

UNIVERSIDAD AUTÓNOMA DE  
NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Laboratorio de Biomecánica  
Práctica 3: Diseño de la estructura de un panorámico

INSTRUCTOR@: YADIRA MORENO VERA  
MARTES V1 BRIGADA 204

**Equipo #2:**

Matricula	Nombre	Carrera
1991908	Covarrubias Becerril Brian Eduardo	IMTC
1991966	Luis Javier Rodríguez Vizcarra	IMTC
1925324	Rubén Cantú Espinoza	IMTC
1926098	Ángel Fernando Mexquitic Rodríguez	IMTC
1895460	Elmer Javier Delgadillo García	IMTC
1847932	Francisco Javier Velazco Rivas	IMTC

## Objetivo

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s).

## Nombre y definición de la forma geometría

Un espectacular es un anuncio publicitario de gran formato, normalmente se utiliza para la publicidad gráfica de cualquier ámbito, son de diversos tamaños y se colocan en puntos estratégicos de una ciudad para su mayor visibilidad.

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

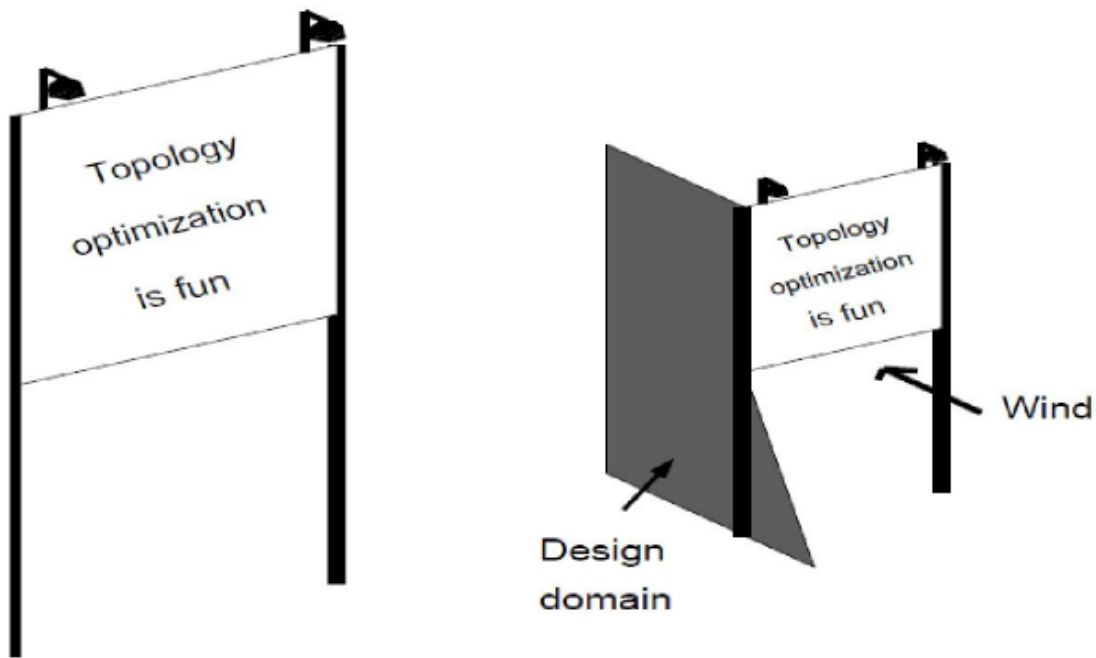


Figura 3.1 Imagen del Panorámico

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.

