



**FIME**



**UANL**

Unicersidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

## PRÁCTICA #4

Docente: Yadira Moreno Vera

Brigada: 109 Hora: N5 Día: lunes

Materia: Laboratorio de Biomecánica

Integrantes del equipo:

Nombre	Matricula:	Carrera:
Aarón Lozano Aguilar	1844469	IMTC
Montserrat Granados Salinas	1817165	IMTC
Eunice Carolina Méndez Sosa	1851345	IMTC
Eimie Carolina Pereda Sánchez	1915035	IMTC
Aida Mata Moreno	1734743	IMTC

Semestre: Agosto-Diciembre 2022

## Objetivo:

Hacer el análisis del tipo de cable que se utilizan en los teleféricos considerando cargas para obtener puntos críticos y obtener una respuesta a través de su simulación.

## Marco Teórico

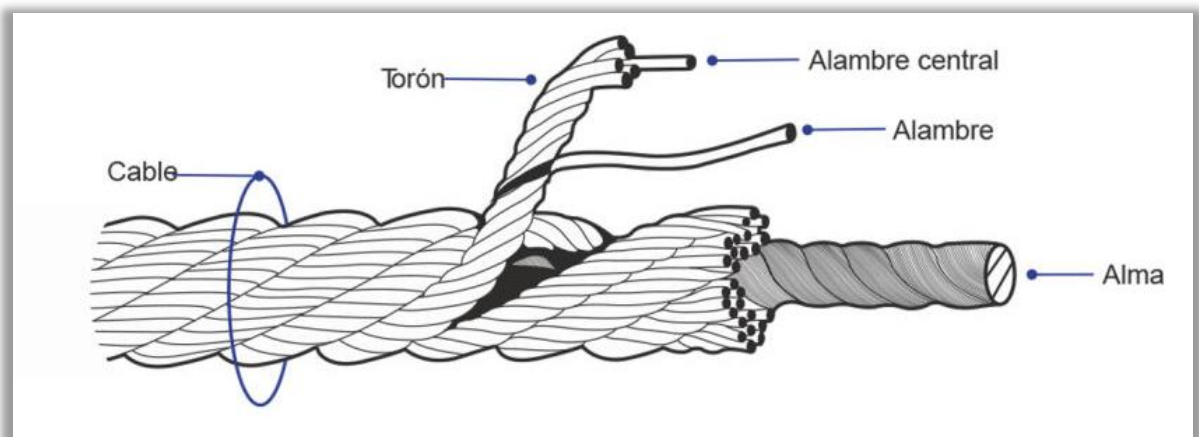
El teleférico es un sistema de transporte aéreo constituido por cabinas colgadas de una serie de cables que se encargan de hacer avanzar a las unidades a través de las estaciones. Este medio de transporte se ve expuesto no solo al peso de los pasajeros y al del mismo teleférico, sino también a las corrientes de aire que pueden llegar a alterar la estabilidad de este.

De una estación a la otra hay los siguientes cables:

- **El cable portante**, que sostiene el peso de la cabina y que debe ser no sólo potente, sino sin juntas, es decir, de una sola pieza.
- **El cable movable**, que está enganchado a la cabina y la arrastra.
- **El cable de freno**, que aguanta la cabina en caso de que el cable movable se rompa.
- **El cable de socorro**, que, en dicha circunstancia de ruptura, lleva la cabina hasta la estación

## Tipo de cable

Para selección correcta del cable portante posee gran importancia dentro de cualquier proyecto de teleférico, ya que éste es quien da movimiento y soporte a las cabinas. Para una correcta elección deben considerarse varios factores, los más importantes son, el peso de la cabina cargada, esfuerzo ejercido por el peso del propio cable, tensión del cable, fuerzas de frenado y aceleración. Los cables están formados por un núcleo de acero o fibra y varias capas de acero trenzado, entrelazados en forma helicoidal.



