

Practica No.3

Equipo 6

Merary Castillo Sanchez 1895677
Juan Javier missael Castillo Ruiz 1884560
Heber Adrián Casillas Gutiérrez 1894878
Luis Mateo Landa Rivera 1909998
Bruno Mendoza Palomo 1992283
Juan Erasmo Guerrero Treviño 1903220

18 de octubre de 2022

Resumen

1. Objetivo

Presentar una propuesta análisis de formas y de la programación para la ejecución de la optimización de características de trabajo específicas

2. Introducción

En esta practica veremos lo que es un diseño de la estructura de un panorámico (anuncios espectaculares) las cuales son susceptible a los efectos de las acciones de empuje de viento debido a su configuración geométrica, tomando como punto principal de estudio el pedestal de soporte del mismo. Este tipo de estructura aparentemente es muy sencilla porque cuenta solamente con tres partes principales que son: la mampara, el pedestal y la cimentación. 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, 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. Debido a que seria cosas de meses calcular tanto factores usaremos un diseño minimalista en la cual se expondrá una valla publicitaria, en donde se tomaran solo las estructuras principales como la valla y el marco de la estructura del panorámico donde haremos los cálculos necesarios para probar sus esfuerzos.

3. Estado del arte

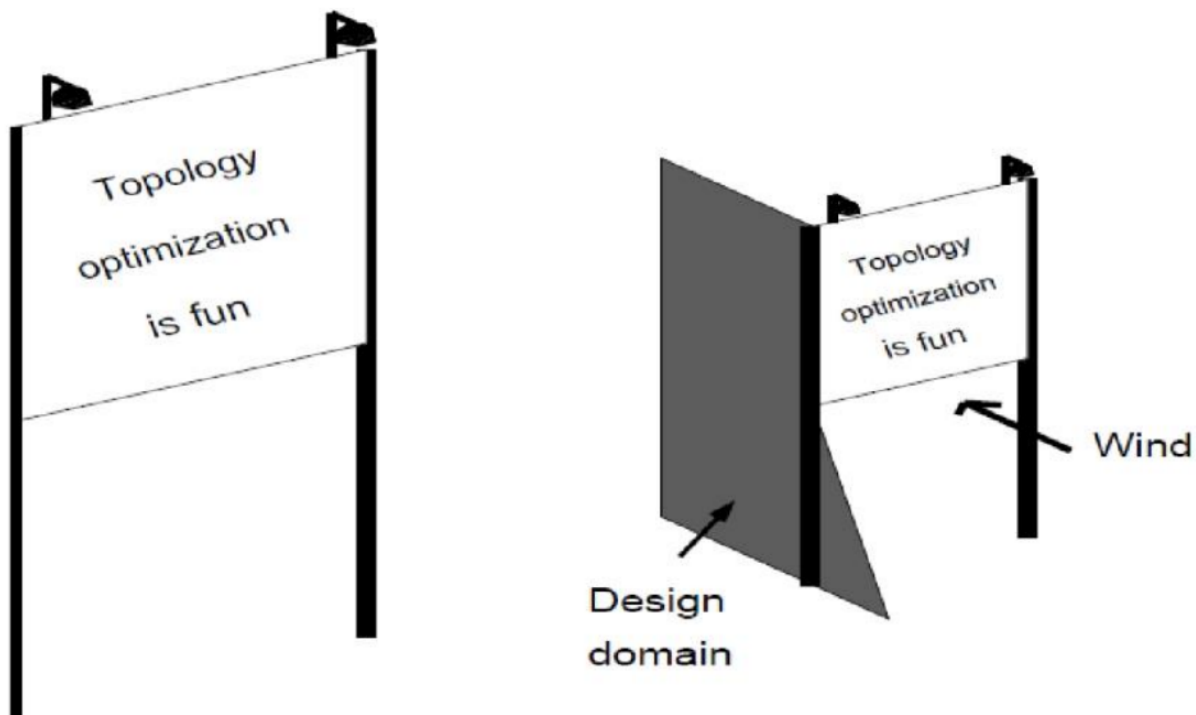
En esta práctica se busca realizar un diseño de la estructura de un panorámico definiendo estructuras funcionales y seguras utilizando la optimización topológica para hacer este diseño. Los panorámicos se enfrentan a distintas cargas como las cargas muertas, vivas, de viento o sismo. Estas estructuras suelen ser muy variadas tanto en ubicación, forma, soporte, geometría o dimensiones. Una característica esencial de los panorámicos es la forma en que se conectan los elementos para darle estabilidad al sistema, lo que a su vez define la forma en que debe fabricarse e instalarse la estructura. Normalmente la ubicación de estos es en lugares de mucho tránsito de personas, donde puedan llegar a un sector de población más alto lo que los lleva a verse a lo largo de las vías terrestres más transitadas y con más densidad en las áreas urbanas. Por lo tanto, a pesar de que estas estructuras no son tan importantes por sí mismas, debe de garantizarse su seguridad porque