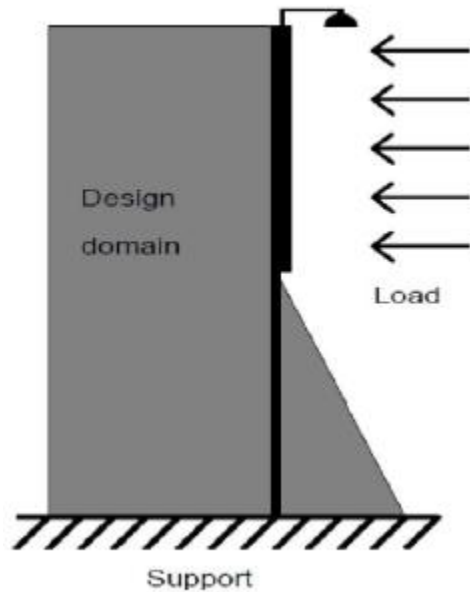


Figura 3.2 Espacio de diseño

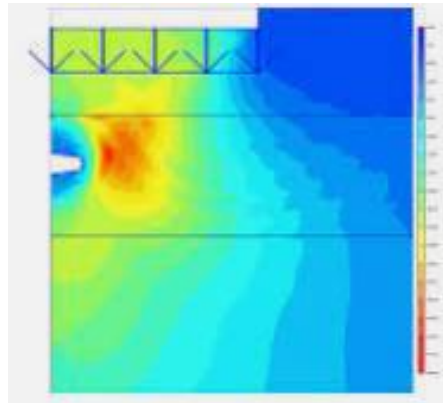
### Estado del arte

Como ya lo mencionamos anteriormente, la optimización topológica es una técnica englobada dentro del campo de análisis estructural, basada en el análisis mecánico de un componente, sistema o estructura. Tiene como objetivo el aligeramiento estructural manteniendo las características y propiedades mecánicas del componente principal. A diferencia de las demás optimizaciones, esta ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial, como la industria aeroespacial.

La optimización topológica, en resumen, consiste en utilizar un software concreto para “eliminar” el material que no posee los soportes. Algunos programas utilizados para realizar esta tarea son Within Labs, Inspire, Tosca, Ansys, Netfabb, etc.

El método de elementos finitos consiste en proponer que un número infinito de variables desconocidas, sean sustituidas por un número limitado de elementos de comportamiento bien definido. Esas divisiones pueden tener diferentes formas, tales como triangular, cuadrangular, entre otros, dependiendo del tipo y tamaño del problema. Como el número de elementos es limitado, son llamados de “elementos finitos” – palabra que da nombre al método. Los elementos finitos están conectados entre sí por puntos, que se llaman nodos o puntos nodales. Al conjunto de todos estos ítems – elementos y nodos – se lo denomina malla. Debido a las subdivisiones de la geometría, las ecuaciones matemáticas que rigen el comportamiento físico no se resolverán de una manera exacta, sino aproximada por este método numérico. La precisión de los Métodos dos Elementos Finitos depende de la cantidad de nodos

y elementos, del tamaño y de los tipos de elementos de la malla. Por lo tanto, cuanto menor sea el tamaño y mayor el número de elementos en una malla, más precisos serán los resultados del análisis.



### **Propuesta de diseño de la geometría, alcances y limitaciones**

Para el diseño del panorámico que solicita la práctica tomaremos ciertas consideraciones y un diseño propuesto por el instructor.



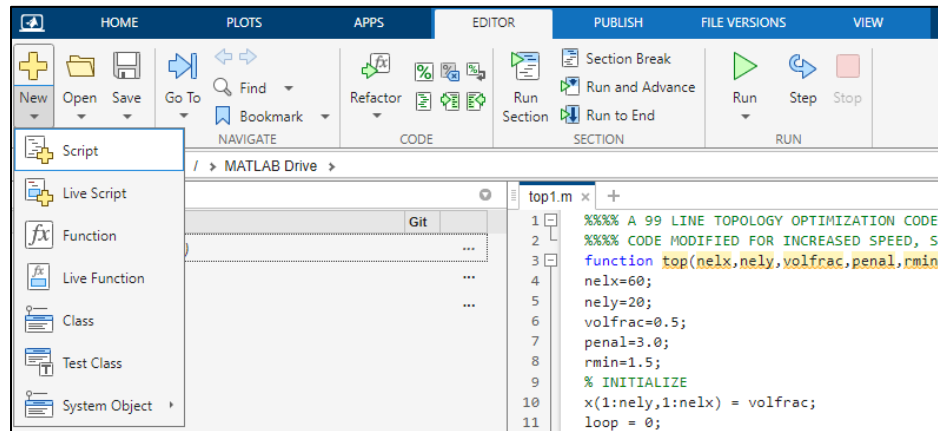
Tendremos 5 cargas y los apoyos tendrán ciertas restricciones en los ejes “x” y “y”.

