



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO CIVIL

**Elaboración de la herramienta Cable-Structures para ser incorporadas al
VLEE de la UTPL**

TRABAJO DE TITULACIÓN

AUTORA: Peláez Peláez, Daniela del Cisne.

DIRECTOR: Duque Yaguache, Edwin Patricio, Msc.

CODIRECTOR: Quiñones, Santiago, Ing.

LOJA ECUADOR

2017



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

2017

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Msc.

Duque Yaguache, Edwin Patricio.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Elaboración de la herramienta Cable-Structures para ser incorporadas al VLEE de la UTPL, realizado por Daniela del Cisne Peláez Peláez, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, octubre de 2017

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Daniela del Cisne Peláez Peláez declaro ser autora del presente trabajo de titulación: Elaboración de la herramienta Cable-Structures para ser incorporadas al VLEE de la UTPL, de la Titulación Ingeniería Civil, siendo el Msc. Duque Yaguache Edwin Patricio director y el Ing. Santiago Quiñones codirector del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la universidad la propiedad intelectual de investigaciones, trabajos científicos o técnico y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la universidad”

f)

Autor: Daniela del Cisne Peláez Peláez

Cédula: 1105338071

DEDICATORIA

A mis padres, Nelson G. Peláez A. y María P. Peláez R. Mí fuente de inspiración.

A mis hermanos, Leonardo y Nelson. Gracias por cuidar de mí.

Los amó...

Daniela C. Peláez P.

AGRADECIMIENTO

Primeramente, agradezco a Dios por la vida, mi familia y por ayudarme a cumplir este objetivo.

A mis padres, Nelson G. Peláez A. y María P. Peláez R., a mis hermanos, en especial a Leonardo y Nelson, por sus sabios consejos, la confianza y apoyo incondicional, que hicieron posible la culminación de mis estudios.

A mi grupo de amigos que de una u otra forma colaboraron para llegar a feliz término de la presente meta.

A Jonathan V. por su comprensión y cariño.

Al Msc. Edwin P. Duque Y. que gracias a su valiosa dirección ha sido posible concluir con el presente trabajo.

Expreso mis reconocidos agradecimientos como signo de gratitud, a la Universidad Técnica Particular de Loja, y a los Ingenieros que conforman el departamento de Geología y Minas e Ingeniería Civil, que en su tiempo aportaron con sus conocimientos durante el transcurso de mi vida universitaria

Daniela C. Peláez P.

TABLA DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN	II
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	III
DEDICATORIA.....	IV
AGRADECIMIENTO	V
TABLA DE CONTENIDOS.....	VI
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XIV
RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN.....	3
OBJETIVOS	5
A. GENERAL.....	5
B. ESPECÍFICOS	5
CAPITULO I.....	6
1. MARCO TEÓRICO	6
1.1. LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA (VLEE)	7
1.1.1. <i>Arquitectura del VLEE</i>	7
1.1.1.1. Capa de presentación.....	8
1.1.1.2. Capa de negocios.	8
1.1.1.3. Capa de datos.....	8
1.1.2. <i>Interfaz gráfica del VLEE</i>	8
1.2. PROGRAMACIÓN.....	9
1.2.1. <i>Código fuente</i>	9
1.2.2. <i>Diagramas de flujo</i>	10
1.3. CABLE DE ACERO	11
1.3.1. <i>Las estructuras del cable de acero</i>	12
1.3.1.1. La estructura transversal de los cordones.....	12
1.3.1.2. Cordones de alambre de igual diámetro.	12
1.3.1.3. Cordones de alambre de diferente diámetro e igual paso.	13
1.3.1.3.1. Estructura Seale – S.....	13
1.3.1.3.2. La estructura Warrington.....	13

1.3.1.3.3.	La estructura Filler-Wire (Relleno)	14
1.3.2.	<i>Las características de los cables.</i>	14
1.3.2.1.	Flexión	14
1.3.2.2.	Tracción	14
1.3.2.3.	Abrasión	15
1.3.2.4.	Corrosión	15
1.3.2.5.	Giro	15
1.3.2.6.	Aplastamiento	15
1.3.2.7.	Golpes, sacudidas y vibraciones	15
1.4.	PARTICULARIDADES DE LA CONSTRUCCIÓN MEDIANTE CABLES ESTRUCTURALES	15
1.5.	ANÁLISIS DE LOS CABLES	16
1.5.1.	<i>Cable “parabólico”.</i>	16
1.5.2.	<i>Cable “Catenaria”.</i>	19
1.5.3.	<i>Cable “Cargas Puntuales”.</i>	21
1.5.4.	<i>Cable “Catenaria Elástica.</i>	23
1.5.4.1.	Catenaria elástica con respuesta a una carga puntual	29
1.5.4.2.	Catenaria elástica con respuesta a dos o más cargas puntuales	34
CAPITULO II.....		38
2. METODOLOGÍA		38
2.1.	GENERALIDADES	39
2.2.	ELABORACIÓN DE HERRAMIENTAS	40
2.2.1.	<i>Laboratorio de “Cables”</i>	40
2.2.1.1.	Programa Master del Laboratorio de “Cables”	40
2.2.1.1.1.	Código Master del Laboratorio de “Cables”	42
2.2.2.	<i>Experimento 1: “Cable Parabólico”.</i>	43
2.2.2.1.	Programa Master del experimento “Cable Parabólico”	43
2.2.2.1.1.	Código Master del experimento “Cable Parabólico”	45
2.2.2.2.	Caso1: $H_o < H_f$	45
2.2.2.3.	Caso2: $H_o > H_f$	46
2.2.2.4.	Caso3: $H_o = H_f$	48
2.2.3.	<i>Experimento 2: “Cable Catenaria”.</i>	49
2.2.3.1.	Programa Master del experimento “Cable Catenaria”	50
2.2.3.1.1.	Código Master del experimento “Cable Catenaria”	52
2.2.3.2.	Caso1: Flecha	52
2.2.3.3.	Caso2: Longitud	54
2.2.4.	<i>Experimento 3: “Cable con Cargas puntuales”.</i>	55
2.2.4.1.	Programa Master del experimento “Cable con cargas Puntuales”	56
2.2.4.1.1.	Código Master del experimento “Cable Catenaria”	58
2.2.4.2.	Caso1: Tramo 1	59
2.2.4.3.	Caso2: Tramo 2	60

2.2.4.4.	Caso2: Tramo 3.....	62
2.2.5.	<i>Experimento 4: “Cable Catenaria Elástica”</i>	63
2.2.5.1.	Programa Master del experimento “Cable Catenaria Elástica”	64
2.2.5.1.1.	Código Master del experimento “Cable Catenaria Elástica”	66
CAPITULO III.....		67
3. CASO DE ESTUDIO		67
3.1.	INTRODUCCIÓN.....	68
3.2.	METROCABLE DE MEDELLÍN	69
3.3.	METROCABLE DE MEDELLÍN – “LÍNEA K”	70
3.3.1.	<i>Localización</i>	71
3.3.2.	<i>Características técnicas de la línea “K” – Metrocable de Medellín.</i>	71
3.3.3.	<i>Solución numérica de la línea K – Metrocable de Medellín</i>	72
3.3.3.1.	Datos de entrada para la herramienta	72
3.3.3.2.	Resultados de salida.....	74
3.3.3.2.1.	Por tramo.....	74
3.3.3.2.2.	Todos los tramos.....	74
3.3.3.2.3.	Deformada del cable de un Vano.....	76
3.3.3.2.4.	Deformada del cable de todos los Vanos.....	77
3.3.3.3.	Análisis de la “línea K” – Metrocable de Medellín.....	79
3.3.3.3.1.	Tensión estática S_E	79
3.3.3.3.2.	Tensión dinámica S_D	79
3.3.3.3.3.	Tensión máxima en el cable de acero	80
3.3.3.3.4.	Determinación de la fecha máxima del cable.	80
CONCLUSIONES		82
RECOMENDACIONES		83
BIBLIOGRAFÍA		84
ANEXOS		87
A.	CÓDIGOS DE LOS EXPERIMENTOS	88
A1)	<i>Códigos del experimento 1 – Cable Parabólico</i>	88
A2)	<i>Códigos del experimento 2 – Cable Catenaria</i>	116
A3)	<i>Códigos del experimento 3 – Cable con Cargas Puntuales.</i>	131
A4)	<i>Códigos del experimento 4 – Cable Catenaria Elástica</i>	243
B.	PLATAFORMA VIRTUAL DE LOS EXPERIMENTOS.....	277
	<i>Experimento 1 – Cable Parabólico</i>	277
	<i>Experimento 2 – Cable Catenaria</i>	279
	<i>Experimento 3 – Cable con Cargas Puntuales</i>	281

C.	MODELO DEL TEXTO GUÍA DE PRÁCTICAS DE LOS EXPERIMENTOS.....	283
D.	MODELO DEL MANUAL DE LOS EXPERIMENTOS.....	284

ÍNDICE DE FIGURAS

Figura 1. Esquema de la Arquitectura del VLEE	7
Figura 2. Interfaz Gráfica de la Plataforma Virtual del GRISS-UTPL (HOME)	8
Figura 3. Interfaz Gráfica de la Plataforma virtual del GRISS-UTPL – Laboratorios	9
Figura 4. Parte de un Código Fuente de una herramienta – Laboratorio Cables.....	10
Figura 5. Diagrama de flujo – ejemplo.....	10
Figura 6: Componentes del cable.....	11
Figura 7. Cordones de alambre de igual diámetro	12
Figura 8. Cordones de alambre de igual diámetro	13
Figura 9. Cordones con estructura SEALE.....	13
Figura 10. Cordón con estructura Warrington	14
Figura 11. Cordón con estructura Filler-Wire.....	14
Figura 12: Puente colgante de acero del cable – Puente de Bailey	16
Figura 13. Cable sometido a una carga uniformemente distribuida a lo largo de una línea recta horizontal.	17
Figura 14: (a) Diagrama de un cable suspendido. (b). Diagrama de cuerpo libre del cable entre $x=0$ y una posición x arbitraria.....	18
Figura 15. Catenaria de una cadena	19
Figura 16. Un cable sometido a una carga uniformemente distribuida en toda su longitud. .	20
Figura 17. (a) N pesos suspendidos de un cable. (b) Primer diagrama de cuerpo libre. (c) Segundo diagrama de cuerpo libre.....	22
Figura 18. Coordenadas y geometría de la Catenaria Elástica	23
Figura 19. Equilibrio estático de un segmento del cable.	24
Figura 20. Equilibrio estático de un segmento del cable con una carga puntual.....	29
Figura 21. Equilibrio estático de un segmento del cable.	34
Figura 22. Esquema de la Arquitectura del Sistema	39
Figura 23. Diagrama de flujo del Programa Master del Laboratorio de Cables.....	41
Figura 24. Código master del Laboratorio de Cables.	42
Figura 25. Diagrama de flujo del Programa Master del experimento “Cable Parabólico”	44
Figura 26. Código master del experimento 1 – Cable parabólico.....	45
Figura 27. Esquema del experimento “Cable Parabólico” – Caso 1.....	45
Figura 28. Diagrama de flujo de las funciones del caso 1 del experimento “Cable Parabólico”	46
Figura 29. Esquema del experimento “Cable Parabólico” – Caso 2.....	47
Figura 30. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Parabólico”	48

Figura 31. Esquema del experimento “Cable Parabólico” – Caso 3.....	48
Figura 32. Diagrama de flujo de las funciones del caso 3 del experimento “Cable Parabólico”	49
Figura 33. Diagrama de flujo del Programa Master del experimento “Cable Catenaria”	51
Figura 34. Código master del experimento 2 – Cable Catenaria.....	52
Figura 35. Esquema del experimento “Cable Catenaria – Caso 1	52
Figura 36. Diagrama de flujo de las funciones del caso 1 del experimento “Cable Catenaria”	53
Figura 37. Esquema del experimento “Cable Catenaria – Caso 1	54
Figura 38. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”	55
Figura 39. Diagrama de flujo del Programa Master del experimento “Cable con Cargas Puntuales”	57
Figura 40. Código master del experimento 3 – Cable con Cargas Puntuales.....	58
Figura 41. Esquema del experimento “Cable Cargas Puntuales” – Caso 1: Tramo 1.....	59
Figura 42. Diagrama de flujo de las funciones del caso 1 del experimento “Cable Catenaria”	60
Figura 43. Esquema del experimento “Cable Catenaria – Caso 1	60
Figura 44. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”	61
Figura 45. Esquema del experimento “Cable Catenaria – Caso 1	62
Figura 46. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”	63
Figura 47. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”	63
Figura 48. Diagrama de flujo del Programa Master del experimento “Cable Catenaria Elástica”	65
Figura 49. Código master del experimento 4 – Cable Catenaria Elástica	66
Figura 50. Mapa de la ubicación de las líneas con tecnología Cable aéreo.	70
Figura 51. Esquema de la línea k – Metrocable Medellín.....	70
Figura 52. Esquema de la ubicación de la Línea K – Metrocable Medellín.....	71
Figura 53. Estructura del cable – Línea K – Metrocable Medellín	72
Figura 54. Archivo de Texto de los datos de entrada	73
Figura 55. Resultados de un Tramo	74
Figura 56. Resultados de todos los puntos de conexión de los tramos del archivo de texto. 76	
Figura 57. La deformada del cable de un vano del metrocable de Medellín – línea k.	77
Figura 58. La deformada de todos los tramos	78

Figura 59. Coordenadas de la carga (Máxima deflexión del cable)	80
Figura 60. Esquema de la carga (Máxima deflexión del cable)	81
Figura 61. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso1”	88
Figura 62. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso2”	89
Figura 63. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso3”	90
Figura 64. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso1”	94
Figura 65. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso2”	98
Figura 66. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso3”	101
Figura 67. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso1”	105
Figura 68. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso2”	109
Figura 69. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso3”	115
Figura 70. Código del experimento 2 – Función “LeerCableCatenCaso1”	116
Figura 71. Código del experimento 2 – Función “LeerCableCatenCaso2”	117
Figura 72. Código del experimento 2 – Función “CableCatenCaso1”	120
Figura 73. Código del experimento 2 – Función “CableCatenCaso2”	122
Figura 74. Código del experimento 2 – Función “SalidaCableCatenCaso1”	126
Figura 75. Código del experimento 2 – Función “SalidaCableCatenCaso2”	130
Figura 76. Código del experimento 3 – Función “LeerCablesCargasPuntuales1”	132
Figura 77. Código del experimento 3 – Función “LeerCablesCargasPuntuales2”	135
Figura 78. Código del experimento 3 – Función “LeerCablesCargasPuntuales3”	139
Figura 79. Código del experimento 3 – Función “CablesCargasPuntuales1”	159
Figura 80. Código del experimento 3 – Función “CablesCargasPuntuales2”	164
Figura 81. Código del experimento 3 – Función “CablesCargasPuntuales3”	182
Figura 82. Código del experimento 3 – Función “SalidaCablesCargasPuntuales1”	193
Figura 83. Código del experimento 3 – Función “SalidaCablesCargasPuntuales2”	213
Figura 84. Código del experimento 3 – Función “SalidaCablesCargasPuntuales3”	242
Figura 85. Código del experimento 4 – Función “LeerCatElas2”	243
Figura 86. Código del experimento 4 – Función “CatElas2”	246

Figura 87. Código del experimento 4– Función “SalidaCatElas2”	252
Figura 88. Código del experimento 4 – Función “Matriz3”	254
Figura 89. Código del experimento 4 – Función “Matriz2”	262
Figura 90. Código del experimento 4 – Función “Matriz1”	262
Figura 91. Código del experimento 4 – Función “MAGADIMEN”	263
Figura 92. Código del experimento 1 – Función “TensionCE”	265
Figura 93. Código del experimento 4 – Función “Carga 1”	266
Figura 94. Código del experimento 4 – Función “CoorxCE”	269
Figura 95. Código del experimento 4– Función “GraficaCE”	270
Figura 96. Código del experimento 4 – Función “CoorzCE”	275
Figura 97. Código del experimento 4 – Función “TolF”	276
Figura 98. Código del experimento 4 – Función “Tolx”	276
Figura 99. Interfaz gráfica del experimento 1 – configuración de datos de entrada con respecto al caso 1.	277
Figura 100. Interfaz gráfica del experimento 1 – configuración de los resultados de análisis con respecto al caso 1 y graficas.	277
Figura 101. Archivo de Texto de los resultados generados por el análisis del experimento 1.	278
Figura 102. Interfaz gráfica del experimento 2 – configuración de datos de entrada con respecto al caso 1 Flecha.....	279
Figura 103. Interfaz gráfica del experimento 1 – configuración de los resultados de análisis con respecto al caso 1 y graficas.	279
Figura 104. Archivo de Texto de los resultados generados por el análisis del experimento 2.	280
Figura 105. Interfaz gráfica de los datos de entrada para el tramo 1 del experimento 3. ...	281
Figura 106. Interfaz gráfica de los resultados generados por el análisis del experimento 3. Grafica cuando el cable está en posicion incial. Resultados de las reacciones en los apoyos.	281
Figura 107. Interfaz gráfica de los resultados generados por el análisis del experimento 3. Grafica con respecto a la deformada del cable. Resultados de las tensiones en el tramo correspondiente.....	281
Figura 108. Archivo de texto de los resultados generados por el análisis del experimento 3.	282

ÍNDICE DE TABLAS

Tabla 1. Nombre de la Funciones que corresponden a los experimentos.	41
Tabla 2. Datos de Entrada y de Salida del experimento “Cable Parabólico”.....	43
Tabla 3. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable Parabólico”.....	44
Tabla 4. Datos de Entrada y de Salida del experimento “Cable Catenaria”.....	50
Tabla 5. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable Catenaria”	51
Tabla 6. Datos de Entrada y de Salida del experimento “Cable con Cargas puntuales”.....	55
Tabla 7. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable con Cargas Puntuales”.....	57
Tabla 8. Datos de Entrada y de Salida del experimento “Cable Catenaria Elástica”.	64
Tabla 9. Nombre de la Funciones que corresponden a los Tramos del Experimento “Cable Catenaria Elástica”	66
Tabla 10. Origen y Destino de las líneas del metrocable (Teleférico) de Medellín.	69
Tabla 11. Datos de entrada de los 18 tramos de la línea K – Metrocable de Medellín.	71

RESUMEN

El Grupo de Investigación de Ingeniería Sísmica y Sismología de la UTPL ha generado un Laboratorio Virtual de Ingeniería Sísmica (VLEE por sus siglas en inglés) (<http://www.ingenieriasismica.utpl.edu.ec/?q=laboratorios>), el mismo que cuenta con varios sublaboratorios referentes a sistemas de protección sísmica, edificios, puentes, entre otros. Los elementos estructurales como “cables” o “tensores” no han sido abordados hasta ahora, es por esto que el presente trabajo de fin de titulación está enfocado en generar el Laboratorio de cables denominado “Cable Structures” el que permite realizar análisis estático de cables sometidos a cargas puntuales y distribuidas de distinta configuración.

Con la finalización del presente TFT, se desarrolló los códigos fuente en el lenguaje Basic para los 4 experimentos: Cable Parabólico, Cable Catenaria, Cable con Cargas Puntuales y Cable Catenaria Elástica. De esta manera el laboratorio de Cables se constituye como una herramienta eficaz para los estudiantes, así como de análisis de proyectos reales. Frente a la aplicación de la herramienta “Cable Catenaria Elástica” se puede evidenciar en el proyecto “Metrocable Medellín – línea k” al modelar la estructura del cable de el Caso de Estudio.

Palabras claves: Laboratorio Virtual, Herramienta, Cables, Catenaria, Catenaria Elástica.

ABSTRACT

The seismic engineering and Seismology Research Group of the UTPL has generated a Virtual seismic Engineering Laboratory (VLEE) (<http://www.ingenieriasismica.utpl.edu.ec/?q=laboratorios>), which has several Sub laboratories related to seismic protection systems, buildings, bridges, among others. The structural elements such as "cables" or "tensors" have not been addressed so far, this is why the present final degree project is focused on generating the cable lab called "Cable structures" which allows to perform static analysis of Cables subject to punctual and distributed loads of different configuration.

With the completion of this FDP, the source codes were developed in the Basic language for the 4 experiments: Parabolic cable, catenary cable, cable with point loads and elastic catenary cable. In this way the Cables laboratory is an effective tool for students, as well as analysis of real projects. In front of the application of the tool "Elastic Catenary Cable" can be evidenced in the project "Metrocable Medellín - line k" when modeling the cable structure of the Case Study.

Key words: Virtual Laboratory, Tool, Cables, Catenary, Elastic Catenary.

INTRODUCCIÓN

Los sistemas estructurales se clasifican en sistema de forma activa, de vector activo, de sección activa, de superficie activa y de altura activa. Los sistemas de forma activa se caracterizan por el material flexible, por actuar de acuerdo al estado de cargas y a la estabilización de la forma (Diez, 2005). En general se puede clasificar a los sistemas de forma activa en cuatro tipos de estructuras: de cables, en tienda, neumáticas y de arcos (Engel, 2001). El desarrollo de las estructuras de cables es fascinante. Lo atestiguan, no solamente las distintas áreas donde se han aplicado como en la ingeniería petrolera, la ingeniería eléctrica, la ingeniería industrial, la pesca, entre otros; sino también en el campo de la ingeniería civil.

A lo largo de la historia se han utilizado los cables a nivel mundial. En 1834, el ing. Wilhelm Albert empleo los primeros cables en la minería en Alemania; entre 1846 y 1888, el ing. Andrew Smith obtuvo la primera patente para la fabricación del cable de acero (Pons, 2013). La fabricación de cables es posible con distintos metales, en cuanto al acero es una opción favorable. El cable de acero es un dispositivo de alambres retorcidos, formados helicoidalmente con la intención de crear una cuerda de metal (Serrano N. y Castro F., 2009), capaz de resistir esfuerzos a tensión pura. Diseñados y acabados con el fin de que trabajen integralmente entre ellos (CAMESA, 2008), se conserva las cualidades propias de flexibilidad.

Los primeros usos estructurales de los cables se iniciaron con los puentes colgantes, vigas y cerchas. Los puentes colgantes y atirantados pueden alcanzar luces mayores de 1000 m, como Akashi-Kaikyo (1998), el Golden Gate (1937), y Sutong de China (2008) (Goicolea, 2012). Las vigas de cable, en la actualidad, son una alternativa eficiente para estructuras destinadas a soportar una fachada o una cubierta, como por ejemplo el estadio Slaski en Polonia y el estadio ciudad de la Plata en Argentina.

Las estructuras realizadas a partir de cables estructurales se clasifican en: estructuras de cables lineales, de cables planos y de cables tridimensionales. Las estructuras de cables lineales se aplican en las líneas de transmisión de energía eléctrica, las catenarias de trenes de alta velocidad, puentes colgantes, arcos, sistemas de transporte por cables, etc.; del mismo modo, las estructuras de cables planas se utilizan en cubiertas de edificios; y, por último, las estructuras de cables tridimensionales que se emplean en estructuras de tensegridad (Such, 2008).

La construcción mediante cables estructurales presenta una serie de particularidades. Entre ellos, la capacidad de trabajo a tracción debido a la elevada resistencia de los aceros con que están confeccionados los cables (Pons, 2013). El aprovechamiento de la capacidad resistente

máxima del material al no causar pandeos (CAMESA, 2008; Pons, 2013; Serrano N. & Castro F., 2009). Salvar grandes luces y cambiar los cables con otras obras estructurales, como muros, teleféricos, arcos o pórticos.

Los cables tienen la capacidad de soportar y transmitir cargas entre sus elementos. Por la pequeña rigidez, tiende a ser susceptible a experimentar grandes movimientos debido a las cargas concentradas y los efectos dinámicos (Pallares M. y Rodríguez C., 2008). El análisis dinámico de estructuras de cable es realizado en función de la matriz de rigidez tangente (Volokh, Vilnay, y Averbuch, 2003), mientras que, el análisis estático de las estructuras de cable es no lineal (Kim et al., 2003).

La presente investigación se refiere a la elaboración de la herramienta “Cable-Structures” para ser incorporadas al Laboratorio Virtual de Ingeniería Sísmica (VLEE) de la Universidad Técnica Particular de Loja (UTPL, <http://www.ingenieriasismica.utpl.edu.ec/?q=laboratorios>). El VLEE es una herramienta poderosa en cuanto al análisis, diseño y simulación de estructuras tradicionales como puentes, edificios etc. Los elementos estructurales como “cables” o “tensores” no se abordan en el mismo. Por ello, el objetivo de este trabajo es generar códigos fuentes, un manual de uso correspondiente a la herramienta informática y un texto guía para prácticas de laboratorio. Los alumnos y profesionales pueden ejecutar las herramientas al estar interesados en conocer el comportamiento de cables estructurales bajo acciones estáticas.

Para alcanzar los objetivos planteados se realizó la revisión del estado de arte con relación a la teoría de cables. El estudio incluye la elaboración de herramientas informáticas de casos específicos del comportamiento de cables. Las herramientas son desarrolladas para realizar los análisis estáticos, los casos específicos son cuatro: cargas uniformemente distribuidas a lo largo de líneas rectas (ejemplo: Puente colgante), cargas distribuidas uniformemente a lo largo de cables (ejemplo: La catenaria), cable sometido a cargas concentradas (ejemplo: Teleférico) y Catenaria Elástica.





La investigación consta de cuatro capítulos, el primer capítulo se establece el marco teórico referente a los aspectos generales del VLEE y a la teoría de los cables, como, análisis y especificaciones; el segundo capítulo se indica la metodología y el tercer capítulo se presenta el caso de estudio “Monocable de Medellín – Línea K”. A continuación, conclusiones y recomendaciones. Posterior a ello, se presenta la bibliografía y anexos, de esta manera se respalda la investigación.

OBJETIVOS

A. GENERAL

Generar la herramienta informática “Cable-Structures” para ser incorporadas en el VLEE.

B. ESPECÍFICOS

-  Generar códigos fuente en lenguaje Basic.
-  Generar un manual de uso de la herramienta informática.
-  Generar un texto guía para prácticas de laboratorio.
-  Con la herramienta generada realizar estudios de caso relevantes.

CAPITULO I

1. MARCO TEÓRICO

1.1. Laboratorio virtual de ingeniería sísmica (VLEE)

El VLEE es una herramienta poderosa en cuanto al análisis, diseño y simulación de diversos sistemas estructurales, y actualmente cuenta con los siguientes laboratorios:

- Sistemas de protección sísmica
- Dinámica de estructuras
- Edificios
- Elementos de hormigón armado
- Puentes
- Cables (en proceso)

Para tener acceso a los laboratorios, el Grupo de Investigación de Ingeniería Sísmica y Sismologías de la UTPL (GRISS UTPL) ha generado una plataforma virtual (<http://www.ingenieriasismica.utpl.edu.ec/?q=laboratorios>), la misma que se describe a continuación.

1.1.1. Arquitectura del VLEE.

El laboratorio cuenta con una Arquitectura distribuida en Tres Capas, con tecnología ASP y un servidor web IIS.



Figura 1. Esquema de la Arquitectura del VLEE

Fuente: Autor

Elaboración: Autor

El ASP (Active Server Pages) es una página HTML que contiene scripts que son procesados por el servidor web IIS antes de ser enviada al cliente web (Perovich y Vignaga, s/f).

El servidor IIS (Internet Information Services) autentica la solicitud si fuera necesario y, a continuación, busca el recurso solicitado, es un servidor web de aplicación de páginas dinámicas y un conjunto de servicios para el sistema operativo Microsoft Windows (Perovich y Vignaga, s/f).

1.1.1.1. Capa de presentación.

El usuario interactúa con la página web del Laboratorio virtual. Es responsable de la presentación de los datos, enviar la información a la capa de negocios para su procesamiento, recibir los resultados del procesamiento de la capa de negocios y presentar los resultados en la interfaz gráfica para el usuario (Coti Colop, 2003; Perovich y Vignaga, s/f). La programación de la capa es el formateo de la información enviada por el servidor y la captura de las acciones realizadas por el cliente. (Vara Mesa, López Sanz, y Verde Marín, 2014)

1.1.1.2. Capa de negocios.

El servidor web IIS responsable de recibir la información de entrada de la capa de presentación, interactuar con los servicios de datos para ejecutar las operaciones de negocios y lógica, y enviar la respuesta procesada a la capa de presentación (Coti Colop, 2003).

1.1.1.3. Capa de datos.

Un sistema físico (computador) en el que está instalado el sistema de gestión de base de datos (SGBD) (Marco Besteiro y Miguel Rodríguez, s/f). Esta capa envía la información a la capa de negocios para que sea procesada e ingresada en objetos según necesite (Vargas del Valle y Maltés Granados, s/f). Es responsable de almacenar, recuperar y mantener la integridad de los datos (Coti Colop, 2003).

1.1.2. Interfaz gráfica del VLEE.

Una interfaz de usuario es el aspecto que se muestra en un programa, en el modo de interactuar con el usuario el lenguaje visual debe ser claro, los contenidos ordenados y comprensibles (Blanco, 2002; Velasco, Sánchez, Laureano, y Mora, 2009).

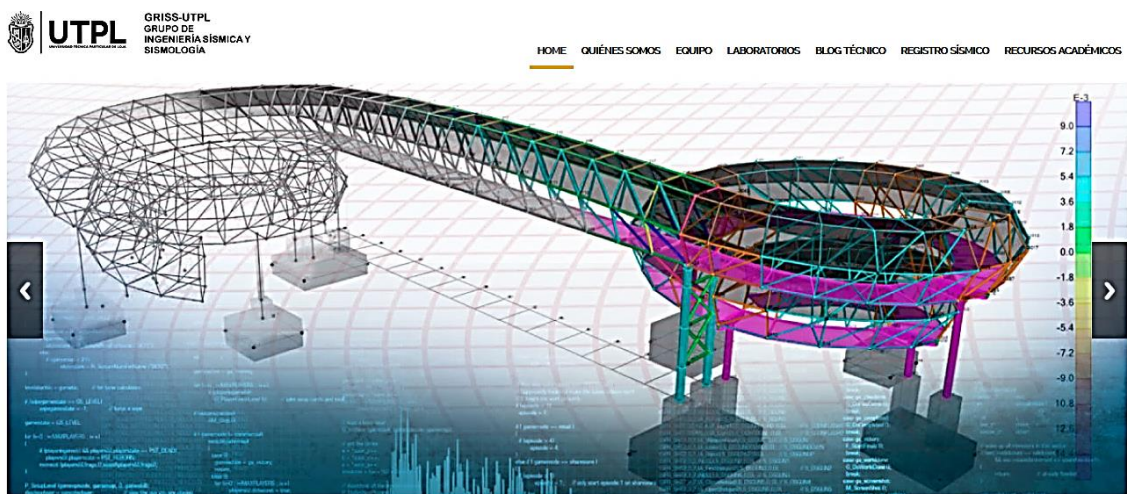


Figura 2. Interfaz Gráfica de la Plataforma Virtual del GRISS-UTPL (HOME)

Fuente: GRISS-UTPL

Elaboración: Autor

En la figura 2, se muestra la interfaz gráfica de usuario correspondiente al GRISS-UTPL. La plataforma virtual cuenta con laboratorios, un blog técnico, registro sísmico y recursos académicos.



Figura 3. Interfaz Gráfica de la Plataforma virtual del GRISS-UTPL – Laboratorios

Fuente: GRISS-UTPL

Elaboración: Autor

En la figura 3, se presenta la interfaz gráfica de usuario correspondiente al GRISS-UTPL de los distintos laboratorios virtuales del VLEE.

1.2. Programación

Blanco (2002) define a la programación como el conjunto de operaciones que lleva a cabo un ordenador para proporcionar un determinado resultado. Para ello es importante elegir un lenguaje de programación, y existe una infinidad de lenguajes: Visual Basic, Visual C++, Java, Pascal, etc.

Todo lenguaje de programación se compone de un conjunto más o menos extenso de palabras claves y símbolos, que forman la denominada sintaxis del lenguaje, y una serie de normas o reglas para el correcto uso y combinación de tales palabras y símbolos (Blanco, 2002).

1.2.1. Código fuente.

El código fuente de un software o programa informático es el conjunto de líneas de texto que pautan las instrucciones que debe seguir un ordenador para ejecutar ese programa (SoftDoit, s/f) y es similar a un texto que se genera en Block de Notas.


```

C:\Users\DELL i7\De...\ProgramaMasterCableParabolico.m Page 1

1 function [] = ProgramaMasterCableParabolico (ruta, nameFileUser,
nameFileExperimento, nameFileUserEnglish)
2 %Experimento 1: Cable Parabólico
3 switch c1(2)
4     case 1 %Caso 1: Ho<Hf
5         [Ho,Hf,L,Wo,DT] = LeerCableParabCasol(ruta, nombre Archivo);
6         [Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf,
DTto, DTf, str, x, y, A, B, D, wx, wy] = CableParabCasol(Ho,Hf,L,Wo,DT);
7         SalidaCableParabCasol(Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax,
Tten, Nten, S, Lo, Lf, DTto, DTf, str, x, y, A, B, D, wx, wy, ruta,
nameFileUser, nameFileExperimento, nameFileUserEnglish);

```

Figura 4. Parte de un Código Fuente de una herramienta – Laboratorio Cables

Fuente: Autor

Elaboración: Autor

1.2.2. Diagramas de flujo.

Blanco (2002) define a un diagrama de flujo como una representación gráfica, basada en símbolos, de los pasos que debe realizar un algoritmo.

Un algoritmo es el conjunto de acciones a realizar para resolver un determinado problema (Blanco, 2002).

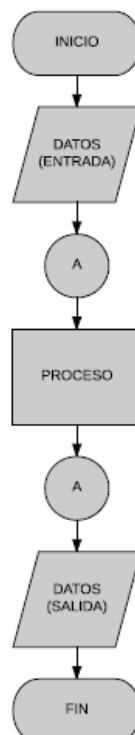


Figura 5. Diagrama de flujo – ejemplo

Fuente: Autor

Elaboración: Autor

1.3. Cable de acero

Serrano N. y Castro F. (2009) definen el cable de acero como un dispositivo de alambres retorcidos, formados helicoidalmente con la intención de crear una cuerda de metal, capaz de resistir esfuerzos a tensión pura. Diseñados y acabados con el fin de que trabajen integralmente entre ellos (CAMESA, 2008), se conserva las cualidades propias de flexibilidad.

Portales P. (2013) muestra las siguientes propiedades que debe obtener el cable:

- Formación de un cuerpo único de dar respuesta a los esfuerzos de tracción.
- Flexibilidad suficiente para enrollamiento, lo cual facilita el transporte y la puesta en obra de grandes longitudes, sin necesidad de efectuar empalmes.

Componentes del cable de acero:

- Alambre
- Torón
- Alma

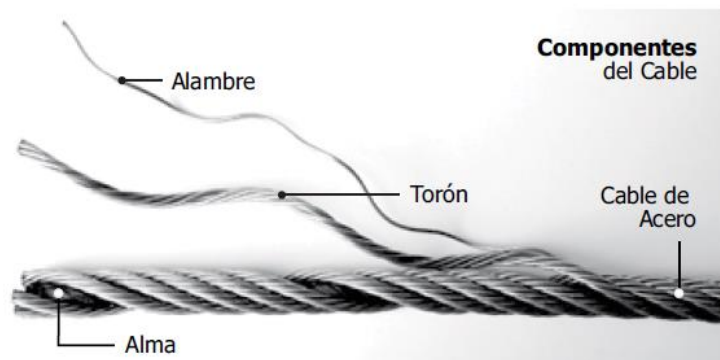


Figura 6: Componentes del cable
Fuente: (EMCOCABLES, s/f)
Elaboración: (EMCOCABLES, s/f)

El elemento básico de un cable son los alambres, el torón es un conjunto de alambres enrollados en torno a un central; y el alma sirve de soporte al conjunto de torones enredados alrededor de sí mismo, hasta conseguir la sección deseada del cable (Pons, 2013).

Los alambres son hilos delgados cuyos diámetros pueden oscilar entre decimas de milímetro y los cinco milímetros (Pons, 2013). El torón es un alambre que sirve de soporte para el resto de alambres y del mismo diámetro. El alma es un torón que sirve de soporte para el resto de torones puede ser del mismo diámetro o distinto, y de acero o textil.

El cable de acero es el elemento ideal en las estructuras para cubrir grandes distancias, por la combinación de su resistencia a la tracción del acero y la eficiencia de la tracción simple (Salvadori, 2005).

1.3.1. Las estructuras del cable de acero.

De acuerdo a Serrano N. y Castro F. (2009) denominan la siguiente estructuras del cable de acero:

1.3.1.1. *La estructura transversal de los cordones.*

La estructura transversal de los cordones se relaciona con el sistema de trenzado longitudinal de sus alambres. En el cordoneado, que es donde se produce este efecto, hemos visto que los cordones pueden ser:

- De igual ángulo: El mismo ángulo de hélice, la redistribución de la carga total a que está sometido el cable puede aportar por igual en cada uno de los alambres.
- De igual paso: El mismo paso de hélice y los ángulos distintos, la redistribución de la carga total que se somete el cable dará lugar a distintos esfuerzos en los alambres de las diferentes capas.

1.3.1.2. *Cordones de alambre de igual diámetro.*

Las pertenecientes a este grupo son los ya denominados “normales”. Como ejemplos se muestra a continuación en la figura 7, donde los cordones están conformados con alma metálica. Almas metálicas de un alambre y una capa de 6 alambres. El otro cordón de tres capas de 6, 12 y 18 alambres.

La composición de los mismos es

1+6	7 alambres
1+6+12+18	37 alambres

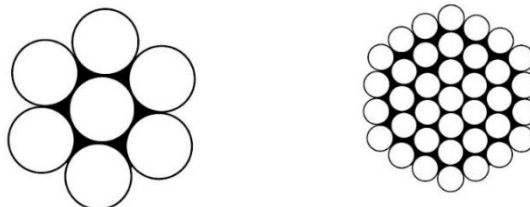


Figura 7. Cordones de alambre de igual diámetro
Fuente: Serrano N. y Castro F. (2009)
Elaboración: Autor

A diferencia de la figura 7 en la figura 8 se tiene cordones compuesto de un alma textil cubierta por una y dos capas de alambres de igual diámetro. Almas textiles. Un cordón con una capa de 12 alambres y otro con dos capas de 12 y 18 alambres.

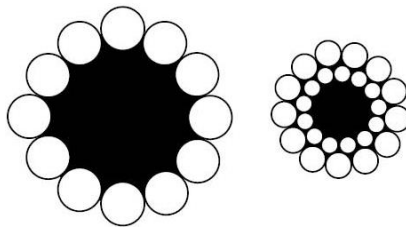


Figura 8. Cordones de alambre de igual diámetro
Fuente: Serrano N. y Castro F. (2009)
Elaboración: Autor

1.3.1.3. **Cordones de alambre de diferente diámetro e igual paso.**

1.3.1.3.1. **Estructura Seale – S.**

La disposición en la que todas las capas de alambre, o al menos las dos últimas, llevan el mismo número de hilos, con diferentes diámetros. La Construcción de la última capa tiene los alambres de mayor diámetro que la capa interior, dándole al Torón mayor resistencia a la abrasión (EMCOCABLES, s/f). En la Figura 9 la composición del cordón se describe como (1) + N +N (S). En la figura 9, 1+9+9(S) y 1+6+12+12 (S)

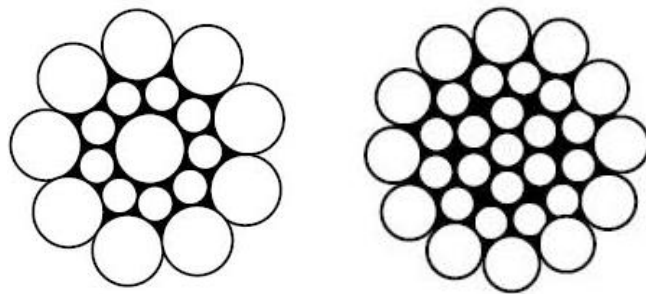


Figura 9. Cordones con estructura SEALE
Fuente: Serrano N. y Castro F. (2009)
Elaboración: Autor

1.3.1.3.2. **La estructura Warrington.**

Son cordones de dos capas de alambres, de forma que la segunda tiene doble número que la primera. Pero estos alambres de la segunda capa no son todos iguales, sino de dos diámetros diferentes repartidos alternativamente uno grueso y uno fino, de manera que el fino se apoya sobre la cresta de la capa inferior. Se caracteriza por tener una capa exterior formada por alambres de dos diámetros diferentes, al alternar su posición dentro de la corona (EMCOCABLES, s/f). En la figura 10, la composición del cordón se describe como (1) + N + (N+N) (W). En la figura 4, 1+6+ (6+6) (W).