su colapso podría afectar a muchas personas. Por dicha razón, los rótulos deben ser diseñados de una forma conservadora para proteger las vidas de las personas y también evitar pérdidas económicas directas, como costo de la estructura misma y daños a otros inmuebles e indirectas, como los tiempos muertos de tránsito vehicular o paralizar momentáneamente áreas comerciales. Mediante la optimización topológica se busca la reducción del volumen de la estructura del panorámico y que soporte las cargas a las que este está sometido como sería la carga por viento mediante movimientos horizontales de masas de aire provocados por las diferencias de presión en las distintas zonas de la atmósfera y a la rotación terrestre. Estos movimientos ocurren constantemente; sin embargo, para el diseño estructural interesan esencialmente los vientos que tienen velocidades muy grandes.

4. Desarrollo

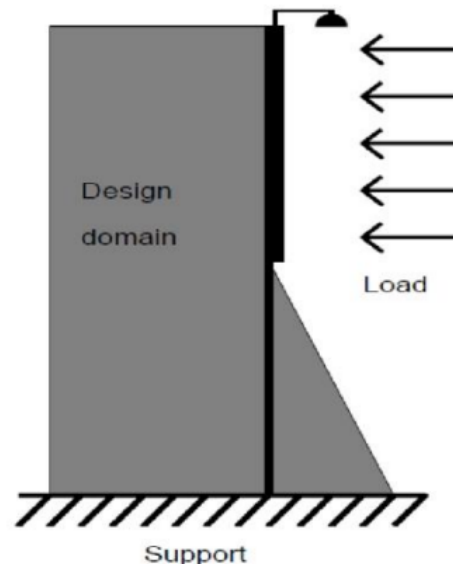
¿Cómo interpreto la tabla de geometría de un cuadro de bicicleta?

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 abajo.

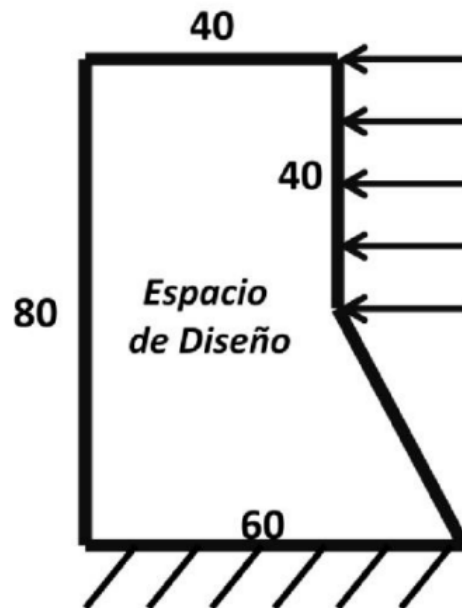


En la figura se puede ver el espacio de diseño para esta práctica. Se espera una fracción volumétrica aproximada

de 0.20 por ciento del espacio de diseño. Supongamos que el panorámico es muy rígido, y sus patas son del mismo material que el marco.



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:



5. Pasos de la programación

Procedimiento de la programación

Fuerzas múltiples.

Ingresamos las 5 fuerzas y cambiar el anclaje del espacio de diseño a otra posición se tiene que cambiar la línea con la instrucción FE-ANALYSI

Esta es la instrucción original:

```
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),1); U = zeros(2*(nely+1)*(nelx+1),1);
```

Se modifica y queda así

```
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;
        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
```

Esta parte del código corresponde a El cual debe ser reemplazado por

```
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],i)
    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
```

Las siguientes líneas son reemplazadas:

```
F(2,1) = -1;
fixeddofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
```

Por los siguiente

```
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 = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
```

Empotramiento diagonal (elementos pasivos)