Mediante la implementación de un código de optimización adecuado para la resolución de esta pieza, podremos observar el resultado de la geometría final y obtener un diseño con las mismas características y propiedades del diseño original, pero con un peso menor. Este método no cuenta con limitaciones, se pueden adecuar sinfín de diseños y obtener optimizaciones deseadas de acuerdo a las necesidades que tengamos. La optimización topológica es una técnica englobada dentro del campo de análisis estructural, basada en el análisis mecánico de un componente, sistema o estructura.

## Pasos del desarrollo de la programación

Para simular en Matlab el código propuesto de optimización topológica, debemos llevar a cabo los siguientes pasos:

- 1) Abrir Matlab, ya sea la aplicación de escritorio o directamente desde la página oficial usando su repositorio online.
- 2) Crear un nuevo script, para ello seleccionamos la opción en la barra de herramientas.



- 3) Una vez hecho esto, solo tenemos que copiar el código proporcionado que se localiza en la página indicada por el instructor.
- 4) Ya con el código copiado (99 Line Topology Optimization Code), tenemos que guardar primero el documento, en este caso como se utilizó Matlab online, se guardará en la nube de la cuenta que esta iniciada.
- 5) Antes de ejecutar la simulación, es necesario modificar ciertas líneas del código, las cuales se mostrarán a continuación:

1. Código original:

```
66 %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
67 function [U]=FE(nelx,nely,x,penal)
68 [KE] = lk;
69 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
70 F = sparse(2*(nely+1)*(nelx+1),1);
71 U = zeros(2*(nely+1)*(nelx+1),1);
```

Código modificado:

```
66 %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
67 function [U]=FE(nelx,nely,x,penal)
68 [KE] = lk;
69 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
70 F = sparse(2*(nely+1)*(nelx+1),5);
71 U = sparse(2*(nely+1)*(nelx+1),5);
```

## 2. Código original:

```
17 for ely = 1:nely
18     for elx = 1:nelx
19         n1 = (nely+1)*(elx-1)+ely;
20         n2 = (nely+1)* elx +ely;
21         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);
22         c = c + x(ely,elx)^penal*Ue'*KE*Ue;
23         dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
24     end
25 end
```

## Código modificado:

```
17 for ely = 1:nely
18     for elx = 1:nelx
19         n1 = (nely+1)*(elx-1)+ely;
20         n2 = (nely+1)* elx +ely;
21         for i=1:5
22             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],i);
23             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
24             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
25         end
26     end
27 end
```

## 3. Código original

```
82 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
83 F(2,1) = -1;
84 fixeddofs = union([1:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
85 alldofs = [1:2*(nely+1)*(nelx+1)];
86 freedofs = setdiff(alldofs,fixeddofs);
```

## Código modificado

```
82 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
83 F(2*(nelx)*(nely+1)+2,1)=1;
84 F(2*(nelx)*(nely+1)+20,1)=1;
85 F(2*(nelx)*(nely+1)+40,1)=1;
86 F(2*(nelx)*(nely+1)+60,1)=1;
87 F(2*(nelx)*(nely+1)+80,1)=1;
88 fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1);2*(nely+1);
89 alldofs = [1:2*(nely+1)*(nelx+1)];
90 freedofs = setdiff(alldofs,fixeddofs);
```

- 6) Para crear el empotramiento también es necesario modificar el código, por lo que quedaría de la siguiente manera:

Código original:

```
1 %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
2 %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3 function top1(nelx,nely,volfrac,penal,rmin);
4 % INITIALIZE
5 x(1:nely,1:nelx) = volfrac;
6 loop = 0;
7 change = 1.;
8 % START ITERATION
```

Código modificado:

```
1 %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
2 %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3 function top1(nelx,nely,volfrac,penal,rmin);
4 % INITIALIZE
5 x(1:nely,1:nelx) = volfrac;
6 for ely = 1:nely
7     for elx = 41:nelx
8         if elx - 20 < (ely/2)
9             passive(ely,elx)=0;
10        else
11            passive(ely,elx)=1;
12        end
13    end
14 end
15 x(find(passive))=0.001;
16 loop = 0;
17 change = 1.;
18 % START ITERATION
```