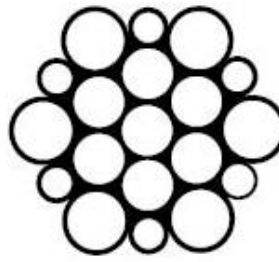


Figura 10. Cordón con estructura Warrington
Fuente: Serrano N. y Castro F. (2009)
Elaboración: Autor

1.3.1.3.3. La estructura Filler-Wire (Relleno).

Se distingue por llevar entre dos capas de alambres otros hilos más finos para rellenar los huecos existentes entre las mismas. Es decir, tres capas de alambres, donde la tercera tiene doble número que la primera, y la segunda, el mismo número de alambres que la primera. Se construye este tipo de cordón, cuando la utilización del cable exige una mayor sección metálica y más capacidad de resistencia al aplastamiento (EMCOCABLES, s/f). La composición del cordón se describe como $(1) + N + N + 2N$ (R), en la fig. 10, 1+6+6+12.

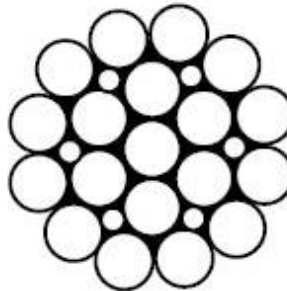


Figura 11. Cordón con estructura Filler-Wire
Fuente: Serrano N. y Castro F. (2009)
Elaboración: Autor

1.3.2. Las características de los cables.

A continuación Serrano N. y Castro F. (2009) exponen las características de los cables:

1.3.2.1. Flexión.

La flexión como un cúmulo de esfuerzos a fatiga creada para tener que seguir el cable una línea no recta, por ejemplo, al pasar por poleas o tambores de arrollamiento.

1.3.2.2. Tracción

Tracción es un conjunto de esfuerzos longitudinales que debe soportar el cable debido a las cargas estáticas, que además son incrementadas, en los esfuerzos dinámicos que se originan debido a la inercia de las masas a mover.

1.3.2.3. *Abrasión*

Abrasión es básicamente, debido a los roces ya sea con el suelo, con plataforma de un camión, con poleas, etc.; es decir, con todo lo que roza la superficie.

Abrasión: se refiere al empeoramiento de las condiciones de trabajo ya sea porque el cable opera en un ambiente cálido o a un aumento de temperatura originado por unos roces efectuados a alta velocidad o por ser muy repetitivos.

1.3.2.4. *Corrosión*

Corrosión: agresión originada como consecuencia del medio ambiente en que debe operar el cable ya sea por vapores ácidos, agua marina o por el simple contacto con la atmosfera al trabajar al aire libre.

1.3.2.5. *Giro.*

Giro: siempre que un cable se desplace una carga no guiada, el poco o mucho momento torsor del cable produce giro alrededor de su eje.

1.3.2.6. *Aplastamiento.*

Aplastamiento: cuando el cable se enrolla y soporta carga, necesariamente sufre una sollicitación de aplastamiento que puede modificar la estructura interna del cable.

1.3.2.7. *Golpes, sacudidas y vibraciones.*

Golpes, sacudidas y vibraciones: en esta agresión, además de los golpes propiamente dichos, también incluyen las sacudidas debidas a una o varias maniobras que pueden producirse al operar la máquina. De entre ellas mencionaremos las aceleraciones, cambios de velocidad, frenados, etc.

1.4. *Particularidades de la construcción mediante cables estructurales*

A continuación se presenta una serie de particularidades positivas que Portales P. (2013) definió en función de las características mecánicas de los cables de acero:

- Gran capacidad de trabajo a la tracción debido a la elevada resistencia de los aceros que está confeccionado los cables, y se utilizan aceros con limites elásticos del orden 2000 N/mm².
- Máximo aprovechamiento de la capacidad resistente del material al no producirse pandeos, puesto que el material esta traccionado.
- Facilidad para alcanzar grandes luces debido tanto a la forma de trabajo como a la elevada resistencia unitaria de los cables.

- Posibilidad de combinar las estructuras de cables con otras soluciones constructivas, como muro, mástiles, arcos o pórticos.

1.5. Análisis de los Cables

Con frecuencia los sistemas formados por cables se denominan sistemas funiculares (Salvadori, 2005). Por la resistencia, poco peso y flexibilidad, las cuerdas y cables se utilizan para soportar cargas y transmitir fuerzas en estructuras, máquinas y vehículos. (Bedford y Fowler, 2008).

Según el diccionario RAE: funiculares proviene de latín “*funicŭlus*”, cuyo significado es cuerda.

En las siguientes secciones se determinarán los cables con cargas distribuidas: Cable “Parabólico” y Cable “Catenaria”; el Cable con Cargas Puntuales y la Catenaria Elástica.

Beer y Cornwell (2010) concretan que el cable está sometido a cargas distribuidas, cuando las acciones externas sobre el cable no se concentran en unos pocos, sino que están repartidas en forma continua a lo largo del mismo.

La flecha optima de una catenaria es aproximadamente un tercio de su luz, para esa relación de flecha a luz, la catenaria y la parábola son curvas similares (Salvadori, 2005)

1.5.1. Cable “parabólico”.

Un cable con un gran número de cargas iguales separadas horizontalmente a distancias iguales, cargas uniformemente distribuidas a lo largo de líneas rectas, se aproxima a una curva geométrica, la parábola. (Salvadori, 2005).

El cable principal de un puente colgante es el ejemplo de un cable parabólico (figura 12), la carga transmitida al cable principal por el gran número de cables verticales se puede representar como una carga distribuida (Bedford y Fowler, 2008).



Figura 12: Puente colgante de acero del cable – Puente de Bailey
Fuente: CHINA HARZONE INDUSTRY CORP., LTD. (s/f). Puente colgante de acero del cable de la estructura simple para el río más largo de los palmos. Recuperado de <http://spanish.steel-trussbridge.com/sale-580516-simple-structure-steel-cable-suspension-bridge-for-longest-spans-river.html>
Elaboración: CHINA HARZONE INDUSTRY CORP., LTD. (s/f).

Para el análisis del cable parabólico se debe considerar la siguiente hipótesis (Bedford y Fowler, 2008; Beer y Cornwell, 2010):

- El peso del cable y la sección es despreciable frente a las cargas que actúan sobre el cable.
- Las fuerzas que actúan sobre el cable son verticales.
- El origen del sistema coordenado está en el punto más bajo del cable.
- La curva descrita por el cable es la parábola $y = \frac{1}{2}ax^2$.
- Cable perfectamente flexible y no se extiende, su longitud es constante, antes y después de la carga.
- No ofrece resistencia a flexión, la fuerza de tensión que actúan en el cable es tangente a los puntos a lo largo de su longitud.

En esta sección se determinará la forma y la variación de la tensión en un cable cargado de esta forma (Figura 13) y las fórmulas están basadas en Bedford y Fowler (2008).

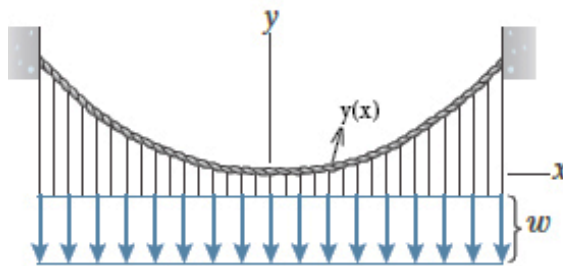


Figura 13. Cable sometido a una carga uniformemente distribuida a lo largo de una línea recta horizontal.

Fuente: Bedford & Fowler (2008)

Elaboración: Bedford & Fowler (2008)

Forma

$$y = \frac{1}{2}ax^2$$

(Ecuación 1)

Donde:

Y: es la flecha o altura del punto más bajo del cable hasta el punto inicial o final del cable. La flecha óptima para un cable parabólico es igual a una tercera parte de luz (Salvadori, 2005). Unidades en metros (m).

X: es la distancia horizontal (vano) entre el punto inicial y final del cable. El valor es positivo cuando la distancia horizontal es desde el eje 0 (cero) hasta el punto final del cable. Y el valor es negativo desde el punto inicial del cable hasta el eje. Unidades en metros (m).

a: es un parámetro. Unidades en m^{-1} .

$$a = \frac{w}{T_o}$$

(Ecuación 2)

Donde:

W: es la magnitud de la carga distribuida. Unidades en $\frac{KN}{m}$.

To: es la tensión en el cable en su punto más bajo, o también denominada, tensión horizontal. La tensión es mínima en el punto más bajo del cable ($x=0$) y crece en forma monótona con su distancia al punto más bajo. Unidades en KN .

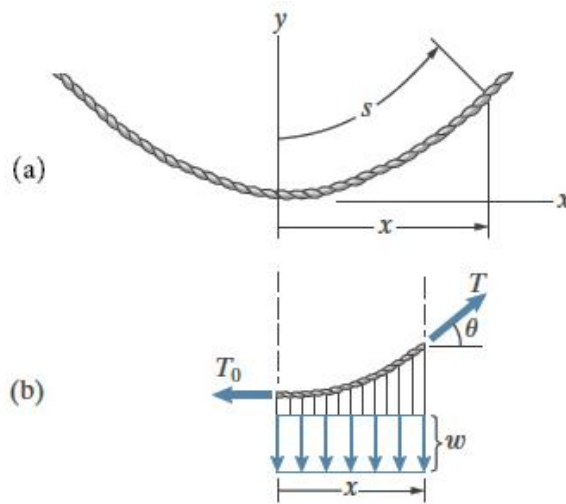


Figura 14: (a) Diagrama de un cable suspendido. (b). Diagrama de cuerpo libre del cable entre $x=0$ y una posición x arbitraria
Fuente: Bedford & Fowler (2008)
Elaboración: Bedford & Fowler (2008)

En la figura 14.a puede verse el diagrama de un cable suspendido con su deformada, el cable adopta la forma de un arco de parábola. Y en la figura 14.b. se presenta el diagrama de un tramo del cable suspendido, de esta forma el origen coordenado se coloca en el punto más bajo del cable, donde la tensión horizontal es T_o .

Tensión

$$T = T_o \sqrt{1 + a^2 x^2}$$

(Ecuación 3)

Donde:

T: Tensión a lo largo del cable. Unidades en KN .

Longitud

$$s = \frac{1}{2} \left\{ x\sqrt{1+a^2x^2} + \frac{1}{a} \ln \left[ax + \sqrt{1+a^2x^2} \right] \right\} \quad (\text{Ecuación 4})$$

Donde:

S: es la longitud del cable. Unidades en metros (m).

1.5.2. Cable “Catenaria”.

Un cable sometido a la acción de su propio peso, carga distribuida a lo largo del cable, La curva funicular difiere de una parábola, si bien posee la misma configuración (Salvadori, 2005). Fue propuesto por Jacob Bernoulli, y la solución fue dada por Johan Bernoulli, Huygens y Leibniz (1961) (Museros Romero, 2017).



Figura 15. Catenaria de una cadena

Fuente: Eliatron Dixit, Tito. (2009). Tito Eliatron Dixit. Recuperado de: <http://eliatron.blogspot.com/2009/05/la-curva-catenaria-cadenas-trenes-y.html>

Elaboración: Eliatron Dixit, Tito. (2009)

Para el análisis de la catenaria se debe considerar la siguiente hipótesis (Bedford y Fowler, 2008; Beer y Cornwell, 2010):

- La sección es despreciable.
- Se admite que el peso w por metro de cable es constante, no se desprecia el peso propio ante el resto de acciones, sino que este constituye, de hecho, la única acción.
- El origen del sistema coordenado está en el punto más bajo del cable.
- Cable perfectamente flexible y no se extiende, su longitud es constante, antes y después de la carga.
- No ofrece resistencia a flexión, la fuerza de tensión que actúan en el cable es tangente a los puntos a lo largo de su longitud.

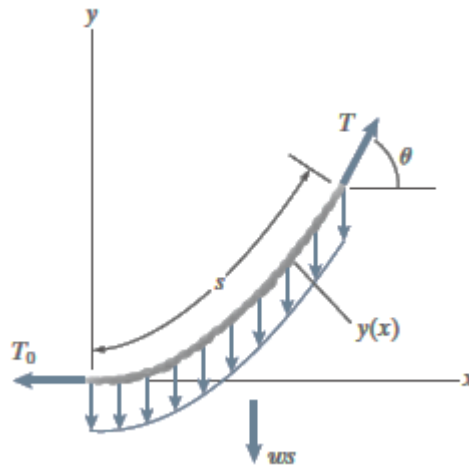


Figura 16. Un cable sometido a una carga uniformemente distribuida en toda su longitud.

Fuente: Bedford & Fowler (2008)

Elaboración: Bedford & Fowler (2008)

Forma

$$y = \frac{1}{2}(\cosh ax - 1)$$

(Ecuación 5)

Donde:

Y=es la flecha o altura del punto más bajo del cable hasta el punto inicial o final del cable. La flecha óptima para un cable parabólico es igual a una tercera parte de luz (Salvadori, 2005). Unidades en metros (m).

X=es la distancia horizontal (vano) entre el punto inicial y final del cable. El valor es positivo cuando la distancia horizontal es desde el eje 0 hasta el punto final del cable. Y el valor es negativo desde el punto inicial del cable hasta el eje. Unidades en metros (m).

Parámetro: a

$$a = \frac{w}{T_o}$$

(Ecuación 6)

Donde:

W: es la magnitud de la carga distribuida. Peso propio del cable. Unidades en kg/m.

To: es la tensión en el cable en su punto más bajo, o también denominada, tensión horizontal. La tensión es mínima en el punto más bajo del cable (x=0) y crece en forma monótona con su distancia al punto más bajo. Unidades en N.

Tensión

$$T = T_o \cosh(ax) \quad (\text{Ecuación 7})$$

Donde:

T: Tensión a lo largo del cable. Unidades en N.

Longitud

$$S = \frac{\sinh(ax)}{a}. \quad (\text{Ecuación 8})$$

Donde:

S: es la longitud del cable. Unidades en metros (m).

1.5.3. Cable “Cargas Puntuales”.

Cables sometidos a cargas concentradas, se considera el caso de número arbitrario N de cuerpos suspendidos de un cable (Figura 16) (Beer y Cornwell, 2010).

Para el análisis del cable con cargas puntuales se debe considerar la siguiente hipótesis (Bedford y Fowler, 2008; Beer y Cornwell, 2010):

- Las fuerzas que actúan sobre el cable son verticales
- Las líneas de acción son conocidas y las abscisas.
- El peso del cable es despreciable frente a las cargas que actúan sobre él.
- El cable es flexible, y por lo tanto en los puntos donde actúa cualquier carga concentrada o reacción puede admitirse que existe una articulación perfecta.
- Los segmentos son rectos, entre las cargas y los puntos de conexión del cable; es decir, el cable es poligonal.

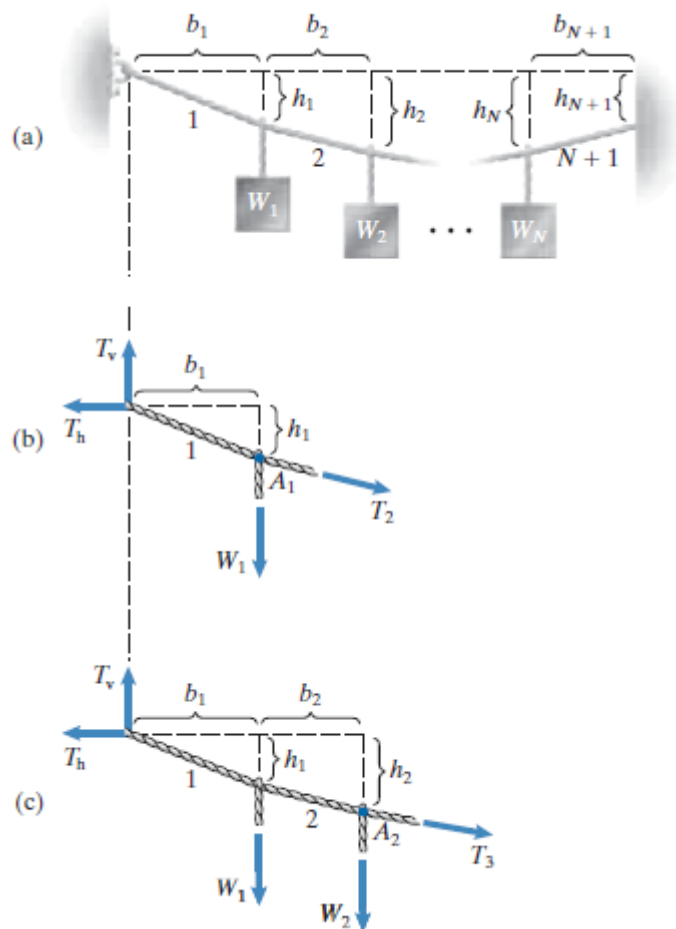


Figura 17. (a) N pesos suspendidos de un cable. (b) Primer diagrama de supero libre. (c) Segundo diagrama de cuerpo libre.
 Fuente: (Bedford & Fowler, 2008)
 Elaboración: (Bedford & Fowler, 2008)

Beer y Cornwell (2010) muestran el procedimiento de cálculo para determinar las posiciones de la carga, las reacciones en los apoyos, la deformada y las tensiones del cable. De este modo se puede calcular al seguir los pasos mostrados a continuación:

- Se conocen las distancias horizontales b_1, b_2, \dots, b_{N+1} .
- La distancia vertical h_{N+1} que especifica la posición de la conexión derecha del cable, es un valor conocido.
- Se tiene tres objetivos: 1) determinar la configuración (forma) del cable al calcular las distancias verticales h_1, h_2, \dots, h_N que especifican las posiciones de los puntos de unión de las cargas, 2) determinar la posición de las cargas, y 3) determinar las tensiones en los segmentos $1, 2, \dots, N+1$ del cable.
- Se comienza por dibujar un diagrama de cuerpo libre, se corta el cable en su punto de conexión izquierdo y justo a la derecha del peso W_1 (Figura 16.b).

- Se descompone la tensión del cable en el punto de conexión izquierdo en sus componentes horizontal y vertical, T_h y T_v , se suma los momentos respecto al punto de unión A_1 , se obtiene la ecuación:

$$\sum M_{\text{punto}A_1} = h_1 T_h - b_1 T_v = 0 \quad (\text{Ecuación 9})$$

- El siguiente paso es dibujar un diagrama de cuerpo libre al cortar el cable en su punto de conexión izquierdo y justo a la derecha del peso w_2 (figura 16.c). se suma los momentos con respecto a A_2 , se obtiene.

$$\sum M_{\text{punto}A_2} = h_2 T_h - (b_1 + b_2) T_v - b_2 W_1 = 0 \quad (\text{Ecuación 10})$$

- Se procede de esta manera, el corte del cable justo a la derecha de cada uno de los N pesos, se obtiene N ecuaciones.

1.5.4. Cable “Catenaria Elástica.

Los cables no resisten esfuerzos de flexión y de compresión, presentan una resistencia a la tracción con una determinada ley de cargas, y se deforma de modo que los esfuerzos de tracción generados resistan la carga aplicada. (Pallares Muñoz y Rodríguez Calderón, 2008).

Según Merritt (1992), Pallares M. y Rodríguez C. (2008), afirman que la eficiencia estructural del cable se debe a la uniformidad de los esfuerzos de tracción en el área del cable y a su variación pequeña a lo largo del eje longitudinal.

El problema de la Catenaria Elástica, corresponde a la respuesta estática de un cable suspendido entre dos soportes rígidos, es de sección transversal constante cuando está descargado y está compuesto por un material homogéneo que es linealmente elástico (Irvine, 1981). El enfoque utilizado a continuación, es el propuesto por Irvine (1981).

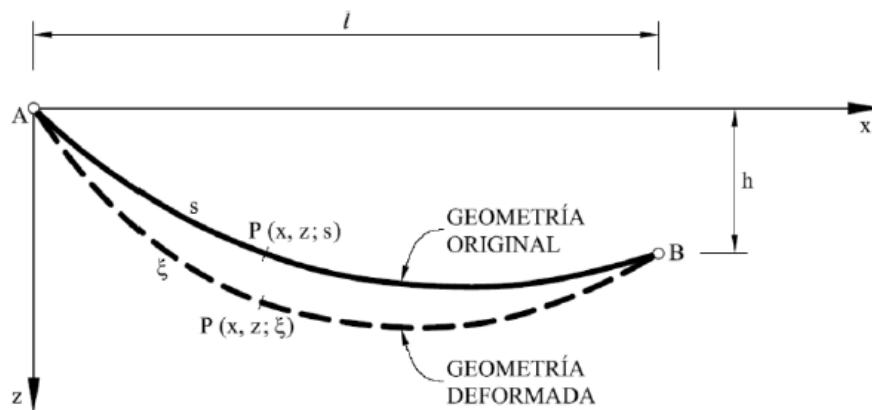


Figura 18. Coordenadas y geometría de la Catenaria Elástica
Fuente: Pallares M. & Rodríguez C. (2008)
Elaboración: Pallares M. & Rodríguez C. (2008)

En la figura 17, el cable está suspendido entre dos puntos fijos A y B, que tienen coordenadas cartesianas $(0,0)$ y (l,h) respectivamente.

Donde:

l : Tramo del cable (luz).

h : Desplazamiento vertical relativo de los extremos.

L_o : La longitud no estirada del cable (Geometría original). No es mucho mayor que $(\sqrt{l^2 + h^2})$, y tampoco puede ser mucho menor, para que la ley de Hooke no sea quebrantada.

P : Un punto en el cable, tiene coordenadas lagrangiana $(x, z; s)$ en el perfil sin tensión (Geometría original).

s : La longitud del cable desde el origen hasta ese punto (P), cuando la geometría es original.

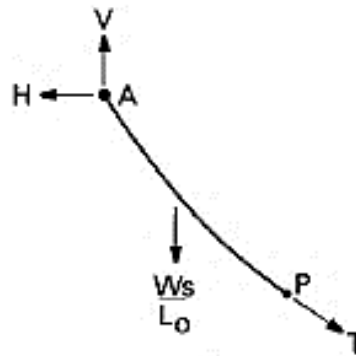


Figura 19. Equilibrio estático de un segmento del cable.
Fuente: Irvine (1981)
Elaboración: Irvine (1981)

En la figura 18, se muestra el equilibrio estático de las fuerzas horizontales y verticales de un segmento del cable.

$$T \frac{dx}{dp} = H \quad (\text{Ecuación 11})$$

$$T \frac{dz}{dp} = V - W \frac{s}{L_o} \quad (\text{Ecuación 12})$$

En que, debido a la conservación de la masa, el peso de ese tramo del perfil del cable mostrados es $W \frac{s}{L_o}$.

Donde:

W : Peso propio del cable (mgL_o).

V : La reacción vertical en el soporte en A.

H : La reacción horizontal en el soporte en A o la componente horizontal constante de la tensión del cable.

Una relación constitutiva que es una expresión matemáticamente consistente de la ley de Hooke es.

$$T = EA_o \left(\frac{dp}{ds} - 1 \right) \quad (\text{Ecuación 13})$$

Donde:

E : Módulo de Young.

A_o : Área de la sección transversal uniforme en el perfil de la geometría original.

Las condiciones finales en los soportes de cable A y B.

$$\begin{aligned} x = 0, \quad z = 0, \quad p = 0 \quad \text{at} \quad s = 0 \\ x = l, \quad z = h, \quad P = L \quad \text{at} \quad s = L_o \end{aligned}$$

Solución para $T=T(s)$

$$T(s) = \left\{ H^2 + \left(V - W \frac{s}{L_o} \right)^2 \right\}^{1/2} \quad (\text{Ecuación 14})$$

Solución para $x=x(s)$

$$x(s) = \frac{Hs}{EA_o} + \frac{HL_o}{EA_o} \left[\sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left\{ \frac{V - W \frac{s}{L_o}}{H} \right\} \right] \quad (\text{Ecuación 15})$$

Solución para $z=z(s)$

$$z(s) = \frac{Ws}{EA_o} \left(\frac{V}{W} - \frac{s}{2L_o} \right) + \frac{HL_o}{W} \left[\left\{ 1 + \left(\frac{V}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - W \frac{s}{L_o}}{H} \right)^2 \right\}^{1/2} \right] \quad (\text{Ecuación 16})$$

Solución para H y V

En la derivación de las soluciones para x y z, ecuación 15 y ecuación 16, respectivamente. Se usaron las condiciones de s=0. Ecuación trascendental en H y V.

$$l = \frac{HL_o}{EA_o} + \frac{HL_o}{W} \left\{ \sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left(\frac{V-W}{H} \right) \right\} \quad (\text{Ecuación 17})$$

Ecuación algebraica en H y V.

$$h = \frac{WL_o}{EA_o} \left(\frac{V}{W} - \frac{1}{2} \right) + \frac{HL_o}{W} \left[\left\{ 1 + \left(\frac{V}{H} \right)^2 \right\}^{\frac{1}{2}} - \left\{ 1 + \left(\frac{V-W}{H} \right)^2 \right\}^{\frac{1}{2}} \right] \quad (\text{Ecuación 18})$$

Ecuación algebraica en H y V.

Para la solución simultanea de estas ecuaciones, los métodos numéricos (Newton Raphson) son necesarios. Una vez resuelta la solución de H y V, obtenidos, pueden sustituirse en las soluciones T, x y z, de modo que la respuesta estructural del cable quede completamente caracterizada (Pallares M. y Rodríguez C., 2008).

Solución Matriz Jacobiana

La matriz Jacobiana puede calcularse analíticamente. Y los parámetros l, h, L_o, A_o, E, W son datos conocidos.

$$J(H, V) = \begin{Bmatrix} \frac{\partial f}{\partial H} & \frac{\partial f}{\partial V} \\ \frac{\partial g}{\partial H} & \frac{\partial g}{\partial V} \end{Bmatrix} \quad (\text{Ecuación 19})$$

Donde:

$$\frac{\partial f}{\partial H} = \frac{L_o}{E * A_o} + \frac{L}{W} * \left[\sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left(\frac{V-W}{H} \right) \right] + \frac{L}{W * H} \left[\frac{V-W}{\sqrt{1 + \left(\frac{V-W}{H} \right)^2}} - \frac{V}{\sqrt{1 + \left(\frac{V}{H} \right)^2}} \right] \quad (\text{Ecuación 20})$$

$$\frac{\partial f}{\partial V} = \frac{L}{W} \left[\left(1 + \left(\frac{V}{H} \right)^2 \right)^{-1/2} - \left(1 + \left(\frac{V-W}{H} \right)^2 \right)^{-1/2} \right] \quad (\text{Ecuación 21})$$

$$\frac{\partial g}{\partial H} = \frac{L}{W} \left[\left(1 + \left(\frac{V}{H} \right)^2 \right)^{1/2} - \left(1 + \left(\frac{V-W}{H} \right)^2 \right)^{1/2} \right] - \frac{L}{W * H^2} \left[\frac{V^2}{\left(1 + \left(\frac{V}{H} \right)^2 \right)^{1/2}} - \frac{(V-W)^2}{\left(1 + \left(\frac{V-W}{H} \right)^2 \right)^{1/2}} \right] \quad (\text{Ecuación 22})$$

$$\frac{\partial g}{\partial V} = \frac{L_o}{E * A_o} + \frac{L_o}{W * H} \left[\frac{V}{\left(1 + \left(\frac{V}{H} \right)^2 \right)^{1/2}} - \frac{(V-W)}{\left(1 + \left(\frac{V-W}{H} \right)^2 \right)^{1/2}} \right] \quad (\text{Ecuación 23})$$

Planteamiento de la solución numérica

Para el desarrollo de la solución, se realiza en tres etapas, basado en Pallares M. y Rodríguez C. (2008).

1) Planteamiento del método de Newton-Raphson, adaptado al problema de la catenaria.

Primer paso: definir el vector de incógnita x .

$$\bar{x} = \begin{Bmatrix} H \\ V \end{Bmatrix} \quad (\text{Ecuación 24})$$

Segundo paso: valores de inicio de acuerdo a la versión del problema de catenaria planteada.

$$V = \frac{W}{2} = x_2 \quad (\text{Ecuación 25})$$

$$H \approx \frac{W}{2\sqrt{6}} \frac{l}{L_o} \sqrt{\frac{l}{L_o - l}} = x_1 \quad (\text{Ecuación 26})$$

$$\bar{x}^0 = \begin{Bmatrix} H \\ V \end{Bmatrix} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \quad (\text{Ecuación 27})$$

Tercer paso: la matriz Jacobiana está definida por:

$$J(\bar{x}^k) = J \left(\begin{Bmatrix} H \\ V \end{Bmatrix}^k \right) = \begin{Bmatrix} \frac{\partial f}{\partial H} & \frac{\partial f}{\partial V} \\ \frac{\partial g}{\partial H} & \frac{\partial g}{\partial V} \end{Bmatrix} \quad (\text{Ecuación 28})$$

Cuarto paso: el vector de funciones no lineales.

$$\bar{f}(\bar{x}^k) = \begin{Bmatrix} f(H, V) - l \\ g(H, V) - h \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (\text{Ecuación 29})$$

Quinto paso: ecuación lineal clásica del método Newton Raphson.

$$\begin{Bmatrix} H \\ V \end{Bmatrix}^{k+1} = \begin{Bmatrix} H \\ V \end{Bmatrix}^k + \begin{Bmatrix} \Delta H \\ \Delta V \end{Bmatrix}^{k+1} \quad (\text{Ecuación 30})$$

Por supuesto la ecuación requiere de criterios de convergencia que permitan finalizar satisfactoriamente el proceso iterativo. Apartado II.

II) Solución explícita del sistema de ecuaciones lineales del algoritmo de Newton-Raphson.

Primera formulación

$$\Delta H = \frac{-f(H,V) + l - \frac{\delta f}{\delta V} \Delta V}{\frac{\delta f}{\delta H}} \quad (\text{Ecuación 31})$$

$$\Delta V = \frac{\frac{\delta g}{\delta H} f(H,V) - \frac{\delta g}{\delta H} l - \frac{\delta f}{\delta H} g(H,V) + \frac{\delta f}{\delta H} h}{-\frac{\delta f}{\delta V} \frac{\delta g}{\delta H} + \frac{\delta g}{\delta V} \frac{\delta f}{\delta H}} \quad (\text{Ecuación 32})$$

III) Criterios de convergencia de la solución.

Se establecen dos criterios de convergencia, al aplicar la norma euclídea a los vectores obtenidos. Como los vectores son solo de dos términos, la norma euclídea puede implementarse explícitamente.

$$\|\bar{x}\| = \sqrt{\sum_{i=1}^2 x_i^2} \quad (\text{Ecuación 33})$$

La formulación del primer criterio sobre el error relativo del vector solución es,

$$r^k = \frac{\|\bar{x}^{-k} - \bar{x}^{-k+1}\|}{\|\bar{x}^{-k+1}\|} \leq \text{tol}_x \quad (\text{Ecuación 34})$$

Donde, la tolerancia tol x debe escogerse adecuadamente lo cual depende de la precisión de la máquina y de las variables.

La formulación del segundo criterio de convergencia se aplica a la función de ecuaciones no lineales así:

$$\|\bar{f}(\bar{x}^{-k})\| \leq \text{tol}_f \quad (\text{Ecuación 35})$$

Donde, la tolerancia $\text{tol } x$ y $\text{tol } f$ debe escogerse adecuadamente lo cual depende de la precisión de la máquina y de las variables.

1.5.4.1. Catenaria elástica con respuesta a una carga puntual

Irvine (1981) supone, además, del peso propio del cable (sección 1.5.4.), una carga vertical concentrada F_1 , que cuelga a una distancia s_1 del perfil, con poco esfuerzo adicional, figura 20.

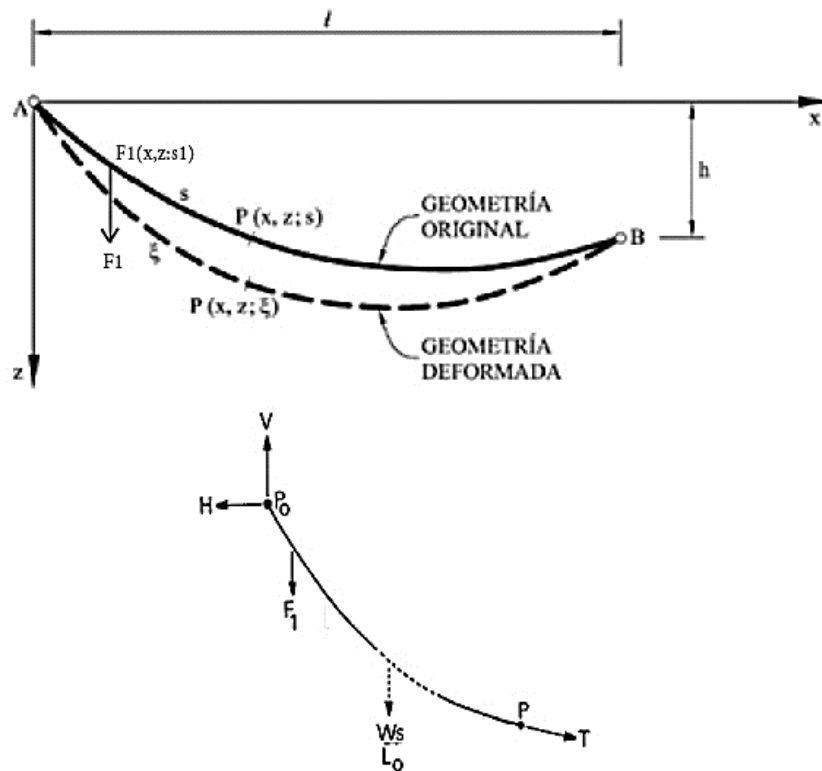


Figura 20. Equilibrio estático de un segmento del cable con una carga puntual.
Fuente: Irvine (1981)
Elaboración: Irvine (1981)

En la figura 20, el cable está suspendido entre dos puntos fijos A y B, que tienen coordenadas cartesianas $(0,0)$ y (l,h) respectivamente con una carga puntual en su longitud.

Donde:

F_1 : Es una carga puntual, que se encuentra a lo largo del cable en la posición s_1 .

s_1 : es la abscisa de la carga F_1 en la longitud del cable.

La primera ecuación (11) de equilibrio es inalterada, sección 1.5.4.

La segunda ecuación (12) de equilibrio es alterada por la carga vertical F_1 , sección 1.5.4.

Cuando, $0 \leq s \leq s_1$

$$T \frac{dz}{dp} = V - W \frac{s}{L_o}$$

Y para $s_1 \leq s \leq L_o$

$$T \frac{dz}{dp} = V - F_1 - W \frac{s}{L_o} \quad (\text{Ecuación 36})$$

Solución para T=T(s)

Cuando, $0 \leq s \leq s_1$

$$T(s) = \left\{ H^2 + \left(V - W \frac{s}{L_o} \right)^2 \right\}^{1/2}$$

Y para $s_1 \leq s \leq L_o$

$$T(s) = \left\{ H^2 + \left(V - F_1 - W \frac{s}{L_o} \right)^2 \right\}^{1/2} \quad (\text{Ecuación 37})$$

Solución para x=x(s)

Cuando, $0 \leq s \leq s_1$

$$x(s) = \frac{Hs}{EA_o} + \frac{HL_o}{EA_o} \left[\sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left\{ \frac{V - W \frac{s}{L_o}}{H} \right\} \right]$$

Y para $s_1 \leq s \leq L_o$

$$x(s) = \frac{Hs}{EA_o} + \frac{HL_o}{W} \left[\sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left\{ \frac{V - F_1 - W \frac{s}{L_o}}{H} \right\} + \sinh^{-1} \left\{ \frac{V - F_1 - W \frac{s_1}{L_o}}{H} \right\} - \sinh^{-1} \left\{ \frac{V - W \frac{s_1}{L_o}}{H} \right\} \right] \quad (\text{Ecuación 38})$$

Solución para z=z(s)

Cuando, $0 \leq s \leq s_1$

$$z(s) = \frac{Ws}{EA_o} \left(\frac{V}{W} - \frac{s}{2L_o} \right) + \frac{HL_o}{W} \left[\left\{ 1 + \left(\frac{V}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - W \frac{s}{L_o}}{H} \right)^2 \right\}^{1/2} \right]$$

Y para $s_1 \leq s \leq L_o$

$$z(s) = \frac{Ws}{EA_o} \left(\frac{V}{W} - \frac{s}{2L_o} \right) + \frac{HL_o}{W} \left[\left\{ 1 + \left(\frac{V}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - F_1 - W \frac{s}{L_o}}{H} \right)^2 \right\}^{1/2} + \frac{F_1}{H} \frac{W}{EA_o} \left(\frac{s_1}{L_o} - \frac{s}{L_o} \right) + \left\{ 1 + \left(\frac{V - F_1 - W \frac{s_1}{L_o}}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - W \frac{s_1}{L_o}}{H} \right)^2 \right\}^{1/2} \right]$$

(Ecuación 39)

Solución para H y V

Para las ecuaciones H y V pueden ser encontradas $x=l, z=h, s=L_o$. En la derivación de las soluciones para x y z, ecuación (38) y ecuación (39), respectivamente. Ecuación trascendental en H y V.

$$l = \frac{HL_o}{EA_o} + \frac{HL_o}{W} \left\{ \sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left(\frac{V - F - W}{H} \right) + \sinh^{-1} \left(\frac{V - F - W \frac{s_1}{L_o}}{H} \right) - \sinh^{-1} \left(\frac{V - W \frac{s_1}{L_o}}{H} \right) \right\}$$

(Ecuación 40)

$$z(s) = \frac{WL_o}{EA_o} \left(\frac{V}{W} - \frac{1}{2} \right) + \frac{HL_o}{W} \left[\left\{ 1 + \left(\frac{V}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - F_1 - W}{H} \right)^2 \right\}^{1/2} + \frac{F_1}{H} \frac{W}{EA_o} \left(\frac{s_1}{L_o} - 1 \right) + \left\{ 1 + \left(\frac{V - F_1 - W \frac{s_1}{L_o}}{H} \right)^2 \right\}^{1/2} - \left\{ 1 + \left(\frac{V - W \frac{s_1}{L_o}}{H} \right)^2 \right\}^{1/2} \right] \quad \text{(Ecuación 41)}$$

Para la solución simultanea de estas ecuaciones, los métodos numéricos (Newton Raphson) son necesarios. Una vez resuelta la solución de H y V, obtenidos, pueden sustituirse en las

soluciones T, x y z, de modo que la respuesta estructural del cable quede completamente caracterizada (Pallares M. y Rodríguez C., 2008).

Solución Matriz Jacobiana

La matriz Jacobiana puede calcularse analíticamente. Y los parámetros $l, h, L_o, A_o, E, W, s_1, F_1$ son datos conocidos.

$$J(H, V) = \begin{Bmatrix} \frac{\partial f}{\partial H} & \frac{\partial f}{\partial V} \\ \frac{\partial g}{\partial H} & \frac{\partial g}{\partial V} \end{Bmatrix}$$

Donde:

$$\begin{aligned} \frac{\partial f}{\partial H} = & \frac{L_o}{E * A_o} + \frac{L_o}{W} \left[\sinh^{-1} \left(\frac{V}{H} \right) - \sinh^{-1} \left(\frac{F - V + \frac{W * s_1}{L_o}}{H} \right) - \sinh^{-1} \left(\frac{V - \frac{W * s_1}{L_o}}{H} \right) \right. \\ & \left. + \sinh^{-1} \left(\frac{F - V + W}{H} \right) \right] \\ & + \frac{L_o}{W * H} \left[\frac{V - \frac{W * s_1}{L_o}}{\sqrt{1 + \left(\frac{V - \frac{W * s_1}{L_o}}{H} \right)^2}} - \frac{F - V + W}{\sqrt{1 + \left(\frac{F - V + W}{H} \right)^2}} - \frac{V}{\sqrt{1 + \left(\frac{V}{H} \right)^2}} + \frac{F - V + \frac{W * s_1}{L_o}}{\sqrt{1 + \left(\frac{F - V + \frac{W * s_1}{L_o}}{H} \right)^2}} \right] \end{aligned} \quad \text{(Ecuación 42)}$$

$$\frac{\partial f}{\partial H} = \frac{L_o}{W} \left[\left(1 + \left(\frac{F - V + W}{H} \right)^2 \right)^{-1/2} - \left(1 + \left(\frac{V}{H} \right)^2 \right)^{-1/2} \right. \\ \left. - \left(1 + \left(\frac{F - V + \frac{W * s_1}{L_o}}{H} \right)^2 \right)^{-1/2} + \left(1 + \left(\frac{V - \frac{W * s_1}{L_o}}{H} \right)^2 \right)^{-1/2} \right] \quad \text{(Ecuación 43)}$$

$$\frac{\partial g}{\partial H} = \frac{L_o}{W} \left[\sqrt{1 + \left(\frac{V}{H}\right)^2} + \sqrt{1 + \left(\frac{F - V + \frac{W * S}{L_o}}{H}\right)^2} - \sqrt{1 + \left(\frac{V - \frac{W * S}{L_o}}{H}\right)^2} \right]$$

$$- \frac{L_o}{W * H^2} \left[\frac{V^2}{\sqrt{1 + \left(\frac{V}{H}\right)^2}} - \frac{\left(V - \frac{W * S}{L_o}\right)^2}{\sqrt{1 + \left(\frac{V - \frac{W * S}{L_o}}{H}\right)^2}} - \frac{(F - V + W)^2}{\sqrt{1 + \left(\frac{F - V + W}{H}\right)^2}} \right]$$

$$+ \frac{\left(F - V + \frac{W * S}{L_o}\right)^2}{\sqrt{1 + \left(\frac{F - V + \frac{W * S}{L_o}}{H}\right)^2}} + \frac{F * W * H}{E * A_o} \left(\frac{S}{L_o} - 1\right)$$

(Ecuación 44)

$$\frac{\partial g}{\partial V} = \frac{L}{A_o * E} + \frac{L}{H * W} \left[\frac{V}{\sqrt{1 + \left(\frac{V}{H}\right)^2}} - \frac{\left(V - \frac{W * S}{L_o}\right)}{\sqrt{1 + \left(\frac{V - \frac{W * S}{L_o}}{H}\right)^2}} \right]$$

$$+ \frac{(F - V + W)}{\sqrt{1 + \left(\frac{F - V + W}{H}\right)^2}} - \frac{\left(F - V + \frac{W * S}{L_o}\right)}{\sqrt{1 + \left(\frac{F - V + \frac{W * S}{L_o}}{H}\right)^2}}$$

(Ecuación 45)

Con respecto al Planteamiento de la solución numérica, se realiza igual que la Catenaria Elástica, se debe seguir los mismos pasos.