## Diseño

El teleférico de la figura 1 necesita un refuerzo en su apoyo. Sugiera un refuerzo según la información dada en la figura 1.1

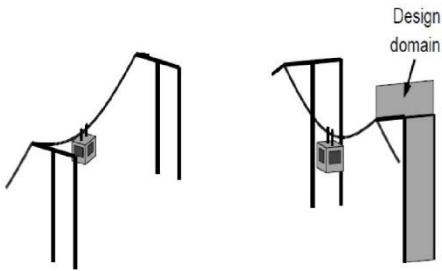


Figura 1

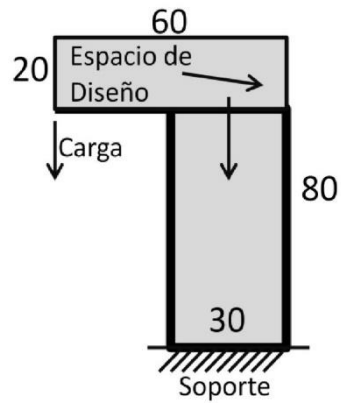


Figura 1.1

Al cuidador del teleférico también le gustaría que se hicieran mejoras para que la estructura pueda llevar dos teleféricos a la vez, como se ilustra en la figura 1.2. Este último caso implica considerar múltiples cargas.

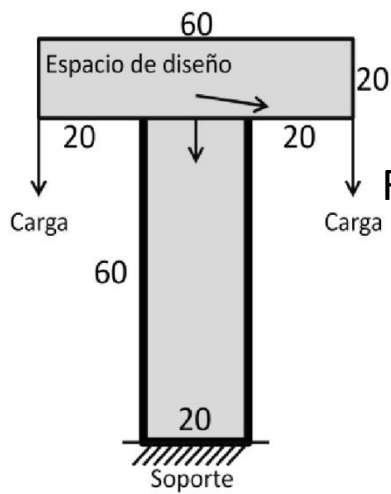


Figura 1.2

## Implementación Código

Llamamos a los elementos finitos en este pasivo vacío, y definimos una matriz con ceros en elementos libres y seres en pasiva, y para eso escribimos las siguientes líneas de código entre las líneas 5 y 6 del código original.

Para hacer esta práctica aplicamos el código de optimización en MATLAB, en donde le hacemos algunas modificaciones para hacer los cálculos correctos hacia nuestro cable de teleférico. Se hace la modificación del código, en donde se da la declaración del vacío en la figura:

## Código original

```
3 function top(nelx,nely,volfrac,penal,rmin);
4 % INITIALIZE
5 - x(1:nely,1:nelx) = volfrac;
6 - loop = 0;
7 - change = 1.;
```

## Código Nuevo

```
3 function new_pr42_f(nelx,nely,volfrac,penal,rmin);
4 % INITIALIZE
5 - x(1:nely,1:nelx) = volfrac;
6 - for ely = 1:nely
7 -     for elx = 1:nelx
8 -         if ely>21
9 -             if elx<31
10 -                 passive(ely,elx) = 1;
11 -             else
12 -                 passive(ely,elx) = 0;
13 -             end
14 -         end
15 -     end
16 - end
17 - x(find(passive))=0.001;
18 - loop = 0; change = 1.;
```

Ésta es una matriz pasiva  $nely \times nelx$  con ceros en los elementos libres de cambiar y unos en los elementos fijos en cero se puede definir en el programa principal y transferir a la subrutina OC).

Por lo tanto, modificamos las líneas de la 17 a la 23 del código original

## Código original

```

14 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
15 - [KE] = 1k;
16 - c = 0.;
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 Nuevo