Las siguientes líneas se modificaron y/o agregaron para crear la incrustación diagonal, o crear el espacio en blanco para recrear la incrustación en la parte inferior derecha, en la sección del archivo de uso de código de 99 líneas que analiza los elementos pasivos. Hacer un círculo es el ejemplo proporcionado aquí, pero también necesitamos un rectángulo y un triángulo para hacer esto.

```
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
    loop = loop + 1;
    xold = x;
```

Líneas modificadas

```
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
%Declarando vacio
for ely = 1:nely
    for elx = 1:nelx
        if (((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) & (ely <(1+nely*0.5))) & (elx
>(1+nelx)*0.6666))
            passive(ely,elx) = 1;
        else
            passive(ely,elx) = 0;
        end
    end
end
x(find(passive))=0.001;
change = 1.;
% START ITERATION
```

Además, originalmente la siguiente parte del código estaba así:

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

```
function [xnew]=OC(nelx,nely,x,volfrac,dc)

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;
```

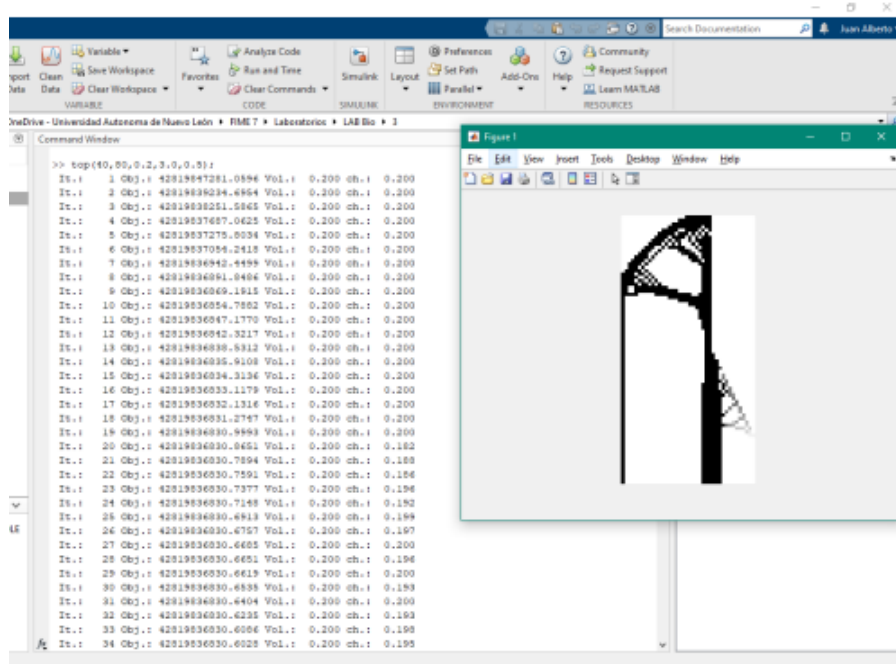
Estas tres partes se modifican y quedan de la siguiente manera:

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)

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;
```

6. Resultado de la optimización



Codigo completo

```

##### A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 ###
##### CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND ###
function top(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
%Declarando vacio
for ely = 1:nely
    for elx = 1:nelx
        if ((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) |(ely <(1+nely*0.5))) & (elx>(1+nelx)*0.5)
6666)
            passive(ely,elx) = 1;
        else
            passive(ely,elx) = 0;
        end
    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(nelx,nely,x,penal);
    % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
    [KE] = lk;
    c = 0.;
    for ely = 1:nely
        for elx = 1:nelx
            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
    end

% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);
% 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
##### 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(l.,min(x+move,x.*sqrt(-dc./lmid)))));
    xnew(find(passive)) = 0.001;
    if sum(sum(xnew)) - volfrac*nex*nely > 0;
        l1 = lmid;
    else
        l2 = lmid;
    end
end
end
##### MESH-INDEPENDENCY FILTER #####
function [dcn]=check(nex,nely,rmin,x,dc)
dcn=zeros(nely,nex);
for i = 1:nex
    for j = 1:nely
        sum=0.0;
        for k = max(i-floor(rmin),1):min(i+floor(rmin),nex)
            for l = max(j-floor(rmin),1):min(j+floor(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
end