La escritura de soluciones se hará progresivamente más tediosa a medida que aumenta el número de cargas puntuales aplicadas. En la sección 1.5.4.2. se presenta con los resultados generales.

1.5.4.2. Catenaria elástica con respuesta a dos o más cargas puntuales

Irvine (1981) propone una solución para la catenaria elástica con varias cargas puntuales a lo largo de la longitud del cable.

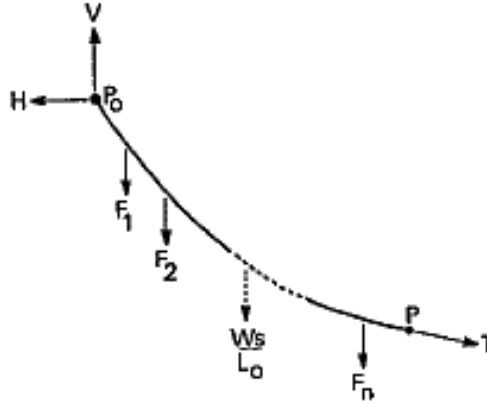


Figura 21. Equilibrio estático de un segmento del cable.
Fuente: Irvine (1981)
Elaboración: Irvine (1981)

La figura 21, proporciona la afirmación del equilibrio vertical en un punto P sobre el perfil tensado.

$$T \frac{dz}{dp} = V - \sum_{i=0}^n F_i - W \frac{s}{L_o} \quad (\text{Ecuación 46})$$

Donde:

n : se elige de manera que P se encuentre entre P_n y P_{n+1} , para $n = 0, 1, \dots, N$.

Se aplica N cargas puntuales, y F_o , se define como $F_o = 0$, de modo que los resultados generales incluirán el caso del cable solo bajo peso propio (cuando $N = 0$, la catenaria elástica). La afirmación del equilibrio horizontal es la misma que antes. El extremo del cable ahora se etiqueta convenientemente P_n y P_{n+1} , y las condiciones finales.

Con vistas a una solución concisa, se introducen las siguientes cantidades adimensionales:

$\xi = \frac{x}{L_o}$ Coordenada horizontal del perfil tensado

$\eta = \frac{z}{L_o}$ Coordenada vertical del perfil tensado

$\sigma = \frac{s}{L_o}$ Coordenada lagrangiana del perfil no estirado

$\sigma_n = \frac{s_n}{L_o}$ Coordenada de carga lagrangiana ($n = 1, 2, \dots, N$)

$\delta = \frac{h}{L_o}$ Desplazamiento vertical relativo de los puntos extremos,

$\gamma = \frac{l}{L_o}$ Relación de aspecto del cable

$\tau = \frac{T}{W}$ Tensión del cable

$\chi = \frac{H}{W}$ Reacción horizontal en soportes P_o, P_{n+1}

$\phi = \frac{V}{W}$ Reacción vertical en apoyo P_o

$\psi_n = \frac{F_n}{W}$ Aplicaron cargas verticales concentradas ($n = 1, 2, \dots, N$)

$\beta = \frac{W}{EA_o}$ Factor de flexibilidad

Definimos $\sigma_0 = 0, \psi_{-1} = 0$, y $\psi_0 = 0$, y utilizamos la suma parcial

$$\psi_n = \sum_{j=1}^n \psi_j$$

Así que eso $\psi_{-1} = 0$, al igual que ψ_0 . Por lo tanto, tenemos:

 **Solución para la tensión del cable** $\tau = \tau(\sigma)$

$$\tau(\sigma) = \left\{ \chi^2 + (\phi - \psi_n - \sigma)^2 \right\}^{1/2} \quad (\text{Ecuación 47})$$

Que se aferra a $\sigma_n < \sigma < \sigma_{n+1}$ para $n = 0, 1, \dots, N$.

 **Solución para la coordenada horizontal del perfil** $\xi = \xi(\sigma)$

$$\xi(\sigma) = \chi \left[\begin{aligned} &\beta \alpha + \sinh^{-1} \left(\frac{\phi}{\chi} \right) - \sinh^{-1} \left(\frac{\phi - \psi_n - \sigma}{\chi} \right) \\ &+ \sum_{i=0}^n \left\{ \sinh^{-1} \left(\frac{\phi - \psi_i - \sigma_i}{\chi} \right) - \sinh^{-1} \left(\frac{\phi - \psi_{i-1} - \sigma_i}{\chi} \right) \right\} \end{aligned} \right] \quad (\text{Ecuación 48})$$

Que se sostiene en $\sigma_n < \sigma < \sigma_{n+1}$ para $n = 0, 1, \dots, N$.

✚ Solución para la coordenada vertical del perfil $\eta = \eta(\sigma)$

$$\eta(\sigma) = \beta\sigma\left(\phi - \frac{\sigma}{2}\right) + \left\{\chi^2 + \phi^2\right\}^{1/2} - \left\{\chi^2 + (\phi - \psi_n - \sigma)^2\right\}^{1/2} \quad (Ecuación 49)$$

$$+ \sum_{i=0}^n \left[\beta\psi_i(\sigma_i - \sigma) + \left\{\chi^2 + (\phi - \psi_i - \sigma_i)^2\right\}^{1/2} - \left\{\chi^2 + (\phi - \psi_{i-1} - \sigma_i)^2\right\}^{1/2} \right]$$

Que también se mantiene en $\sigma_n < \sigma < \sigma_{n+1}$ para $n = 0, 1, \dots, N$.

✚ Solución para χ y ϕ

$$\frac{\gamma}{\chi} - \beta = \sinh^{-1}\left(\frac{\phi}{\chi}\right) - \sinh^{-1}\left(\frac{\phi - \psi_N - 1}{\chi}\right) + \sum_{i=0}^N \left\{ \sinh^{-1}\left(\frac{\phi - \psi_i - \sigma_i}{\chi}\right) - \sinh^{-1}\left(\frac{\phi - \psi_{i-1} - \sigma_i}{\chi}\right) \right\} \quad (Ecuación 50)$$

$$\delta = \beta\left(\sigma - \frac{1}{2}\right) + \left(\chi^2 + \phi^2\right)^{1/2} - \left\{\chi^2 + (\phi - \psi_N - 1)^2\right\}^{1/2} + \sum_{i=0}^N \left[\beta\psi_i(\sigma_i - 1) + \left\{\chi^2 + (\phi - \psi_i - \sigma_i)^2\right\}^{1/2} - \left\{\chi^2 + (\phi - \psi_{i-1} - \sigma_i)^2\right\}^{1/2} \right] \quad (Ecuación 51)$$

Los valores de estos dos parámetros pueden determinarse a través del método de newton bidimensional modificado. Las ecuaciones (50) y (51) constituyen una solución exacta para la respuesta estática de un cable elástico bajo su propio peso y N cargas concentradas.

Una vez resuelta la solución de χ y ϕ , obtenidos, pueden sustituirse en las soluciones T, x y z, después de que, dejen de ser adimensionales, de modo que la respuesta estructural del cable quede completamente caracterizada (Pallares M. y Rodríguez C., 2008).

✚ Solución Matriz Jacobiana

La matriz Jacobiana puede calcularse analíticamente. Y los parámetros son datos conocidos.

$l, h, L_o, A_o, E, W, (s_1, s_2, \dots, s_N), (F_1, F_2, \dots, F_N)$

$$J(H, V) = \begin{Bmatrix} \frac{\partial f}{\partial H} & \frac{\partial f}{\partial V} \\ \frac{\partial g}{\partial H} & \frac{\partial g}{\partial V} \end{Bmatrix}$$

Donde:

$$\frac{\partial \gamma}{\partial \chi} = \beta + \sinh^{-1}\left(\frac{\phi}{\chi}\right) + \sinh^{-1}\left(\frac{\psi_N - \phi + 1}{\chi}\right) + \sum_{i=0}^N \left\{ \begin{array}{l} \sinh^{-1}\left(\frac{\psi_{i-1} - \phi + \sigma_i}{\chi}\right) \\ - \sinh^{-1}\left(\frac{\sigma_i - \phi - \psi_i}{\chi}\right) \end{array} \right\}$$

$$- \frac{1}{\chi} \left\{ \begin{array}{l} \left[\sum_{i=0}^N \frac{\psi_{i-1} - \phi + \sigma_i}{\sqrt{1 + \left(\frac{\psi_{i-1} - \phi + \sigma_i}{\chi}\right)^2}} \right] - \left[\sum_{i=0}^N \frac{\sigma_i - \phi - \psi_i}{\sqrt{1 + \left(\frac{\sigma_i - \phi - \psi_i}{\chi}\right)^2}} \right] \\ + \left[\frac{\psi_N - \phi + 1}{\sqrt{1 + \left(\frac{\psi_N - \phi + 1}{\chi}\right)^2}} \right] + \left[\frac{\phi}{\sqrt{1 + \left(\frac{\phi}{\chi}\right)^2}} \right] \end{array} \right\}$$

(Ecuación 52)

$$\frac{\partial \gamma}{\partial \phi} = - \left\{ \begin{array}{l} \left[\sum_{i=0}^N \frac{1}{\sqrt{1 + \left(\frac{\psi_{i-1} - \phi + \sigma_i}{\chi}\right)^2}} \right] - \left[\sum_{i=0}^N \frac{1}{\sqrt{1 + \left(\frac{\sigma_i - \phi + \psi_i}{\chi}\right)^2}} \right] \\ + \left[\frac{1}{\sqrt{1 + \left(\frac{\psi_N - \phi + 1}{\chi}\right)^2}} \right] - \left[\frac{1}{\sqrt{1 + \left(\frac{\phi}{\chi}\right)^2}} \right] \end{array} \right\}$$

(Ecuación 53)

$$\frac{\partial \delta}{\partial \chi} = \left[\sum_{i=0}^N \frac{\chi}{\sqrt{\chi^2 + (\sigma_i - \phi + \psi_i)^2}} \right] - \left[\sum_{i=0}^N \frac{\chi}{\sqrt{\chi^2 + (\psi_{i-1} - \phi + \sigma_i)^2}} \right]$$

$$- \left[\frac{\chi}{\sqrt{\chi^2 + (\psi_N - \phi + 1)^2}} \right] + \left[\frac{\chi}{\sqrt{\chi^2 + \phi^2}} \right]$$

(Ecuación 54)

$$\frac{\partial \delta}{\partial \phi} = \beta - \left[\sum_{i=0}^N \frac{\phi}{\sqrt{\chi^2 + (\psi_{i-1} - \phi + \sigma_i)^2}} \right] + \left[\frac{\psi_N - \phi + 1}{\sqrt{\chi^2 + (\psi_N - \phi + 1)^2}} \right] + \left[\sum_{i=0}^N \frac{\psi_{i-1}}{\sqrt{\chi^2 + (\psi_{i-1} - \phi + \sigma_i)^2}} \right]$$

$$- \left[\sum_{i=0}^N \frac{\sigma_i}{\sqrt{\chi^2 + (\sigma_i - \phi + \psi_i)^2}} \right] - \left[\sum_{i=0}^N \frac{\psi_i}{\sqrt{\chi^2 + (\sigma_i - \phi + \psi_i)^2}} \right] + \left[\sum_{i=0}^N \frac{\phi}{\sqrt{\chi^2 + (\sigma_i - \phi + \psi_i)^2}} \right]$$

$$+ \left[\frac{\phi}{\sqrt{\chi^2 + \phi^2}} \right] + \left[\sum_{i=0}^N \frac{\sigma_i}{\sqrt{\chi^2 + (\psi_{i-1} - \phi + \sigma_i)^2}} \right]$$

(Ecuación 55)

CAPITULO II
2. METODOLOGÍA

2.1. Generalidades

Para el desarrollo de la presente investigación es necesario hacer énfasis en la Arquitectura del Sistema, lo cual involucra la combinación de varias etapas, presentadas en el siguiente esquema:

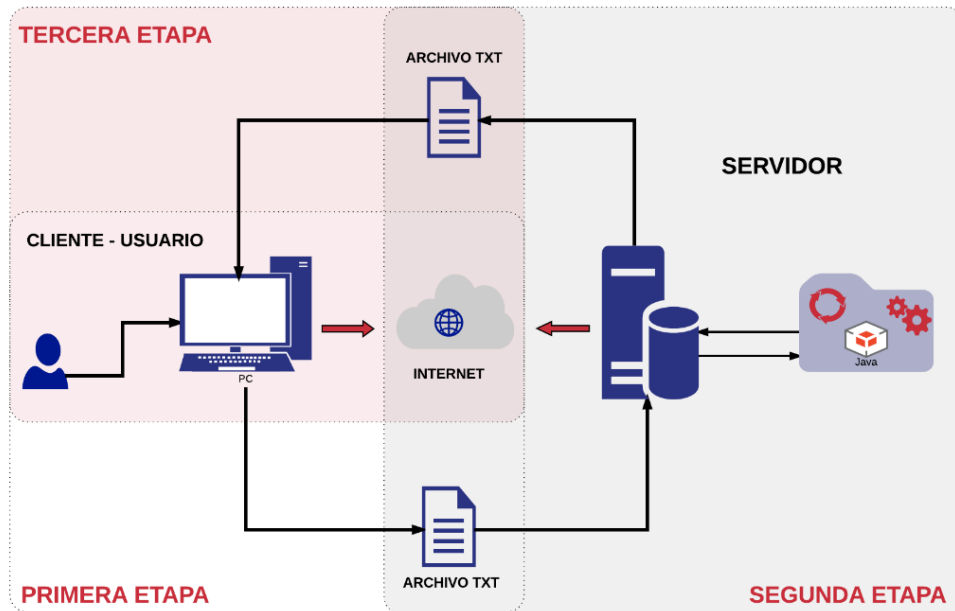


Figura 22. Esquema de la Arquitectura del Sistema
Fuente: Serrano, P., y Castro, D. (2008).
Elaboración: Autor

Primera Etapa: El cliente interactúa con la plataforma virtual del GRISS-UTPL correspondiente al Laboratorio de Cables.

La interfaz gráfica de usuario presenta al cliente los experimentos disponibles (experimento 1, 2, 3, ..., n), el cliente configura el experimento y al momento de ejecutar solicita la información. Por medio de la Tecnología ASP, se permite la ejecución de la configuración, se transforma en un archivo de texto (.TXT) los datos de entrada del experimento y se envía el archivo al servidor.

Segunda etapa: La gestión del Servidor.

El servidor recibe el archivo de texto, se guarda en la base de datos de VLEE, específicamente en la carpeta del Laboratorio de Cables. En el servidor es lo que se genera el análisis para dicho experimento por medio del Código Fuente (Lenguaje Basic). El código fuente está estructurado para que lea el archivo de texto, realice el análisis matemático y presente los datos de salida, se transforma en un archivo de texto.

Tercera etapa: El cliente Visualiza en la Plataforma Virtual los resultados.

La página web recibe el archivo de texto y lo transforma en un formato entendible para la interfaz gráfica de usuario. Y el cliente visualizará los resultados (datos, gráficas, ..., etc.) en la plataforma virtual.

Para alcanzar el objetivo, el proyecto de estudio corresponde a partir de la segunda etapa, en la realización de Código fuente en el Lenguaje Basic para cada uno de los experimentos del Laboratorio de Cables.

2.2. Elaboración de Herramientas

2.2.1. Laboratorio de “Cables”

En este Laboratorio el usuario encontrará experimentos que le permitan analizar el comportamiento no lineal geométrico de estructuras de cables, además, simular la respuesta estructural del cable suspendido entre dos soportes rígidos bajo el efecto de su propio peso, cargas verticales, concentradas y/o distribuidas.

El laboratorio de Cables cuenta hasta la actualidad con los siguientes experimentos:

- Experimento 1: Cable Parabólico
- Experimento 2: Catenaria
- Experimento 3: Cable con Cargas Puntuales
- Experimento 4: Catenaria Elástica

2.2.1.1. Programa Master del Laboratorio de “Cables”.

El programa master del laboratorio de cables, hace referencia, a la función *ProgramaMasterLaboratorioCables*, es la herramienta líder para la gestión del laboratorio de cables. Al solicitar la función, el proceso es el siguiente:

- Se inicia la tarea, el control es de la función.
- Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada.
- Identifica la función que corresponde al experimento ejecutado.
- La función del experimento ejecutado realiza su proceso específico.
- Identifica los archivos de texto generados por los experimentos, los cuales, corresponden a los datos de salida (resultados).

outputExperimento.txt: es un archivo de texto, que contiene los resultados.

Estos datos están destinados para la interfaz gráfica y el usuario lo visualizara.

outputUser.txt: es un archivo de texto, que contiene los resultados en el idioma español. Estos datos están destinados para el usuario, a través de un botón en la interfaz gráfica del experimento se descargarán los resultados.

outputUserEnglish.txt: A diferencia del archivo de texto anterior, los datos serán en el idioma inglés.

- Se finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *Servidor*.

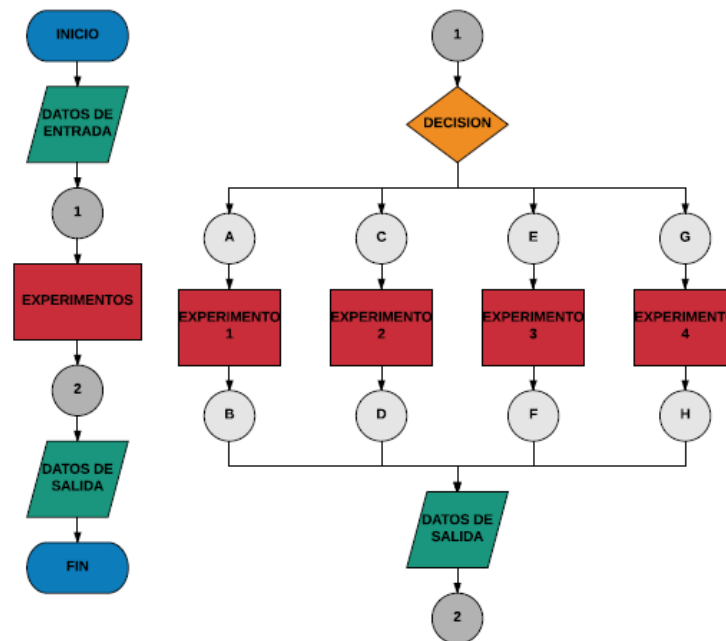


Figura 23. Diagrama de flujo del Programa Master del Laboratorio de Cables.

Fuente: Autor

Elaborado por: Autor

En la figura 23, se muestra el proceso de la función *ProgramaMasterLaboratorioCables* (figura 23-A). Entre el conector 1 y conector 2, existe un proceso que corresponde a la ejecución de los experimentos (figura 23-B); en donde, se identifica el experimento ejecutado con su respectiva función y cada uno tiene sus conectores de enlace que almacenan el proceso de análisis de dicho experimento a través de otras funciones.

Tabla 1. Nombre de la Funciones que corresponden a los experimentos.

Número del Experimento	Nombre del Experimento	Nombre de la Función	Conector
Experimento 1	Cable - Parabólico	ProgramaMasterCableParabolico	A-B
Experimento 2	Cable - Catenaria	ProgramaMasterCableCatenaria	C-D
Experimento 3	Cable - Cargas puntuales	ProgramaMasterCableCargasPuntuales	E-F
Experimento 4	Cacle - Catenaria Elástica	ProgramaMasterCableCatenariaElastica	G-H

Fuente: Autor

Elaborado por: Autor

2.2.1.1.1. Código Master del Laboratorio de "Cables".

Los datos de entrada corresponden:

Ruta: La variable contiene la dirección donde se encuentra el archivo input.txt

NameFileUser: nombre del archivo de texto de resultados para el usuario.

NameFileExperimento: nombre del archivo de texto de resultados de para el servidor.

NameFileUserEnglish: nombre del archivo de texto de resultados para el usuario en el idioma inglés.

Los datos de salida corresponden a los archivos de texto

NameFileUser: La variable contiene el archivo de texto de resultados para el usuario (nameFileUser.txt).

NameFileExperimento: La variable contiene el archivo de texto de resultados de para el servidor (NameFileExperimento.txt).

NameFileUserEnglish: La variable contiene el archivo de texto de resultados para el usuario en el idioma inglés (nameFileUserEnglish.txt).

```
C:\Users\DELL i7\...\ProgramaMasterLaboratorioCables.m Page
1
3 de agosto de 2017 19:14:25
1 function [] = ProgramaMasterLaboratorioCables(ruta)
2     ruta='C:\Users\DELL i7\Desktop\ProgramaMasterCables\MASTER\';
3     nombreArchivo='input.txt';
4     nameFileUser = 'outputUser.txt';
5     nameFileExperimento = 'outputExperimento.txt';
6     nameFileUserEnglish = 'outputUserEnglish.txt';
7     fid = fopen([ruta nombreArchivo ], 'r');
8     formato='%f32';
9     ma=textscan(fid,formato);
10    c1=ma{1}; % c1=columna #1 de la matriz ma
11
12    %%
13    switch c1(1)
14        case 1 %Experimento 1: Cable Parabólico
15            ProgramaMasterCableParabolico(ruta,nameFileUser,nameFileExperimento,
16            nameFileUserEnglish)
17        case 2 %Experimento 2: Catenaria
18            ProgramaMasterCableCatenaria(ruta,nameFileUser,nameFileExperimento,
19            nameFileUserEnglish)
20        case 3 %Experimento 3: Cable con Cargas Discretas
21            ProgramaMasterCableCargasPuntuales(ruta,nameFileUser,
22            nameFileExperimento,nameFileUserEnglish)
23        case 4 %Experimento 4: Catenaria Elástica
24            ProgramaMasterCablesCatenariaElastica(ruta,nameFileUser,
25            nameFileExperimento,nameFileUserEnglish)
26    end
27    %Cierra todas las ventanas
28    fclose(fid);
29    end
```

Figura 24. Código master del Laboratorio de Cables.

Fuente: Autor

Elaboración: Autor

2.2.2. Experimento 1: “Cable Parabólico”.

En el Análisis estático de Cables se considera una carga uniformemente distribuida a lo largo de una línea recta, se ignora el peso del cable y la sección transversal. Con el objetivo de obtener la respuesta de un cable suspendido entre dos soportes rígidos con su deformada, la variación de tensión en toda su longitud y las reacciones en los apoyos. El origen del sistema coordenado está en el punto más bajo del cable.

Tabla 2. Datos de Entrada y de Salida del experimento “Cable Parabólico”.

DESCRIPCIÓN	SÍMBOLO	UNIDAD
DATOS DE ENTRADA		
Altura o flecha del punto inicial (O)	Ho	m
Altura o flecha del punto final (F)	Hf	m
Vano o distancia horizontal entre el punto inicial (O) y el punto final (F)	L	m
Carga distribuida a lo largo de una línea recta	Wo	KN/m
Distancia entre tensores	DT	m
DATOS DE SALIDA (RESULTADOS)		
Tensión mínima del cable	Tmin	KN
Tensión del cable en el punto inicial (O)	To	KN
Tensión del cable en el punto final (F)	Tf	KN
Tensión máxima del cable	Tmax	KN
Tensión en los tensores	Tten	KN
Número de tensores	Nten	#
Longitud del cable	S	m
Distancia horizontal o vano entre el punto inicial (O) y el eje	Lo	m
Distancia horizontal o vano entre el punto final (F) y el eje	Lf	m
Distancia horizontal entre el punto inicial (O) y el primer tensor	DTto	m
Distancia horizontal entre el punto final (F) y el ultimo tensor	DTf	m

Fuente: Autor

Elaboración: Autor

Por la variación de nivel entre los dos soportes rígidos, existen tres casos:

Caso 1: $H_o < H_f$

Caso 2: $H_o > H_f$

Caso 3: $H_o = H_f$

2.2.2.1. Programa Master del experimento “Cable Parabólico”.

El programa master del experimento “Cable Parabólico”, hace referencia, a la función *ProgramaMasterCableParabolico*, es la herramienta líder para la gestión del experimento. Al solicitar la función, el proceso es el siguiente:

- Se inicia la tarea, el control es de la función del experimento.
- Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada.
- Identifica la función que corresponde al caso (Caso1,2,3) que sea ejecutado.
- La función del caso ejecutado realiza su proceso específico.
- Identifica los archivos de texto generados por el caso ejecutado, lo cual, corresponde a los datos de salida (resultados).

outputExperimento.txt

outputUser.txt

outputUserEnglish.txt

- Se finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, función *ProgramaMasterLaboratorioCables*

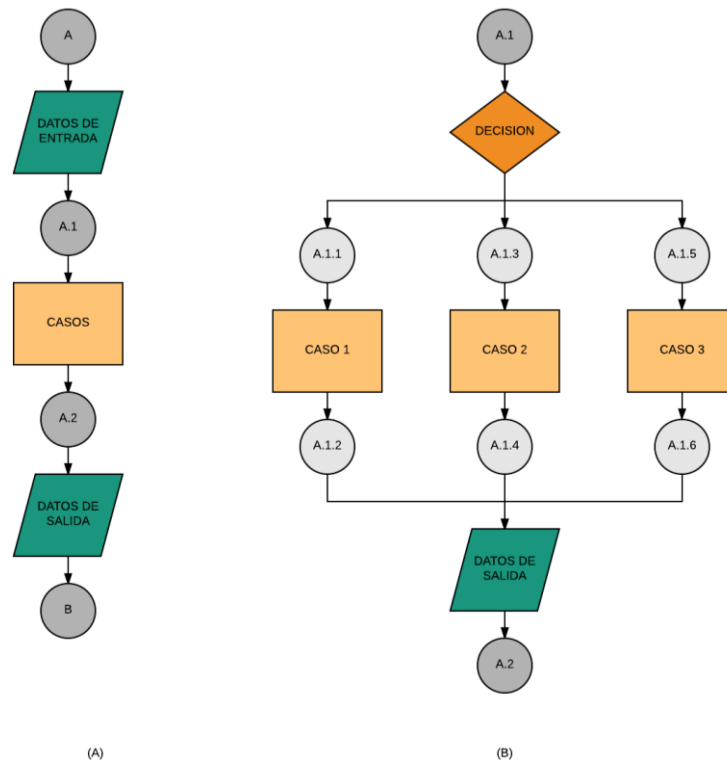


Figura 25. Diagrama de flujo del Programa Master del experimento “Cable Parabólico”

Fuente: Autor

Elaboración: Autor

En la figura 25, se muestra el proceso de la función *ProgramaMasterCableParabolico* (figura 25-A). Entre el conector A.1 y conector A.2, existe un proceso que corresponde a la ejecución de los casos (figura 25-B); en donde, se identifica el caso ejecutado con su respectiva función y cada uno tiene sus conectores de enlace que almacenan el proceso de análisis de dicho caso a través de otras funciones.

Tabla 3. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable Parabólico”

Número de Caso	Nombre del Experimento	Nombre de la Función	Conector
Caso 1	Caso 1: $H_o < H_F$	LeerCableParabCaso1 CableParabCaso1 SalidaCableParabCaso1	A.1.1 - A.1.2
Caso 2	Caso 2: $H_o > H_F$	LeerCableParabCaso2 CableParabCaso2 SalidaCableParabCaso2	A.1.3 - A.1.4
Caso 3	Caso 3: $H_o = H_F$	LeerCableParabCaso3 CableParabCaso3 SalidaCableParabCaso3	A.1.5 - A.1.6

Fuente: Autor

Elaboración: Autor

2.2.2.1.1. Código Master del experimento “Cable Parabólico”.

```
C:\Users\DELL i7\De...\ProgramaMasterCableParabolico.m Page 1

1 function [] =
ProgramaMasterCableParabolico(ruta,nameFileUser,nameFileExperimento,
nameFileUserEnglish)
2 %Experimento 1: Cable Parabólico
3 switch cl(2)
4     case 1 %Caso 1: Ho<Hf
5         [Ho,Hf,L,Wo,DT] = LeerCableParabCaso1(ruta, nombreArchivo);
6         [Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf, DTo, DTf,
str, x, y, A, B, D, wx, wy] = CableParabCaso1(Ho,Hf,L,Wo,DT);
7         SalidaCableParabCaso1(Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten,
S, Lo, Lf, DTo, DTf, str, x, y, A, B, D, wx, wy, ruta, nameFileUser,
nameFileExperimento, nameFileUserEnglish);
8     case 2 %Caso 2: Ho>Hf
9         [Ho,Hf,L,Wo,DT] = LeerCableParabCaso2 (ruta, nombreArchivo);
10        [Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf, DTo, DTf,
str, x, y, A, B, D, wx, wy] = CableParabCaso2(Ho,Hf,L,Wo,DT);
11        SalidaCableParabCaso2(Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten,
S, Lo, Lf, DTo, DTf, str, x, y, A, B, D, wx, wy, ruta, nameFileUser,
nameFileExperimento, nameFileUserEnglish);
12    case 3 %Caso 3: Ho=Hf
13        [H,L,Wo,DT] = LeerCableParabCaso3 (ruta, nombreArchivo);
14        [H,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf, DTo, DTf,
str, x, y, A, B, D, wx, wy] = CableParabCaso3(H,L,Wo,DT);
15        SalidaCableParabCaso3(H,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S,
Lo, Lf, DTo, DTf, str, x, y, A, B, D, wx, wy, ruta, nameFileUser,
nameFileExperimento, nameFileUserEnglish);
16 end
17 %Cierra todas las ventanas
18 fclose(fid);
19 end
```

Figura 26. Código master del experimento 1 – Cable parabólico.

Fuente: Autor

Elaboración: Autor

2.2.2.2. Caso1: $H_o < H_f$.

Un cable suspendido entre dos soportes rígidos a distinto nivel. La flecha en el punto inicial (O) es MENOR a la flecha en el punto final (F), $H_o < H_f$, (Figura 27).

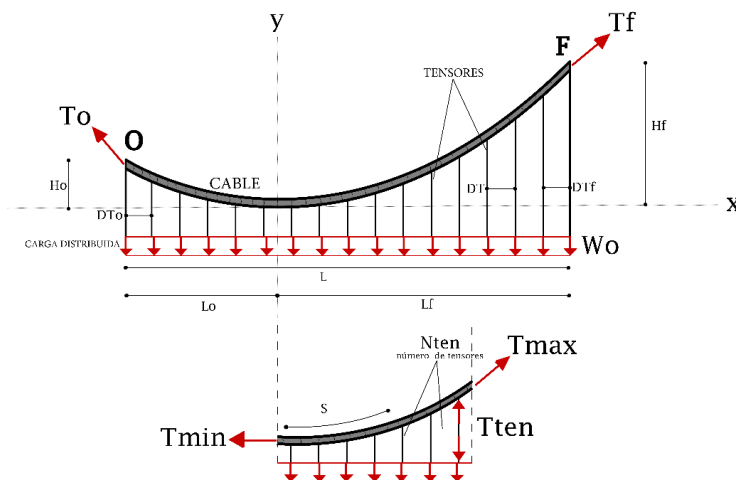


Figura 27. Esquema del experimento “Cable Parabólico” – Caso 1

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 28):

- Se inicia la tarea con la función *LeerCableParabCaso1*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableParabCaso1*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableParabCaso1*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableParabolico*.

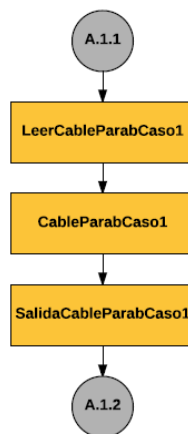


Figura 28. Diagrama de flujo de las funciones del caso 1 del experimento "Cable Parabólico"

Fuente: Autor

Elaboración: Autor

2.2.2.3. Caso2: $H_o > H_f$.

Un cable suspendido entre dos soportes rígidos a distinto nivel. La flecha en el punto inicial (O) es MAYOR a la flecha en el punto final (F), $H_o > H_f$.

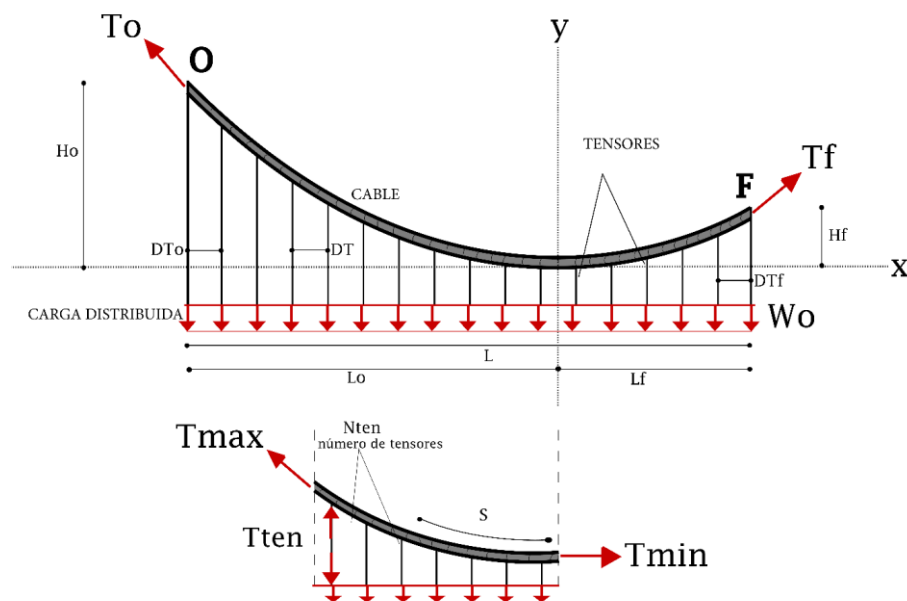


Figura 29. Esquema del experimento “Cable Parabólico” – Caso 2

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso2, contienen líneas de códigos que realizan una tarea específica (figura 30):

- Se inicia la tarea con la función *LeerCableParabCaso2*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableParabCaso2*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableParabCaso2*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableParabolico*.

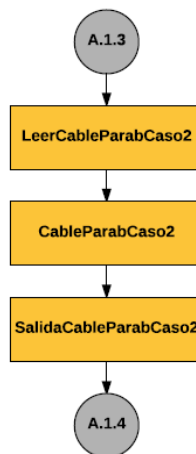


Figura 30. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Parabólico”

Fuente: Autor

Elaboración: Autor

2.2.2.4. Caso3: $H_o = H_f$.

Un cable suspendido entre dos soportes rígidos al mismo nivel. La flecha en el punto inicial (O) es IGUAL a la flecha en el punto final (F), $H_o = H_f$.

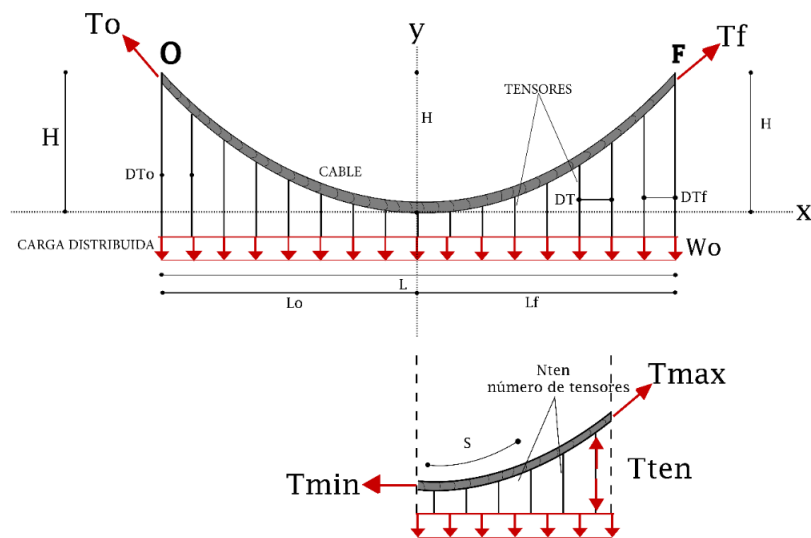


Figura 31. Esquema del experimento “Cable Parabólico” – Caso 3

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso3, contienen líneas de códigos que realizan una tarea específica (figura 32):

- Se inicia la tarea con la función *LeerCableParabCaso3*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableParabCaso3*

- Recibe las variables de datos de entrada

Realiza el análisis matemático del experimento

Devuelve las variables de datos de salida
- Después, con la función *SalidaCableParabCaso3*

Recibe las variables de datos de salida

Genera los archivos de texto correspondiente a los datos de salida (resultados).

outputExperimento.txt

outputUser.txt

outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableParabolico*.

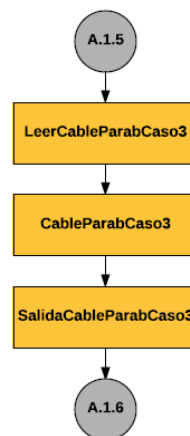


Figura 32. Diagrama de flujo de las funciones del caso 3 del experimento “Cable Parabólico”
 Fuente: Autor
 Elaboración: Autor

2.2.3. Experimento 2: “Cable Catenaria”.

En el análisis estático de cables se considera el peso de un cable uniformemente distribuida a lo largo del cable, se ignora la sección transversal. Para obtener la respuesta de un cable suspendido entre dos soportes rígidos como su deformada, la variación de tensión en toda su longitud y las reacciones en los apoyos. El origen del sistema coordenado está en el punto más bajo del cable. Y el peso del cable no se desprecia.

Por la configuración de datos disponibles, flecha o la distancia horizontal, existen dos casos:

- Caso 1: Flecha
- Caso 2: Longitud

Tabla 4. Datos de Entrada y de Salida del experimento “Cable Catenaria”.

DESCRIPCIÓN	SÍMBOLO	UNIDAD
DATOS DE ENTRADA		
CASO 1: FLECHA		
Esfuerzo de ruptura	ER	N
Factor de seguridad	FS	#
Carga distribuida	Wo	Kg/m
Altura del soporte (O)	Ho	m
Altura del soporte (F)	Hf	m
Cota	Hm	m
Cota	b	m
CASO 2: LONGITUD		
Esfuerzo de ruptura	ER	N
Factor de seguridad	FS	#
Carga distribuida	Wo	Kg/m
Vano o distancia horizontal entre el punto inicial (O) y el punto final (F)	L	m
Cota	Ym	m
DATOS DE SALIDA (RESULTADOS)		
Vano o distancia horizontal entre el punto inicial (O) y el punto final (F)	L	m
Distancia horizontal o vano entre el punto inicial (O) y el eje	Lo	m
Distancia horizontal o vano entre el punto final (F) y el eje	Lf	m
Altura o flecha del punto inicial (O)	Yo	m
Altura o flecha del punto final (F)	Yf	m
Tensión del cable en el punto inicial (O)	To	KN
Tensión del cable en el punto final (F)	Tf	KN
Tensión máxima que soporta el cable	TmaxCable	KN
Tensión máxima del cable	Tmax	KN
Tensión mínima del cable	Tmin	KN
Longitud del cable	S	m

Fuente: Autor

Elaboración: Autor

2.2.3.1. Programa Master del experimento “Cable Catenaria”.

El programa master del experimento “Cable Catenaria”, hace referencia, a la función *ProgramaMasterCableCatenaria*, es la herramienta líder para la gestión del experimento. Al solicitar la función, el proceso es el siguiente:

- Se inicia la tarea, el control es de la función del experimento.
- Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada.
- Identifica la función que corresponde al caso (Caso1 y 2) que sea ejecutado.
- La función del caso ejecutado realiza su proceso específico.
- Identifica los archivos de texto generados por el caso ejecutado, lo cual, corresponde a los datos de salida (resultados).

outputExperimento.txt

outputUser.txt

outputUserEnglish.txt

- Se finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, función *ProgramaMasterLaboratorioCables*

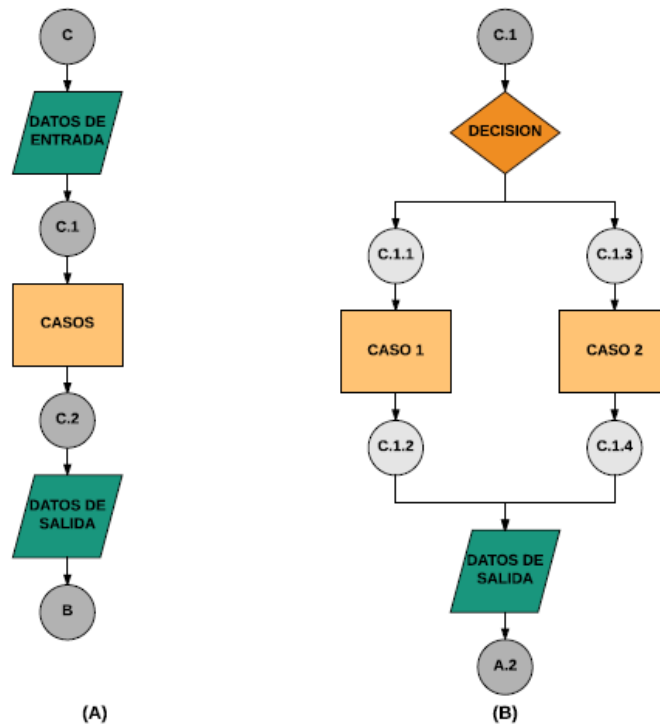


Figura 33. Diagrama de flujo del Programa Master del experimento “Cable Catenaria”
Fuente: Autor
Elaboración: Autor

En la figura33, se muestra el proceso de la función *ProgramaMasterCableCatenaria* (figura 33-A). Entre el conector C.1 y conector C.2, existe un proceso que corresponde a la ejecución de los casos (figura 33-B); en donde, se identifica el caso ejecutado con su respectiva función y cada uno tiene sus conectores de enlace que almacenan el proceso de análisis de dicho caso a través de otras funciones.