```

27 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
28 - [KE] = 1k;
29 - c = 0.;
30 - for ely = 1:nely
31 -     for elx = 1:nelx
32 -         n1 = (nely+1)*(elx-1)+ely;
33 -         n2 = (nely+1)* elx +ely;
34 -         dc(ely,elx)=0.;
35 -         for i=1:2
36 -             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);
37 -             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
38 -             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
39 -         end
40 -     end
41 - end

```

## Declaración Fuerzas

### Punto 1

Cambiamos la línea 80 y 81 del código original por las siguientes líneas que modifican las fuerzas en nuestro teleférico. (Solo para el primer caso)

$$F(40,1) = -1.;$$

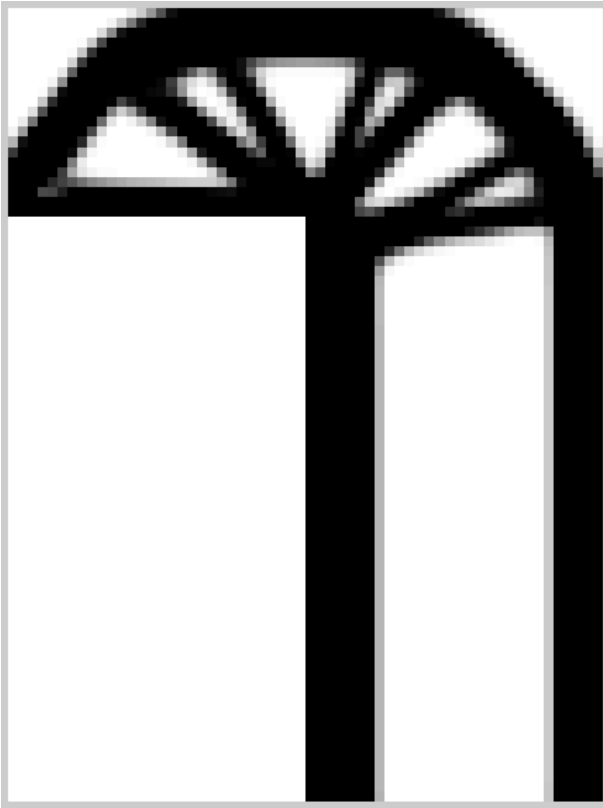
$$\text{fixeddofs} = 2*(\text{nely}+1):2*(\text{nely}+1):2*(\text{nelx}+1)*(\text{nely}+1)$$

```
79 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
80 - F(2,1) = -1;
81 - fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
82 - alldofs = [1:2*(nely+1)*(nelx+1)];
83 - freedofs = setdiff(alldofs, fixeddofs);
```

```
94 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
95 - F(40,1) = -1;
96 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
97 - alldofs = [1:2*(nely+1)*(nelx+1)];
98 - freedofs = setdiff(alldofs, fixeddofs);
```

## Resultado

- Imagen 1. Resultado obtenido usando el software MATLAB, podemos ver como se obtiene una simulación de la carga, y la figura del teleférico.



## Parte 1 Código

```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER
1999 %%%
function top(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<31
                passive(ely,elx) = 1;
            else
                passive(ely,elx) = 0;
            end
        end
    end
end
x(find(passive))=0.001;
loop = 0;change = 1.;
% START ITERATION
while change > 0.01C
    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;
            dc(ely,elx)=0.;
```



```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND,
OCTOBER 1999 %%%
function top(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<31
                passive(ely,elx) = 1;
            else
                passive(ely,elx) = 0;
            end
        end
    end
end
x(find(passive))=0.001;
loop = 0;change = 1.;
% START ITERATION
while change > 0.01C
    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;
            dc(ely,elx)=0.;

```

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND,
OCTOBER 1999 %%%
function top(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<31
                passive(ely,elx) = 1;
            else
                passive(ely,elx) = 0;
            end
        end
    end
end
x(find(passive))=0.001;
loop = 0;change = 1.;
% START ITERATION
while change > 0.01C
    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;
            dc(ely,elx)=0.;

```

## Punto 2

Cambiamos la línea 80 y 81 del código original por las siguientes líneas. (Solo para el segundo caso)

F(40,1) = -1.; F(9760,2)=1.;

fixeddofs = 2\*(nely+1):2\*(nely+1):2\*(nelx+1)\*(nely+1)

## Código original

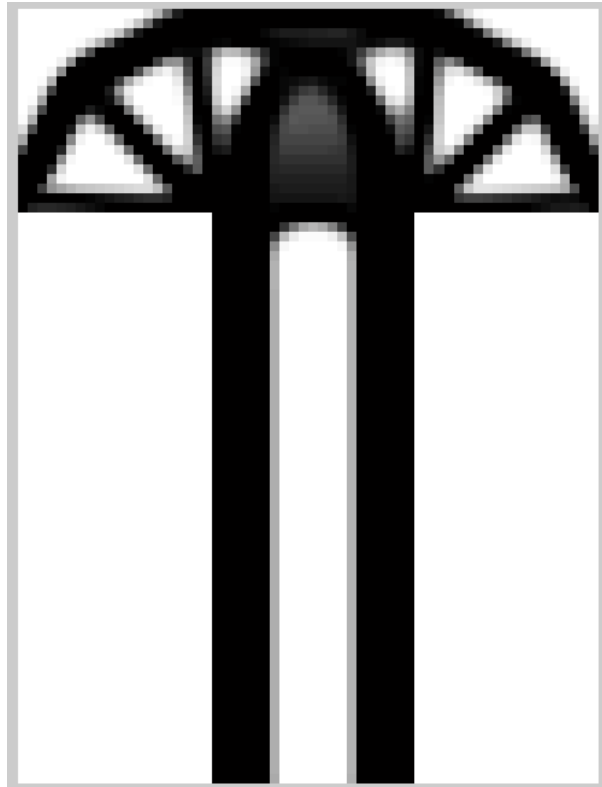
```
79 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
80 - F(2,1) = -1;
81 - fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
82 - alldofs = [1:2*(nely+1)*(nelx+1)];
83 - freedofs = setdiff(alldofs, fixeddofs);
```

