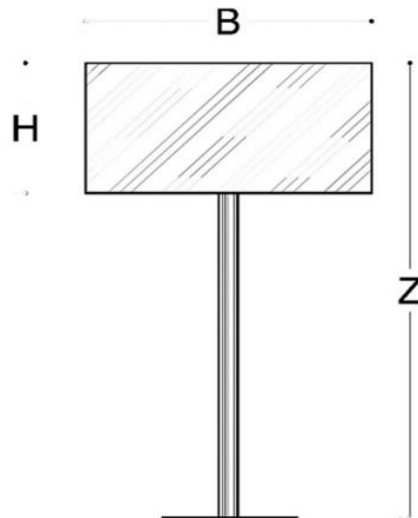
En la metodología se abordan algunas consideraciones como: las secciones de mampara según la altura del anuncio, la sección del tubo pedestal, velocidades de viento y los materiales.



**Figura 3.-**Espacio de Diseño

### 3. Propuesta de diseño de geometría, alcances y limitaciones.

Se plantean las dimensiones de la mampara elevada (figura 2), la cual tendrá como altura del anuncio de 5, 10, 15, 20 y 25 metros, con mamparas con dimensiones donde la relación base y altura de como resultado 2; como se muestra en la tabla correspondiente. Las velocidades del viento que están actuando en la mampara y tubo del pedestal, serán tomadas con velocidades mínima de 80km/hr y un máximo de 80 km/hr, con incrementos de 10 km/hr.



**Figura 4.-Geometría de un Panorámico**

**Tabla 1. Secciones de mampara**

Propuesta de Secciones de Mampara Elevada			
Altura anuncio z (m)	Sección Mampara		Relación H/B
	B (m)	H(m)	
5.00	3.00	1.50	2.00
	2.50	1.25	2.00
	2.00	1.00	2.00
10.00	5.00	2.50	2.00
	4.00	2.00	2.00
	3.00	1.50	2.00
15.00	6.50	3.25	2.00
	5.00	2.50	2.00
	4.00	2.00	2.00
20.00	7.50	3.75	2.00
	6.50	3.25	2.00
	5.00	2.50	2.00
25.00	10.00	5.00	2.00
	8.50	4.25	2.00
	7.50	3.75	2.00

La propuesta para la sección de tubo de pedestal la cual se optó de sección circular hueca está en función de asegurar la estabilidad de la estructura tomando en cuenta para eso la relación de esbeltez (RE).

$$R_E = \frac{kl}{r}$$

donde:

RE = Relación de Esbeltez

k = factor de longitud efectiva

l = longitud del elemento

r = radio de giro

De acuerdo con la fórmula de Relación de esbeltez, se dieron valores a cada término para obtener las propuestas del pedestal, de la siguiente manera:

- La longitud del elemento, l, se propuso de acuerdo con la altura de los anuncios a analizar; 5 metros, 10 metros, 15 metros, 20 metros y 25 metros respectivamente.
- El factor de longitud efectiva se tomó de la tabla C-A-7.1 del Reglamento AISC 360-16, donde caemos en el tipo de caso (e) y se toma como valor de “k” igual a 2.
- Para el radio de giro se realizó un concentrado de todos los perfiles de sección hueca en una tabla de Excel y se vinculó los valores de longitud, factor de longitud efectiva y los radios de giro de cada elemento; y de esta manera obtener relaciones de esbeltez para cada perfil.

Tabla 2. Relación de secciones circulares según altura de anuncio

Altura Anuncio		5.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	120
273.00	9.27	9.33	107.18	≤	120
I = 6693.37		cm <sup>3</sup>			
Altura Anuncio		10.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	120
508.00	15.09	17.44	114.68	≤	120
I = 71032.8		cm <sup>3</sup>			
Altura Anuncio		15.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	120
762.00	19.05	26.28	114.16	≤	120
I = 306983.94		cm <sup>3</sup>			
Altura Anuncio		20.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	120
1,067.00	25.40	36.84	108.58	≤	120
I = 1127849.85		cm <sup>3</sup>			
Altura Anuncio		25.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	120
1,219.00	25.40	42.21	118.46	≤	120
I = 169627.35		cm <sup>3</sup>			
Altura Anuncio		5.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	200
168.00	7.11	5.70	175.44	≤	200
I = 6693.37		cm <sup>3</sup>			
Altura Anuncio		10.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	200
324.00	10.31	11.09	180.34	≤	200
I = 12498.8		cm <sup>3</sup>			
Altura Anuncio		15.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	200
457.00	14.27	15.66	191.57	≤	200
I = 48680.2		cm <sup>3</sup>			
Altura Anuncio		20.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	200
610.00	17.48	20.96	190.84	≤	200
I = 142917		cm <sup>3</sup>			
Altura Anuncio		25.00 M			
Tubo Circular		Cedula 40			
D (mm)	t (mm)	r (cm)	RE	≤	200
762.00	19.05	26.28	190.26	≤	200
I = 306983.94		cm <sup>3</sup>			