Tabla 5. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable Catenaria”

Número de Caso	Nombre del Experimento	Nombre de la Función	Conector
Caso 1	Caso 1: Flecha	LeerCableCatenCaso1 CableCatenCaso1 SalidaCableCatenCaso1	C.1.1 - C.1.2
Caso 2	Caso 2: Longitud	LeerCableCatenCaso2 CableCatenCaso2 SalidaCableCatenCaso2	C.1.3 - C.1.4

Fuente: Autor
Elaboración: Autor

2.2.3.1.1. Código Master del experimento “Cable Catenaria”.

```
C:\Users\DELL i7\Des...\ProgramaMasterCableCatenaria.m Page
1
3 de agosto de 2017 10:06:43
1 function [] =
ProgramaMasterCableCatenaria(ruta,nameFileUser,nameFileExperimento,
nameFileUserEnglish)
2 %Experimento 1: Cable Catenaria
3 switch c1(1)
4     case 1 %Caso 1: Flecha
5         [FS,ER,Wo,Ho,Hf,Hm,b] = LeerCableCatenCaso1(ruta, nombreArchivo);
6
[FS,ER,Wo,Ho,Hf,Hm,b,Yo,Yf,L,Lo,Lf,To,Tf,Th,TmaxCable,Tmax,Tmin,S,a,wx,wy,
x,y,str] = CableCatenCaso1(FS,ER,Wo,Ho,Hf,Hm,b);
7
SalidaCableCatenCaso1(FS,ER,Wo,Ho,Hf,Hm,b,Yo,Yf,L,Lo,Lf,To,Tf,Th,TmaxCable,
Tmax,Tmin,S,a,wx,wy,x,y,str, ruta, nameFileUser, nameFileExperimento,
nameFileUserEnglish);
8     case 2 %Caso 2: Longitud
9         [FS,ER,Wo,L,Ym] = LeerCableCatenCaso2(ruta, nombreArchivo);
10        [FS,ER,Wo,L,Ym,Ho,Hf,To,Tf,Th,Yo, Yf,
TmaxCable,Tmax,Tmin,Lo,Lf,S,a,wx,wy,
x,y,str] = CableCatenCaso2(FS,ER,Wo,L,Ym);
11        SalidaCableCatenCaso2(FS,ER,Wo,L,Ym,Ho,Hf,To,Tf,Th,Yo, Yf,
TmaxCable,Tmax,
Tmin,Lo,Lf,S,a,wx,wy,x,y,str, ruta, nameFileUser, nameFileExperimento,
nameFileUserEnglish);
12 end
13 %Cierra todas las ventanas
14 fclose(fid);
15
16 end
```

Figura 34. Código master del experimento 2 – Cable Catenaria

Fuente: Autor

Elaboración: Autor

2.2.3.2. Caso1: Flecha.

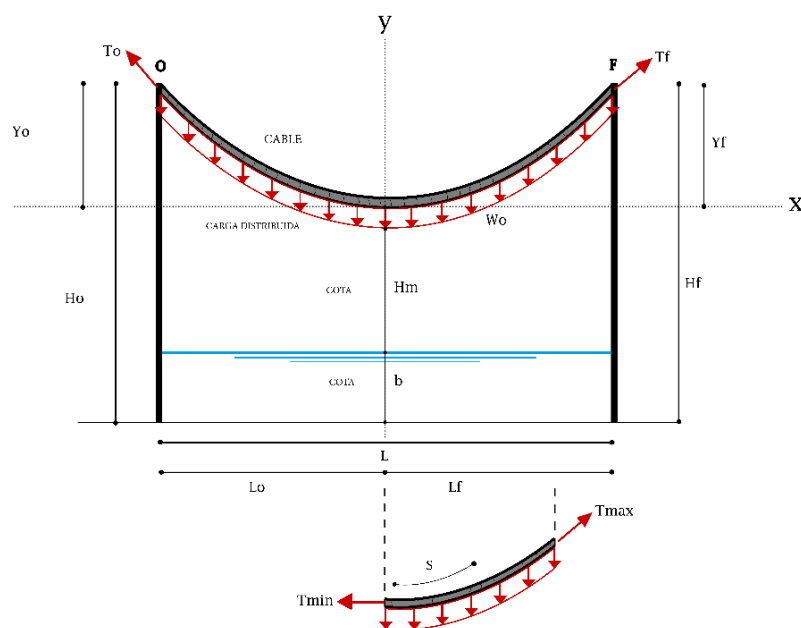


Figura 35. Esquema del experimento “Cable Catenaria – Caso 1

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 36):

- Se inicia la tarea con la función *LeerCableCatenCaso1*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCatenCaso1*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableCatenCaso1*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableCatenaria*.

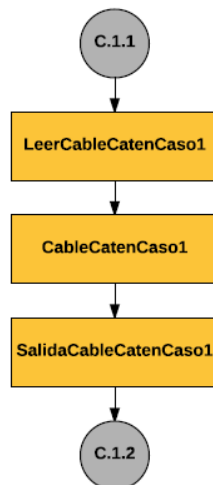


Figura 36. Diagrama de flujo de las funciones del caso 1 del experimento "Cable Catenaria"

Fuente: Autor

Elaboración: Autor

2.2.3.3. Caso2: Longitud.

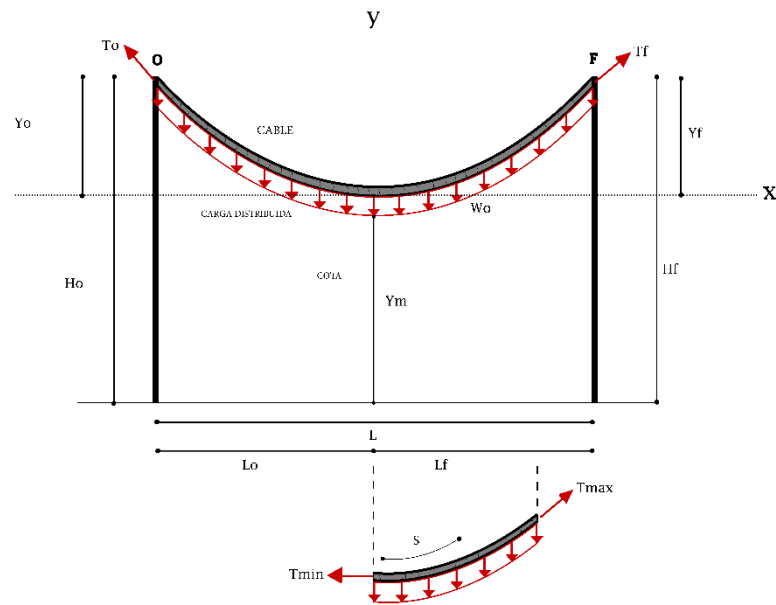


Figura 37. Esquema del experimento "Cable Catenaria – Caso 1

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 38):

- Se inicia la tarea con la función *LeerCableCatenCaso2*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCatenCaso2*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableCatenCaso2*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableCatenaria*

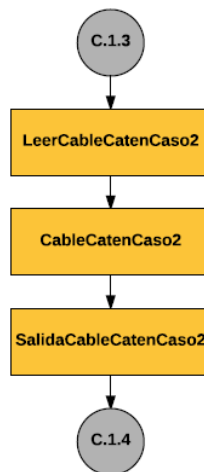


Figura 38. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”

Fuente: Autor

Elaboración: Autor

2.2.4. Experimento 3: “Cable con Cargas puntuales”.

En el análisis estático de cables se considera cargas puntuales, se ignora el peso del cable y la sección transversal. Para obtener la respuesta de un cable suspendido entre dos soportes rígidos como su deformada, la variación de tensión en toda su longitud y las reacciones en los apoyos. Adquiere una geometría tal que en cada punto de aplicación de una carga se forma un cambio de curvatura del cable.

Por la configuración de los tramos existen tres casos:

- Caso 1: Tramo AB
- Caso 2: Tramo BC
- Caso 3: Tramo CD

Tabla 6. Datos de Entrada y de Salida del experimento “Cable con Cargas puntuales”.

DESCRIPCIÓN	SÍMBOLO	UNIDAD
DATOS DE ENTRADA		
CASO 1: TRAMO AB		
Número de cargas	NWAB	#
Cargas	WAB	KN
Distancias	DAB	m
Desnivel de la primera carga	H1AB	m
Desnivel entre A y B	YB	m
CASO 2: TRAMO BC		
Número de cargas	NWBC	#
Cargas	WBC	KN
Distancias	DBC	m
Desnivel de la primera carga	H1BC	m
Desnivel entre B Y C	YC	m
CASO 3: TRAMO CD		
Número de cargas	NWCD	#
Cargas	WCD	KN

Distancias	DCD	m
Desnivel de la primera carga	H1CD	m
Desnivel entre C Y D	YD	m
DATOS DE SALIDA (RESULTADOS)		
FUERZAS RESULTANTES		
Reacción en el apoyo A	A	KN
Reacción en del eje x del apoyo A	Ax	KN
Reacción en del eje y del apoyo A	Ay	KN
Reacción en el apoyo B	B	KN
Reacción en del eje x del apoyo B	Bx	KN
Reacción en del eje y del apoyo B	By	KN
Reacción en el apoyo C	C	KN
Reacción en del eje x del apoyo C	Cx	KN
Reacción en del eje y del apoyo C	Cy	KN
Reacción en el apoyo D	D	KN
Reacción en del eje x del apoyo D	Dx	KN
Reacción en del eje y del apoyo D	Dy	KN
TRAMO AB		
Tensión máxima del cable	Tmax	KN
Tensión mínima del cable	Tmin	KN
Longitud del vano	L	M
Longitud del cable	S	M
TRAMO BC		
Tensión máxima del cable	Tmax	KN
Tensión mínima del cable	Tmin	KN
Longitud del vano	L	M
Longitud del cable	S	M
TRAMO CD		
Tensión máxima del cable	Tmax	KN
Tensión mínima del cable	Tmin	KN
Longitud del vano	L	M
Longitud del cable	S	M

Fuente: Autor

Elaboración: Autor

2.2.4.1. Programa Master del experimento “Cable con cargas Puntuales”.

El programa master del experimento “Cable con Cargas Puntuales”, hace referencia, a la función *ProgramaMasterCableCargasPuntuales*, es la herramienta líder para la gestión del experimento. Al solicitar la función, el proceso es el siguiente:

- Se inicia la tarea, el control es de la función del experimento.
- Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada.
- Identifica la función que corresponde al caso (Caso1,2 y 3) que sea ejecutado.
- La función del caso ejecutado realiza su proceso específico.
- Identifica los archivos de texto generados por el caso ejecutado, lo cual, corresponde a los datos de salida (resultados).

outputExperimento.txt

outputUser.txt

outputUserEnglish.txt

- Se finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, función *ProgramaMasterLaboratorioCables*

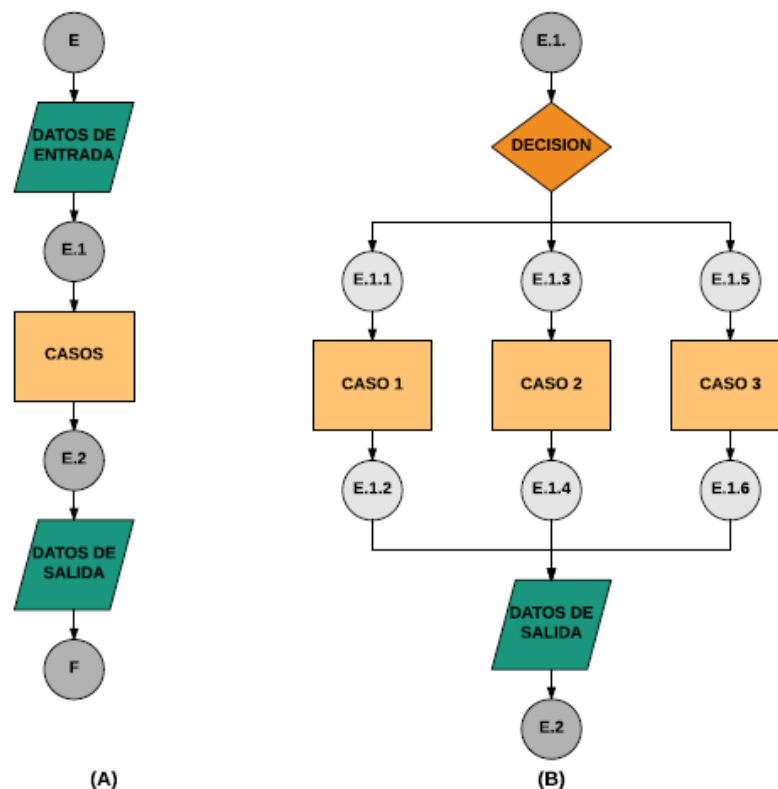


Figura 39. Diagrama de flujo del Programa Master del experimento “Cable con Cargas Puntuales”

Fuente: Autor

Elaboración: Autor

En la figura 39, se muestra el proceso de la función *ProgramaMasterCableCatenaria* (figura 39-A). Entre el conector E.1 y conector E.2, existe un proceso que corresponde a la ejecución de los casos (figura 39-B); en donde, se identifica el caso ejecutado con su respectiva función y cada uno tiene sus conectores de enlace que almacenan el proceso de análisis de dicho caso a través de otras funciones.

Tabla 7. Nombre de la Funciones que corresponden a los Casos del Experimento “Cable con Cargas Puntuales”

Número de Caso	Nombre del Experimento	Nombre de la Función	Conector
Caso 1	Tramo 1	LeerCableCargaPuntual1 CableCargaPuntual1 SalidaCableCargaPuntual1	E.1.1 - E.1.2
Caso 2	Tramo 2	LeerCableCargaPuntual2 CableCableCargaPuntual2 SalidaCableCargaPuntual2	E.1.3 - E.1.4
Caso 3	Tramo 3	LeerCableCargaPuntual3 CableCableCargaPuntual3 SalidaCableCargaPuntual3	E.1.5 - E.1.6

Fuente: Autor

Elaboración: Autor

2.2.4.1.1. Código Master del experimento “Cable Catenaria”.

```
C:\Users\DELL ... \ProgramaMasterCableCargasPuntuales.m Page
13 de agosto de 2017 16:43:11
1 function [] = ProgramaMasterCableCargasPuntuales(ruta,nameFileUser,
nameFileExperimento,nameFileUserEnglish)
2     %% Programa Master del experimento Cables con Cargas puntuales
3     switch cl(1)
4         case 1 %Caso1: 1 Tramo
5             [NW,W,D,h1,YB] = LeerCableCargasPuntuales1(ruta, nombreArchivo);
6             [A, Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S,
x,y,xx,yy,str,NW,W,D,h1,
YB] = CableCargasPuntuales1 (NW,W,D,h1,YB);
7             SalidaCableCargasPuntuales1(A, Ax, Ay, B, Bx, By, T, Tmax, Tmin,
h, L,
S, x,y,xx,yy,str,NW,W,D,h1,YB, ruta, nameFileUser, nameFileExperimento,
nameFileUserEnglish);
8         case 2 %Caso2: 2 Tramos
9             [NT, NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC] =
LeerCableCargasPuntuales2(ruta, nombreArchivo);
10            [TA, TAx, TAY, TBAB,TBBC, TB, TBx, TBy,TC,TCx,TCy, TAB, TABmax,
TABmin,
hAB, LAB, SAB,TBC, TBCmax, TBCmin, hBC, LBC, SBC,str,NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,
WBC,DBC,h1BC,YC,x,y,xx,yy] = CableCargasPuntuales2 (NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,
DBC,h1BC,YC);
11            SalidaCableCargasPuntuales2(TA, TAx, TAY, TBAB,TBBC, TB, TBx,
TBy,TC,
TCx,TCy, TAB, TABmax, TABmin, hAB, LAB, SAB,TBC, TBCmax, TBCmin, hBC, LBC,
SBC,str,NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,x,y,xx,yy, ruta, nameFileUser,
nameFileExperimento, nameFileUserEnglish) ;
12        case 3 %Caso3: 3 Tramos
13            [NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,NWCD,WCD,DCD,h1CD,YD] =
LeerCableCargasPuntuales3(ruta, nombreArchivo);
14            [TA, TAx, TAY, TBAB,TBBC, TB, TBx, TBy,TCBC,TCCD,TC,TCx,TCy, TD,
TDx,
TDy, TAB, TABmax, TABmin, hAB, LAB, SAB,TBC, TBCmax, TBCmin, hBC, LBC, SBC,
TCD,
TCDmax, TCDmin, hCD, LCD, SCD,str,NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,NWCD,
WCD,DCD,h1CD,YD,x,y,xx,yy] = CableCargasPuntuales3 (NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,
DBC,h1BC,YC,NWCD,WCD,DCD,h1CD,YD);
15            SalidaCableCargasPuntuales3(TA, TAx, TAY, TBAB,TBBC, TB, TBx,
TBy,TCBC,
TCCD,TC,TCx,TCy, TD, Tex, TDy, TAB, TABmax, TABmin, hAB, LAB, SAB,TBC, TBCmax,
TBCmin,
hBC, LBC, SBC, TCD, TCDmax, TCDmin, hCD, LCD, SCD,str,NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,
WBC,DBC,h1BC,YC,NWCD,WCD,DCD,h1CD,YD,x,y,xx,yy, ruta, nameFileUser,
nameFileExperimento,nameFileUserEnglish) ;
16        end
17
18 % Cierra todas las ventanas con gráficos
19 fclose(fid);
20 %close
21 end
```

Figura 40. Código master del experimento 3 – Cable con Cargas Puntuales

Fuente: Autor

Elaboración: Autor

2.2.4.2. Caso1: Tramo 1

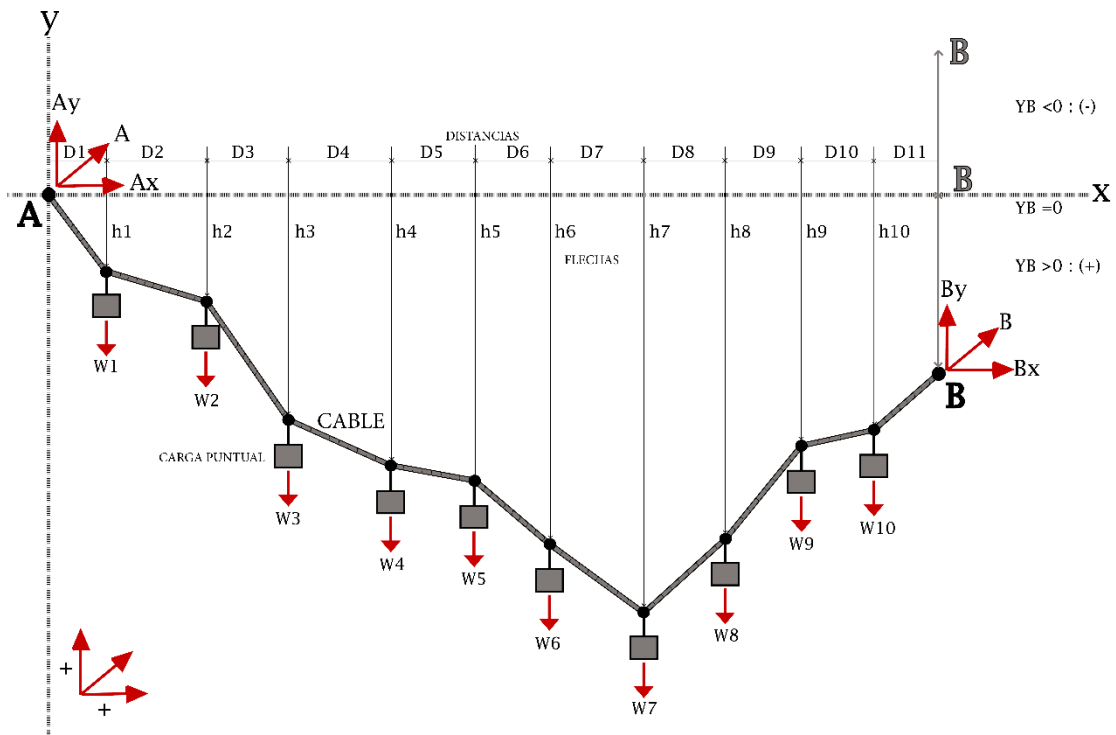


Figura 41. Esquema del experimento “Cable Cargas Puntuales” – Caso 1: Tramo 1

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 42):

- Se inicia la tarea con la función *LeerCableCargasPuntuales1*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCargasPuntuales1*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableCargasPuntuales1*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableCargasPuntuales*.

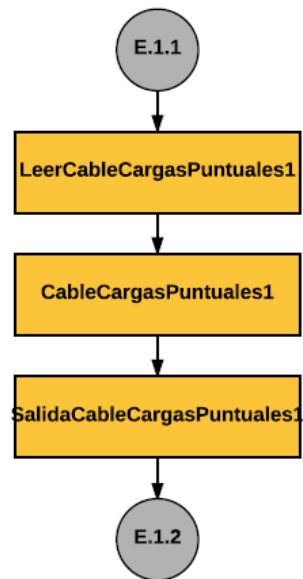


Figura 42. Diagrama de flujo de las funciones del caso 1 del experimento “Cable Catenaria”
Fuente: Autor
Elaboración: Autor

2.2.4.3. Caso2: Tramo 2.

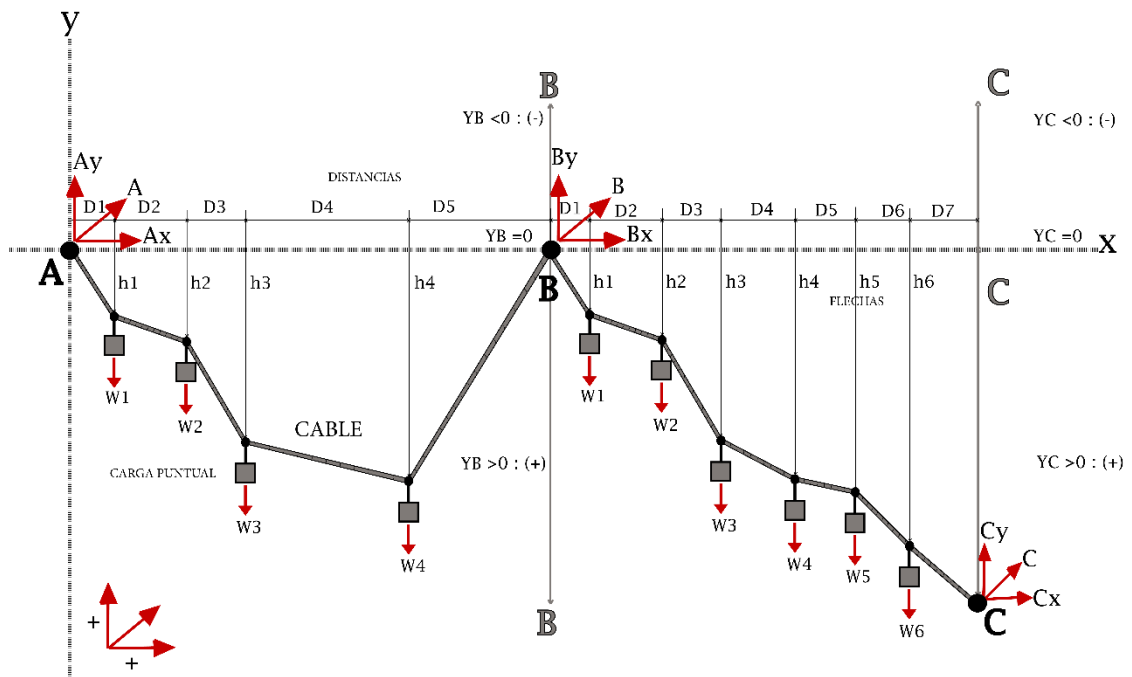


Figura 43. Esquema del experimento “Cable Catenaria – Caso 1”
Fuente: Autor
Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 44):

- Se inicia la tarea con la función *LeerCableCargasPuntuales2*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCargasPuntuales2*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableCargasPuntuales2*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableCargasPuntuales*

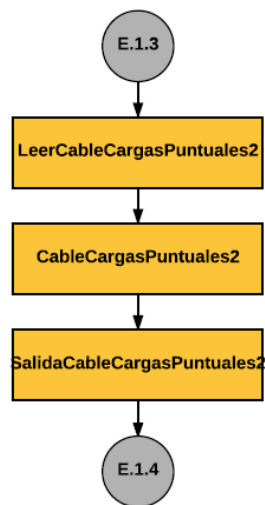


Figura 44. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”

Fuente: Autor

Elaboración: Autor

2.2.4.4. Caso2: Tramo 3.

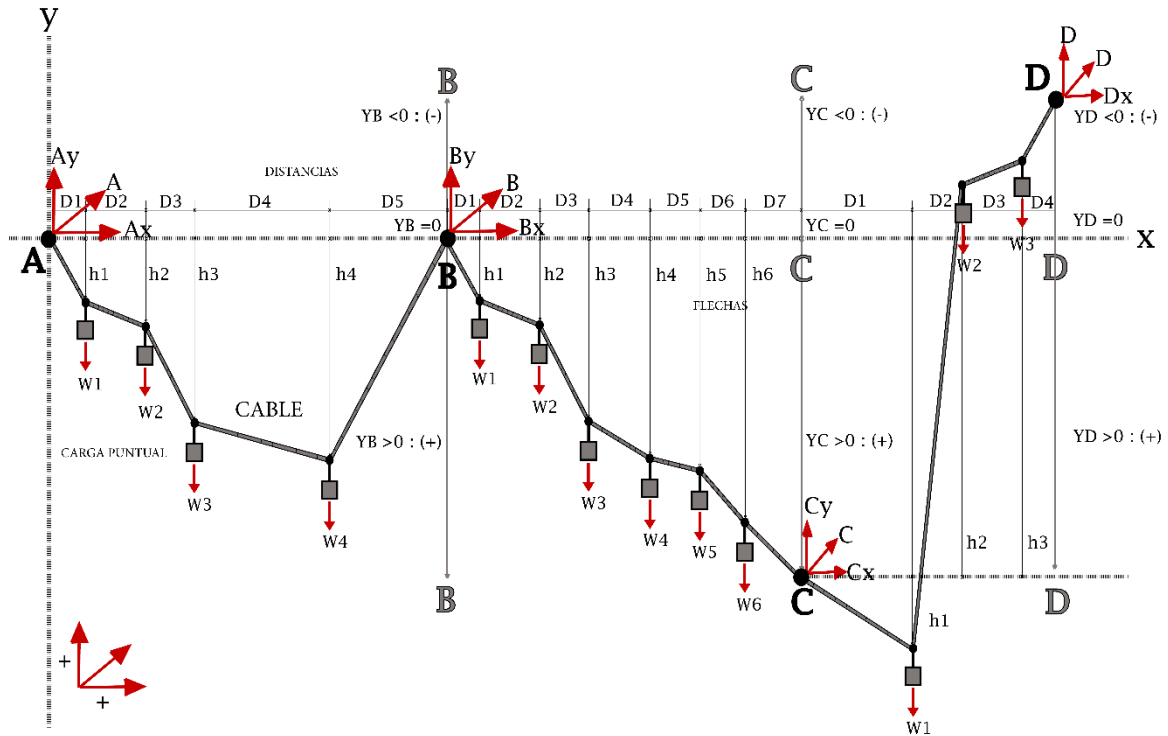


Figura 45. Esquema del experimento "Cable Catenaria – Caso 1

Fuente: Autor

Elaboración: Autor

Las funciones con respecto al caso1, contienen líneas de códigos que realizan una tarea específica (figura 46):

- Se inicia la tarea con la función *LeerCableCargasPuntuales3*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCargasPuntuales3*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida
- Después, con la función *SalidaCableCargasPuntuales3*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterCableCargasPuntuales*

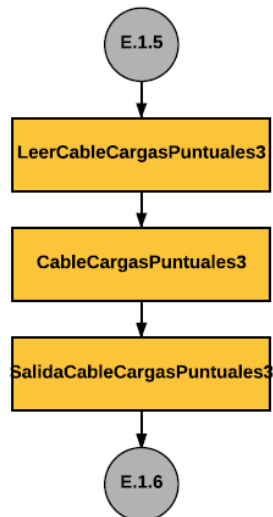


Figura 46. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”

Fuente: Autor

Elaboración: Autor

2.2.5. Experimento 4: “Cable Catenaria Elástica”

En el análisis de la respuesta de La Catenaria Elástica a muchas cargas puntuales, se considera el peso propio de un cable (Catenaria) como una carga uniformemente distribuida en toda su longitud, dos o más cargas puntuales verticales situados a una distancia, sección transversal constante y compuesto por un material homogéneo linealmente elástico. Para obtener la respuesta de un cable suspendido entre dos soportes rígidos como su deformada, la variación de tensión en toda su longitud y las reacciones en los apoyos.

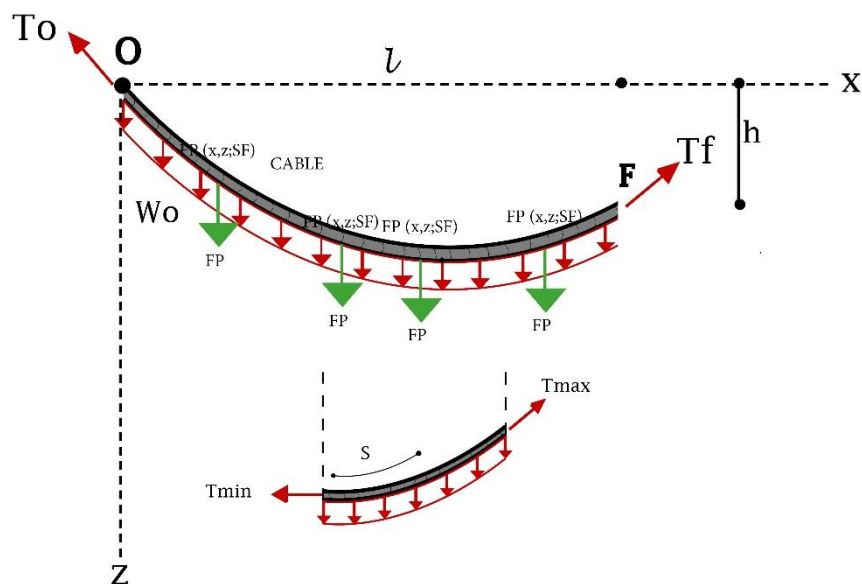


Figura 47. Diagrama de flujo de las funciones del caso 2 del experimento “Cable Catenaria”

Fuente: Autor

Elaboración: Autor

Tabla 8. Datos de Entrada y de Salida del experimento “Cable Catenaria Elástica”.

DESCRIPCIÓN	SÍMBOLO	UNIDAD
DATOS DE ENTRADA		
CASO 1: TRAMO AB		
Tramo del cable (Luz)	l	m
Desplazamiento vertical relativo de los extremos	h	m
La longitud no estirada del cable (Geometría original)	L_o	m
Peso propio del cable por unidad de longitud	w_o	$\frac{KN}{m}$
Área de la sección transversal uniforme en el perfil de la geometría original	A_o	m^2
Módulo de Young	E	$\frac{KN}{m^2}$
Número de Tramos	NT	#
Número de cargas por tramo	NFP	#
Cargas	FP	KN
Abscisa de las cargas a lo largo del cable	SF	m
DATOS DE SALIDA (RESULTADOS POR TRAMO)		
Reacción en el apoyo inicial (O)	O	KN
Reacción en del eje x del apoyo (O)	Ho	KN
Reacción en del eje y del apoyo (O)	Vo	KN
Reacción en el apoyo (F)	F	KN
Reacción en del eje x del apoyo (F)	Hf	KN
Reacción en del eje y del apoyo (F)	Vf	KN
Tensión máxima del cable	Tmax	KN
Tensión mínima del cable	Tmin	KN
RESULTADOS DEL PUNTO MÁS BAJO		
Tensión	Tp	KN
Abscisa a lo largo del cable	Sp	M
Coordenada en el eje x	Xp	M
Coordenada en el eje z	Zp	m
RESULTADOS DONDE SE LOCALIZA LA CARGA		
Carga	FP	KN
Abscisa de la carga a lo largo del cable	S	m
Coordenada en eje x	X(s)	m
Coordenada en eje y	Z(s)	m
Tensión	T	KN

Fuente: Autor

Elaboración: Autor

2.2.5.1. Programa Master del experimento “Cable Catenaria Elástica”.

El programa master del experimento “Cable Catenaria Elástica”, hace referencia, a la función *ProgramaMasterCableCatenariaElastica*, es la herramienta líder para la gestión del experimento. Al solicitar la función, el proceso es el siguiente:

- Se inicia la tarea con la función *LeerCableCatenariaElastica*
Lee el archivo de texto (Input.txt), correspondientes a los datos de entrada (variables).
- Seguidamente con la función *CableCatenariaElastica*
Recibe las variables de datos de entrada
Realiza el análisis matemático del experimento
Devuelve las variables de datos de salida

- Después, con la función *SalidaCableCatenariaElastica*
Recibe las variables de datos de salida
Genera los archivos de texto correspondiente a los datos de salida (resultados).
outputExperimento.txt
outputUser.txt
outputUserEnglish.txt
- Finaliza la tarea, el control es devuelto al punto desde el cual la función fue llamada, *ProgramaMasterLaboratorioCables*.

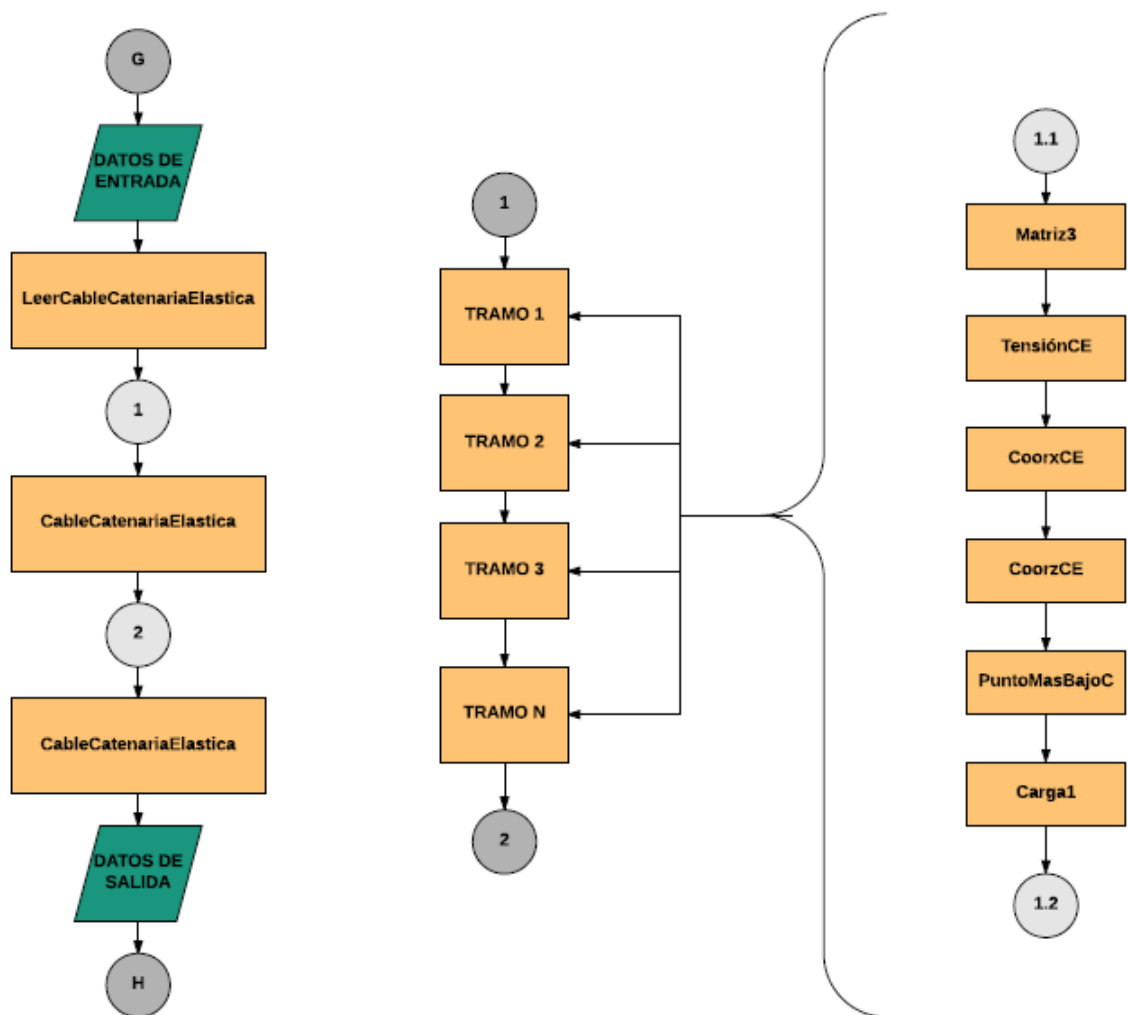


Figura 48. Diagrama de flujo del Programa Master del experimento “Cable Catenaria Elástica”
Fuente: Autor
Elaboración: Autor

En la figura 48, se muestra el proceso de la función *ProgramaMasterCableCatenariaElastica*. Entre el conector 1 y conector 2, existe un proceso que corresponde a la ejecución de varias funciones con su procedimiento. Cada Tramo tiene su configuración específica.

Tabla 9. Nombre de la Funciones que corresponden a los Tramos del Experimento “Cable Catenaria Elástica”

Función Nivel 1	Función Nivel 2	Función Nivel 3
LeerCableCatenariaElastica		
CableCatenariaElastica	Matriz3	Matriz1
		MAGADIMEN
		Matriz2
		tolx
		tolf
	TensionCE	
	CoorxCE	
	CoorzCE	
	PuntoMasBajoCable	
	Carga1	
SalidaCableCatenariaElastica		

Fuente: Autor

Elaboración: Autor

2.2.5.1.1. Código Master del experimento “Cable Catenaria Elástica”.

```
C:\Users\DE...\ProgramaMasterCablesCatenariaElastica.m
Page 1
3 de agosto de 2017 21:32:28
1 function [] =
ProgramaMasterCablesCatenariaElastica(ruta,nameFileUser,
nameFileExperimento,nameFileUserEnglish)
2     %% Experimento de la Catenaria Elástica
3     switch c1(1)
4         case 1
5             [NTra] = LeerCableCatenariaElastica (ruta,
nombreArchivo);
6             [r7,r13,r15,XG,ZG,RCR,T,str]= CableCatenariaElastica
(NTra);
7             SalidaCableCatenariaElastica(NTra,r7,r13,r15,XG,ZG,RCR,T,str,ruta,
nameFileUser, nameFileExperimento,nameFileUserEnglish);
8         end
9         fclose(fid);
10
11     % Cierra todas las ventanas con gráficos
12     %close
13 end
```

Figura 49. Código master del experimento 4 – Cable Catenaria Elástica

Fuente: Autor

Elaborado por: Autor

CAPITULO III

3. CASO DE ESTUDIO

3.1. Introducción

El uso de los cables en la formación de estructuras ha sido estratégico. Las estructuras de cables se clasifican en lineales, planas y tridimensionales. Las estructuras de cables lineales tienen la característica de avanzar en una dimensión (Such, 2008). En general se ha utilizado habitualmente en las líneas de transmisión de energía eléctrica, catenarias de trenes de alta velocidad, puentes colgantes, arcos y sistemas de transporte por cables.

Los sistemas de transporte por cable son esenciales para transportar a pequeñas distancias con grandes desniveles o topografía excesivamente irregular. El transporte por cable se clasifica por el tipo de instalación, cuando se emplean cables metálicos, situados a lo largo del recorrido para constituir la vía de circulación de los vehículos o para transmitir a los mismos un esfuerzo motor o frenante. (Orro Arcay, Novales Ordax, y Rodríguez Bugarín, 2003). Los tipos de transporte por cable son tres: funiculares, teleféricos y telesquíes.