## Código nuevo

```
96 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
97 - F(40,1) = -1.; F(9760,2)=1.;
98 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
99 - alldofs = [1:2*(nely+1)*(nelx+1)];
100 - freedofs = setdiff(alldofs, fixeddofs);
```

## Resultado

- Imagen 2. Resultado obtenido de la simulación utilizando el software MATLAB, con una simulación de 2 fuerzas.



## Parte 2 Código

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%
function top(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<21
                passive(ely,elx) = 1;
            elseif elx>41
                passive(ely,elx)=1;
            else
                passive(ely,elx) = 0;
            end
        end
    end
end
x(find(passive))=0.001;
loop = 0; 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;
            dc(ely,elx)=0.;
```

```

for i=1:2
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
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(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

```

```

%%%%%%%%%% 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),2); U =
sparse(2*(nely+1)*(nelx+1),2);
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(40,1) = -1.; F(9760,2)=1.;
fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nelx+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)];

```



## **Conclusiones**

**Aarón Lozano Aguilar 1844469 IMTC**

Está práctica fue demasiado útil e interesante para ver cómo se comporta y qué tipos de cables se usan en un teleférico dependiendo de sus cargas, así como también analizando sus puntos críticos y obteniendo la simulación de este, se puede observar que en los resultados de los casos propuestos se tiene una geométrica muy similar entre ellos. En el caso de dos cargas, este como las fuerzas son aplicadas en opuestos simétricos la forma de pieza es simétrica en el eje Y

**Montserrat Granados Salinas 1817165 IMTC**

A lo largo de esta práctica se tuvo una mejor comprensión de la optimización de diseño, al contrario de las demás practicas esta vez se optimizó un soporte para cable de un teleférico; se hicieron dos diseños optimizados para observar cómo afectan las diferentes cargas aplicadas en cada uno de ellos, me ayudó a comprender mejor cómo funciona el código del programa no solo en cosas como prótesis si no también en panorámicos o teleféricos y poder apreciar su diseño generativo de manera correcta dando los mejores resultados posibles.

**Eunice Carolina Méndez Sosa 1851345 IMTC**

Lo que aprendí en esta práctica es sobre como con MATLAB se pueden hacer distintos diseños para problemas de objetos que son prácticos en la vida cotidiana, a mi parecer a sido un programa muy eficaz que nos hace mucho más fácil el análisis, poco a poco se va entendiendo mejor y con esto se pudo encontrar el tipo de cable que se podría implementar en un teleférico.

**Eimie Carolina Pereda Sánchez 1915035 IMTC**

En esta práctica se trato acerca de un análisis sobre el tipo de cable que utilizan los teleféricos apoyándonos en su diseño, sus esfuerzos, fuerza y con el mismo código 99 que se ha estado usando a lo largo de las practicas encontrar esas incógnitas cambiando solo lo necesario al código para que vaya referido al nuevo problema que tenemos.

**Aida Mata Moreno 1734743 IMTC**

Concluyo con que un cable de un teleférico consta de mas cables que se usan para diferentes movimientos, además que se siguió un diseño inicial del teleférico y a partir de ahí nosotros desarrollamos el análisis en donde pudimos dar con el resultado querido.

## **Referencias Bibliográficas**

- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- Blu Cactus. (2021). Estructura de un panorámico. octubre 2021, de Blu Cactus Digital Marketing Sitio web: <https://www.blucactus.com.mx/cuanto-cuesta-estructura-panoramico/>