Se tomaran ciertas consideraciones para la solución de esta práctica: 5 cargas, los apoyos tendrán restricciones en “X”, “Y” y el espacio de diseño para esta práctica será de:

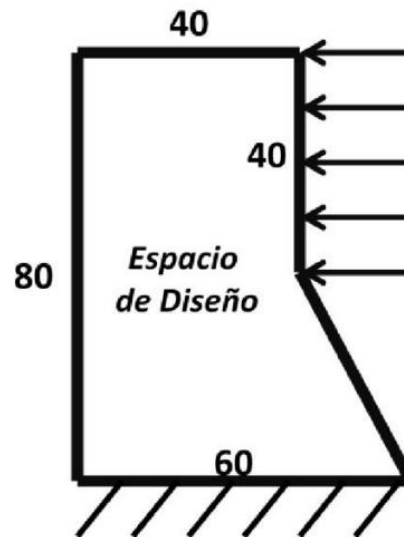


Figura 5.-Espacio de Diseño

#### 4. Pasos del desarrollo de la programación.

Se va a utilizar el código de la práctica 1 dándole unos cambios necesarios para poder implementar de manera exitosa la nueva implementación, la cual en este caso es el panorámico.

Código original

```

1      %%%% PRACTICA 3 LABORATORIO BIOMECANICA %%%
2      function toppl(nelx,nely,volfrac,penal,rmin)
3          % INITIALIZE
4          x(1:nely,1:nelx) = volfrac;
5          loop = 0;
6          change = 1.;
7          % START ITERATION
8          while change > 0.01
9              loop = loop + 1;
10             xold = x;
11             % FE-ANALYSIS
12             [U]=FE(nelx,nely,x,penal);
13             % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
14             [KE] = lk;
15             c = 0.;
16             for ely = 1:nely
17                 for elx = 1:nelx
18                     n1 = (nely+1)*(elx-1)+ely;
19                     n2 = (nely+1)* elx +ely;
20                     Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],1);
21                     c = c + x(ely,elx)^penal*Ue'*KE*Ue;
22                     dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
23                 end
24             end
25             % FILTERING OF SENSITIVITIES
26             [dc] = check(nelx,nely,rmin,x,dc);

```

```

27 - % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
28 - [x] = OC(nelx,nely,x,volfrac,dc);
29 - % PRINT RESULTS
30 - change = max(max(abs(x-xold)));
31 - disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
32 - ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
33 - ' ch.: ' sprintf('%6.3f',change )])
34 - % PLOT DENSITIES
35 - colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
36 - end
37 - %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
38 - function [xnew]=OC(nelx,nely,x,volfrac,dc) l1 = 0; l2 = 100000; move = 0.2;
39 - while (l2-l1 > 1e-4)
40 -     lmid = 0.5*(l2+l1);
41 -     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid))));
42 -     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
43 -         l1 = lmid;
44 -     else
45 -         l2 = lmid;
46 -     end
47 - end
48 - %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
49 - function [dcn]=check(nelx,nely,rmin,x,dc)
50 -     dcn=zeros(nely,nelx);
51 -     for i = 1:nelx
52 -         for j = 1:nely
53 -             sum=0.0;
54 -             for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
55 -                 for l = max(j-round(rmin),1):min(j+round(rmin), nely)
56 -                     fac = rmin-sqrt((i-k)^2+(j-l)^2);
57 -                     sum = sum+max(0,fac);
58 -                     dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
59 -                 end
60 -             end
61 -             dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
62 -         end
63 -     end
64 - %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
65 - function [U]=FE(nelx,nely,x,penal)
66 -     [KE] = lk;
67 -     K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
68 -     F = sparse(2*(nely+1)*(nelx+1),1); U =sparse(2*(nely+1)*(nelx+1),1);
69 -     for ely = 1:nely
70 -         for elx = 1:nelx
71 -             n1 = (nely+1)*(elx-1)+ely;
72 -             n2 = (nely+1)* elx +ely;
73 -             edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
74 -             K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
75 -         end
76 -     end
77 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)

```

```

78 - F(2,1) = -1;
79 - fixeddofs = union([1:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
80 - alldofs = [1:2*(nely+1)*(nelx+1)];
81 - freedofs = setdiff(alldofs, fixeddofs);
82 - % SOLVING 127
83 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
84 - U(fixeddofs,:) = 0;
85 - %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
86 - function [KE]=lk
87 - E = 1.;
88 - nu = 0.3;
89 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
90 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
91 - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
92 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
93 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
94 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
95 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
96 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
97 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
98 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
99 - .