Orro Arcay et al. (2003) define a el teleférico por la instalación de transporte, los vehículos se encuentran suspendidos de uno o más cables. El diccionario de la RAE define a el teleférico como un sistema de transporte en que los vehículos van suspendidos de un cable de tracción y se emplea principalmente para salvar grandes diferencias de altitud. Según el tipo de cables el teleférico se divide en dos, Bicable y Monocable.

Los teleféricos monocables poseen un cable transportador que tiene la función de un cable portante y cable tractor, los vehículos se conectan al anillo de cable mediante mordazas (Orro Arcay et al., 2003). El cable transportador soporta la carga y transmite la fuerza para el movimiento, se conoce también como portador tractor. (Orro Arcay, Novales Ordax, y Rodríguez Bugarín, 2003). El cable portante constituye la vía de circulación y soporta la carga, también se conoce como cable carril o portador. (Orro Arcay et al., 2003). Y el cable tractor transmite la fuerza para el movimiento, también se conoce como cable tracción. (Orro Arcay et al., 2003).

En América existe alrededor de 60 teleféricos de gran relevancia. Colombia se caracteriza por tener 7 sistemas de transporte por cable: 4 Teleféricos, 1 Miocable, 1 Cable aéreo y 1 Metrocable. El Metrocable de Medellín es un sistema de transporte aéreo del tipo teleférico y un subtipo de Monocable. El Metrocable de Medellín está compuesto por tres líneas de servicio comercial: Línea J, Línea K, Línea L, Línea M y Línea H, juntas suman una extensión total de 11.92 kilómetros (“Metrocable Medellín”, 2017).

En el capítulo presente se analiza el caso de estudio “Metrocable de Medellín – Línea K” a través del experimento 4 “Cable Catenaria Elástica” del Laboratorio de Cables de VLEE. En el análisis de la respuesta, se considera el peso propio de un cable (catenaria) como una

carga uniformemente distribuida en toda su longitud, dos o más cargas puntuales verticales (cabinas) situados a una distancia, sección transversal constante y compuesto por un material homogéneo linealmente elástico. Con el objeto de obtener su deformada, la variación de tensión en toda su longitud y las reacciones en los apoyos.

3.2. Metrocable de Medellín

Para una integración intermodal referente al transporte urbano de Medellín, es clave la combinación de bus, cable y metro. La tecnología utilizada en las 10 líneas es la siguiente:

- Línea A: Tecnología Férreo.
- Línea B: Tecnología Férreo
- Línea J: Tecnología Cable aéreo
- Línea K: Tecnología Cable aéreo (Caso de Estudio)
- Línea L: Tecnología Cable aéreo
- Línea M: Tecnología Cable aéreo
- Línea H: Tecnología Cable aéreo
- Línea 1 bus: Tecnología de Bus Articulado
- Línea 2 bus: Tecnología de Bus Padrón
- Línea T-A: Tecnología Tranvía

Son 5 las líneas del metrocable de Medellín con tecnología de Cable aéreo. El caso de estudio es la línea K y su recorrido es de la estación de Santo Domingo Savio a la estación de Acevedo. En la tabla 10 se presenta el recorrido de las líneas.

Tabla 10. Origen y Destino de las líneas del metrocable (Teleférico) de Medellín.

Línea	Origen	Destino
Línea K	<ul style="list-style-type: none"> • Santo Domingo Savio • Acevedo 	<ul style="list-style-type: none"> • Acevedo • Santo Domingo Savio
Línea J	<ul style="list-style-type: none"> • San Javier • La Aurora 	<ul style="list-style-type: none"> • La Aurora • San Javier
Línea L	<ul style="list-style-type: none"> • Santo Domingo Savio • Arví 	<ul style="list-style-type: none"> • Arví • Santo Domingo Savio
Línea M	<ul style="list-style-type: none"> • Miraflores • Trece de noviembre 	<ul style="list-style-type: none"> • Trece de noviembre • Miraflores
Línea H	<ul style="list-style-type: none"> • Alejandro E • La sierra 	<ul style="list-style-type: none"> • La sierra • Alejandro E

Fuente: Autor

Elaboración: Autor

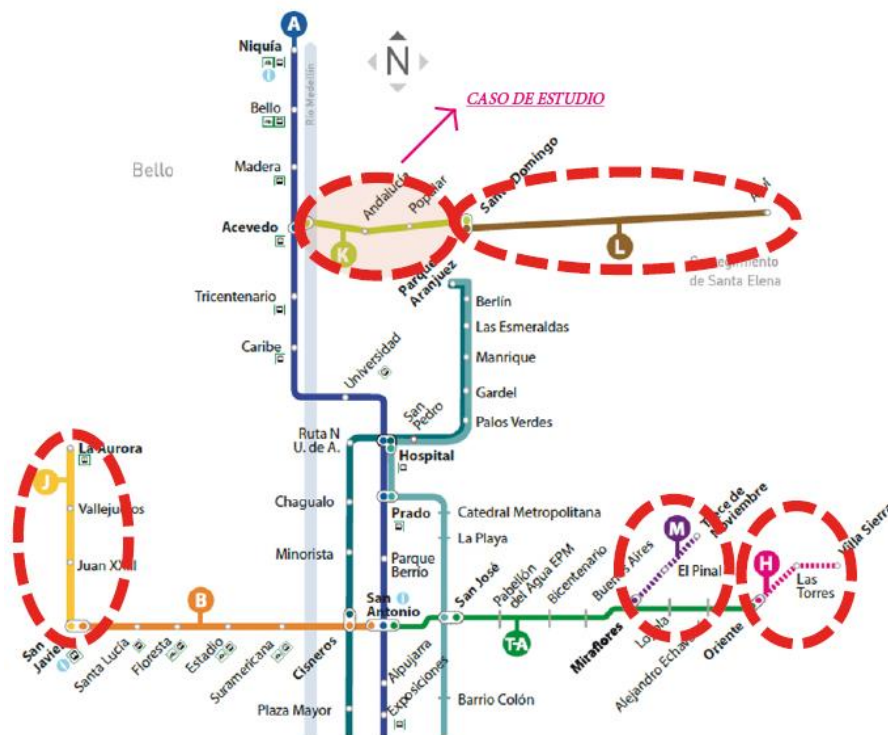


Figura 50. Mapa de la ubicación de las líneas con tecnología Cable aéreo.

Fuente: Metro de Medellín Ltda. (2017)

Elaboración: Metro de Medellín Ltda. (2017)

3.3. Metrocable de Medellín – “Línea K”

La línea K tiene una longitud de trazado de 2072 m, lo cual cubre un desnivel aproximadamente de 400 m. El recorrido inicia en la estación Santo Domingo Savio hasta la estación Acevedo. Cuenta con 4 estaciones y 93 cabinas, con una demanda de 25000 pasajeros por día.

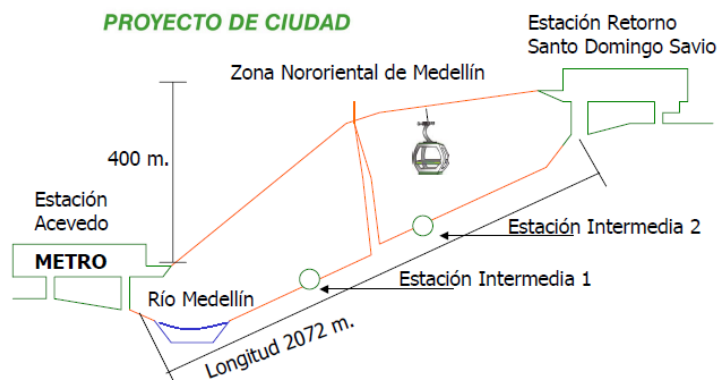


Figura 51. Esquema de la línea k – Metrocable Medellín

Fuente: Metro de Medellín Ltda. (2004)

Elaboración: Metro de Medellín Ltda. (2004)

3.3.1. Localización

El proyecto se encuentra ubicado en la localidad de la ciudad de Medellín Colombia y el tamo objeto de estudio, La línea K, atraviesa las comunas número 2 Santa Cruz y número 1 Popular, desde la estación intercambiadora Acevedo hasta la estación de transferencia Santo Domingo Savio que brinda la posibilidad de intercambio con la Línea L.



Figura 52. Esquema de la ubicación de la Línea K – Metrocable Medellín.

Fuente: Metro de Medellín Ltda. (2004)

Elaboración: Metro de Medellín Ltda. (2004)

3.3.2. Características técnicas de la línea “K” – Metrocable de Medellín.

El Metrocable de Medellín - línea k tiene un sistema tipo: Telecabinas Pinza Desenganchable.

Tabla 11. Datos de entrada de los 18 tramos de la línea K – Metrocable de Medellín.

DESCRIPCIÓN	SÍMBOLO	UNIDAD	MAGNITUD
DATOS DE ENTRADA			
Longitud del trazado	L	m	2072
Desnivel		m	400
Pendiente Promedio de la línea	m	%	20
Número de pilonas de sostenimiento		#	20
Factor de Seguridad	FS	#	5
Características del cable			
Diámetro del cable (alma compactada)	d	mm	51
Peso del cable	W	Ton	42
Área de la sección transversal en el perfil de la geometría original	A_o	m^2	0.00204
Módulo de Young	E	$\frac{KN}{m^2}$	4.9*E7
Peso propio del cable por unidad de longitud	w_o	$\frac{KN}{m}$	0.1129
Resistencia a la rotura	Rmin	KN	2000
Cabinas			
Número de cabinas		#	93
Distancia entre cabinas	d	m	60
Peso de la Cabina	FP	KN	6.86
Peso de las personas en la cabina: 8 personas sentadas y dos de pie	FP	KN	7.35
Peso de la cabina más las personas	FP	KN	14.21
Velocidad de las cabinas	V	m/s	5
Frecuencia	F	s	12
Aceleración	a	m/s2	0.4166
Tramos			
Número de tramos		#	19

Distancia entre pilonas (Luz)	l	m	107
Desplazamiento vertical relativo de los extremos	h	m	21
La longitud no estirada del cable (Geometría original)	L_o	m	108.5
El número de cabinas por tramo		#	1
La carga de la cabina con las personas	FP	KN	14.21
Distancia donde se encuentra la carga	S	m	54.5

Fuente: Metro de Medellín Ltda. (2004)

Elaboración: Autor

3.3.3. Solución numérica de la línea K – Metrocable de Medellín

La modelación realizada para el análisis del sistema estructural del cable del teleférico de la línea K – Metrocable Medellín, es la configuración original y básica representada en la figura siguiente.

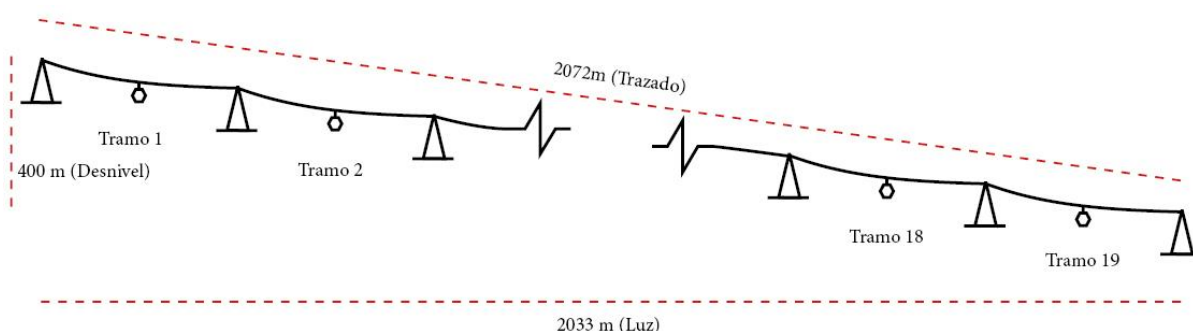


Figura 53. Estructura del cable – Línea K – Metrocable Medellín

Fuente: autor

Elaboración: Autor

Se ha resuelto numéricamente el problema del análisis del cable de acero con la herramienta “Catenaria Elástica” del laboratorio VLEE. El programa obtiene como solución la tensión del cable cuando este está sometido a las cargas de trabajo descritas previamente, las reacciones en el apoyo y la deformada.

Se analizó 19 tramos, correspondiente al teleférico, cada tramo tiene la misma configuración: luz a salvar de 107 metros, los apoyos tienen un desnivel de 21 m y la cabina se encuentra a la mitad del cable.

3.3.3.1. Datos de entrada para la herramienta

Propiedades del cable:

Diámetro: $\phi = 51 \rightarrow mm$

Área: $A = 0.000204 \rightarrow m^2$

Módulo de elasticidad: $E = 49E^7 \rightarrow \frac{KN}{m^2}$

Modelación de la estructura:

Número de tramos: #19

Por tramo se tiene:

Longitud a salvar de cada tramo: $l = 107 \rightarrow m$

Desnivel de cada tramo: $h = 21 \rightarrow m$

Largo inicial del cable: $L_o = 108.5 \rightarrow m$

Cargas sobre cada tramo:

Numero de cargas: #1

Magnitud de la carga: $FP = 14.21 \rightarrow KN$

Abscisa de la carga: $s = 54.5 \rightarrow m$

Archivo de texto de los resultados para el usuario:

LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA (VLEE)			
GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y SISMOLOGÍA (GRISS-UTPL)			
DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA CIVIL			
SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y CONSTRUCCIÓN			
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA (UTPL)			

LABORATORIO: CABLES			
EXPERIMENTO: Catenaria Elástica			
FECHA : 01-Sep-2017 09:20:41			
Autores : Msc. Duque, E.; Ing. Quiñonez, S.; Pelaez, D.			

TRAMO 1			

INPUT DATA			

DESCRIPTION	SYMBOL	UNITY	MAGNITUDE
Distancia Horizontal entre los puntos (O) - (F)	l	m	107
Diferencia de nivel entre los puntos (O) - (F)	h	m	21
Propiedades del Cable			
Longitud inicial entre los puntos (O)-(F)	Lo	m	108.5
Área de la geometría original	Ao	m^2	0.000204
Módulo de Young	E	KN/m^2	49000000
Peso por unidad de longitud	Wo	KN/m	0.11
Cargas			
Número de cargas	NFP	#	1
Carga 1	FP	KN	14.21
Posición de la carga 1 a lo largo - cable	SF	m	54.5

Figura 54. Archivo de Texto de los datos de entrada

Fuente: Autor.

Elaborado: Autor

En la figura 54, se muestra la primera parte del archivo de texto que imprime la herramienta “Cable Catenaria elástica” del laboratorio VLEE, después del análisis. La primera parte del

archivo de texto contiene un encabezado, nombre del experimento, fecha, autores y los datos de entrada. De igual forma, el archivo está disponible en español e inglés.

3.3.3.2. Resultados de salida.

Una vez ejecutada la herramienta, se muestran los resultados, figura 49 y figura 50.

3.3.3.2.1. Por tramo.

ANALYSIS OUPUT				
DESCRIPTION		SYMBOL	UNITY	MAGNITUDE
Reacción horizontal del punto	(O)	Ho	KN	98.0322
Reacción vertical del punto	(O)	Vo	KN	32.4789
Reacción en el punto	(O)	O	KN	103.2724
Reacción horizontal del punto	(F)	Hf	KN	-98.0322
Reacción vertical del punto	(F)	Vf	KN	6.3339
Reacción en el punto	(F)	F	KN	98.2366
Tensión máxima del cable		Tmax	KN	103.2724
Tensión mínima del cable		Tmin	KN	98.2366
Resultados del punto más bajo				
Tensión		Tp	KN	98.2366
Abscisa a lo largo del cable		Sp	m	108.5
Coordenada en el eje x		Xp	m	107
Coordenada en el eje z		Zp	m	21
Resultados donde se localiza la carga				
Carga	S	X(s)	Z(s)	Tensión
[KN]	[m]	[m]	[m]	[KN/m]
0	0	0	0	103.2724
14.21	54.5	52.7199	15.85	98.7976

Figura 55. Resultados de un Tramo

Fuente: Autor.

Elaborado: Autor

En la figura 55, se muestra la segunda parte del archivo de texto que imprime la herramienta “Cable Catenaria elástica” del laboratorio VLEE, después del análisis. La segunda parte del archivo de texto contiene las reacciones en los apoyos (punto Inicial “O” y punto final “F”), las tensiones en el cable (Tensión máxima y Tensión mínima), los datos del punto más bajo de la geometría del cable (Tensión, abscisa, coordenadas en el eje “X” y “Z”) y los resultados donde se localiza la carga (magnitud de la carga, abscisa en la longitud del cable, coordenada “X” y “Z”, y Tensión).

3.3.3.2.2. Todos los tramos.

Las Reacciones en los apoyos (puntos de conexión del cable), 20 pilonas. En la figura 56, se muestra la tercera parte del archivo de texto que imprime la herramienta “Cable Catenaria elástica” del laboratorio VLEE, después del análisis. La tercera parte del archivo de texto contiene las reacciones en los apoyos, cada punto de conexión, en la proyección del eje “X”, del eje “Y”, y la resultante.

Punto 1			
Reacción - proyección X	KN	98.0322	
Reacción - proyección Y	KN	32.4789	
Reacción	KN	103.2724	
Punto 2			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 3			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 4			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 5			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 6			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 7			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 8			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 9			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 10			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 11			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	

Punto 12			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 13			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 14			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 15			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 16			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 18			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 19			
Reacción - proyección X	KN	0	
Reacción - proyección Y	KN	38.8127	
Reacción	KN	38.8127	
Punto 20			
Reacción - proyección X	KN	-98.0322	
Reacción - proyección Y	KN	6.3339	
Reacción	KN	98.2366	

Figura 56. Resultados de todos los puntos de conexión de los tramos del archivo de texto.
Fuente: Autor.
Elaborado: Autor

3.3.3.2.3. Deformada del cable de un Vano

En la figura 57, se presenta la deformada del cable, que corresponde a un vano. El cable está suspendido con una luz de 107 m. y los apoyos con un desnivel de 21 m. En la sección 3.3.3.3.4. se presenta el análisis numérico para determinar la flecha máxima del cable y si cumple con la flecha máxima requerida.

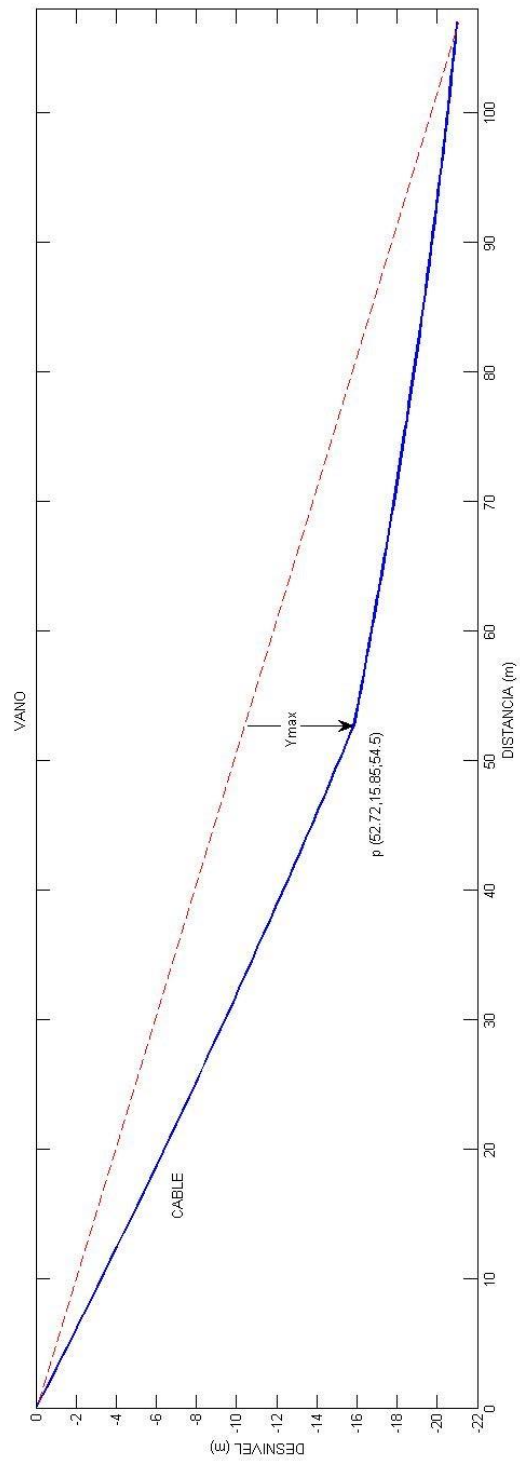


Figura 57. La deformada del cable de un vano del metrocable de Medellín – línea k.
Fuente: Autor.
Elaborado: Autor

3.3.3.2.4. Deformada del cable de todos los Vanos

En la figura 58, se presenta la deformada del cable, que corresponde a los 19 vanos. El cable tiene una longitud de trazado de 2072 m y un desnivel de 400 m, entre la primera pylona (estación Santo Domingo Savio) y la última pylona (Estación Acevedo).

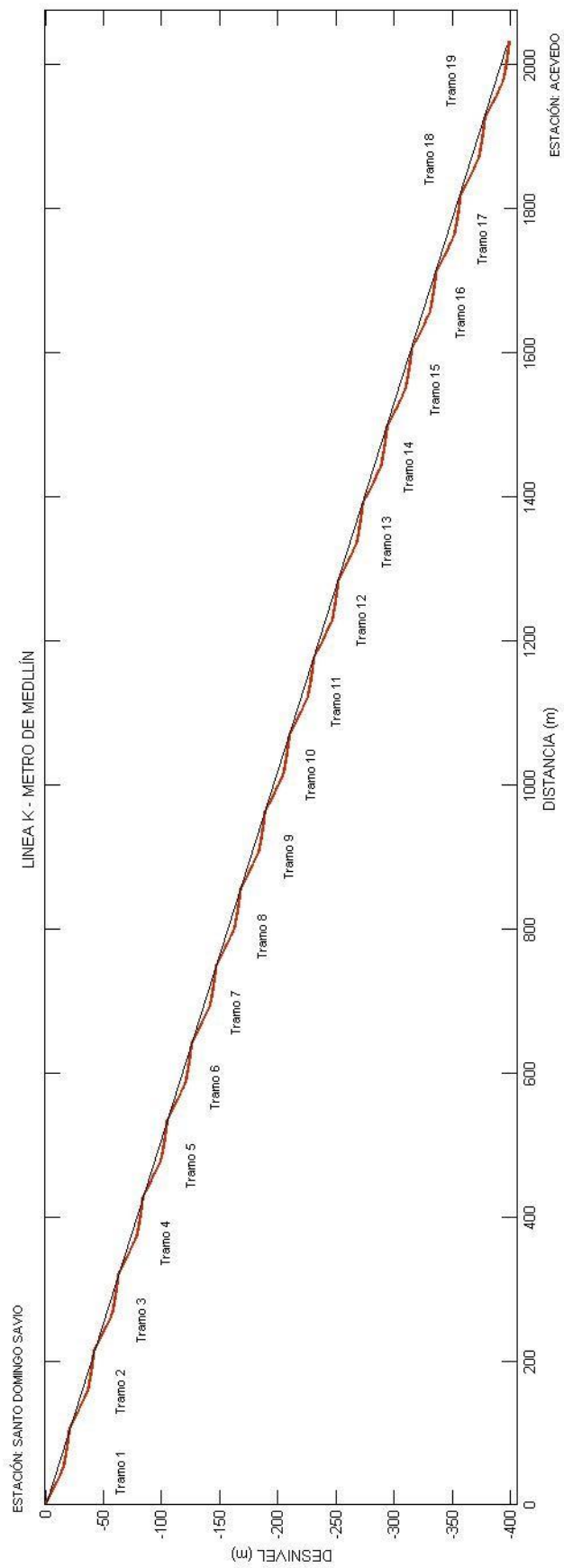


Figura 58. La deformada de todos los tramos
Fuente: Autor.
Elaborado: Autor

3.3.3.3. *Análisis de la “línea K” – Metrocable de Medellín.*

El análisis dependerá de las características mecánicas del cable, así como las configuraciones del sistema, las tensiones y flecha máxima admisible del cable.

3.3.3.3.1. *Tensión estática S_E .*

La carga o tensión estática se origina por la carga de trabajo segura a transportar por el cable de acero, correspondiente a la carga de peso propio del cable y a las cargas puntuales (cabinas cargadas).

$$S_E = \frac{T}{A} \quad (\text{Ecuación 56})$$

Donde:

S_E : Tensión estática del cable de acero $\left(\frac{KN}{m^2}\right)$

T : Tracción o tensión originada por las cargas de trabajo (KN)

A : Sección de acero del cable (m^2)

$$S_E = \frac{T}{A} = \frac{103.27 \rightarrow KN}{0.000204 \rightarrow m^2} = 50.62E^4 \frac{KN}{m^2}$$

3.3.3.3.2. *Tensión dinámica S_D .*

Se origina al desplazar las cargas a la velocidad prevista. Esta tensión se determina como una parte o fracción del análisis estático, mediante la siguiente expresión.

$$S_D = \frac{a}{g} * S_E \quad (\text{Ecuación 57})$$

Donde:

S_D : Tensión dinámica del cable de acero $\left(\frac{KN}{m^2}\right)$

S_E : Tensión estática del cable de acero $\left(\frac{KN}{m^2}\right)$

a : aceleración del cable, originada por enganche y desacople de las cabinas $\left(\frac{m}{s^2}\right)$

g : aceleración de gravedad $\left(\frac{m}{s^2}\right)$

$$S_D = \frac{a}{g} * S_E = \left(\frac{0.4166 \rightarrow \frac{m}{s^2}}{9.81 \rightarrow \frac{m}{s^2}} \right) * 50.62E^4 \rightarrow \frac{KN}{m^2} = 2.15E^4 \rightarrow \frac{KN}{m^2}$$

3.3.3.3.3. Tensión máxima en el cable de acero

La tensión admisible se determina de la división entre la tensión de ruptura mínima y el factor de seguridad, debe cumplir, que la tensión máxima del cable sea menor o igual que la tensión admisible

$$S_E + S_D \leq \left(\frac{R_{\min}}{A} \right) * \frac{1}{FS} \quad (\text{Ecuación 58})$$

$$T \left(1 + \frac{a}{g} \right) \leq \left(\frac{R_{\min}}{FS} \right) \quad (\text{Ecuación 59})$$

Donde

R_{\min} : Resistencia mínima a la rotura garantizada (KN)

$$S_E + S_D \leq \left(\frac{R_{\min}}{A} \right) * \frac{1}{FS}$$

$$50.62E^4 \rightarrow \frac{KN}{m^2} + 2.15E^4 \frac{KN}{m^2} \leq \left(\frac{2000 \rightarrow KN}{0.000204 \rightarrow m^2} \right) * \frac{1}{5} \quad // \text{Si cumple}$$

$$52.77E^4 \rightarrow \frac{KN}{m^2} \leq 196E^4 \frac{KN}{m^2}$$

$$T \left(1 + \frac{a}{g} \right) \leq \left(\frac{R_{\min}}{FS} \right)$$

$$103.27 \rightarrow KN \left(1 + \frac{0.4166 \rightarrow \frac{m}{s^2}}{9.81 \rightarrow \frac{m}{s^2}} \right) \leq \left(\frac{2000 \rightarrow KN}{5} \right) \quad // \text{Si cumple}$$

$$107.66 \rightarrow KN \leq 400 \rightarrow KN$$

La resistencia de ruptura garantizada para el tipo de cable elegido de diámetro 51 mm. El cable resiste satisfactoriamente las solicitaciones de trabajo.

3.3.3.3.4. Determinación de la flecha máxima del cable.

Por geometría se puede determinar que el punto más bajo para el presente caso de estudio es el punto "F" (Apoyo en el punto de conexión final del cable). La flecha máxima se la determina en base a la teoría de cables descrita en la sección 1.5.4. (Catenaria Elástica). De este modo, la herramienta calcula las coordenadas de la carga, las mismas que se indican en la figura 59.

Carga [KN]	S [m]	X(s) [m]	Z(s) [m]	Tensión [KN/m]
0	0	0	0	103.2724
14.21	54.5	52.7199	15.85	98.7976

Figura 59. Coordenadas de la carga (Máxima deflexión del cable)

Fuente: Autor.

Elaborado: Autor

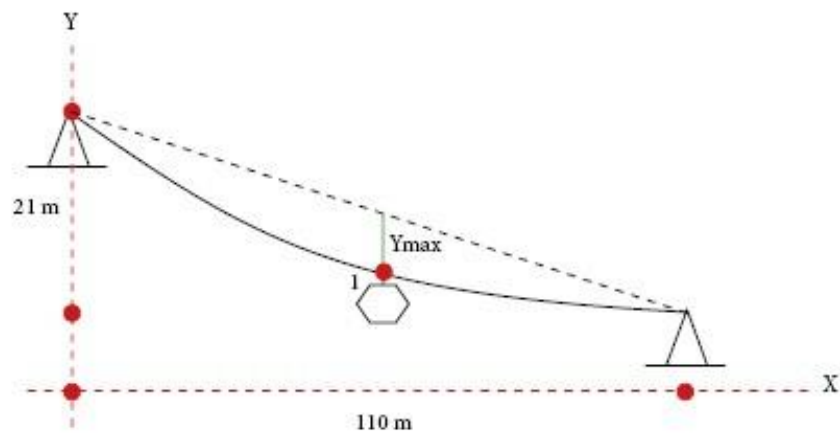


Figura 60. Esquema de la carga (Máxima deflexión del cable)
Fuente: Autor.
Elaborado: Autor

Para la determinación de la flecha se la realiza con la siguiente expresión:

$$Y_{\max} = h - \text{coord}_z - \frac{(l - \text{coord}_x)h}{l} \quad (\text{Ecuación 60})$$

$$Y_{\max} = 21 - 15.85 - \frac{(107 - 52.72)21}{107}$$

$$Y_{\max} = 5.50 \rightarrow m$$

La flecha máxima admisible es menor a 5.5% – 6% de l

$$Y_{\max} \leq [5.5 - 6\%]l \quad (\text{Ecuación 61})$$

$$5.50 \rightarrow m \leq 0.06 * 107 \rightarrow m$$

$$5.50 \rightarrow \leq 6.42 \rightarrow m$$

Si cumple

Se cumple cabalmente el requerimiento de flecha máxima.

CONCLUSIONES

Al culminar el presente Trabajo de Fin de Titulación se expresa las siguientes conclusiones:

- ✚ Se ha generado el módulo de “Cable Estructuras” perteneciente al VLEE. Este laboratorio permite el análisis numérico de los siguientes experimentos: “Cable Parabólico”, “Cable Catenaria”, “Cable con cargas puntuales” y “Cable Catenaria Elástica”. Asimismo, por medio de la interfaz se puede visualizar los resultados, gráficas y descargar archivos.
- ✚ El laboratorio virtual de “Cables Estructuras” ofrece las posibilidades de estar disponible para estudiantes, docentes y profesionales en la plataforma virtual del GRISS UTPL. En particular es una herramienta digital con la ventaja de ser accesible y gratuita.
- ✚ Los experimentos diseñados adecuadamente se convierten en un instrumento de apoyo para que los estudiantes desarrollen competencias, al afrontar y resolver problemas en conocimientos avanzados referente a Cables. De igual forma los fundamentos teóricos de cada experimento corresponden a Estructuras de Cables y son estratégicamente elegidos para contribuir en el aprendizaje de los estudiantes.
- ✚ El módulo del cable se constituye como una herramienta de estudio para estudiantes, así como de análisis de proyectos reales como se puede evidenciar en el proyecto “Metrocable Medellín – línea k”.
- ✚ En base al Caso de Estudio, Metrocable de Medellín – Línea K. Se realizó la aplicación de la herramienta, “Cable Catenaria Elástica”, con el objeto de modelar la estructura del cable del teleférico y se determinó con los datos planteados la deformada del cable, la tensión que presenta el cable y las reacciones en los apoyos. Se establece que la herramienta es útil y se puede aplicar a distintas configuraciones de estructuras formadas por cables, como es el caso de un teleférico.

RECOMENDACIONES

Habiéndose determinado la necesidad de desarrollar herramientas para el Laboratorio Virtual de Estructuras de Cables se hacen las siguientes recomendaciones:

- ✚ Profundizar en los temas de los experimentos del laboratorio Virtual de estructuras de Cables, a través de los fundamentos teóricos se recomienda:
 - Dinámica del cable suspendido.
 - La teoría lineal de las vibraciones libres de un cable suspendido.
 - La respuesta dinámica lineal de un Cable suspendido.
 - Las teorías no lineales de un cable.
 - Superficies tridimensionales
 - Análisis estático y dinámico de las membranas suspendidas.
- ✚ Generar nuevos experimentos, con las herramientas correspondientes, que aporten a la formación del estudiante.
- ✚ Se recomienda la actualización constante de la plataforma virtual del laboratorio, de la misma forma las herramientas informáticas que el mismo posee.
- ✚ Realizar la aplicación y el análisis a los distintos proyectos reales con los experimentos existentes del laboratorio de Cables.

BIBLIOGRAFÍA

- Barberis, M. (2001). *Análisis Estructural de Cables, Herramienta computacional y Aplicaciones*. Universidad Nacional de Córdoba.
- Bedford, A., & Fowler, W. (2008). CABLES. En *Mecánica para ingeniería: Estática* (5ta ed., p. 656). México: Pearson.
- Beer, F., & Cornwell, P. (2010). *Mecánica Vectorial para ingenieros: estática*. Valencia, España: McGraw-Hill Interamericana.
- Blanco, L. M. (2002). *Fundamentos de Programación con Visual Basic. NET*. Madrid (España): Eidos.
- CAMESA. (2008). Cable de Acero. wirecoworldgroup. Recuperado a partir de www.camesawire.com
- Center for History and New Media. (s/f). Guía rápida. Recuperado a partir de http://zotero.org/support/quick_start_guide
- Coti Colop, B. M. (2003, noviembre). *Reglas del Negocio en Arquitectura Tres Capas*. Universidad de San Carlos de Guatemala, Guatemala.
- Cubillos, C. (s/f). *Arquitectura Cliente/Servidor*. Chile.
- Diez, G. (2005). *Diseño Estructural en Arquitectura: Introducción* (la ed.). Buenos Aires: Nobuko.
- EMCOCABLES. (s/f). CABLES. Emcocables. Recuperado a partir de <http://www.emcocables.com/distribuidores.html>
- Engel, H. (2001). *Sistemas de estructuras* (2a.).
- Fábregas Acosta, E. (2013, abril). *Plataformas Interactivas de Experimentación Virtual y Remota: Aplicaciones de Control y Robótica*. Universidad Nacional de Educación a Distancia, Madrid.
- Goicolea, J. (2012). *Cálculo de Cables*. Madrid.
- González Martín, Ó. (1999, 2000). *Arquitecturas de Sistemas de Bases de Datos*.
- Irvine, H. M. (1981). Statics of a suspended Cable. En *Cable Structures*. The MIT Press.
- Irvine, H. M., & Sinclair, G. B. (1975). THE SUSPENDED ELASTIC CABLE UNDER THE ACTION OF CONCENTRATED VERTICAL LOADS. *Int. J. Solid Structures*, 12, 309–317.
- Marco Besteiro, & Miguel Rodríguez. (s/f). ASP.NET. Introducción a las aplicaciones WEB.
- Mejía, A. (2011). *Diseño y Construcción de un sistema de Transporte de carga por medio de cables para topografía de gran pendiente*. Universidad EAFIT, Medellín.
- Metro de Medellín Ltda. (2004). *Ejemplo de Intermodalidad: Sistema de transporte por cable aéreo integrado al metro de Medellín*. Diapositivas, Medellín.
- Metrocable Medellín. (2017, julio 29). En *Wikipedia, la enciclopedia libre*. Recuperado a partir

de

[https://es.wikipedia.org/w/index.php?title=Metrocable_\(Medell%C3%ADn\)&oldid=100790372](https://es.wikipedia.org/w/index.php?title=Metrocable_(Medell%C3%ADn)&oldid=100790372)

Monleón Cremades, S. (2014). *Curso de concepción de puentes. Vol. I: panorámica general de puentes*. Editorial de la Universidad Politécnica de Valencia. Recuperado a partir de

<http://site.ebrary.com/lib/bibliotecautplsp/detail.action?docID=10995713&p00=puede+colgante>

Museros Romero, P. (2017). *Mecánica: Estática y Cálculo vectorial*. Valencia, España: Universidad Politécnica de Valencia.

Orro Arcay, A., Novales Ordax, M., & Rodríguez Bugarín, M. (2003). *Transporte por Cable*. Coruña: Tórculo Artes Gráficas.

Pallares M., M. R., & Rodríguez C., W. (2008). Validación de la formulación numérica de la catenaria elástica con ANSYS. En *Matemáticas: Enseñanza Universitaria* (Vol. XVI, pp. 63–85). Colombia: Escuela Regional de Matemáticas Universidad del Valle.

Perles, P. (2009). Estructuras de tracción. En *Temas de estructuras especiales* (p. 263). Editorial Nobuko.

Perovich, D., & Vignaga, A. (s/f). *Arquitecturas y Tecnologías para el desarrollo de Aplicaciones Web*. Universidad de la República, Facultad de Ingeniería, Instituto de Computación, Montevideo, Uruguay.

Ponce, A., & Ponce, R. (2013). *Diseño y simulación de un teleférico con capacidad de transportación para 8 personas y un recorrido de 1 KM*. Universidad Politécnica Salesiana, Quito.

Pons, A. (2013). *Analizando la construcción* (la.). Barcelona: Universidad Politécnica de Catalunya.

Salvadori, M. (2005). *Estructuras para Arquitectos*. Buenos Aires: Nobuko.

Serrano N., P., & Castro F., D. (2009). *Los Cables de Acero y sus Aplicaciones* (Universidad de Cantabria). Santander. España.

SoftDoit. (s/f). ¿Qué es el código fuente? Recuperado a partir de <https://www.softwaredoit.es/definicion/definicion-codigo-fuente.html>

Suarez, V. A. (s/f). Desarrollo del Laboratorio virtual de Ingeniería Sísmica Aplicado al diseño Estructural Basado en Desplazamientos.

Such, M. (2008). *Métodos generalizados para el cálculo de estructuras de cables y simulación de la interacción dinámica catenaria pantógrafo según la norma europea EN50318*. Universidad Pontificia Comillas de Madrid, Madrid.

Vara Mesa, J. M., López Sanz, M., & Verde Marín, J. (2014). Selección de arquitectura y herramientas de programación. En *Desarrollo web en entorno servidor* (p. 293). RAMA

Editorial.

Vargas del Valle, R. J., & Maltés Granados, J. P. (s/f). Programación en Capas, 5.

Velasco, P., Sánchez, L., Laureano, A., & Mora, M. (2009). Un diseño de interfaz: tomando en cuenta los estilos de aprendizaje (p. 11). Presentado en XXII Congreso Nacional y VIII Congreso Internacional de Informática y Computación de la ANIEI, Ensenada, Baja California.