Código original:

```
40 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
41 [x] = OC(nelx,nely,x,volfrac,dc);
50 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
51 function [xnew]=OC(nelx,nely,x,volfrac,dc)
```

Código modificado:

```
40 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
41 [x] = OC(nelx,nely,x,volfrac,dc,passive);
50 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
51 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
```

Código original:

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

## Código modificado

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

## Código completo después de las modificaciones

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, JANUARY 2000
```

```
%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND
```

```
function topp3(nex,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nex) = volfrac;
loop = 0;
%Declarando vacio
for ely = 1:nely
for elx = 1:nex
if ((ely-(nely*0.5)<(2*elx)-(1.36*nex)) | (ely <(1+nely*0.5))) & (elx
>(1+nex)*0.6666))
passive(ely,elx) = 1;
else
passive(ely,elx) = 0;
end
end
end
x(find(passive))=0.001;
change = 1.;
% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
% FE-ANALYSIS
[U]=FE(nex,nely,x,penal);
%13 OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[KE] = lk;
c = 0.;
for ely = 1:nely
for elx = 1:nex
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely; %19
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) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
end
end
end
%25 FILTERING OF SENSITIVITIES
[dc] = check(nex,nely,rmin,x,dc);
%27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
```



```

[x] = OC(nelx,nely,x,volfrac,dc,passive);
%29 PRINT RESULTS
change = max(max(abs(x-xold)));
disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change )])
% PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-
6);
end
%40 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)

l1 = 0; l2 = 100000; move = 0.2;
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;
l1 = lmid;
else
l2 = lmid;
end
end
%%%%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%%%%%
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
for j = 1:nely
sum=0.0;
for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
for l = max(j-round(rmin),1):min(j+round(rmin), nely)
fac = rmin-sqrt((i-k)^2+(j-l)^2);
sum = sum+max(0,fac);
dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
end
end
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
end
%65 %%%%%%%%%% 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 =zeros(2*(nely+1)*(nelx+1),5);
for ely = 1:nely
for elx = 1:nelx
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely;
edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
end
end
% 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;

```

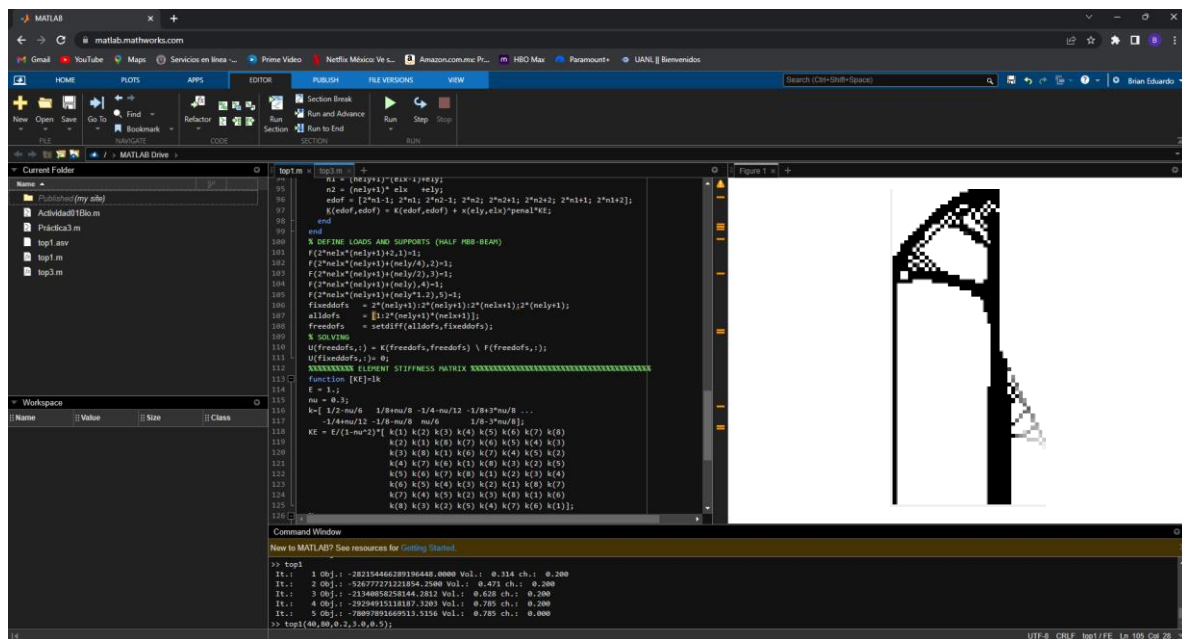
```