```

### Cambios para fuerzas múltiples

Se tiene que editar el script para poder ingresar las fuerzas necesarias, observando vemos que tenemos 5 y para cambiar el anclaje del espacio de diseño a otra posición se tiene que cambiar con la instrucción `fixeddofs` modificando ciertas líneas

```

%%%%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%%%%
function [U]=FE(nelx,nely,x,penal)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1), 5);
U = sparse(2*(nely+1)*(nelx+1), 5);

```

Figura 1. Fragmento del código con líneas modificadas

```

for ely = 1:nely
for elx = 1:nelx
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely;
Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],1);
c = c + x(ely,elx)^penal*Ue'*KE*Ue;
dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
end
end

```

Figura 2. Fragmento del código original



```

]for ely = 1:nely
]for elx = 1:nelx
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx +ely;
    dc(ely,elx)= 0.;
]for i =1:5
    Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],1);
    c = c + x(ely,elx)^penal*Ue'*KE*Ue;
    dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
-end
-end
-end

```

Figura 3. Fragmento del código con líneas modificadas

```

% DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
F(2,1) = -1;
fixeddofs =union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);

```

Figura 4. Fragmento del código original

```

% DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
F(2*(nelx)*(nely+1)+2,1) = 1;
F(2*(nelx)*(nely+1)+(nely/4),2) = 1;
F(2*(nelx)*(nely+1)+(nely/2),3) = 1;
F(2*(nelx)*(nely+1)+(nely),4) = 1;
F(2*(nelx)*(nely+1)+(nely*1.2),5) = 1;
fixeddofs =union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);

```

Figura 5. Fragmento del código con líneas modificadas

### Empotramiento diagonal (elementos pasivos)

Para crear el empotramiento diagonal, o crear el espacio en blanco para recrear el empotramiento en la parte inferior derecha; en el archivo del uso del código de 99 líneas existe una sección donde se habla de elementos pasivos el cual sirve de ayuda para determinar un espacio en blanco, necesitamos modificar el código original para poder recrear el espacio conocido con los siguientes cambios en la codificación para llegar al resultado del panorámico.

```

%%% PRACTICA 3 LABORATORIO BIOMECHANICA 2 %%%
]function toppl(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION

```

Figura 6. Fragmento del código original

```

for ely = 1:nely
for elx = 41:nelx
    if elx -20 < (ely/2)
        passive(ely,elx)=0;
    else
        passive(ely,elx)=1;
    end
end
end
end

```

Figura 7. Fragmento del código con líneas modificadas

```

% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc);

##### OPTIMALITY CRITERIA UPDATE #####
function [xnew]=OC(nelx,nely,x,volfrac,dc) l1 = 0; l2 = 100000; move = 0.2;

```

Figura 8. Fragmento del código original

```

% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);

##### OPTIMALITY CRITERIA UPDATE #####
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)

```

Figura 9. Fragmento del código con líneas modificadas

```

while (l2-l1 > 1e-4)
lmid = 0.5*(l2+l1);
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
if sum(sum(xnew)) - volfrac*nelx*nely > 0;

```

Figura 10. Fragmento del código original

```

while (l2-l1 > 1e-4)
lmid = 0.5*(l2+l1);
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
xnew(find(passive)) = 0.001;
if sum(sum(xnew)) - volfrac*nelx*nely > 0;