ANEXOS

A. Códigos de los experimentos

A1) Códigos del experimento 1 – Cable Parabólico.

```
function [Ho,Hf,L,Wo,DT] = LeerCableParabCaso1 (ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la%
% función de CableAnalisisEstatico1, que hace referencia al Análisis      %
% Estático de Cables con Cargas distribuidas uniformemente a lo largo de  %
% líneas rectas cuando Ho<Hf.                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                    %
% Para crear una columna más en el archivo txt lo realizamos de la        %
% siguiente manera:                                                         %
% cl=ma{1} donde cl= número de columna; m{1}=posicion de la columna        %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces       %
% cl(1)= "posiciones" (columna 1 fila 1)                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lee el nombre del archivo donde se encuentra los Datos de entrada
fid =fopen([ruta nombreArchivo ],'r');
formato='%f32';
% Se crea la matriz
ma=textscan(fid,formato);
% cl=columna #1 de la matriz ma
cl=ma{1};
%%Datos de Entrada
%Altura o flecha del punto inicial (O)
Ho=cl(3);
%Altura o flecha del punto inicial (F)
Hf=cl(4);
%Distancia horizontal (Vano) entre el punto Inicial y Final
L=cl(5);
%Carga Distribuida
Wo=cl(6);
%Distancia de los tensores
DT=cl(7);
fclose(fid);
end
```

Figura 61. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso1”
Fuente: Autor

```

funcion [Ho,Hf,L,Wo,DT] = LeerCableParabCaso2 (ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la%
% función de CableAnalisisEstatico1, que hace referencia al Análisis      %
% Estático de Cables con Cargas distribuidas uniformemente a lo largo de %
% líneas rectas cuando Ho>Hf.                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                  %
% Para crear una columna más en el archivo txt lo realizamos de la      %
% siguiente manera:                                                       %
% cl=ma{1} donde cl= número de columna; m{1}=posicion de la columna      %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces      %
% cl(1)= "posiciones" (columna 1 fila 1)                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lee el nombre del archivo donde se encuentra los Datos de entrada
fid =fopen([ruta nombreArchivo ],'r');
formato='%f32';
% Se crea la matriz
ma=textscan(fid,formato);
% cl=columna #1 de la matriz ma
cl=ma{1};
%%Datos de Entrada
%Altura o flecha del punto inicial (O)
Ho=cl(3);
%Altura o flecha del punto inicial (F)
Hf=cl(4);
%Distancia horizontal (Vano) entre el punto Inicial y Final
L=cl(5);
%Carga Distribuida
Wo=cl(6);
%Distancia de los tensores
DT=cl(7);
fclose(fid);
end

```

Figura 62. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso2”
Fuente: Autor


```

function [H,L,Wo,DT] = LeerCableParabCaso3 (ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la%
% función de CableAnálisisEstatico1, que hace referencia al Análisis      %
% Estático de Cables con Cargas distribuidas uniformemente a lo largo de %
% líneas rectas cuando Ho=Hf.                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                  %
% Para crear una columna más en el archivo txt lo realizamos de la      %
% siguiente manera:                                                       %
% cl=ma{1} donde cl= número de columna; m{1}=posicion de la columna      %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces      %
% cl(1)= "posiciones" (columna 1 fila 1)                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lee el nombre del archivo donde se encuentra los Datos de entrada
fid =fopen([ruta nombreArchivo ],'r');
formato='%f32';
% Se crea la matriz
ma=textscan(fid,formato);
% cl=columna #1 de la matriz ma
cl=ma{1};
%%Datos de Entrada
%Altura o flecha del punto inicial(O) o Altura o flecha del punto inicial(F)
H=cl(3);
%Distancia horizontal (Vano) entre el punto Inicial y Final
L=cl(4);
%Carga Distribuida
Wo=cl(5);
%Distancia de los tensores
DT=cl(6);
fclose(fid);
end

```

Figura 63. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “LeerCableParabCaso3”
Fuente: Autor

```

function [Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf, DTo, DTf,
str, x, y, A, B, D, wx, wy] = CableParabCasol(Ho,Hf,L,Wo,DT)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           ANÁLISIS ESTÁTICO DE CABLES           %
%   CARGAS DISTRIBUIDAS UNIFORMEMENTE A LO LARGO DE LINEAS RECTAS   %
%   %
%   Un cable suspendido está sometido a una carga vertical distribuida %
%   uniformemente en toda su longitud. El origen del sistema coordenado se %
%   encuentra en el punto más bajo del cable. %
%   La curva descrita por el cable es la catenaria. %
%   %
%           CASO 1:   Ho<Hf   %
%   %
%   Variables de entrada: %
%   %
%   Ho   = Altura desde el punto más bajo del cable hasta el punto O (m) %
%   Hf   = Altura desde el punto más bajo del cable hasta el punto F (m) %
%   L    = Distancia o el vano entre el punto O y el punto F (m) %
%   Wo   = Carga distribuida uniforme. a lo largo de líneas rectas (KN/m) %
%   DT   = Distancia entre los tensores (m) %
%   %
%   Variables de salida: %
%   %
%   To   = Tensión en el punto O (KN) %
%   Tf   = Tensión en el punto F (KN) %
%   Tmax = Tensión máxima entre la tensión del punto O y punto F (KN) %
%   Tmin = Tensión en el punto más bajo del cable, cuando m=0 (KN) %
%   Tten = Tensión en los tensores del cable (KN) %
%   S    = Longitud del cable medida desde el punto O hasta el punto F (m) %
%   Lo   = Distancia entre punto más bajo del cable hasta el punto O (m) %
%   Lf   = Distancia entre punto más bajo del cable hasta el punto F (m) %
%   DTo  = Distancia entre el primer tensor y el punto O (m) %
%   DTf  = Distancia entre el último tensor y el punto F (m) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----Desarrollo del Análisis-----
%% cuando Ho=0

if Ho==0
    %Lo: es la distancia entre el punto inicial hasta el punto con de
    %máxima deformación x (-)
    Lo=0;
    %Lf: es la distancia desde el punto con de máxima deformación hasta el
    %punto final x (+)
    Lf=L;
    %a=es un parámetro.
    u=2*Hf/L^2;
    % Tmin: es la tensión en el cable en el punto con de máxima deformación
    Tmin=Wo/u;
    % To: es la tensión en el cable en el punto inicial.
    To=Tmin*sqrt(1+u^2*Lo^2);
    % Tf: es la tensión en el cable en el punto final.
    Tf=Tmin*sqrt(1+u^2*Lf^2);
    % Tmax: es la tensión máxima en el cable.
    Tmax=max(To,Tf);
    % Tten: es la tensión en los tensores.
    Tten=Wo*DT;
    % Nten: es el número de tensores en toda la longitud del cable.
    Nten=floor(L/DT);
    % Dto: es la distancia entre el punto inicial y el primer tensor.
    DTo=(L-(Nten-1)*DT)/2;
    % Dtf: es la distancia entre el último tensor y el punto final.
    DTf=DTo;
    % So: Longitud desde el punto inicial al punto con de máxima deformación.
    So=abs(1/2*(Lo*sqrt(1+u^2*Lo^2)+1/u*log(u*Lo+sqrt(1+u^2*Lo^2))));
    % Sf: Longitud desde el punto con máxima de formación hasta el punto final
    Sf=abs(1/2*(Lf*sqrt(1+u^2*Lf^2)+1/u*log(u*Lf+sqrt(1+u^2*Lf^2))));

```

```

    % Longitud total del cable.
    S=So+Sf;
%% caso contrario Ho>0
else
    %La curva descrita por el cable es la parábola  $Y=1/2*a*x^2$ :por lo tanto
    %entre dos puntos para conocer la distancia x es:  $a*x^2+b*x+c=0$ 
    a=1-Hf/Ho;
    b=2*L;
    c=L^2;
    d=b^2-4*a*c;
    if d>0
        x1=(-b+sqrt(d))/(2*a);
        %disp('valor positivo de la raiz')
    elseif d==0
        x1=-b/(2*a);
        %disp('la raiz vale 0')
    else
        x1=(-b+1i*sqrt(-d))/(2*a);
        %disp('la raiz es negativa')
    end
    %Lo: distancia entre el punt inicial hasta el punto con de máxima
    %deformación x (-)
    Lo=x1;
    %Lf: distancia desde el punt con de máxima deformación hasta el punto
    %final x (+)
    Lf=L+Lo;
    %a=es un parámetro.
    u=2*Ho/Lo^2;
    % Tmin: es la tensión en el cable en el punto con de máxima deformación
    Tmin=Wo/u;
    % To: es la tensión en el cable en el punto inicial.
    To=Tmin*sqrt(1+u^2*Lo^2);
    % Tf: es la tensión en el cable en el punto final.
    Tf=Tmin*sqrt(1+u^2*Lf^2);
    % Tmax: es la tensión máxima en el cable.
    Tmax=max(To,Tf);
    % Tten: es la tension en los tensores.
    Tten=Wo*DT;
    % Nten: es el número de tensores en toda la longitud del cable.
    Nten=floor(L/DT);
    % Dto: es la distancia entre el punto inicial y el primer tensor.
    Dto=(L-(Nten-1)*DT)/2;
    % Dtf: es la distancia entre el último tensor y el punto final.
    Dtf=Dto;
    % So: Longitud desde el punto inicial al punt con de máxima de formación
    So=abs(1/2*(Lo*sqrt(1+u^2*Lo^2)+1/u*log(u*Lo+sqrt(1+u^2*Lo^2))));
    % Longitud desde el punt con de máxima de formación hasta el punto final
    Sf=abs(1/2*(Lf*sqrt(1+u^2*Lf^2)+1/u*log(u*Lf+sqrt(1+u^2*Lf^2))));
    % Longitud total del cable.
    S=So+Sf;
%% finalización del bucle
end
% El CABLE

    %La curva descrita por el cable es la parábola  $y=1/2*a*x^2$  donde:  $y=Ho$   $x=Lo$ 
    u=2*Hf/Lf^2;
    % x: coordenadas en x para el cable limites [Lo,Lf]
    x=Lo:0.05:Lf;
    % y: ecuación de la curva descrita por el cable es la parábola en funcion
    de x
    y=u/2*x.^2;
    %Determinación de los datos o puntos en "Y"
    % Es un proceso repetitivo hasta el número de tensores
    b=Nten;
    bcol=1;
    B=ones(b,bcol);

```

```

for p=1:Nten
    B(p)=-DT;
end
nrows =Nten;
ncols =1;
A= ones(nrows,ncols);
dy=DT;
DTY=dy;
for r = 1:nrows
    if r == 1
        xxy=Lo+DTo;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    elseif r>=2 && r< Nten
        for v=2:r
            if v==2
                pty=DTo+DTY;
                DTY=pty;
            elseif v>2
                DTY=pty+dy-DTo;
            end
        end
        xxy=Lo+DTY;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    else
        xxy=Lf-DTf;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    end
end

% C: es una matriz de los tensores que corresponde a la primera fila datos
% de y (en la curva) y en la segunda fila datos de y (en la línea recta)
%C=horcate(A,B);
%Determinación de los datos de "X"
% Es un proceso repetitivo hasta el número de tensores
drows=Nten;
dcol=1;
D=ones(drows,dcol);
dx=DT;
DTX=dx;
for rx = 1:drows
    if rx == 1
        xxx=Lo+DTo;
        D(rx) =xxx;
    elseif rx>=2 && rx< Nten
        for vx=2:rx
            if vx==2
                ptx=DTo+DTX;
                DTX=ptx;
            elseif vx>2
                DTX=ptx+dx-DTo;
            end
        end
        xxx=Lo+DTX;
        D(rx) =xxx;
    else
        xxx=Lf-DTf;
        D(rx) =xxx;
    end
end

% ()
A=A';
B=B';
D=D';

```

```

    % E: es una matriz de los tensores que corresponde a la primera
    fila datos
    % de x (en la curva) y en la segunda fila datos de x (en la línea
    recta)
    %E=horzcat(D,D);
    % xx: son las coordenadas en x
    %xx=E';
    % yy: son las coordenadas en y
    %yy=C';
    %subplot(222),plot(xx,yy,'r')
    %title('TENSORES');
    %axis off
    %%
    %figura la carga distribuida
    % Do: es un parámetro
    Do=(Hf-Ho)*4/5;
    % wx: es la coordenada x para graficar la carga distribuida
    wx=[Lo Lf Lo];
    % wy: es la coordenada y para graficar la carga distribuida
    wy=[-DT -DT -Do -Do];
    %subplot(223),plot(wx,wy,'B')
    %title('CARGA DISTRIBUIDA');
    %limites de la grafica
    %wxmin=Lo-DT/2;
    %wxmax=Lf+DT/2;
    %wymn=- (Do+DT/2);
    %wymax=0;
    %axis([wxmin wxmax wymn wymax])
    %patch(wx,wy,'r')
    %axis off
    %%
    % Todas las Graficas
    %subplot(224),plot(x,y,'k',xx,yy,'r',wx,wy,'B')
    %title('CABLE - TENSORES - CARGA DISTRIBUIDA');
    %patch(wx,wy,'r')
    %axis([-L+Lf-DT/2 Lf+DT/2 -Do-DT Hf])
    %axis off
    %%
    %denominación de la herramienta : programa 1
    %%
    % fecha y hora
    str = datestr(now);
end

```

Figura 64. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso1”
Fuente: Autor

```

function [Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax, Tten, Nten, S, Lo, Lf, DTo, DTf,
str, x, y, A, B, D, wx, wy] = CableParabCaso2(Ho,Hf,L,Wo,DT)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               ANÁLISIS ESTÁTICO DE CABLES
%
%          CARGAS DISTRIBUIDAS UNIFORMEMENTE A LO LARGO DE LINEAS RECTAS
%
%
% Un cable suspendido está sometido a una carga vertical distribuida
% uniformemente en toda
% su longitud. El origen del sistema coordenado se encuentra en el punto
% másbajo del cable.
% La curva descrita por el cable es la catenaria.
%
%
%                               CASO 2:   Ho>Hf
%
%
%          [Tmin To Tf Tmax Tten S] = CCDULLR1(Ho,Hf,L,Wo,DT)
%
%
% Variables de entrada:
%
%
% Ho   = Altura desde el punto más bajo del cable hasta el punto O (m)
% Hf   = Altura desde el punto más bajo del cable hasta el punto F (m)
% L    = Distancia o el vano entre el punto O y el punto F (m)
% Wo   = Carga distribuida uniforme a lo largo de líneas rectas (KN/m)
% DT   = Distancia entre los tensores (m)
%
% Variables de salida:
%
%
% To   = Tensión en el punto O (KN)
% Tf   = Tensión en el punto F (KN)
% Tmax = Tensión máxima entre la tensión del punto O y punto F (KN)
% Tmin = Tensión en el punto más bajo del cable, cuando la me=0 (KN)
% Tten = Tensión en los tensores del cable (KN)
% S    = Longitud del cable medida desde el punto O hasta el punto F (m)
% Lo   = Distancia entre punto más bajo del cable hasta el punto O (m)
% Lf   = Distancia entre punto más bajo del cable hasta el punto F (m)
% DTo  = Distancia entre el primer tensor y el punto O (m)
% DTf  = Distancia entre el último tensor y el punto F (m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----Desarrollo del Análisis-----
%% cuando Ho=0
if Ho==0

    %Lo: es la distancia entre el punto inicial hasta el punto con de
    %máxima deformación x (-)
    Lo=L;
    %Lf: es la distancia desde el punto con de máxima deformación hasta el
    %punto final x (+)
    Lf=0;
    %a=es un parámetro.
    u=2*Ho/L^2;
    % Tmin: es la tensión en el cable en el punto con de máxima deformación
    Tmin=Wo/u;
    % To: es la tensión en el cable en el punto inicial.
    To=Tmin*sqrt(1+u^2*Lo^2);
    % Tf: es la tensión en el cable en el punto final.
    Tf=Tmin*sqrt(1+u^2*Lf^2);
    % Tmax: es la tensión máxima en el cable.
    Tmax=max(To,Tf);
    % Tten: es la tension en los tensores.
    Tten=Wo*DT;
    % Nten: es el número de tensores en toda la longitud del cable.
    Nten=floor(L/DT);
    % Dto: es la distancia entre el punto inicial y el primer tensor.
    DTo=(L-(Nten-1)*DT)/2;
    % Dtf: es la distancia entre el último tensor y el punto final.
    DTf=DTo;
    % So: Longitud desde el punto inicial al punto con de máxima deformación

```

```

So=abs(1/2*(Lo*sqrt(1+u^2*Lo^2)+1/u*log(u*Lo+sqrt(1+u^2*Lo^2))));
% Sf: Longitud desde el punt con máxima de formación hasta el punt final
Sf=abs(1/2*(Lf*sqrt(1+u^2*Lf^2)+1/u*log(u*Lf+sqrt(1+u^2*Lf^2))));
% Longitud total del cable.
S=So+Sf;
%% caso contrario Ho>0
else
    %La curva descrita por el cable es la parábola  $Y=1/2*a*x^2$ : por lo tanto
    %entre dos puntos para conocer la distancia x es:  $a*x^2+b*x+c=0$ 
    a=1-Hf/Ho;
    b=2*L;
    c=L^2;
    d=b^2-4*a*c;
    if d>0
        x1=(-b+sqrt(d))/(2*a);
        %disp('valor positivo de la raiz')
    elseif d==0
        x1=-b/(2*a);
        %disp('la raiz vale 0')
    else
        x1=(-b+1i*sqrt(-d))/(2*a);
        %disp('la raiz es negativa')
    end
    %Lo: es la distancia entre el punto inicial hasta el punto con de
    % máxima deformación x (-)
    Lo=x1;
    %Lf: es la distancia desde el punto con de máxima deformación hasta el
    %punto final x (+)
    Lf=L+Lo;
    %a=es un parámetro.
    u=2*Ho/Lo^2;
    % Tmin: es la tensión en el cable en el punto con de máxima deformación
    Tmin=Wo/u;
    % To: es la tensión en el cable en el punto inicial.
    To=Tmin*sqrt(1+u^2*Lo^2);
    % Tf: es la tensión en el cable en el punto final.
    Tf=Tmin*sqrt(1+u^2*Lf^2);
    % Tmax: es la tensión máxima en el cable.
    Tmax=max(To,Tf);
    % Tten: es la tension en los tensores.
    Tten=Wo*DT;
    % Nten: es el número de tensores en toda la longitud del cable.
    Nten=floor(L/DT);
    % Dto: es la distancia entre el punto inicial y el primer tensor.
    Dto=(L-(Nten-1)*DT)/2;
    % Dtf: es la distancia entre el último tensor y el punto final.
    Dtf=Dto;
    % So: Longitud desde el punto inicial al punt con de máxima de formación
    So=abs(1/2*(Lo*sqrt(1+u^2*Lo^2)+1/u*log(u*Lo+sqrt(1+u^2*Lo^2))));
    % Sf: Longitud desde el punt con máxima de formación hasta el punt final
    Sf=abs(1/2*(Lf*sqrt(1+u^2*Lf^2)+1/u*log(u*Lf+sqrt(1+u^2*Lf^2))));
    % Longitud total del cable.
    S=So+Sf;
end
% El CABLE
%La curva descrita por el cable es la parábola  $y=1/2*a*x^2$  donde:  $y=Ho$   $x=Lo$ 
u=2*Ho/Lo^2;
% x: coordenadas en x para el cable limites [Lo,Lf]
x=Lo:0.05:Lf;
% y: ecuación de la curva descrita por el cable es la parábola en
%funcion de x
y=u/2*x.^2;
%FIGURA DE LOS TENSORES
%Determinación de los datos o puntos en "Y"
% Es un proceso repetitivo hasta el número de tensores
b=Nten;
bcol=1;

```

```

B=ones(b,bcol);
for p=1:Nten
    B(p)=-DT;
end
nrows =Nten;
ncols =1;
A = ones(nrows,ncols);
dy=DT;
DTY=dy;
for r = 1:nrows
    if r == 1
        xxy=Lo+DTo;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    elseif r>=2 && r< Nten
        for v=2:r
            if v==2
                pty=DTo+DTY;
                DTY=pty;
            elseif v>2
                DTY=pty+dy-DTo;
            elseif v>2
                DTY=pty+dy-DTo;
            end
        end
        xxy=Lo+DTY;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    else
        xxy=Lf-DTf;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    end
end
% C: es una matriz de los tensores que corresponde a la primera fila datos
% de y (en la curva) y en la segunda fila datos de y (en la línea recta)
%C=[A,B];
    %Determinación de los datos de "X"
    % Es un proceso repetitivo hasta el número de tensores
drows=Nten;
dcol=1;
D=ones(drows,dcol);
dx=DT;
DTX=dx;
for rx = 1:drows
    if rx == 1
        xxx=Lo+DTo;
        D(rx) =xxx;
    elseif rx>=2 && rx< Nten
        for vx=2:rx
            if vx==2
                ptx=DTo+DTX;
                DTX=ptx;
            elseif vx>2
                DTX=ptx+dx-DTo;
            end
        end
        xxx=Lo+DTX;
        D(rx) =xxx;
    else
        xxx=Lf-DTf;
        D(rx) =xxx;
    end
end
A=A';
B=B';
D=D';

```



```

%figura la carga distribuida
% Do: es un parámetro
Do=(Ho-Hf)*4/5;
% wx: es la coordenada x para graficar la carga distribuida
wx=[Lo Lf Lo];
% wy: es la coordenada y para graficar la carga distribuida
wy=[-DT -DT -Do -Do];
%subplot(223),plot(wx,wy,'B')
%title('CARGA DISTRIBUIDA');
%límites de la grafica
%wxmin=Lo-DT/2;
%wxmax=Lf+DT/2;
%wymn=- (Do+DT/2);
%wymax=0;
%axis([wxmin wxmax wymn wymax])
%patch(wx,wy,'r')
%axis off
% Todas las Graficas
%subplot(224),plot(x,y,'k',xx,yy,'r',wx,wy,'B')
%title('CABLE - TENSORES - CARGA DISTRIBUIDA');
%patch(wx,wy,'r')
%axis([-L+Lf-DT/2 Lf+DT/2 -Do-DT Ho])
%axis off
%%
%denominación de la herramienta : programa 2
%%
% fecha y hora
str = datestr(now);
end

```

Figura 65. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso2”
Fuente: Autor


```

% Dtf: es la distancia entre el último tensor y el punto final.
Dtf=DT0;
% So: Longitud desde el punto inicial al punt con de máxima de formación
So=abs(1/2*(Lo*sqrt(1+u^2*Lo^2)+1/u*log(u*Lo+sqrt(1+u^2*Lo^2))));
% Sf: Longitud desde el punto con máxima de formación hasta el punt final
Sf=abs(1/2*(Lf*sqrt(1+u^2*Lf^2)+1/u*log(u*Lf+sqrt(1+u^2*Lf^2))));
% Longitud total del cable.
S=So+Sf;
%-----FIGURAS-GRAFICAS-----
% El CABLE
%La curva descrita por el cable es la paráb  $y=1/2*a*x^2$  donde: $y=H_0$   $x=L_0$ 
u=2*H0/L0^2;
% x: coordenadas en x para el cable limites [L0,Lf]
x=L0:0.05:Lf;
% y: ecuación de la curva descrita por el cable es la paráb en
%funcion de x
y=u/2*x.^2;
y=u/2*x.^2;
%subplot(221),plot(x,y,'K')
%grid on;
% generar una gráfica en done la curva tiene un estilo de sombra
%hline1 = plot(x,y,'K');
%ax = gca;
%hline2 = line(x+.06,y,...
% 'LineWidth',4,...
% 'Color',[.5 .5 .5],...
% 'Parent',ax);
%set(gca,'Children',[hline1 hline2])
%axis off
%title('CABLE');
%%
%FIGURA DE LOS TENSORES
%Determinación de los datos o puntos en "Y"
% Es un proceso repetitivo hasta el número de tensores
b=Nten;
bcol=1;
B=ones(b,bcol);
for p=1:Nten
    B(p)=-DT;
end
nrows =Nten;
ncols =1;
A = ones(nrows,ncols);
dy=DT;
DTY=dy;
for r = 1:nrows
    if r == 1
        xxy=L0+DT0;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    elseif r>=2 && r< Nten
        for v=2:r
            if v==2
                pty=DT0+DTY;
                DTY=pty;
            elseif v>2
                DTY=pty+dy-DT0;
            end
        end
        xxy=L0+DTY;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    else
        xxy=Lf-DTf;
        yy=u/2*(xxy)^2;
        A(r) =yy;
    end
end

```

```

end
% C: es una matriz de los tensores que corresponde a la primera fila datos
% de y (en la curva) y en la segunda fila datos de y (en la línea recta)
% C=[A,B];
    %Determinación de los datos de "X"
    % Es un proceso repetitivo hasta el número de tensores
    drows=Nten;
    dcol=1;
    D=ones(drows,dcol);
    dx=DT;
    DTX=dx;
    for rx = 1:drows
        if rx == 1
            xxx=Lo+DTo;
            D(rx) =xxx;
        elseif rx>=2 && rx< Nten
            for vx=2:rx
                if vx==2
                    ptx=DTo+DTX;
                    DTX=ptx;
                elseif vx>2
                    DTX=ptx+dx-DTo;
                end
            end
            xxx=Lo+DTX;
            D(rx) =xxx;
        else
            xxx=Lf-DTf;
            D(rx) =xxx;
        end
    end
    A=A';
    B=B';
    D=D';
% E: es una matriz de los tensores que corresponde a la primera fila datos
% de x (en la curva) y en la segunda fila datos de x (en la línea recta)
    %E=[D,D];
    % xx: son las coordenadas en x
    %xx=E';
    % yy: son las coordenadas en y
    %yy=C';
    %subplot(222),plot(xx,yy,'r')
    %title('TENSOIRES');
    %axis off
    %figura la carga distribuida
    % Do: es un parámetro
    Do=(H)*4/5;
    % wx: es la coordenada x para graficar la carga distribuida
    wx=[Lo Lf Lo];
    % wy: es la coordenada y para graficar la carga distribuida
    wy=[-DT -DT -Do -Do];
    %subplot(223),plot(wx,wy,'B')
    %title('CARGA DISTRIBUIDA');
    %límites de la grafica
    %wxmin=Lo-DT/2;
    %wymn=- (Do+DT/2);
    %wymax=0;
    %axis([wxmin wxmax wymn wymax])
    %patch(wx,wy,'r')
    %axis off
    % fecha y hora
    str = date;
end

```

Figura 66. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “CableParabCaso3”
Fuente: Autor

```

function [Tmin] = SalidaCableParabCasol(Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax,
Tten, Nten, S, Lo, Lf, DTo, DTf, str, x, y, A, B, D, wx, wy, ruta,
nameFileUser, nameFileExperimento,nameFileUserEnglish)
%%
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SÍSMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, ' DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, ' SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE PARABÓLICO\n');
fprintf(text_tiket, 'CASO 1. : Ho<Hf : Flecha en el punto O es menor a la
flecha del punto F\n');
fprintf(text_tiket,[ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores : Msc. Duque, E., Ing. Quiñones. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
| | | \n');
fprintf(text_tiket,[ ' Altura o flecha del punto O | Ho | m
| ' num2str(Ho) '\n']);
fprintf(text_tiket,[ ' Altura o flecha del punto F | Hf | m
| ' num2str(Hf) '\n']);
fprintf(text_tiket,[ ' Vano entre el punto O y F | L | m
| ' num2str(L) '\n']);
fprintf(text_tiket,[ ' Carga distribuida línea recta | Wo | KN
| ' num2str(Wo) '\n']);
fprintf(text_tiket,[ ' Distancia entre tensores | DT | m
| ' num2str(DT) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL| UNITY

```

```

| MAGNITUDE\n');
fprintf(text_tiket,
'
|_____|\n');
fprintf(text_tiket, [ ' Tensión mínima | Tmin | KN
| ' num2str(Tmin) '\n']];
fprintf(text_tiket, [ ' Tensión punto O | To | KN
| ' num2str(To) '\n']];
fprintf(text_tiket, [ ' Tensión punto F | Tf | KN
| ' num2str(Tf) '\n']];
fprintf(text_tiket, [ ' Tensión máxima | Tmax | KN
| ' num2str(Tmax) '\n']];
fprintf(text_tiket, [ ' Tensión en los Tensores | Tten | KN
| ' num2str(Tten) '\n']];
fprintf(text_tiket, [ ' Número de Tensores | Nten | #
| ' num2str(Nten) '\n']];
fprintf(text_tiket, [ ' Longitud del Cable | S | m
| ' num2str(S) '\n']];
fprintf(text_tiket, [ ' Distancia o vano punto O | Lo | m
| ' num2str(Lo) '\n']];
fprintf(text_tiket, [ ' Distancia o vano punto F | Lf | m
| ' num2str(Lf) '\n']];
fprintf(text_tiket, [ ' Dist. punto O y Tensor | DTo | m
| ' num2str(DTo) '\n']];
fprintf(text_tiket, [ ' Dist. punto F y Tensor | DTf | m
| ' num2str(DTf) '\n']];
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
|_____|\n');
);
fprintf(text_tiket,
'
|_____|\n');
);
fclose(text_tiket);
%%
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'
|_____|\n');
);
fprintf(text_tiket,
'
|_____|\n');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, ' DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, ' SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORY: CABLES \n');
fprintf(text_tiket, 'EXPERIMENT: PARABOLIC CABLE\n');
fprintf(text_tiket, 'CASE 1. : Ho<Hf : Arrow at point (O) is less than the
arrow of point F\n');
fprintf(text_tiket, [ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, [ 'Authors : Msc. Duque, E., Ing. Quiñonez, S. Pélaez, D.,
\n']);
fprintf(text_tiket,
'
|_____|\n'

```



```

text_tiket=fopen([ruta nameFileExperimento], 'a');
%%fprintf(text_tiket, [ 'DATOS DE SALIDA \n']);
fprintf(text_tiket, [ 'Tmin ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ 'To ' num2str(To) '\n']);
fprintf(text_tiket, [ 'Tf ' num2str(Tf) '\n']);
fprintf(text_tiket, [ 'Tmax ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ 'Tten ' num2str(Tten) '\n']);
fprintf(text_tiket, [ 'Nten ' num2str(Nten) '\n']);
fprintf(text_tiket, [ 'S ' num2str(S) '\n']);
fprintf(text_tiket, [ 'Lo ' num2str(Lo) '\n']);
fprintf(text_tiket, [ 'Lf ' num2str(Lf) '\n']);
fprintf(text_tiket, [ 'DTo ' num2str(DTo) '\n']);
fprintf(text_tiket, [ 'DTf ' num2str(DTf) '\n']);
%fprintf(text_tiket, [ 'CABLE \n']);
fprintf(text_tiket, ['x' ' ']);
dlmwrite([ruta nameFileExperimento], x, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket, ['y' ' ']);
dlmwrite([ruta nameFileExperimento], y, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
dlmwrite([ruta nameFileExperimento], y, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
%fprintf(text_tiket, [ 'Tensores \n']);
fprintf(text_tiket, [ 'xx' ' ']);
dlmwrite([ruta nameFileExperimento], (A(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ']);
dlmwrite([ruta nameFileExperimento], (B(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ']);
dlmwrite([ruta nameFileExperimento], (D(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
%fprintf(text_tiket, [ 'CARGA DISTRIBUIDA \n']);
fprintf(text_tiket, [ 'wx' ' ']);
dlmwrite([ruta nameFileExperimento], (wx(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'wy' ' ']);
dlmwrite([ruta nameFileExperimento], (wy(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
%fprintf(text_tiket, [ 'CABLES-TENSORES-CARGA DISTRIBUIDORA\n']);
fclose(text_tiket);
end

```

Figura 67. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso1”
Fuente: Autor


```

function [Tmin] = SalidaCableParabCaso2(Ho,Hf,L,Wo,DT, Tmin, To, Tf, Tmax,
Tten, Nten, S, Lo, Lf, DTO, DTf, str, x, y, A, B, D, wx, wy, ruta,
nameFileUser, nameFileExperimento,nameFileUserEnglish)
    text_tiket=fopen([ruta nameFileUser], 'wt');
    fprintf(text_tiket,
    '
    );
    fprintf(text_tiket,
    '
    );
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '          LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '          GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SISMOLOGIA (GRISS-UTPL)\n');
    fprintf(text_tiket, '          DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
    fprintf(text_tiket, '          SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
    fprintf(text_tiket, '          UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
    fprintf(text_tiket, ' -----
-----\n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, 'LABORATORIO: CABLES \n');
    fprintf(text_tiket, 'EXPERIMENTO: CABLE PARABÓLICO\n');
    fprintf(text_tiket, 'CASO    2. : Ho>Hf : Flecha en el punto O es mayor a la
flecha del punto F\n');
    fprintf(text_tiket, [ 'FECHA      : ' num2str(str) '\n']);
    fprintf(text_tiket, 'Autores    : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
    fprintf(text_tiket,
    '
    );
    fprintf(text_tiket,
    '
    );
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, 'INPUT DATA \n');
    fprintf(text_tiket, '_____ \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '          DESCRIPTION                      |SYMBOL| UNITY
| MAGNITUDE\n');
    fprintf(text_tiket,
    '
    );
    fprintf(text_tiket, [ '          Altura o flecha del punto O                      | Ho   |  m
| ' num2str(Ho) '\n']);
    fprintf(text_tiket, [ '          Altura o flecha del punto F                      | Hf   |  m
| ' num2str(Hf) '\n']);
    fprintf(text_tiket, [ '          Vano entre el punto O y F                      | L    |  m
| ' num2str(L) '\n']);
    fprintf(text_tiket, [ '          Carga distribuida linea recta                  | Wo   |  KN
| ' num2str(Wo) '\n']);
    fprintf(text_tiket, [ '          Distancia entre tensores                      | DT   |  m
| ' num2str(DT) '\n']);
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, 'ANALYSIS OUPUT \n');
    fprintf(text_tiket, '_____ \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '          DESCRIPTION                      |SYMBOL| UNITY

```

```

| MAGNITUDE\n');
fprintf(text_tiket,
'
|_____|_____|_____|\n');
fprintf(text_tiket, [ ' Tensión mínima | Tmin | KN
| ' num2str(Tmin) '\n']];
fprintf(text_tiket, [ ' Tensión punto O | To | KN
| ' num2str(To) '\n']];
fprintf(text_tiket, [ ' Tensión punto F | Tf | KN
| ' num2str(Tf) '\n']];
fprintf(text_tiket, [ ' Tensión máxima | Tmax | KN
| ' num2str(Tmax) '\n']];
fprintf(text_tiket, [ ' Tensión en los Tensores | Tten | KN
| ' num2str(Tten) '\n']];
fprintf(text_tiket, [ ' Número de Tensores | Nten | #
| ' num2str(Nten) '\n']];
fprintf(text_tiket, [ ' Longitud del Cable | S | m
| ' num2str(S) '\n']];
fprintf(text_tiket, [ ' Distancia o vano punto O | Lo | m
| ' num2str(Lo) '\n']];
fprintf(text_tiket, [ ' Distancia o vano punto F | Lf | m
| ' num2str(Lf) '\n']];
fprintf(text_tiket, [ ' Dist. punto O y Tensor | DTo | m
| ' num2str(DTo) '\n']];
fprintf(text_tiket, [ ' Dist. punto F y Tensor | DTf | m
| ' num2str(DTf) '\n']];
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
|_____|\n'
);
fprintf(text_tiket,
'
|_____|\n'
);
%%
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
%%
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');

fprintf(text_tiket,
'
|_____|\n'
);
fprintf(text_tiket,
'
|_____|\n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, ' DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, ' SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORY: CABLES \n');
fprintf(text_tiket, 'EXPERIMENT: PARABOLIC CABLE\n');
fprintf(text_tiket, 'CASE 2. : Ho>Hf : Arrow at point (O) is greater than
the arrow of point F\n');
fprintf(text_tiket, [ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Authors : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');

```

```

fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'_____ \n');
fprintf(text_tiket, [ ' Height or arrow of starting point (O) | Ho | m
| ' num2str(Ho) '\n' ]);
fprintf(text_tiket, [ ' Height or arrow of end point (F) | Hf | m
| ' num2str(Hf) '\n' ]);
fprintf(text_tiket, [ ' Vain between start and end point | L | m
| ' num2str(L) '\n' ]);
fprintf(text_tiket, [ ' Load distributed in a straight line | Wo | KN
| ' num2str(Wo) '\n' ]);
fprintf(text_tiket, [ ' Distance between tensioners | DT | m
| ' num2str(DT) '\n' ]);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'_____ \n');
fprintf(text_tiket, [ ' Minimum tension | Tmin
| ' num2str(Tmin) '\n' ]);
fprintf(text_tiket,
'_____ \n');
fprintf(text_tiket, [ ' Minimum tension | Tmin | KN
| ' num2str(Tmin) '\n' ]);
fprintf(text_tiket, [ ' Maximun tension | To | KN
| ' num2str(To) '\n' ]);
fprintf(text_tiket, [ ' Tension at the starting point (O) | Tf | KN
| ' num2str(Tf) '\n' ]);
fprintf(text_tiket, [ ' Tension at the end point (F) | Tmax | KN
| ' num2str(Tmax) '\n' ]);
fprintf(text_tiket, [ ' Tension on the tensioners | Tten | KN
| ' num2str(Tten) '\n' ]);
fprintf(text_tiket, [ ' Number of tensioners | Nten | #
| ' num2str(Nten) '\n' ]);
fprintf(text_tiket, [ ' Cable length | S | m
| ' num2str(S) '\n' ]);
fprintf(text_tiket, [ ' Distance to the starting point (O) | Lo | m
| ' num2str(Lo) '\n' ]);
fprintf(text_tiket, [ ' Distance to the end point (F) | Lf | m
| ' num2str(Lf) '\n' ]);
fprintf(text_tiket, [ ' Distance: starting point and tensioner | DTo | m
| ' num2str(DTo) '\n' ]);
fprintf(text_tiket, [ ' Distance: end point and tensioner | DTf | m
| ' num2str(DTf) '\n' ]);
fprintf(text_tiket, ' \n');
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket,
'_____ \n'
);

```

```

);
fclose(text_tiket);
%%
%%% DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento], 'wt');
text_tiket=fopen([ruta nameFileExperimento], 'a');
fprintf(text_tiket, [ 'DATOS DE SALIDA \n']);
fprintf(text_tiket, [ 'Tmin ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ 'To ' num2str(To) '\n']);
fprintf(text_tiket, [ 'Tf ' num2str(Tf) '\n']);
fprintf(text_tiket, [ 'Tmax ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ 'Tten ' num2str(Tten) '\n']);
fprintf(text_tiket, [ 'Nten ' num2str(Nten) '\n']);
fprintf(text_tiket, [ 'S ' num2str(S) '\n']);
fprintf(text_tiket, [ 'Lo ' num2str(Lo) '\n']);
fprintf(text_tiket, [ 'Lf ' num2str(Lf) '\n']);
fprintf(text_tiket, [ 'DTo ' num2str(DTo) '\n']);
fprintf(text_tiket, [ 'DTf ' num2str(DTf) '\n']);
fprintf(text_tiket, [ 'CABLE \n']);
fprintf(text_tiket, ['x' ' ']);
dlmwrite([ruta nameFileExperimento], x, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket, ['y' ' ']);
dlmwrite([ruta nameFileExperimento], y, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'Tensores \n']);
fprintf(text_tiket, [ 'xx' ' ']);
dlmwrite([ruta nameFileExperimento], (A(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ']);
dlmwrite([ruta nameFileExperimento], (B(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ']);
dlmwrite([ruta nameFileExperimento], (D(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'CARGA DISTRIBUIDA \n']);
fprintf(text_tiket, [ 'wx' ' ']);
dlmwrite([ruta nameFileExperimento], (wx(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'wy' ' ']);
dlmwrite([ruta nameFileExperimento], (wy(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'CABLES-TENSORES-CARGA DISTRIBUIDORA\n']);
fclose(text_tiket);
end

```

Figura 68. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso2”
Fuente: Autor

```

function [Tmin] = SalidaCableParabCaso3(H,L,Wo,DT, Tmin, To, Tf, Tmax, Tten,
Nten, S, Lo, Lf, DTo, DTf, str, x, y, A, B, D, wx, wy, ruta, nameFileUser,
nameFileExperimento,nameFileUserEnglish)
%%
    text_tiket=fopen([ruta nameFileUser], 'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '    GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SÍSMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, '          DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, '          SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, '          UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE PARABÓLICO\n');
fprintf(text_tiket, 'CASO    3. : Ho=Hf : Flecha en el punto O es igual a la
flecha del punto F\n');
fprintf(text_tiket, [ 'FECHA      : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores    : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
          DESCRIPTION          |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
| _____ | _____ | _____ \n');
fprintf(text_tiket, [ '    Altura o flecha          | H    | m
| ' num2str(H) '\n']);
fprintf(text_tiket, [ '    Vano entre el punto O y F          | L    | m
| ' num2str(L) '\n']);
fprintf(text_tiket, [ '    Carga distribuida linea recta          | Wo   | KN
| ' num2str(Wo) '\n']);
fprintf(text_tiket, [ '    Distancia entre tensores          | DT   | m
| ' num2str(DT) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');

```

```

fprintf(text_tiket, '
| MAGNITUDE\n');
fprintf(text_tiket,
'
| | | | | \n');
fprintf(text_tiket, [ ' Tensión mínima | Tmin | KN
| ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ ' Tensión punto O | To | KN
| ' num2str(To) '\n']);
fprintf(text_tiket, [ ' Tensión punto F | Tf | KN
| ' num2str(Tf) '\n']);
fprintf(text_tiket, [ ' Tensión máxima | Tmax | KN
| ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ ' Tensión en los Tensores | Tten | KN
| ' num2str(Tten) '\n']);
fprintf(text_tiket, [ ' Número de Tensores | Nten | #
| ' num2str(Nten) '\n']);
fprintf(text_tiket, [ ' Longitud del Cable | S | m
| ' num2str(S) '\n']);
fprintf(text_tiket, [ ' Distancia o vano punto O | Lo | m
| ' num2str(Lo) '\n']);
fprintf(text_tiket, [ ' Distancia o vano punto F | Lf | m
| ' num2str(Lf) '\n']);
fprintf(text_tiket, [ ' Dist. punto O y Tensor | DTo | m
| ' num2str(DTo) '\n']);
fprintf(text_tiket, [ ' Dist. punto F y Tensor | DTf | m
| ' num2str(DTf) '\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
| | | | | \n');
);
fprintf(text_tiket,
'
| | | | | \n');
);
fclose(text_tiket);
%%
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'
| | | | | \n');
);
fprintf(text_tiket,
'
| | | | | \n');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, ' DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, ' SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORY: CABLES \n');
fprintf(text_tiket, ' EXPERIMENT: PARABOLIC CABLE\n');
fprintf(text_tiket, ' CASE 3. : Ho>Hf : Arrow at point (O) is greater than
the arrow of point F\n');
fprintf(text_tiket, [ ' FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, ' Authors : Msc. Duque, E., Ing. Quiñonez, S. Pélaez, D.,
\n');
fprintf(text_tiket,
'
| | | | | \n');