##### FE-ANALYSIS #####
function [U]=FE(nex,nely,x,penal)
[KE] = lk;
K = sparse(2*(nex+1)*(nely+1), 2*(nex+1)*(nely+1));
F = sparse(2*(nex+1)*(nely+1),5);
U = zeros(2*(nex+1)*(nely+1),5);
for elx = 1:nex
    for ely = 1:nely
        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
end
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2*nex*(nely+1)+2,1) = 1;
F(2*nex*(nely+1)+(nely/4),2) = 1;
F(2*nex*(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
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)];

```

7. Conclusión

Heber Adrián Casillas Gutiérrez 1894878:

En esta practica tuve la oportunidad de ver los esquemas paramétricos de una valla publicitaria aprendiendo de las partes de las cuales están conformadas para poder tener una mejor idea de su funcionamiento logrando así tener una idea del diseño de la forma geométrica que debe de tener una valla publicitaria tomando en cuenta a los efectos del viento que afectan al objeto pudiendo así determinar el como reacciona la valla para poder pensar en mejores soluciones para mejorar su aguante ante diversas situaciones.

Luis Mateo Landa Rivera 1909998:

Para esta tercera practica, pudimos analizar el diseño de una valla publicitaria utilizando nuevamente el código de las 99 líneas de optimización topológica de Matlab, realizando diferentes suposiciones se realizo el diseño en dos dimensiones, dichas suposiciones fueron que la valla sea muy rígida, que se aplicarían 5 cargas de manera transversal al diseño el cual se considera que esta empotrado al suelo, con las medidas apropiadas, se modificó el código en Matlab para saber cómo quedaría su diseño ya optimizado visto desde un lateral tomando en cuenta solo coordenadas de "x" y "y".

Juan Erasmo Guerrero Treviño 1903220:

En esta práctica se analizo el diseño de un panoramico, el cual sabemos que se enfrenta a diversas cargas posibles que pueden variar en cuanto a direccion, magnitud y fuerza. Tal como son el viento, sismos, algún animal que se estrelle, etc. Se tomaron en cuenta algunas restricciones en los ejes "xz "z", para despues emplear el código en matlab y así crear una optimización de la pieza.

Juan Javier Missael Castillo Ruiz 1884560:

El problema es el diseño de una valla publicitaria que está expuesta a un campo de viento. El soporte de la valla publicitaria necesita algún tipo de marco de soporte para hacerlo más rígido. En la figura del espacio de trabajo se da una ilustración bidimensional de las cargas y el apoyo. Una fracción de volumen de aproximadamente 0,2 del dominio de diseño. Supusimos que la valla publicitaria es muy rígida 1 y las patas disponibles están hechas del mismo material que el marco. En este informe de laboratorio, demostramos el trabajo que se completó a través de la práctica con Matlab. Descubrimos que esto tomó más tiempo en completarse que los experimentos anteriores porque el software tuvo que pasar por un proceso para maximizar sus esfuerzos, y también notamos que los espacios en blanco, que son componentes pasivos, debían tenerse en cuenta para el diagrama. Podemos inferir de nuestro trabajo que, aunque comúnmente se asume que algo se ignora porque es un espacio en blanco dentro de un sistema de intentos, esto no debería ser así y debemos darle valor para el diseño ideal del diagrama

Bruno Mendoza Palomo 1992283:

En esta práctica se utilizó el diseño de una valla publicitaria, gracias a los esquemas paramétricos se puede observar como esta funciona, de que esta conformada, entre otras cosas, igual con esto podemos conseguir encontrar mejores soluciones de diseño para que sus propiedades sean mejores y por lo tanto esta sea más resistente.

Merary Castillo Sanchez 1895677:

El diseño para la realización de cualquier objeto es fundamental para la realización de su buen trabajo sin tener

error alguno , ya que en la programación utilizamos matlab que es una herramienta que me ha servido bastante tanto a mi como a los de mi equipo y nos ayuda a realizar un trabajo en equipo fundamental y coordinado

8. Bibliografias

- [1] J. Cruz S, J. F. Grajales, J. (2010). ANÁLISIS PARAMÉTRICO DE ANUNCIOS ESPECTACULARES SUJETOS A LA ACCIÓN DEL VIENTO. PAKBOL Recuperado 18 de octubre de 2022, de <https://ingenieria.unach.mx/images/Articulos/77.pdf>
- [2] This Matlab code was written by Ole Sigmund, Department of Solid Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark. Please send your comments to the author: sigmund@fam.dtu.dk
- [3] ".^ 99 line topology optimization code written in Matlab" by Ole Sigmund (2001), Structural and Multidisciplinary Optimization, Vol 21, pp. 120–127. The code as well as a postscript version of the paper can be