```

Figura 11. Fragmento del código con líneas modificadas

## Código completo modificado

```
1  %%%% PRACTICA 3 LABORATORIO BIOMECANICA %%%
2  function topp3(nelx,nely,volfrac,penal,rmin);
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  loop = 0;
6  % Declaracion de vacio %
7  for ely = 1:nely
8  for elx = 1:nelx
9      if ((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) & (ely<(1+nely*0.5))) & (elx >(1+nelx)*0.6666)
10         passive(ely,elx)=1;
11     else
12         passive(ely,elx)=0;
13     end
14 end
15 end
16 x(find(passive))=0.001;
17 change = 1.;
18 % START ITERATION
19 while change > 0.01
20     loop = loop + 1;
21     xold = x;
22     % FE-ANALYSIS
23     [U]=FE(nelx,nely,x,penal);
24     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
25     [KE] = lk;
26     c = 0.;
27     for ely = 1:nely
28     for elx = 1:nelx
29         n1 = (nely+1)*(elx-1)+ely;
30         n2 = (nely+1)* elx +ely; %19
31         dc(ely,elx)=0.;
32         for i = 1:5
33             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],1);
34             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
35             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
36         end
37     end
38 end
39 % FILTERING OF SENSITIVITIES
40 [dc] = check(nelx,nely,rmin,x,dc);
41 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
42 [x] = OC(nelx,nely,x,volfrac,dc,passive);
43 % PRINT RESULTS
44 change = max(max(abs(x-xold)));
45 disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
46 ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
47 ' ch.: ' sprintf('%6.3f',change )])
48 % PLOT DENSITIES
49 colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
50 end
```

```

52      %%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%%
53      function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
54      -   l1 = 0; l2 = 100000; move = 0.2;
55      -   while (l2-l1 > 1e-4)
56      -       lmid = 0.5*(l2+l1);
57      -       xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
58      -       xnew(find(passive)) = 0.001;
59      -       if sum(sum(xnew)) - volfrac*nelx*nely > 0;
60      -           l1 = lmid;
61      -       else
62      -           l2 = lmid;
63      -       end
64      -   end
65      %%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%%
66      function [dcn]=check(nelx,nely,rmin,x,dc)
67      -   dcn=zeros(nely,nelx);
68      -   for i = 1:nelx
69      -       for j = 1:nely
70      -           sum=0.0;
71      -           for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
72      -               for l = max(j-round(rmin),1):min(j+round(rmin),nely)
73      -                   fac = rmin-sqrt((i-k)^2+(j-l)^2);
74      -                   sum = sum+max(0,fac);
75      -                   dcn(j,i) = dcn(j,i) + max(0,fac)*x(1,k)*dc(1,k);
76      -               end
77      -           end
78      -           dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
79      -       end
80      -   end
81      %%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
82      function [U]=FE(nelx,nely,x,penal)
83      -   [KE] = 1k;
84      -   K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
85      -   F = sparse(2*(nely+1)*(nelx+1),5);
86      -   U = zeros(2*(nely+1)*(nelx+1),5);
87      -   for ely = 1:nely
88      -       for elx = 1:nelx
89      -           n1 = (nely+1)*(elx-1)+ely;
90      -           n2 = (nely+1)* elx +ely;
91      -           edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
92      -           K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93      -       end
94      -   end
95      % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
96      -   F(2*(nelx)*(nely+1)+2,1) = 1;
97      -   F(2*(nelx)*(nely+1)+(nely/4),2) = 1;
98      -   F(2*(nelx)*(nely+1)+(nely/2),3) = 1;
99      -   F(2*(nelx)*(nely+1)+(nely),4) = 1;
100      -   F(2*(nelx)*(nely+1)+(nely*1.2),5) = 1;

```

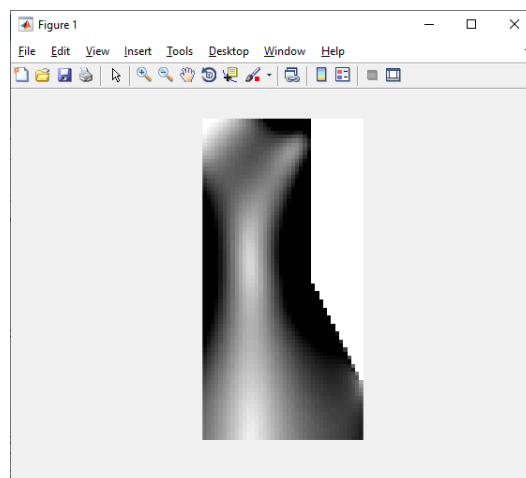
```

102 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
103 - alldofs = [1:2*(nely+1)*(nelx+1)];
104 - freedofs = setdiff(alldofs,fixeddofs);
105 - % SOLVING 127
106 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
107 - U(fixeddofs,:)= 0;
108 - %%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%
109 - function [KE]=lk
110 - E = 1.;
111 - nu = 0.3;
112 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
113 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
114 - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
115 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
116 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
117 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
118 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
119 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
120 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
121 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

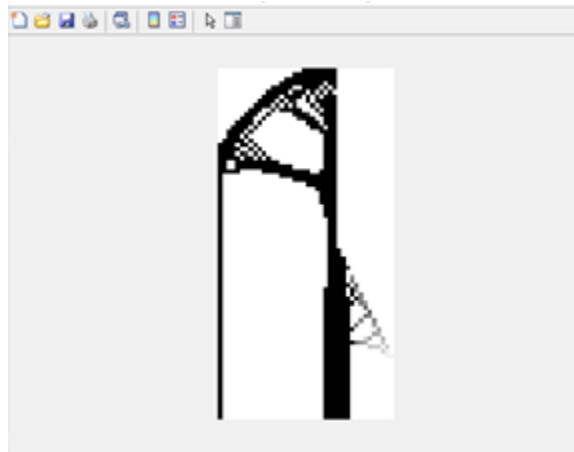
```