```



```

'
);
fprintf(text_tiket,
'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
|_____ |_____ |_____ \n');
fprintf(text_tiket, [ ' Height or arrow | H | m
| ' num2str(H) '\n']);
fprintf(text_tiket, [ ' Vain between start and end point | L | m
| ' num2str(L) '\n']);
fprintf(text_tiket, [ ' Load distributed in a straight line | Wo | KN
| ' num2str(Wo) '\n']);
fprintf(text_tiket, [ ' Distance between tensioners | DT | m
| ' num2str(DT) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
|_____ |_____ |_____ \n');
fprintf(text_tiket, [ ' Minimum tension | Tmin | KN
| ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ ' Maximun tension | To | KN
| ' num2str(To) '\n']);
fprintf(text_tiket, [ ' Tension at the starting point (O) | Tf | KN
| ' num2str(Tf) '\n']);
fprintf(text_tiket, [ ' Tension at the end point (F) | Tmax | KN
| ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ ' Tension on the tensioners | Tten | KN
| ' num2str(Tten) '\n']);
fprintf(text_tiket, [ ' Number of tensioners | Nten | #
'
|_____ \n');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, ' DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, ' SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORY: CABLES \n');
fprintf(text_tiket, 'EXPERIMENT: PARABOLIC CABLE\n');
fprintf(text_tiket, 'CASE 3. : Ho>Hf : Arrow at point (O) is greater than
the arrow of point F\n');

```



```

fclose(text_tiket);
%%
%%% DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento], 'wt');
text_tiket=fopen([ruta nameFileExperimento], 'a');
fprintf(text_tiket, [ 'DATOS DE SALIDA \n' ]);
fprintf(text_tiket, [ 'Tmin ' num2str(Tmin) '\n' ]);
fprintf(text_tiket, [ 'To ' num2str(To) '\n' ]);
fprintf(text_tiket, [ 'Tf ' num2str(Tf) '\n' ]);
fprintf(text_tiket, [ 'Tmax ' num2str(Tmax) '\n' ]);
fprintf(text_tiket, [ 'Tten ' num2str(Tten) '\n' ]);
fprintf(text_tiket, [ 'Nten ' num2str(Nten) '\n' ]);
fprintf(text_tiket, [ 'S ' num2str(S) '\n' ]);
fprintf(text_tiket, [ 'Lo ' num2str(Lo) '\n' ]);
fprintf(text_tiket, [ 'Lf ' num2str(Lf) '\n' ]);
fprintf(text_tiket, [ 'DTo ' num2str(DTo) '\n' ]);
fprintf(text_tiket, [ 'DTf ' num2str(DTf) '\n' ]);
fprintf(text_tiket, [ 'CABLE \n' ]);
fprintf(text_tiket, [ 'x' ' ' ]);
dlmwrite([ruta nameFileExperimento], x, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'y' ' ' ]);
dlmwrite([ruta nameFileExperimento], y, '-append', 'roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'Tensores \n' ]);
fprintf(text_tiket, [ 'xx' ' ' ]);
dlmwrite([ruta nameFileExperimento], (A(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ' ]);
dlmwrite([ruta nameFileExperimento], (B(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'yy' ' ' ]);
dlmwrite([ruta nameFileExperimento], (D(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'CARGA DISTRIBUIDA \n' ]);
fprintf(text_tiket, [ 'wx' ' ' ]);
dlmwrite([ruta nameFileExperimento], (wx(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'wy' ' ' ]);
dlmwrite([ruta nameFileExperimento], (wy(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket, [ 'CABLES-TENSORES-CARGA DISTRIBUIDORA\n' ]);
fclose(text_tiket);
END

```

Figura 69. Código de lectura del experimento 1 “Cable Parabólico” – Funcion “SalidaCableParabCaso3”
Fuente: Autor

A2) Códigos del experimento 2 – Cable Catenaria.

```
function [FS,ER,Wo,Ho,Hf,Hm,b] = LeerCableCatenCaso1(ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la %
% función de CableAnálisisEstaticoCatenaria1, que hace referencia al      %
% Análisis Estático de Cables con Cargas distribuidas uniformemente a lo  %
% largo de cables - Catenaria Caso 1                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                  %
% Para crear una columna más en el archivo txt lo realizamos de la siguien- %
% te manera:                                                              %
% c1=ma{1} donde c1= número de columna; m{1}=posicion de la columna      %
% c1(1)= c1=posición (columna 1); (1)=posición de la fila, entonces      %
% c1(1)= "posiciones" (columna 1 fila 1)                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Código lee archivo txt.
% ruta='C:\Users\DELL i7\Dropbox\SHARED-DANIELA-PELAEZ\DanielaPelaiez-
CableStructure\REVISAR-CAMBIOS'
% nombreArchivo='input.txt';
%%
fid =fopen([ruta nombreArchivo ],'r');
formato='%f';
ma=textscan(fid,formato);
c1=ma{1}; % c1=columna #1 de la matriz ma
Ho=c1(2); %m
Hf=c1(3); %N
ER=c1(4); %N
FS=c1(5); %#
Wo=c1(6); %kg/m
Hm=c1(7); %m
b=c1(8);
fclose(fid);
end
```

Figura 70. Código del experimento 2 – Función “LeerCableCatenCaso1”

Fuente: Autor

```

function [FS,ER,Wo,L,Ym] = LeerCableCatenCaso2(ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la %
% función de CableAnálisisEstáticoCatenaria1, que hace referencia al      %
% Análisis Estático de Cables con Cargas distribuidas uniformemente a lo %
% largo de cables - Catenaria Caso 1                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                  %
% Para crear una columna más en el archivo txt lo realizamos de la siguién- %
% te manera:                                                              %
% cl=ma{1} donde cl= número de columna; m{1}=posición de la columna      %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces      %
% cl(1)= "posiciones" (columna 1 fila 1)                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Código lee archivo txt.
% ruta='C:\Users\DELL i7\Dropbox\SHARED-DANIELA-PELAEZ\DanielaPelaez-
CableStructure\REVISAR-CAMBIOS'
% nombreArchivo='input.txt';
%%

fid =fopen([ruta nombreArchivo ],'r');
formato='%f';
ma=textscan(fid,formato);
cl=ma{1}; % cl=columna #1 de la matriz ma
FS=cl(2); %#
ER=cl(3); %N
Wo=cl(4); %kg/m
L=cl(5); %m
Ym=cl(6); %m
fclose(fid);
end

```

Figura 71. Código del experimento 2 – Función “LeerCableCatenCaso2”

Fuente: Autor

```

function
[FS,ER,Wo,Ho,Hf,Hm,b,Yo,Yf,L,Lo,Lf,To,Tf,Th,TmaxCable,Tmax,Tmin,S,a,wx,wy,x,y,Str] = CableCatenCaso1(FS,ER,Wo,Ho,Hf,Hm,b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               ANÁLISIS ESTÁTICO DE CABLES
%
%   CARGA DISTRIBUIDA UNIFORMEMENTE A LO LARGO DE CABLES (CAECatenariaCaso2)
%
%
%   El peso de un cable lo somete a una carga uniformemente distribuida en toda
su longitud. %
%   Si un cable se somete a fuerzas iguales y paralelas espaciadas
uniformemente, la carga %
%   sobre el cable suele representarse uniformemente distribuida en toda su
longitud. %
%   En esta sección se muestra cómo determinar la forma resultante del cable y
la variación %
%   de su tensión.
%
%   [EzRu,Tmax,Tmin,H,S,Lo,Lf,a,wx,wy,x,y,Str] =
CableAnalisisEstaticoCatenarial(L,FS,ER,Wo) %
%
%   Variables de entrada:
%
%   Ho      = Es la flecha o altura que desea salvar del punto inicial          (m)
%   Hf      = Es la flecha o altura que desea salvar del punto final            (m)
%   Hm      = Flecha a salvar entre el punto de la máxima deformada
%               del cable y el nivel del río o superficie a especificar          (m)
%   b       = La cota hasta la superficie                                     (m)
%   ER      = Esfuerzo de Ruptura del cable                                   (N)
%   FS      = Factor de seguridad                                           (Adimensional)
%   Wo      = Es la magnitud de la carga distribuida                        (kg/m)
%
%   Variables de salida:
%
%   Tmax    = La carga máxima segura de trabajo                            (N)
%   Tmin    = La tensión mínima que soporta el cable, pendiente 0           (N)
%   L       = Vano o Distancia entre los puntos O y P                      (m)
%   S       = La longitud Total del cable                                  (m)
%   Lo      = La longitud Total del cable                                  (m)
%   Lf      = La longitud Total del cable                                  (m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%clc
%close all
%Tmax es la carga máxima segura de trabajo, en función de: Esfuerzo
%de rotura y el factor de seguridad en N
TmaxCable=ER/FS; %N
%Wo es la magnitud de la carga distribuida en toda su longitud o el
%peso del cable... se expresa el cambio de unidades... el usuario
%ingresa en kg/m y la unidad para el análisis se transformará a
% N/m
Wo=(Wo*9.81); %N/m
%pasol:flechas
Yo=Ho-Hm-b;
Yf=Hf-Hm-b;
if Ho<Hf
    %paso2:Identificar la tensión máxima
    Tf=TmaxCable;
    %paso3:Tensión mínima
    Tmin=TmaxCable-Yf*Wo;
    %paso4:determinar a
    a=Wo/Tmin;
    %paso5:determinar Lo
    Lo=-(acosh(a*Yo+1)/a);
    %paso6:determinar Lf
    Lf=(acosh(a*Yf+1)/a);

```

```

    %paso7:determinar L
    L=Lf-Lo;
    %paso8:determinar So
    So=-sinh(a*Lo)/a;
    %paso9:determinar Sf
    Sf=sinh(a*Lf)/a;
    %paso10:determinar S
    S=So+Sf;
    %paso11:determinar Tf
    To=Tmin*cosh(a*Lo);
    %paso12:determinar th
    Th=Tmin;
    %paso13:determinar Tmax
    Tmax=max(To,Tf);
elseif Ho>Hf
    %paso2:Identificar la tensión máxima
    To=TmaxCable;
    %paso3:Tensión mínima
    Tmin=TmaxCable-Yo*Wo;
    %paso4:determinar a
    a=Wo/Tmin;
    %paso5:determinar Lo
    Lo=-(acosh(a*Yo+1)/a);
    %paso6:determinar Lf
    Lf=(acosh(a*Yf+1)/a);
    %paso7:determinar L
    L=Lf-Lo;
    %paso8:determinar So
    So=-sinh(a*Lo)/a;
    %paso9:determinar Sf
    Sf=sinh(a*Lf)/a;
    %paso10:determinar S
    S=So+Sf;
    %paso11:determinar Tf
    Tf=Tmin*cosh(a*Lf);
    %paso12:determinar th
    Th=Tmin;
    %paso13:determinar Tmax
    Tmax=max(To,Tf);
else Ho=Hf;
    %Identificar la tensión máxima
    To=TmaxCable;
    Tf=TmaxCable;
    %paso3:Tensión mínima
    Tmin=TmaxCable-Yo*Wo;
    %paso4:determinar a
    a=Wo/Tmin;
    %paso5:determinar Lo
    Lo=-(acosh(a*Yo+1)/a);
    %paso6:determinar Lf
    Lf=(acosh(a*Yf+1)/a);
    %paso7:determinar L
    L=Lf-Lo;
    %paso8:determinar So
    So=-sinh(a*Lo)/a;
    %paso9:determinar Sf
    Sf=sinh(a*Lf)/a;
    %paso10:determinar S
    S=So+Sf;
    %paso12:determinar th
    Th=Tmin;
    %paso13:determinar Tmax
    Tmax=max(To,Tf);
end

```

```

%% GRAFICOS
% pOSICIÓN INICIAL
%Do=2;
wx=[Lo Lf];
wy=[Ho Hf];
%subplot(221),plot(wx,wy,'r')
%title('CARGA DISTRIBUIDA');
%wxmin=Lo-Do;
%wxmax=Lf+Do;
%wymn=(-Do);
%wymax=Hf+Do;
%axis([wxmin wxmax wymn wymax])
%hline11 = plot(wx,wy,'K');
%ax = gca;
%hline22 = line(wx+.06,wy,...
% 'LineWidth',4,...
% 'Color',[.5 .5 .5],...
% 'Parent',ax);
%set(gca,'Children',[hline11 hline22])
%axis off
%title('CABLE');
%%
%FIGURA DEL CABLE
%figura el cable
x=Lo:0.5:Lf;
y=(cosh(a*x)-1)/a;
%subplot(222),plot(x,y,'K')
%grid on;
%hline1 = plot(x,y,'K');
%ax = gca;
%hline2 = line(x+.06,y,...
% 'LineWidth',4,...
% 'Color',[.5 .5 .5],...
% 'Parent',ax);
%set(gca,'Children',[hline1 hline2])
%axis off
%title('DEFORMADA DEL CABLE');
%%
%deFORMADA Y POSICION INICIAL
%Do=2;
%wxx=[Lo Lf Lo];
%wyy=[Ho Hf Hf Ho];
%subplot(223),plot(wxx,wyy,'r')
%wxmin=Lo-Do;
%wxmax=Lf+Do;
%wymn=(-Do);
%wymax=Hf+Do;
%axis([wxmin wxmax wymn wymax])
%hline11 = plot(wxx,wyy,'K');
%ax = gca;
%hline22 = line(x+.06,y,...
% 'LineWidth',4,...
% 'Color',[.5 .5 .5],...
% 'Parent',ax);
%set(gca,'Children',[hline11 hline22])
%axis off
%title('EL CABLE Y SU DEFORMADA');
%%
% fecha y hora
% str = datestr(now);
%%
end

```

Figura 72. Código del experimento 2 – Función “CableCatenCaso1”

Fuente: Autor

```

function [FS,ER,Wo,L,Ym,Ho,Hf,To,Tf,Th,Yo, Yf,
TmaxCable,Tmax,Tmin,Lo,Lf,S,a,wx,wy,x,y,str] = CableCatenCaso2(FS,ER,Wo,L,Ym)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               ANÁLISIS ESTÁTICO DE CABLES
%
% CARGA DISTRIBUIDA UNIFORMEMENTE A LO LARGO DE CABLES (CAECatenariaCaso2)
%
% El peso de un cable lo somete a una carga uniformemente distribuida en toda
su longitud. %
% Si un cable se somete a fuerzas iguales y paralelas espaciadas
uniformemente, la carga %
% sobre el cable suele representarse uniformemente distribuida en toda su
longitud. %
% En esta sección se muestra cómo determinar la forma resultante del cable y
la variación %
% de su tensión.
%
% [EzRu,Tmax,Tmin,H,S,Lo,Lf,a,wx,wy,x,y,str] =
CableAnalisisEstaticoCatenarial(L,FS,ER,Wo) %
%
% Variables de entrada:
%
% L          = Vano o Distancia entre los puntos O y P                      (m)
% Var        = Es el desnivel entre el punto inicial y el punto final      (m)
% Ym         = Flecha a salvar entre el punto de la máxima deformada
%              del cable y el nivel del río o superficie a especificar      (m)
% ER = Esfuerzo de Ruptura del cable                                         (N)
% FS = Factor de seguridad                                                    (Adimensional)
% Wo = Es la magnitud de la carga distribuida                              (kg/m)
%
% Variables de salida:
%
% Tmax = La carga máxima segura de trabajo                                  (N)
% Tmin = La tensión mínima que soporta el cable, pendiente 0                (N)
% Ho    = Es la flecha o altura que desea salvar del punto inicial           (m)
% Hf    = Es la flecha o altura que desea salvar del punto final             (m)
% S     = La longitud Total del cable                                       (m)
% Lo    = La longitud Total del cable                                       (m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%clc
%close all
%Tmax es la carga máxima segura de trabajo, en función de: Esfuerzo
%de rotura y el factor de seguridad en N
TmaxCable=ER/FS; %N
%Wo es la magnitud de la carga distribuida en toda su longitud o el
%peso del cable... se expresa el cambio de unidades... el usuario
%ingresa en kg/m y la unidad para el análisis se transformará a
% N/m
Wo=(Wo*9.81); %N/m
%determinación de la variable "Tmin"
Th=Ym*Wo; %N
%Determinar la variable "a" en funcion de la carga distribuida y la
%tensión mínima
a=Wo/Th;
%% Procedimiento
%Determinar la Tensión máxima Tmax
Tmax=Th*cosh(a*L/2); %N
%Determinar la Tensión mínima Tmin
Tmin=Th; %N
%H es la flecha o altura a partir del punto con máxima deformación hasta el
%punto inicial (O) o y el punto final (F)
Yo=(cosh(a*L/2)-1)/a; %m
Yf=Yo;
Ho=Yo+Ym; %m
Hf=Yf+Ym; %m
%To es la tensión en el punto inicial (O)
To=Tmin+Wo*Yo; %m

```



```

    %Tf es la tensión en el punto final (F)
    Tf=Tmin+Wo*Yf; %m
    %S es la longitud del cable
    So=sinh(a*L/2)/a; %m
    Sf=sinh(a*L/2)/a; %m
    S=So+Sf; %m
    %Lo es la distancia entre el punto inicial (O) y el punto con la máxima
    %deformada.
    Lo=-L/2; %m
    %Lo es la distancia entre el punto final (F) y el punto con la máxima
    %deformada.
    Lf=-Lo; %m
%% GRAFICOS
% POSICIÓN INICIAL
%Do=2;
wx=[Lo Lf];
wy=[Ho Hf];
subplot(221),plot(wx,wy,'r')
%FIGURA DEL CABLE
    %figura el cable
x=Lo:0.05:Lf;
y=(cosh(a*x)-1)/a;
subplot(222),plot(x,y,'K')
%%
%deFORMADA Y POSICION INICIAL
%Do=2;
%wx=[Lo Lf Lf Lo];
%wy=[Ho Hf Hf Ho];
subplot(223),plot(wx,wy,'r')
%wxmin=Lo-Do;
%wxmax=Lf+Do;
%wymn=(-Do);
%wymax=Hf+Do;
%axis([wxmin wxmax wymn wymax])
%hline11 = plot(wx,wy,'K');
%ax = gca;
%hline22 = line(x+.06,y,...
% 'LineWidth',4,...
% 'Color',[.5 .5 .5],...
% 'Parent',ax);
%set(gca,'Children',[hline11 hline22])
%axis off
%title('EL CABLE Y SU DEFORMADA');
% fecha y hora
    str = datestr(now);
end

```

Figura 73. Código del experimento 2 – Función “CableCatenCaso2”

Fuente: Autor

```

function [Tmin] =
SalidaCableCatenCasol(FS,ER,Wo,Ho,Hf,Hm,b,Yo,Yf,L,Lo,Lf,To,Tf,Th,TmaxCable,Tmax
,Tmin,S,a,wx,wy,x,y,str,ruta,nameFileUser,
nameFileExperimento,nameFileUserEnglish)
%% Resultados en español para el usuario
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SÍSMOLOGIA (GRISS-UTPL) \n');
fprintf(text_tiket, '          DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, '          SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN \n');
fprintf(text_tiket, '          UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'LABORATORIO:          CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO:          CABLE CATENARIA \n');
fprintf(text_tiket, 'CASO          :          #2          | Determina el desnivel, Ho
and Hf          \n');
fprintf(text_tiket, [ 'FECHA          :          ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores          : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '_____\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          DESCRIPTION          |SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'_____|_____|_____|_____\n'
);
fprintf(text_tiket, [ 'Esfuerzo de rotura del cable          | ER |
N          | ' num2str(ER) '\n']);
fprintf(text_tiket, [ 'Factor de Seguridad          | FS |
dimes          | ' num2str(FS) '\n']);
fprintf(text_tiket, [ 'Carga Distribuida del cable          | Wo |
N/m          | ' num2str(Wo) '\n']);
fprintf(text_tiket, [ 'Altura del soporte          (O)          | Ho |
m          | ' num2str(Ho) '\n']);
fprintf(text_tiket, [ 'Altura del soporte          (F)          | Hf |
m          | ' num2str(Hf) '\n']);
fprintf(text_tiket, [ 'Altura: superficie hasta el punto máx. cable          | Hm |
m          | ' num2str(Hm) '\n']);
fprintf(text_tiket, [ 'Altura: hasta la superficie          | b |
m          | ' num2str(b) '\n']);
fprintf(text_tiket, '\n');

```

```

fprintf(text_tiket, '\n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
                                DESCRIPTION                                |SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'_____|_____|_____|_____ \n'
);
fprintf(text_tiket, [ 'Distancia horizontal entre los puntos (O) y (F) | L |
m | ' num2str(L) '\n']);
fprintf(text_tiket, [ 'Distancia horizontal desde el punto (O) al eje | Lo |
m | ' num2str(Lo) '\n']);
fprintf(text_tiket, [ 'Distancia horizontal desde el punto (F) al eje | Lf |
m | ' num2str(Lf) '\n']);
fprintf(text_tiket, [ 'Tensión en el punto incial (O) | To |
N | ' num2str(To) '\n']);
fprintf(text_tiket, [ 'Tensión en el punto final (F) | Tf |
N | ' num2str(Tf) '\n']);
fprintf(text_tiket, [ 'Tensión horizontal | Th |
N | ' num2str(Th) '\n']);
fprintf(text_tiket, [ 'Tensión máxima que soporta el cable |TmaxCa| N
| ' num2str(TmaxCable) '\n']);
fprintf(text_tiket, [ 'Tensión máxima | Tmax |
N | ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ 'Tensión mínima | Tmin |
N | ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ 'Longitud del Cable | S |
m | ' num2str(S) '\n']);
fprintf(text_tiket, [ 'Altura o flecha del punto (O) | Yo |
m | ' num2str(Yo) '\n']);
fprintf(text_tiket, [ 'Altura o flecha del punto (F) | Yf |
m | ' num2str(Yf) '\n']);
fprintf(text_tiket, [ 'Ecuación: Deformada del cable Y=1/'
num2str(a) '*[cosh(' num2str(a) '*x)-1]\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'_____ \n'
);
);
fprintf(text_tiket,
'_____ \n'
);
);
fclose(text_tiket);
%% Resultados en el idioma ingles para el usuario
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'_____ \n'
);
);
fprintf(text_tiket,
'_____ \n'
);
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
                                VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
                                SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, '
                                DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, '
                                SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, '
                                UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, '
                                -----
-----\n');

```

```

fprintf(text_tiket, '      \n');
fprintf(text_tiket, '      \n');
fprintf(text_tiket, 'LABORATORY:   CABLES\n');
fprintf(text_tiket, 'EXPERIMENT:   CATENARY | STATIC ANALYSIS
|\n');
fprintf(text_tiket, 'CASO      :   #1      | Determine the horizontal
distance between O and F\n');
fprintf(text_tiket, [ 'FECHA      :   ' num2str(str) '\n']];
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, 'INPUT DATA      \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
DESCRIPTION
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'_____ | _____ | _____ \n'
);
fprintf(text_tiket, [ 'Cable breakage stress
N      | ' num2str(ER) '\n']];
fprintf(text_tiket, [ 'Security factor
dimes  | ' num2str(FS) '\n']];
fprintf(text_tiket, [ 'Load distributed on the cable
N/m    | ' num2str(Wo) '\n']];
fprintf(text_tiket, [ 'Height of the support (O)
m      | ' num2str(Ho) '\n']];
fprintf(text_tiket, [ 'Height of the support (F)
m      | ' num2str(Hf) '\n']];
fprintf(text_tiket, [ 'Height: surfaces and maximum cable
m      | ' num2str(Hm) '\n']];
fprintf(text_tiket, [ 'Height: Up to the surface
m      | ' num2str(b) '\n']];
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, 'ANALYSIS OUPUT\n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
DESCRIPTION
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'_____ | _____ | _____ \n'
);
fprintf(text_tiket, [ 'Horizontal distance between points (O) and (F) | L      |
m      | ' num2str(L) '\n']];
fprintf(text_tiket, [ 'Horizontal distance from starting point (O)      | Lo     |
m      | ' num2str(Lo) '\n']];
fprintf(text_tiket, [ 'Horizontal distance from end point      (F) | Lf     |
m      | ' num2str(Lf) '\n']];
fprintf(text_tiket, [ 'Tension-initial point (O)      | To     |
N      | ' num2str(To) '\n']];
fprintf(text_tiket, [ 'Tension-end      point (F)      | Tf     |
N      | ' num2str(Tf) '\n']];
fprintf(text_tiket, [ 'Horizontal tension
N      | ' num2str(Th) '\n']];
fprintf(text_tiket, [ 'Maximun tension - supports the cable      |TmaxCa|
N      | ' num2str(TmaxCable) '\n']];
fprintf(text_tiket, [ 'Maximun tension
| Tmax |

```

```

N      | ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ 'Minimun tension                                | Tmin |
N      | ' num2str(Tmin) '\n']);
fprintf(text_tiket,[ 'Cable length                                  | S     |
m      | ' num2str(S) '\n']);
fprintf(text_tiket,[ 'Arrow or point height (O)                      | Yo    |
m      | ' num2str(Yo) '\n']);
fprintf(text_tiket,[ 'Arrow or point height (F)                      | Yf    |
m      | ' num2str(Yf) '\n']);
fprintf(text_tiket,[ 'Equation: deformed cable                      Y=1/'
num2str(a) '*[cosh(' num2str(a) '*x)-1]\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fclose(text_tiket);
%%_DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento], 'wt');
text_tiket=fopen([ruta nameFileExperimento], 'a');
%%fprintf(text_tiket,[ 'DATOS DE SALIDA \n']);
fprintf(text_tiket,[ 'L ' num2str(L) '\n']);
fprintf(text_tiket,[ 'Lo ' num2str(Lo) '\n']);
fprintf(text_tiket,[ 'Lf ' num2str(Lf) '\n']);
fprintf(text_tiket,[ 'Yo ' num2str(Yo) '\n']);
fprintf(text_tiket,[ 'Yf ' num2str(Yf) '\n']);
fprintf(text_tiket,[ 'To ' num2str(To) '\n']);
fprintf(text_tiket,[ 'Tf ' num2str(Tf) '\n']);
fprintf(text_tiket,[ 'TmaxCable ' num2str(TmaxCable) '\n']);
fprintf(text_tiket,[ 'Tmax ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ 'Tmin ' num2str(Tmin) '\n']);
fprintf(text_tiket,[ 'S ' num2str(S) '\n']);
fprintf(text_tiket,[ 'Y=1/a*[cosh(ax)-1] Y=1/' num2str(a) '*[cosh(' num2str(a)
'*x)-1]\n']);
%fprintf(text_tiket,[ 'Posición Inicial del cable \n']);
fprintf(text_tiket,[ 'wx' ' ']);
dlmwrite([ruta nameFileExperimento], wx, '-append', 'roffset', 0, 'delimiter', '
', 'precision', '%.4f')
fprintf(text_tiket,[ 'wy' ' ']);
dlmwrite([ruta nameFileExperimento], wy, '-append', 'roffset', 0, 'delimiter', '
', 'precision', '%.4f')
%fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
%fprintf(text_tiket,[ 'La deformada del Cable \n']);
fprintf(text_tiket,[ 'x' ' ']);
dlmwrite([ruta nameFileExperimento], (x(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket,[ 'y' ' ']);
dlmwrite([ruta nameFileExperimento], (y(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
%fprintf(text_tiket,[ 'Posición Inicial del Cable - Carga Distribuida-
DEformada del Cable\n']);
fclose(text_tiket);
end

```

Figura 74. Código del experimento 2 – Función “SalidaCableCatenCaso1”

Fuente: Autor

```
function [Tmin] = SalidaCableCatenCaso2(FS,ER,Wo,L,Ym,Ho,Hf,To,Tf,Th,Yo,Yf,
TmaxCable,Tmax,Tmin,Lo,Lf,S,a,wx,wy,x,y,str, ruta, nameFileUser,
nameFileExperimento,nameFileUserEnglish)
%% Resultados en español para el usuario
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SISMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, ' DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, ' SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE CATENARIA\n');
fprintf(text_tiket, 'CASO : #2 | Determina el desnivel, Ho
and Hf | \n');
fprintf(text_tiket, [ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '_____\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION | SYMBOL |
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'_____|_____|_____|_____\n'
);
fprintf(text_tiket, [ 'Esfuerzo de rotura del cable | ER |
N | ' num2str(ER) '\n']);
fprintf(text_tiket, [ 'Facotr de Seguridad | FS |
dimes | ' num2str(FS) '\n']);
fprintf(text_tiket, [ 'Carga Distribuida del cable | Wo |
N/m | ' num2str(Wo) '\n']);
fprintf(text_tiket, [ 'Distancia horizontal entre los puntos (O) y (F) | L |
m | ' num2str(L) '\n']);
fprintf(text_tiket, [ 'Altura: superficie y el punto máximo del cable | Ym |
N | ' num2str(Ym) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
```

```

fprintf(text_tiket, '
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, [ 'Altura o flecha del punto (O)
m | ' num2str(Yo) '\n']);
fprintf(text_tiket, [ 'Altura o flecha del punto (F)
m | ' num2str(Yf) '\n']);
fprintf(text_tiket, [ 'Altura del apoyo (O)
m | ' num2str(Ho) '\n']);
fprintf(text_tiket, [ 'Altura del apoyo (F)
m | ' num2str(Hf) '\n']);
fprintf(text_tiket, [ 'Tensión en el punto inicial (O)
N | ' num2str(To) '\n']);
fprintf(text_tiket, [ 'Tensión en el punto final (F)
N | ' num2str(Tf) '\n']);
fprintf(text_tiket, [ 'Tensión máxima que soporta el cable
| ' num2str(TmaxCable) '\n']);
fprintf(text_tiket, [ 'Tensión máxima
N | ' num2str(Tmax) '\n']);
fprintf(text_tiket, [ 'Tensión mínima
N | ' num2str(Tmin) '\n']);
fprintf(text_tiket, [ 'Tensión horizontal
N | ' num2str(Th) '\n']);
fprintf(text_tiket, [ 'Distancia horizontal desde el punto (O) al eje
m | ' num2str(Lo) '\n']);
fprintf(text_tiket, [ 'Distancia horizontal desde el punto (F) al eje
m | ' num2str(Lf) '\n']);
fprintf(text_tiket, [ 'Longitud del Cable
m | ' num2str(S) '\n']);
fprintf(text_tiket, [ 'Ecuación: Deformada del cable
num2str(a) '*[cosh(' num2str(a) '*x)-1]\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fclose(text_tiket);
%% Resultados en el idioma ingles para el usuario
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE)\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, '
DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, '
SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, '
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, '
-----\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, 'LABORATORY: CABLES\n');
fprintf(text_tiket, 'EXPERIMENT: CATENARY | \n');

```



```

fprintf(text_tiket,[ 'Cable length' | S |
m | ' num2str(S) '\n']);
fprintf(text_tiket,[ 'Equation: deformed cable Y=1/'
num2str(a) '*[cosh(' num2str(a) '*x)-1]\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
\n'
);
fprintf(text_tiket,
'
\n'
);
fclose(text_tiket);
%% DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento], 'wt');
text_tiket=fopen([ruta nameFileExperimento], 'a');
fprintf(text_tiket,[ 'Lo ' num2str(Lo) '\n']);
fprintf(text_tiket,[ 'Lf ' num2str(Lf) '\n']);
fprintf(text_tiket,[ 'yo ' num2str(Yo) '\n']);
fprintf(text_tiket,[ 'Yf ' num2str(Yf) '\n']);
fprintf(text_tiket,[ 'Ho ' num2str(Ho) '\n']);
fprintf(text_tiket,[ 'Hf ' num2str(Hf) '\n']);
fprintf(text_tiket,[ 'To ' num2str(To) '\n']);
fprintf(text_tiket,[ 'Tf ' num2str(Tf) '\n']);
fprintf(text_tiket,[ 'TmaxCable ' num2str(TmaxCable) '\n']);
fprintf(text_tiket,[ 'Tmax ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ 'Tmin ' num2str(Tmin) '\n']);
fprintf(text_tiket,[ 'S ' num2str(S) '\n']);
fprintf(text_tiket,[ 'Y=1/a*[cosh(ax)-1] Y=1/' num2str(a) '*[cosh(' num2str(a)
'*x)-1]\n']);
fprintf(text_tiket,[ 'Posición Inicial del cable \n']);
fprintf(text_tiket,[ 'wx' ' ']);
dlmwrite([ruta nameFileExperimento], wx, '-append', 'roffset', 0, 'delimiter', '
', 'precision', '%.4f')
fprintf(text_tiket,[ 'wy' ' ']);
dlmwrite([ruta nameFileExperimento], wy, '-append', 'roffset', 0, 'delimiter', '
', 'precision', '%.4f')
fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
fprintf(text_tiket,[ 'La deformada del Cable \n']);
fprintf(text_tiket,[ 'x' ' ']);
dlmwrite([ruta nameFileExperimento], (x(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket,[ 'y' ' ']);
dlmwrite([ruta nameFileExperimento], (y(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket,[ 'Posición Inicial del Cable - Carga Distribuida-
DEformada del Cable\n']);
fclose(text_tiket);
end

```

Figura 75. Código del experimento 2 – Función “SalidaCableCatenCaso2”

Fuente: Autor

A3) Códigos del experimento 3 – Cable con Cargas Puntuales.

```
function [NW,W,D,h1,YB] = LeerCableCargasPuntuales1(ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la
% función de CableCargasDiscretas1, que hace referencia: Análisis Estático
% de Cables con Cargas puntuales o discretas, detallando específicamente %
% las alturas, las tensiones que se produce a lo largo del cable      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt.                %
% Para crear una columna más en el archivo txt lo realizamos de la siguiente manera: %
% c1=ma{1} donde c1= número de columna; m{1}=posicion de la columna    %
% c1(1)= c1=posición (columna 1); (1)=posición de la fila, entonces   %
% c1(1)= "posiciones" (columna 1 fila 1)                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Código lee archivo txt.
ruta='C:\Users\DELL i7\Dropbox\SHARED-DANIELA-PELAEZ\DanielaPelaez-CableStructure\REVISAR-CAMBIOS'
nombreArchivo='input.txt';
fid =fopen([ruta nombreArchivo ],'r');
formato='%f %f %f %f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f';
ma=textscan(fid,formato);
c1=ma; % c1=columna #1 de la matriz ma
e1=c1{1};
%leer el número de cargas
NW=e1(2); %#
%leer la matriz de cargas
e2=3;
s=NW;
W=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        W (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                W (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                W (b)=a2;
            end
            W (b)=a2;
        end
        W (c)=a2;
    end
    W (c)=a2;
end
W=W';
%leer la matriz de distancias
e2=4;
ss=NW+1;
s=ss;
D=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        D (c)=a2;
```

```

elseif c>=2
    for b=2:c
        if b==2
            a1=c1{b};
            a2=a1(e2);
            D (b)=a2;
        elseif b>2
            a1=c1{b};
            a2=a1(e2);
            D (b)=a2;
        end
        D (b)=a2;
    end
    D (c)=a2;
end
D (c)=a2;
end
D=D';
%Leer la altura 1
h1=e1(5); %m
%Leer el desnivel entre el punto inicial y el punto final
YB=e1(6); %m
fclose(fid);
end

```

Figura 76. Código del experimento 3 – Función “LeerCablesCargasPuntuales1”

Fuente: Autor

```

function [NT, NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC] =
LeerCableCargasPuntuales2(ruta, nombreArchivo)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la%
% función de CableCargasDiscretas1, que hace referencia: Análisis Estático%
% de Cables con Cargas puntuales o discretas, detallando específicamente %
% las alturas, las tensiones que se produce a lo largo del cable %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt. %
% Para crear una columna más en el archivo txt lo realizamos de la siguienn%
% te manera: %
% cl=ma{1} donde cl= número de columna; m{1}=posicion de la columna %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces %
% cl(1)= "posiciones" (columna 1 fila 1) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Código lee archivo txt.
%ruta='C:\Users\DELL i7\Dropbox\SHARED-DANIELA-PELAEZ\DanielaPelaiez-
CableStructure\REVISAR-CAMBIOS'
%nombreArchivo='input.txt';
%%
fid =fopen([ruta nombreArchivo ],'r');
formato='%f %f %f %f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f
%f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f';
ma=textscan(fid,formato);
cl=ma; % cl=columna #1 de la matriz ma
%e1: es la columna 1 de la matriz cl
e1=cl{1};
%leer el número de Tramos
NT=e1(2);
%identificar el número de tramos %% Tramo AB
%leer el número de cargas
NWAB=e1(3);%#
%leer la matriz de cargas
e2=4;
s=NWAB;
WAB=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        WAB (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                WAB (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                WAB (b)=a2;
            end
            WAB (b)=a2;
        end
        WAB (c)=a2;
    end
    WAB (c)=a2;
end
WAB=WAB';
%leer la matriz de distancias
e2=5;

```

```

ss=NWAB+1;
s=ss;
DAB=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        DAB (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                DAB (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                DAB (b)=a2;
            end
            DAB (b)=a2;
        end
        DAB (c)=a2;
    end
    DAB (c)=a2;
end
DAB=DAB';
%Leer la altura 1
h1AB=e1(6); %m
%Leer el desnivel entre el punto inicial y el punto final
YB=e1(7); %m
%% Tramo BC
%leer el número de cargas
NWBC=e1(8);%#
%leer la matriz de cargas
e2=9;
s=NWBC;
WBC=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        WBC (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                WBC (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                WBC (b)=a2;
            end
            WBC (b)=a2;
        end
        WBC (c)=a2;
    end
    WBC (c)=a2;
end
WBC=WBC';
%leer la matriz de distancias
e2=10;
ss=NWBC+1;
s=ss;
DBC=ones(s,1); %KN
for c = 1:s
    if c==1

```

```

        a1=c1{c};
        a2=a1(e2);
        DBC (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                DBC (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                DBC (b)=a2;
            end
            DBC (b)=a2;
        end
        DBC (c)=a2;
    end
    DBC (c)=a2;
end
DBC=DBC';
%Leer la altura 1
h1BC=e1(11); %m
%Leer el desnivel entre el punto incial y el punto final
YC=e1(12); %m
fclose(fid);
end

```

Figura 77. Código del experimento 3 – Función “LeerCablesCargasPuntuales2”

Fuente: Autor

```

function [NT, NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,NWCD,WCD,DCD,h1CD,YD] =
LeerCableCargasPuntuales3(ruta, nombreArchivo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La presente función nos permite leer los datos de entrada referente a la%
% función de CableCargasDiscretas1, que hace referencia: Análisis Estático%
% de Cables con Cargas puntuales o discretas, detallando específicamente %
% las alturas, las tensiones que se produce a lo largo del cable %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nomenclatura
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ma= Matriz de "n" datos ingresados en el archivo txt. %
% Para crear una columna más en el archivo txt lo realizamos de la siguienn%
% te manera: %
% cl=ma{1} donde cl= número de columna; m{1}=posicion de la columna %
% cl(1)= cl=posición (columna 1); (1)=posición de la fila, entonces %
% cl(1)= "posiciones" (columna 1 fila 1) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAMACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Código lee archivo txt.
%ruta='C:\Users\DELL i7\Dropbox\SHARED-DANIELA-PELAEZ\DanielaPelaez-
CableStructure\REVISAR-CAMBIOS'
%nombreArchivo='input.txt';
fid =fopen([ruta nombreArchivo ],'r');
formato='%f %f %f %f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f
%f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f%f %f %f';
ma=textscan(fid,formato);
cl=ma; % cl=columna #1 de la matriz ma
%el: es la columna 1 de la matriz cl
el=cl{1};
%leer el número de Tramos
NT=el(2);
%% Tramo AB
%leer el número de cargas
NWAB=el(3);%#
%leer la matriz de cargas
e2=4;
s=NWAB;
WAB=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        WAB (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                WAB (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                WAB (b)=a2;
            end
            WAB (b)=a2;
        end
        WAB (c)=a2;
    end
    WAB (c)=a2;
end
WAB=WAB';
%leer la matriz de distancias
e2=5;
ss=NWAB+1;
s=ss;
DAB=ones(s,1); %KN
for c = 1:s

```

```

        if c==1
            a1=c1{c};
            a2=a1(e2);
            DAB (c)=a2;
        elseif c>=2
            for b=2:c
                if b==2
                    a1=c1{b};
                    a2=a1(e2);
                    DAB (b)=a2;
                elseif b>2
                    a1=c1{b};
                    a2=a1(e2);
                    DAB (b)=a2;
                end
            end
            DAB (b)=a2;
        end
        DAB (c)=a2;
    end
    DAB=DAB';
    %Leer la altura 1
    h1AB=e1(6); %m
    %Leer el desnivel entre el punto incial y el punto final
    YB=e1(7); %m
%% Tramo BC
    %leer el número de cargas
    NWBC=e1(8);%#
    %leer la matriz de cargas
    e2=9;
    s=NWBC;
    WBC=ones(s,1); %KN
    for c = 1:s
        if c==1
            a1=c1{c};
            a2=a1(e2);
            WBC (c)=a2;
        elseif c>=2
            for b=2:c
                if b==2
                    a1=c1{b};
                    a2=a1(e2);
                    WBC (b)=a2;
                elseif b>2
                    a1=c1{b};
                    a2=a1(e2);
                    WBC (b)=a2;
                end
            end
            WBC (b)=a2;
        end
        WBC (c)=a2;
    end
    WBC=WBC';
    %leer la matriz de distancias
    e2=10;
    ss=NWBC+1;
    s=ss;
    DBC=ones(s,1); %KN
    for c = 1:s
        if c==1
            a1=c1{c};
            a2=a1(e2);
            DBC (c)=a2;
        elseif c>=2