F(2*nex*(nely+1)+(nely),4) = 1;
F(2*nex*(nely+1)+(nely*1.2),5) = 1;
fixeddofs = 2*(nely+1):2*(nely+1):2*(nex+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nex+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING 127
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ELEMENT STIFFNESS MATRIX %%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;

k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

## Resultados de la optimización



## Diagrama de flujo o mapa conceptual Sway

## Conclusiones

- Covarrubias Becerril Brian Eduardo: esta práctica fue parecida a las realizadas anteriormente, consistió en modificar, simular y observar la optimización topológica del código que ya obtuvimos de 99 líneas en el software de Matlab. En este caso se realizó un panorámico, con soporte fijo y sometido a 5 fuerzas. Para cumplir con estas condiciones, primero se realizó una propuesta, después se modificó el código y al final se simuló para ver el diseño optimizado y darnos cuenta de que cumple con las mismas características deseadas pero con un diseño más estético y quitando todo material sobrante.
- Francisco Javier Velazco Rivas. en esta práctica se analizo como el espacio de un anuncio , buscando la manera que la geometría de este mismo sea simetría además de buscar ser más óptimo en todos sus aspectos viéndolo de la manera de programar ,además de tomar como base un código ya dado dándole un cambio general para hacerlo mejor.  
Tomando en cuenta que se debe conservar la idea pero dándole una nueva idea o modelo , es por eso que cada quien dio un punto de vista y creo que optamos por la mejor opción.
- Elmer Javier Delgadillo García: en base a lo sucedido y visto en la presente práctica, se puede destacar que si bien efectuar una programación exitosa es la tarea fundamental para conllevar la situación requerida, en este caso: el diseño de la estructura de un panorámico, la organización del procedimiento y diseño posee una mayor comprensión mediante la visualización gráfica, como lo fue un diagrama de flujo. De esta manera, es más fácil relacionarse con cualquier programa a realizar.
- Luis Javier Rodríguez Vizcarra: En conclusión, esta práctica es similar a las realizadas anteriormente, esta siendo más compleja ya que se estudiaron diferentes casos, también en esta práctica se identificaron los elementos pasivos tomados para el diagrama y por último comentar la importancia que tienen los software para ayudarnos mediante sus diferentes herramientas a resolver diferentes tipos de situaciones como la de este caso la optimización de los diferentes casos planteado.
- Ángel Fernando Mexquitic Rodríguez: para concluir esta práctica considero que es bastante importante llevar una buena programación en el caso del diseño de estructura de un panorámico. Es muy importante este para poder llevar una buena resolución de los problemas planteados y así optimizar de mejor manera los procesos. Viendo a su vez cosas importantes como lo son los diagramas de flujo y de esta manera tener un apoyo visual mayor.

- Rubén Cantú Espinoza: para la elaboración de esta tercer práctica se presentó y se aplicó una propuesta de análisis de formas, con el fin de optimizar el diseño, para ello se tomó en cuenta lo que se vio en las prácticas anteriores donde se definió la forma geométrica de la pieza, el cual vendría siendo un panorámico, y posteriormente se declararon las variables a tomar en cuenta, como lo es el espacio del diseño y las fuerzas aplicadas. Se continuó con el estado del arte donde se analizó la pieza para la optimización de acuerdo a su estructura y se continuó con la implementación del código, por último solo se observaron los resultados.

## Referencias

- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- IMME. (2007). *Una metodología para la optimización estructural de formas usando principios de evolución flexible distribuida*. Scientific Electronic Library Online. [http://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S0376-723X2007000100002](http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0376-723X2007000100002)