## 5. Resultados de la optimización.

A continuación, se muestra el resultado



Esta fue la primera estructura obtenida de la optimización, siendo esta la estructura propuesta.



Posteriormente realizamos una segunda optimización, obteniendo una estructura donde se puede observar cómo se ve reducido el material, pero contiene las mismas propiedades.

## **6. Conclusiones por cada autor**

### **Fátima Montserrat Zarazúa Uribe 1811014**

Es importante realizar una investigación acerca del lugar, el tránsito de personas que verán el panorama y cuales son las normativas por parte de la localidad para colocar la estructura, antes de crear la estructura de nuestro panorámico. En esta práctica nuevamente implementamos el uso del software de Matlab, en donde se creó una codificación mediante una propuesta enfocada al análisis efectuado referente a la geometría de la estructura de nuestro panorámico, dicho programa es una propuesta que permite observar el comportamiento del espectacular considerando las ráfagas de viento.

### **Jair Alejandro Tamayo Ibarra 1815498**

Esta práctica nos ayudó a observar los pequeños detalles que conllevan una estructura que vemos día con día al trasladarnos hacia un lugar, en este caso hablamos sobre un anuncio panorámico, se necesitan muchísimos cálculos para poder ponerlo en una posición adecuada, tanto para su propia resistencia, y también para poder ser visto por mucha gente; en esta práctica también utilizamos el programa de MatLab nuevamente, donde Realizamos una codificación para poder hacer un análisis de la geometría del cartel panorámico.

### **Karen Alexa Pérez Ortiz 1904708**

La elaboración de esta práctica me ha hecho reflexionar y analizar el trabajo que conlleva hacer un anuncio espectacular o panorámicos para que esté no solamente cumpla con el propósito de publicitar algo, sino que también sea duradero y de calidad, ya que si no se siguen las normativas y parámetros correctos para la elaboración de estas estructuras podrían causar graves accidentes, ya sea a personas o a edificaciones cerca del lugar, por ello es necesario conocer su geometría así como también observar cómo distintos fenómenos atmosféricos y físicos que ocurren día a día. De igual forma, se elaboró la programación en Matlab de la propuesta geométrica del panorámico, con el propósito de poder observar cómo se comportaba el anuncio cuando se encuentre con ráfagas de viento.

**Jorge Luis Ávila Hernández 1905338**

En esta práctica de laboratorio realizamos nuevamente un sistema de Matlab, analizando la geometría de un modelo de forma analítica para ver el comportamiento de este para poder producirlo y así ser capaces de efectuarlo en el programa, así como también el lograr a la resolución de éste.

**Ana Cristina Lucio Iracheta 1907905**

En esta práctica 3 realizamos un código en Matlab que modificamos para poder analizar la estructura propuesta y su geometría inicial, posteriormente analizamos la de un nuevo modelo, y de esta manera poder observar el cambio en la estructura y de ser el caso en sus propiedades; así mismo, realizamos una investigación acerca de las estructuras panorámicas.

**Ana Belén Bolaños Carbajal 1908896**

Se investigó acerca del diseño de estructura de los anuncios panorámicos que se establecen en diferentes puntos estratégicos para llamar la atención del público. Se utilizó MatLab para programar el modelo a simular y de esta forma conocer su comportamiento, asimismo analizamos los aspectos necesarios a tomar en cuenta para llevar a cabo este modelo, entre los que se encontraba: la altura del panorámico, materiales a utilizar y también considerando la velocidad a la que van las ráfagas de viento.



## **Referencias Bibliográficas**

ANÁLISIS PARAMÉTRICO DE ANUNCIOS ESPECTACULARES SUJETOS A LA ACCIÓN DEL VIENTO. (2018, diciembre). *Revista Pakbal*, 43. [https://ingenieria.unach.mx/images/Articulos\\_revista/revistapakbal\\_43\\_pag69-77.pdf](https://ingenieria.unach.mx/images/Articulos_revista/revistapakbal_43_pag69-77.pdf)