```



```

        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                DBC (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                DBC (b)=a2;
            end
            DBC (b)=a2;
        end
        DBC (c)=a2;
    end
    DBC (c)=a2;
end
DBC=DBC';
%Leer la altura 1
h1BC=e1(11); %m
%Leer el desnivel entre el punto incial y el punto final
YC=e1(12); %m
%% Tramo CD
%leer el número de cargas
NWCD=e1(13);%#
%leer la matriz de cargas
e2=14;
s=NWCD;
WCD=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a2=a1(e2);
        WCD (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);
                WCD (b)=a2;
            elseif b>2
                a1=c1{b};
                a2=a1(e2);
                WCD (b)=a2;
            end
            WCD (b)=a2;
        end
        WCD (c)=a2;
    end
    WCD (c)=a2;
end
WCD=WCD';
%leer la matriz de distancias
e2=15;
ss=NWCD+1;
s=ss;
DCD=ones(s,1); %KN
for c = 1:s
    if c==1
        a1=c1{c};
        a1=c1{c};
        a2=a1(e2);
        DCD (c)=a2;
    elseif c>=2
        for b=2:c
            if b==2
                a1=c1{b};
                a2=a1(e2);

```

```

        DCD (b)=a2;
    elseif b>2
        a1=c1{b};
        a2=a1(e2);
        DCD (b)=a2;
    end
    DCD (b)=a2;
end
    DCD (c)=a2;
end
    DCD (c)=a2;
end
    DCD=DCD';
    %Leer la altura 1
    h1CD=e1(16); %m
    %Leer el desnivel entre el punto inicial y el punto final
    YD=e1(17); %m
    fclose(fid);
end

```

Figura 78. Código del experimento 3 – Función “LeerCablesCargasPuntuales3”

Fuente: Autor

```

function [A, Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S,
x,y,xx,yy,str,NW,W,D,h1,YB] = CableCargasPuntuales1 (NW,W,D,h1,YB)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determinación de tensiones en cables que soportaban cuerpos suspendidos. %
%Se considera el caso de un número arbitrario N de cuerpos suspendidos de %
%un cable. Se supone que el peso del cable puede despreciarse comparado %
%con los pesos suspendidos, y que el cable es lo bastante flexible para %
%aproximar su forma con una serie de segmentos rectos. %
% %
% %
% Caso 1: El Punto Inicial y Punto Final se encuentran a nivel %
% Caso 2: El Punto Inicial y Punto Final se encuentran a desnivel +YB %
% Caso 3: El Punto Inicial y Punto Final se encuentran a desnivel -YB %
% %
% Variables de entrada: %
% %
% NW: Número de cargas %
% W : Cargas || W1,W2,W3, ...,WN (KN) %
% D : Distancias en x || separación entre el punto inicial y final, %
% y las cargas (m) %
% h1: es la flecha o altura entre el punto inicial y la carga 1 (m) %
% YB: es el desnivel entre el punto inicial y el punto final (m) %
% || YB = 0 cuando es cero: está a nivel el punto inicial y final %
% YB = -1 cuando el valor es negativo: el punto inicial está por %
% debajo del punto final. %
% YB = +1 cuando el valor es positivo: el punto inicial está sobre %
% el punto final. %
% %
% Variables de salida: %
% %
% A : La Reacción (Fuerza) en el punto inicial (KN) %
% Ax : Es la reacción (Fuerza) en x de la reacción (Fuerza) en A (KN) %
% Ay : Es la reacción (Fuerza) en y de la reacción (Fuerza) en A (KN) %
% B : La reacción (Fuerza) en el punto final (KN) %
% Bx : Es la reacción (Fuerza) en x de la reacción (Fuerza) en B (KN) %
% By : Es la reacción (Fuerza) en y de la reacción (Fuerza) en B (KN) %
% T : Es la tensión del cable en los diferentes tramos corresponde %
% diente a las cargas (KN) %
% Tmax: Es la tensión máxima que soporta el cable (KN) %
% Tmin: Es la tensión mínima que soporta el cable (KN) %
% h : Es la flecha o altura || como nivel desde el punto inicial %
% hasta la carga %
% || si el valor es positivo está la carga por debajo del %
% nivel, y si el valor es negativo esta sobre el nivel %
% El nivel es desde el punto inicial una línea recta || %
% L : Es el vano entre el punto inicial y el punto final (m) %
% S : Es la longitud del cable entre el punto inicial y final (m) %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% desarrollo
YB=-YB;
if 0==YB
%% Procedimiento
% Datos: a ser una matriz de [1*NW] se invierte a [NW*1]
% Datos de las cargas puntuales
W=W'; %KN
% Datos de las distancias en x: punto inicial, cargas y punto final
D=D'; %m
% s: es el valor correspondiente al número de cargas NW
s =NW; %#
% a1: es el valor de la sumatoria de todas las distancias
a1=sum(D); %m
% a2: es una matriz de [(NW-1)*1], el valor corresponde a la suma de las
% distancias en forma descendente
a2=ones(s,1); %m
for c = 1:s

```

```

    if c==1
        e=D(c);
        ee=a1-e;
        a2 (c)=ee;
    elseif c>=2 && c==NW
        for b=2:c
            if b==2
                e=D(b);
                ee=ee-e;
                a2 (b)=ee;
            elseif b>2;
                e=D(b);
                ee=ee-e;
                a2 (b)=ee;
            end
            a2(b)=ee;
        end
        a2 (c)=ee;
    else
        end
end
% a3: es una matriz de [(NW-1)*1], los valores corresponden al producto de
% la carga por la distancia en X... para la sumatoria de momentos
a3=ones (s,1); %KN*m
for c = 1:s
    if c==1
        e=a2 (c);
        ee=W(c);
        eee=e*ee;
        a3 (c)=eee;
    elseif c>=2 && c==NW
        for b=2:c
            if b==2
                e=a2 (b);
                ee=W(b);
                eee=e*ee;
                a3 (b)=eee;
            else
                e=a2 (b);
                ee=W(b);
                eee=e*ee;
                a3 (b)=eee;
            end
            a3(b)=eee;
        end
        a3 (c)=eee;
    else
        end
end
% a4: es la sumatoria de la matriz a3
a4=sum(a3); %KN*m
% Con la sumatoria de momentos en el punto B, se despeja la variable Ay,
% en donde: a4 corresponde al producto de las cargas y su distancia en x, y
% a1 corresponde a la distancia en x de Ay.
Ay=a4/a1; %KN
% a5: es la sumatoria de las cargas
a5=sum(W); %KN
% By: es la fuerza en "y" del punto final. Se resuelve con sumatoria de
% momentos
By=a5-Ay; %KN
% Ax: es la fuerza en "x" del punto inicial. en función de la distancia 1
% y el desnivel 1=> h1
% a7: es la distancia 1
a7=D(1); %m
Ax=(a7*Ay)/h1; %KN
% TA: es la tensión en el punto incial
A=sqrt(Ax^2+Ay^2); %KN

```

```

% a8: es la sumatoria de las distancias en formar ascendente
s2=NW-1;
a8=ones(s2,1); %m
for c = 1:s2
    if c==1
        e1=D(c);
        e2=c+1;
        e3=D(e2)+e1;
        a8 (c)=e3;
    elseif c>=2 && c==s2
        for b=2:c
            if b==2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8 (b)=e3;
            elseif b>2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8 (b)=e3;
            end
            a8(b)=e3;
        end
    else
        end
end
% a9: es una matriz de la sumatoria de cargas
s3=NW;
a9=ones(s3,1); %KN
for c = 1:s3
for c = 1:s3
    if c==1
        e1=c;
        e2=W(e1);
        a9 (c)=e2;
    elseif c>=2 && c==s3
        for b=2:c
            if b==2
                e1=b;
                e2=W(e1)+e2;
                a9 (b)=e2;
            elseif b>2
                e1=b;
                e2=W(e1)+e2;
                a9 (b)=e2;
            end
            a9 (b)=e2;
        end
    else
        end
end
% h: es la matriz de los datos de alturas correspondiente a las distancias
% en x y a las cargas
s4=NW;
h=ones(s4,1); %m
for c = 1:s4
    if c==1
        h (c)=h1;
    elseif c>=2 && c==s4
        for b=2:c
            if b==2
                e1=b;
                e2=D(e1);
                e3=b-1;

```

```

        e4=a9(e3);
        e5=e2*e4;
        e6=b-1;
        e7=a8(e6);
        e8=e7*Ay;
        e9=(e8-e5)/Ax;
        h(b)=e9;
    elseif b>2
        e1=b;
        e2=D(e1);
        e3=b-1;
        e4=a9(e3);
        e5=e2*e4+e5;
        e6=b-1;
        e7=a8(e6);
        e8=e7*Ay;
        e9=(e8-e5)/Ax;
        h(b)=e9;
    end
    h(b)=e9;
end
h(c)=e9;
else
end
end
% a10= es una matriz [(NW+1),1] que corresponde a las alturas en "y"
s5=NW+1;
a10=ones(s5,1); %m
for c = 1:s5
    if c==1
e1=0;
        e2=c;
        e3=h(e2);
        e5=abs(e1-e3);
        a10(c)=e5;
    elseif c>=2 && c==s5
        for b=2:c
            if b==2
                e1=b-1;
                e2=b;
                e3=h(e2);
                e4=h(e1);
                e5=abs(e4-e3);
                a10(b)=e5;
            elseif b<s5
                e1=b-1;
                e2=b;
                e3=h(e2);
                e4=h(e1);
                e5=abs(e4-e3);
                a10(b)=e5;
            elseif b==s5
                e1=0;
                e5=abs(e1-e3);
                a10(b)=e5;
            end
            a10(b)=e5;
        end
    end
    a10(c)=e5;
else
end
end
% a11: es un matriz [(NW+1),1] que corresponde a las distancias en "x"
a11=D; %m
% a12: es un matriz [(NW),1] que corresponde a la hipotenusa de a11=x

```

```

% a10=y a12=raiz(a11^2+a10^2)
a12=sqrt(a10.^2+a11.^2); %m
% T: es la matriz de las tensiones
s6=NW+1;
T=ones(s,1); %KN
for c = 1:s6
    if c==1 %primer punto
        e1=h(c);
        if 0==e1
            %sumatoria de fuerzas en x
            e3=Ax;
            e11=e3;
            T (c)=e11;
        elseif 0<e1
            %sumatoria de fuerzas en y
            e3=a10(c);
            e4=a12(c);
            e5=e3/e4;
            e6=Ay;
            e11=e6/e5;
            T (c)=e11;
        end
        T (c)=e11;
    elseif c>=2 && c==s6
        for b=2:c
            if b==2 %Segundo punto
                e2=h(b);
                bb=b-1;
            end
            T (c)=e11;
        end
    elseif c>=2 && c==s6
        for b=2:c
            if b==2 %Segundo punto
                e2=h(b);
                bb=b-1;
                e3=a10(bb);
                e4=a11(bb);
                e5=a12(bb);
                e6=a10(b);
                e8=a12(b);
                e9=T(bb);
                e10=W(bb);
                if 0==e1 && e1==e2
                    %sumatoria de fuerzas en x
                    e11=e9;
                    T (b)=e11;
                elseif 0==e1 && e1<e2
                    %sumatoria de fuerzas en x
                    e11=e9/(e6/e8);
                    T (b)=e11;
                elseif 0<e1 && e1==e2
                    %sumatoria de fuerzas en x
                    e11=e9*e4/e5;
                    T (b)=e11;
                elseif 0<e1 && e1<e2
                    %sumatoria de fuerzas en y
                    e11=((e9*e3/e5)-e10)/(e6/e8);
                    T (b)=e11;
                elseif 0<e1 && e1>e2
                    %sumatoria de fuerzas en y
                    e11=(e10-(e9*e3/e5))/(e6/e8);
                    T (b)=e11;
                end
                T (b)=e11;
            end
            if b<s6 %Tercer punto hasta el penúltimo punto
                bb=b-1;
                bbb=b-2;
            end
        end
    end
end

```

```

e1=h(bbb);
e2=h(bb);
e3=h(b);
e4=a10(bb);
e5=a11(bb);
e6=a12(bb);
e7=a10(b);
e8=a11(b);
e9=a12(b);
e10=W(bb);
e12=T(bb);
if e1==e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12;
    T(b)=e11;
elseif e1<e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12*e5/e6;
    T(b)=e11;

elseif e1<e2 && e2<e3
    %sumatoria de fuerzas en y
    e11=((e12*e4/e6)-e10)/(e7/e9);
    T(b)=e11;
elseif e1<e2 && e2>e3
    %sumatoria de fuerzas en y
    %sumatoria de fuerzas en x
    e11=e12*e5/e6;
    T(b)=e11;
elseif e1<e2 && e2<e3
    %sumatoria de fuerzas en y
    e11=((e12*e4/e6)-e10)/(e7/e9);
    T(b)=e11;
elseif e1<e2 && e2>e3
    %sumatoria de fuerzas en y
    e11=(e10-(e12*e4/e6))/(e7/e9);
    T(b)=e11;
elseif e1==e2 && e2>e3
    %sumatoria de fuerzas en x
    e11=e12/(e8/e9);
    T(b)=e11;
elseif e1==e2 && e2<e3
    %sumatoria de fuerzas en x
    e11=e12/(e8/e9);
    T(b)=e11;
elseif e1>e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12*e5/e6;
    T(b)=e11;
elseif e1>e2 && e2<e3
    %sumatoria de fuerzas en x
    e11=(e12*e5/e6)/(e8/e9);
    T(b)=e11;
elseif e1>e2 && e2>e3
    %sumatoria de fuerzas en y
    e11=(e10+(e12*e4/e6))/(e7/e9);
    T(b)=e11;
end
T(b)=e11;
elseif b==s6
    bb=b-1;
    bbb=b-2;
    e1=h(bbb);
    e2=h(bb);
    e3=0;
    e4=a10(bb);
    e5=a11(bb);

```



```

e6=a12(bb);
e7=a10(b);
e8=a11(b);
e9=a12(b);
e10=W(bb);
e12=T(bb);
if e1==e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12;
    T (b)=e11;
elseif e1<e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12*e5/e6;
    T (b)=e11;
elseif e1<e2 && e2<e3
    %sumatoria de fuerzas en y
    e11=((e12*e4/e6)-e10)/(e7/e9);
    T (b)=e11;

elseif e1<e2 && e2>e3
    %sumatoria de fuerzas en y
    e11=(e10-(e12*e4/e6))/(e7/e9);
    T (b)=e11;

elseif e1==e2 && e2>e3
    %sumatoria de fuerzas en x
    e11=e12/(e8/e9);
    T (b)=e11;
elseif e1==e2 && e2<e3
    %sumatoria de fuerzas en x
    e11=e12/(e8/e9);
    T (b)=e11;
elseif e1>e2 && e2==e3
    %sumatoria de fuerzas en x
    e11=e12*e5/e6;
    T (b)=e11;
elseif e1>e2 && e2<e3
    %sumatoria de fuerzas en x
    e11=(e12*e5/e6)/(e8/e9);
    T (b)=e11;
elseif e1>e2 && e2>e3
    %sumatoria de fuerzas en y
    e11=(e10+(e12*e4/e6))/(e7/e9);
    T (b)=e11;
end
    T (b)=e11;
end
    T (b)=e11;
end
    T (c)=e11;
end

% TBx= es la tension en B. el punto final en la proyección de x
Bx=Ax; %KN
% TB= es la tensión en el punto final
B=sqrt(Bx^2+By^2); %KN
% Tmax= la tensión máxima en el cable
Tmax=max(T);
% Tmin= la tension mínima en el cable
Tmin=min(T);
% L=Longitud a salvar o vano
L=sum(D);
% S=Longitud del cable
S=sum(a12);
elseif 0>YB
    %% Procedimiento

```

```

% Datos: a ser una matriz de [1*NW] se invierte a [NW*1]
% Datos de las cargas puntuales
W=W'; %KN
% Datos de las distancias en x: punto inicial, cargas y punto final
D=D'; %m
% s: es el valor correspondiente al número de cargas NW
s =NW; %#
% a1: es el valor de la sumatoria de todas las distancias
a1=sum(D); %m
% a2: es una matriz de [(NW-1)*1], el valor corresponde a la suma de las
% distancias en forma descendente
a2=ones(s,1); %m
for c = 1:s
    if c==1
        e=D(c);
        ee=a1-e;
        a2 (c)=ee;
    elseif c>=2 && c==NW
        for b=2:c
            if b==2
                e=D(b);
                ee=ee-e;
                a2 (b)=ee;
            elseif b>2;
                e=D(b);
                ee=ee-e;
                a2 (b)=ee;
            end
            a2 (b)=ee;
        end
        a2 (c)=ee;
    else
        end
end
% a3: es una matriz de [(NW-1)*1], los valores corresponden al producto de
% la carga por la distancia en X... para la sumatoria de momentos
a3=ones (s,1); %KN*m
for c = 1:s
    if c==1
        e=a2 (c);
        ee=W(c);
        eee=e*ee;
        a3 (c)=eee;
    elseif c>=2 && c==NW
        for b=2:c
            if b==2
                e=a2 (b);
                ee=W(b);
                eee=e*ee;
                a3 (b)=eee;
            else
                e=a2 (b);
                ee=W(b);
                eee=e*ee;
                a3 (b)=eee;
            end
            a3 (b)=eee;
        end
        a3 (c)=eee;
    else
        end
end
% a4: es la sumatoria de la matriz a3
a4=sum(a3); %KN*m
% v1: es la variable correspondiente a la distancia 1
v1=D(1);
% v2: es la variable correspondiente al producto entre la suma de todas

```

```

% las distancias por la altura 1, dividido para la distancia 1 (v1)
v2=a1*h1/v1;
% v3: es la variable correspondiente a la resta entre la v2 y el
% desnivel en el punto B (YB)
v3=v2+YB;
% Ax: es la variable correspondiente a la reacción en el punto A en
% reacción en x
Ax=a4/v3;
% Ay: es la variable correspondiente a la reacción en el punto A en
% reacción en y
Ay=h1*Ax/v1;
% TA= es la tensión en el punto inicial
A=sqrt(Ax^2+Ay^2); %KN
% Bx: es la variable correspondiente a la reacción en el punto B en
% reacción en x
Bx=Ax;
% By: es la variable correspondiente a la reacción en el punto B en
% reacción en y
% a5: es la sumatoria de las cargas
a5=sum(W); %KN
By=a5-Ay;
% TA= es la tensión en el punto inicial
B=sqrt(Bx^2+By^2); %KN
% a8: es la sumatoria de las distancias en formar ascendente
s2=NW-1;
a8=ones(s2,1); %m
for c = 1:s2
    if c==1
        e1=D(c);
        e2=c+1;
        e3=D(e2)+e1;
        a8(c)=e3;
    elseif c>=2 && c==s2
        for b=2:c
            if b==2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8(b)=e3;
            elseif b>2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8(b)=e3;
            end
        end
        a8(b)=e3;
    end
    a8(c)=e3;
else
end
end
% a9: es una matriz de la sumatoria de cargas
s3=NW;
a9=ones(s3,1); %KN
for c = 1:s3
    if c==1
        e1=c;
        e2=W(e1);
        a9(c)=e2;
    elseif c>=2 && c==s3
        for b=2:c
            if b==2
                e1=b;
                e2=W(e1)+e2;
                a9(b)=e2;
            elseif b>2
                e1=b;

```

```

        e2=W(e1)+e2;
        a9 (b)=e2;
    end
    a9 (b)=e2;
end
    a9 (c)=e2;
else
end
end
% h: es la matriz de los datos de alturas correspondiente a las distancias
% en x y a las cargas
s4=NW;
h=ones(s4,1); %m
for c = 1:s4
    if c==1
        h (c)=h1;
    elseif c>=2 && c==s4
        for b=2:c
            if b==2
                e1=b;
                e2=D(e1);
                e3=b-1;
                e4=a9(e3);
                e5=e2*e4;
                e6=b-1;
                e7=a8(e6);
                e8=e7*Ay;
                e9=(e8-e5)/Ax;
                h (b)=e9;
            elseif b>2
                e1=b;
                e2=D(e1);
                e3=b-1;
                e4=a9(e3);
                e5=e2*e4+e5;
                e6=b-1;
                e7=a8(e6);
                e8=e7*Ay;
                e9=(e8-e5)/Ax;
                h (b)=e9;
            end
            h (b)=e9;
        end
        h (c)=e9;
    else
end
end
% a10= es una matriz [(NW+1),1] que corresponde a las alturas en "y"
s5=NW+1;
a10=ones(s5,1); %m
for c = 1:s5
    if c==1
        e1=0;
        e2=c;
        e3=h(e2);
        e5=abs(e1-e3);
        a10 (c)=e5;
    elseif c>=2 && c==s5
        for b=2:c
            if b==2
                e1=b-1;
                e2=b;

```

```

        e3=h(e2);
        e4=h(e1);
        e5=abs(e4-e3);
        a10(b)=e5;
    elseif b<s5
        e1=b-1;
        e2=b;
        e3=h(e2);
        e4=h(e1);
        e5=abs(e4-e3);
        a10(b)=e5;
    elseif b==s5
        e1=-YB;
        e5=abs(e1-e3);
        a10(b)=e5;
    end
    a10(b)=e5;
end
a10(c)=e5;
else
else
end
end
% a11: es un matriz [(NW+1),1] que corresponde a las distancias en "x"
a11=D; %m
% a12: es un matriz [(NW),1] que corresponde a la hipotenusa de a11=x
% a10=y a12=raiz(a11^2+a10^2)
a12=sqrt(a10.^2+a11.^2); %m
% T: es la matriz de las tensiones
s6=NW+1;
T=ones(s,1); %KN
for c = 1:s6
    if c==1 %primer punto
        e1=h(c);
        if 0==e1
            %sumatoria de fuerzas en x
            e3=Ax;
            e11=e3;
            T(c)=e11;

        elseif 0<e1
            %sumatoria de fuerzas en y
            e3=a10(c);
            e4=a12(c);
            e5=e3/e4;
            e6=Ay;
            e11=e6/e5;
            T(c)=e11;
        end
        T(c)=e11;
    elseif c>=2 && c==s6
        for b=2:c
            if b==2 %Segundo punto
                e2=h(b);
                bb=b-1;
                e3=a10(bb);
                e4=a11(bb);
                e5=a12(bb);
                e6=a10(b);
                e8=a12(b);
                e9=T(bb);
                e10=W(bb);
                if 0==e1 && e1==e2
                    %sumatoria de fuerzas en x

```

```

        e11=e9;
        T (b)=e11;
    elseif 0==e1 && e1<e2
        %sumatoria de fuerzas en x
        e11=e9/(e6/e8);
        T (b)=e11;
    elseif 0<e1 && e1==e2
        %sumatoria de fuerzas en x
        e11=e9*e4/e5;
        T (b)=e11;
    elseif 0<e1 && e1<e2
        %sumatoria de fuerzas en y
        e11=((e9*e3/e5)-e10)/(e6/e8);
        T (b)=e11;
    elseif 0<e1 && e1>e2
        %sumatoria de fuerzas en y
        e11=(e10-(e9*e3/e5))/(e6/e8);
        T (b)=e11;
T (b)=e11;
elseif b<s6 %Tercer punto hasta el penúltimo punto
    bb=b-1;
    bbb=b-2;
    e1=h(bbb);
    e2=h(bb);
    e3=h(b);
    e4=a10(bb);
    e5=a11(bb);
    e6=a12(bb);
    e7=a10(b);
    e8=a11(b);
    e9=a12(b);
    e10=W(bb);
    e12=T(bb);
    if e1==e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12;
        T (b)=e11;
    elseif e1<e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1<e2 && e2<e3
        %sumatoria de fuerzas en y
        e11=((e12*e4/e6)-e10)/(e7/e9);
        T (b)=e11;
    elseif e1<e2 && e2>e3
        %sumatoria de fuerzas en y
        e11=(e10-(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    elseif e1==e2 && e2>e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1==e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1>e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=(e12*e5/e6)/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2>e3
        %sumatoria de fuerzas en y

```

```

        e11=(e10+(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    end
    T (b)=e11;
elseif b==s6
    bb=b-1;
    bbb=b-2;
    e1=h(bbb);
    e2=h(bb);
    e3=-YB;
    e4=a10(bb);
    e5=a11(bb);
    e6=a12(bb);
    e7=a10(b);
    e8=a11(b);
    e9=a12(b);
    e10=W(bb);
    e12=T(bb);
    if e1==e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12;
        T (b)=e11;
    elseif e1<e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1<e2 && e2<e3
        %sumatoria de fuerzas en y
        e11=((e12*e4/e6)-e10)/(e7/e9);
        T (b)=e11;
    elseif e1<e2 && e2>e3
        %sumatoria de fuerzas en y
        e11=(e10-(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    elseif e1==e2 && e2>e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1==e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1>e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=(e12*e5/e6)/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2>e3
        %sumatoria de fuerzas en y
        e11=(e10+(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    end
    T (b)=e11;
end
    T (b)=e11;
end
    T (c)=e11;
end
end
% Tmax= la tensión máxima en el cable
Tmax=max(T);
% Tmin= la tension mínima en el cable
Tmin=min(T);
% L=Longitud a salvar o vano

```

```

L=sum(D);
% S=Longitud del cable
S=sum(a12);
elseif 0<YB
    %% Procedimiento
    % Datos: a ser una matriz de [1*NW] se invierte a [NW*1]
    % Datos de las cargas puntuales
    W=W'; %KN
    % Datos de las distancias en x: punto inicial, cargas y punto final
    D=D'; %m
    % s: es el valor correspondiente al número de cargas NW
    s =NW; %#
    % a1: es el valor de la sumatoria de todas las distancias
    a1=sum(D); %m
    % a2: es una matriz de [(NW-1)*1], el valor corresponde a la suma de las
    % distancias en forma descendente
    a2=ones(s,1); %m
    for c = 1:s
        if c==1
            e=D(c);
            ee=a1-e;
            a2 (c)=ee;
        elseif c>=2 && c==NW
            for b=2:c
                if b==2
                    e=D(b);
                    ee=ee-e;
                    a2 (b)=ee;
                elseif b>2;
                    e=D(b);
                    ee=ee-e;
                    a2 (b)=ee;
                end
            end
            a2(b)=ee;
        end
        a2 (c)=ee;
    end
    else
        end
    end
    % a3: es una matriz de [(NW-1)*1], los valores corresponden al producto de
    % la carga por la distancia en X... para la sumatoria de momentos
    a3=ones (s,1); %KN*m
    for c = 1:s
        if c==1
            e=a2(c);
            ee=W(c);
            eee=e*ee;
            a3 (c)=eee;
        elseif c>=2 && c==NW
            for b=2:c
                if b==2
                    e=a2(b);
                    ee=W(b);
                    eee=e*ee;
                    a3 (b)=eee;
                else
                    e=a2(b);
                    ee=W(b);
                    eee=e*ee;
                    a3 (b)=eee;
                end
            end
            a3(b)=eee;
        end
        a3 (c)=eee;
    end
    else
        end
    end
end

```



```

% a4: es la sumatoria de la matriz a3
a4=sum(a3); %KN*m
% v1: es la variable correspondiente a la distancia 1
v1=D(1);
% v2: es la variable correspondiente a la altura 1, dividido para la
% distancia 1 (v1)
v2=h1/v1*a1;
% v3: es la variable correspondiente a la resta entre la v2 y el
% desnivel en el punto B (YB)
v3=v2+YB;
% Ax: es la variable correspondiente a la reacción en el punto A en
% reacción en x
Ax=a4/v3;
% Ay: es la variable correspondiente a la reacción en el punto A en
% reacción en y
Ay=h1*Ax/v1;
% TA= es la tensión en el punto inicial
A=sqrt(Ax^2+Ay^2); %KN
% Bx: es la variable correspondiente a la reacción en el punto B en
% reacción en x
Bx=Ax;
% By: es la variable correspondiente a la reacción en el punto B en
% reacción en y
% a5: es la sumatoria de las cargas
a5=sum(W); %KN
By=a5-Ay;
% TA= es la tensión en el punto inicial
B=sqrt(Bx^2+By^2); %KN
% a8: es la sumatoria de las distancias en formar ascendente
s2=NW-1;
a8=ones(s2,1); %m
for c = 1:s2
    if c==1
        e1=D(c);
        e2=c+1;
        e3=D(e2)+e1;
        a8(c)=e3;
    elseif c>=2 && c==s2
        for b=2:c
            if b==2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8(b)=e3;
            elseif b>2
                e2=b+1;
                ee2=D(e2);
                e3=e3+ee2;
                a8(b)=e3;
            end
        end
        a8(b)=e3;
    end
    a8(c)=e3;
else
end
end
% a9: es una matriz de la sumatoria de cargas
s3=NW;
a9=ones(s3,1); %KN
for c = 1:s3
    if c==1
        e1=c;
        e2=W(e1);
        a9(c)=e2;
    elseif c>=2 && c==s3
        for b=2:c

```

```

        if b==2
            e1=b;
            e2=W(e1)+e2;
            a9 (b)=e2;
        elseif b>2
            e1=b;
            e2=W(e1)+e2;
            a9 (b)=e2;
        end
        a9 (b)=e2;
    end
    a9 (c)=e2;
else
end
% h: es la matriz de los datos de alturas correspondiente a las distancias
% en x y a las cargas
s4=NW;
h=ones(s4,1); %m
for c = 1:s4
    if c==1
        h (c)=h1;
    elseif c>=2 && c==s4
        for b=2:c
            if b==2
                e1=b;
                e2=D(e1);
                e3=b-1;
                e4=a9 (e3);
                e5=e2*e4;
                e6=b-1;
                e7=a8 (e6);
                e8=e7*Ay;
                e9=(e8-e5)/Ax;
                h (b)=e9;
            elseif b>2
                e1=b;
                e2=D(e1);
                e3=b-1;
                e4=a9 (e3);
                e5=e2*e4+e5;
                e6=b-1;
                e7=a8 (e6);
                e8=e7*Ay;
                e9=(e8-e5)/Ax;
                h (b)=e9;
            end
        end
        h (b)=e9;
    end
    h (c)=e9;
else
end
end
% a10= es una matriz [(NW+1),1] que corresponde a las alturas en "y"
s5=NW+1;
a10=ones(s5,1); %m
for c = 1:s5
    if c==1
        e1=0;
        e2=c;
        e3=h(e2);
        e5=abs(e1-e3);
        a10 (c)=e5;
    elseif c>=2 && c==s5
        for b=2:c
            if b==2

```

```

        e1=b-1;
        e2=b;
        e3=h(e2);
        e4=h(e1);
        e5=abs(e4-e3);
        a10 (b)=e5;
    elseif b<s5
        e1=b-1;
        e2=b;
        e3=h(e2);
        e4=h(e1);
        e5=abs(e4-e3);
        a10 (b)=e5;
    elseif b==s5
        e1=-YB;
        e5=abs(e1-e3);
        a10 (b)=e5;
    end
    a10 (b)=e5;
end
a10 (c)=e5;
else
end
end
% a11: es un matriz [(NW+1),1] que corresponde a las distancias en "x"
a11=D; %m
% a12: es un matriz [(NW),1] que corresponde a la hipotenusa de a11=x
% a10=y a12=raiz(a11^2+a10^2)
a12=sqrt(a10.^2+a11.^2); %m
% T: es la matriz de las tensiones
s6=NW+1;
T=ones(s,1); %KN
for c = 1:s6
    if c==1 %primer punto
        e1=h(c);
        if 0==e1
            %sumatoria de fuerzas en x
            e3=Ax;
            e11=e3;
            T (c)=e11;

        elseif 0<e1
            %sumatoria de fuerzas en y
            e3=a10(c);
            e4=a12(c);
            e5=e3/e4;
            e6=Ay;
            e11=e6/e5;
            T (c)=e11;
        end
        T (c)=e11;
    elseif c>=2 && c==s6
        for b=2:c
            if b==2 %Segundo punto
                e2=h(b);
                bb=b-1;
                e3=a10(bb);
                e4=a11(bb);
                e5=a12(bb);
                e6=a10(b);
                e8=a12(b);
                e9=T(bb);
                e10=W(bb);
                if 0==e1 && e1==e2
                    %sumatoria de fuerzas en x
                    e11=e9;
                    T (b)=e11;
                end
            end
        end
    end
end

```

```

elseif 0==e1 && e1<e2
    %sumatoria de fuerzas en x
    e11=e9/(e6/e8);
    T (b)=e11;
elseif 0<e1 && e1==e2
    %sumatoria de fuerzas en x
    e11=e9*e4/e5;
    T (b)=e11;
elseif 0<e1 && e1<e2
    %sumatoria de fuerzas en y
    e11=((e9*e3/e5)-e10)/(e6/e8);
    T (b)=e11;
end
T (b)=e11;
elseif b<s6 %Tercer punto hasta el penúltimo punto
    bb=b-1;
    bbb=b-2;
    e1=h(bbb);
    e2=h(bb);
    e3=h(b);
    e4=a10(bb);
    e5=a11(bb);
    e6=a12(bb);
    e7=a10(b);
    e8=a11(b);
    e9=a12(b);
    e10=W(bb);
    e12=T(bb);
    if e1==e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12;
        T (b)=e11;
    elseif e1<e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1<e2 && e2<e3
        %sumatoria de fuerzas en y
        e11=((e12*e4/e6)-e10)/(e7/e9);
        T (b)=e11;
    elseif e1<e2 && e2>e3
        %sumatoria de fuerzas en y
        e11=(e10-(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    elseif e1==e2 && e2>e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1==e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=e12/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2==e3
        %sumatoria de fuerzas en x
        e11=e12*e5/e6;
        T (b)=e11;
    elseif e1>e2 && e2<e3
        %sumatoria de fuerzas en x
        e11=(e12*e5/e6)/(e8/e9);
        T (b)=e11;
    elseif e1>e2 && e2>e3
        %sumatoria de fuerzas en y
        e11=(e10+(e12*e4/e6))/(e7/e9);
        T (b)=e11;
    end
    T (b)=e11;
elseif b==s6

```

```

        bb=b-1;
        bbb=b-2;
        e1=h(bbb);
        e2=h(bb);
        e3=YB;
        e4=a10(bb);
        e5=a11(bb);
        e6=a12(bb);
        e7=a10(b);
        e8=a11(b);
        e9=a12(b);
        e10=W(bb);
        e12=T(bb);
        if e1==e2 && e2==e3
            %sumatoria de fuerzas en x
            e11=e12;
            T (b)=e11;
        elseif e1<e2 && e2==e3
            %sumatoria de fuerzas en x
            e11=e12*e5/e6;
            T (b)=e11;
        elseif e1<e2 && e2<e3
            %sumatoria de fuerzas en y
            e11=((e12*e4/e6)-e10)/(e7/e9);
            T (b)=e11;
        elseif e1<e2 && e2>e3
            %sumatoria de fuerzas en y
            e11=(e10-(e12*e4/e6))/(e7/e9);
            T (b)=e11;
        elseif e1==e2 && e2>e3
            %sumatoria de fuerzas en x
            e11=e12/(e8/e9);
            T (b)=e11;
        elseif e1==e2 && e2<e3
            %sumatoria de fuerzas en x
            e11=e12/(e8/e9);
            T (b)=e11;
        elseif e1>e2 && e2==e3
            %sumatoria de fuerzas en x
            e11=e12*e5/e6;
            T (b)=e11;
        elseif e1>e2 && e2<e3
            %sumatoria de fuerzas en x
            e11=(e12*e5/e6)/(e8/e9);
            T (b)=e11;
        elseif e1>e2 && e2>e3
            %sumatoria de fuerzas en y
            e11=(e10+(e12*e4/e6))/(e7/e9);
            T (b)=e11;
        end
        T (b)=e11;
    end
    T (b)=e11;
end
    T (c)=e11;
end
end
% Tmax= la tensión máxima en el cable
Tmax=max(T);
% Tmin= la tension mínima en el cable
Tmin=min(T);
% L=Longitud a salvar o vano
L=sum(D);
% S=Longitud del cable
S=sum(a12);
end
%Posición inicial

```

```

Lo=0;
Lf=L;
Ho=0;
Hf=YB;
x=[Lo Lf ];
y=[Ho Hf ];
s=NW+2;
xx=ones (s,1);
for c = 1:s
    if c==1
        e4=0;
        xx (c)=e4;
    elseif c>=2 && c==s
        for b=2:c
            if b==2
                e1=b-1;
                e2=xx (e1);
                e3=D (e1);
                e4=e2+e3;
                xx (b)=e4;
            elseif b>2
                e1=b-1;
                e2=xx (e1);
                e3=D (e1);
                e4=e2+e3;
            end
            xx (b)=e4;
        end
        xx (c)=e4;
    else
    end
end
s=NW+2;
yy=ones (s,1);
for c = 1:s
    if c==1
        e4=0;
        yy (c)=e4;
    elseif c>=2 && c==s
        for b=2:c
            if b==2
                e1=b-1;
                e4=-h (e1);
                yy (b)=e4;
            elseif b>2 && b<s
                e1=b-1;
                e4=-h (e1);
                yy (b)=e4;
            elseif b==s
                e4=YB;
                yy (b)=e4;
            end
            yy (b)=e4;
        end
        yy (c)=e4;
    else
    end
end
xx=xx';
yy=yy';
YB=-YB;
Ax=-Ax;
str = datestr(now);
end

```

Figura 79. Código del experimento 3 – Función “CablesCargasPuntuales1”
Fuente: Autor

```

function [TA, TAx, TAy, TBAB,TBBC, TB, TBx, TBy,TC,TCx,TCy, TAB, TABmax,
TABmin, hAB, LAB, SAB,TBC, TBCmax, TBCmin, hBC, LBC, SBC,str,NT,
NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,x,y,xx,yy] = CableCargasPuntuales2
(NT, NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determinación de tensiones en cables que soportaban cuerpos suspendidos. %
%Se considera el caso de un número arbitrario N de cuerpos suspendidos de %
%un cable. Se supone que el peso del cable puede despreciarse comparado %
%con los pesos suspendidos, y que el cable es lo bastante flexible para %
%aproximar su forma con una serie de segmentos rectos. %
%
% Caso 1: El Punto Inicial y Punto Final se encuentran a nivel %
% Caso 1: El Punto Inicial y Punto Final se encuentran a desnivel YB %
%
% Variables de entrada: %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DESARROLLO
%% IDENTIFICAR EL NÚMERO DE TRAMO
%% PARA EL TRAMO AB
%% PARA EL TRAMO AB
NW=NWAB;
W=WAB;
D=DAB;
h1=h1AB;
[A, Ax, Ay, B,Bx, By, T, Tmax, Tmin, h, L, S, x,y,xx,yy] =
CableCargasDiscretas1 (NW,W,D,h1,YB);
TA=A;
TAx=Ax;
TAy=Ay;
TBAB=B;
TBx=Bx;
%e1= es igual a la variable TBx
e1=TBx;
TBy=By;
%e2: es igual a la variable TBy
e2=TBy;
TAB=T;
TABmax=Tmax;
TABmin=Tmin;
hAB=h;
LAB=L;
SAB=S;
% PARA GRAFICAS
xAB=x;
yAB=y;
xxAB=xx;
yyAB=yy;
%para reconocer el valor YB=YYBB
YYBB=YB;
YYCC=YC;
%% PARA EL TRAMO BC
if YC==0 && YB==0 && YB==YC
    %caso 1
elseif YB==0 && YC<0 && YB>YC
    %caso 2
elseif YB==0 && YC<0 && YB<YC
    %caso 3
elseif YB<0 && YC==0 && YB<YC
    %caso 4
    YC=-YB;
elseif YB<0 && YC==YB && YC<0
    %caso 5
    YC=0;
elseif YB<0 && YB>YC && YC<0
    %caso 6
    YC=YC-YB;
elseif YB<0 && YC>0 && YB<YC

```

```

        %caso 7
        YC=-YB+YC;
elseif YB>0 && YC==0 && YB>YC
    %caso 8
    YC=-YB;
elseif YB>0 && YC>0 && YB==YC
    %caso 9
    YC=0;
elseif YB>0 && YC>0 && YB<YC
    %caso 10
    YC=YC-YB;
elseif YB>0 && YC<0 && YB>YC
    %caso 11
    YC=-YB+YC;
elseif YB<0 && YC<0 && YB<YC
    %caso 12
    YC=- (YB-YC);
elseif YB>0 && YC>0 && YB>YC
    %caso 13
    YC=(YB-YC);
end
NW=NWBC;
W=WBC;
D=DBC;
h1=h1BC;
YB=YC;
[A,Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S, x,y,xx,yy] =
CableCargasDiscretas1 (NW,W,D,h1,YB);
TBBC=A;
TBx=Ax;
%e3: es igual a la variable TBy
e3=TBx;
TBy=Ay;
%e4: es igual a la variable TBy
e4=TBy;
TC=B;
TCx=Bx;
TCy=By;
TBC=T;
TBCmax=Tmax;
TBCmin=Tmin;
hBC=h;
LBC=L;
SBC=S;
% PARA GRAFICAS
xBC=x;
yBC=y;
xxBC=xx;
yyBC=yy;
%%
YC=YYCC;
YB=YYBB;
%% REACCIÓN EN B
    %sumatoria en la dirección Y
    TBy=e2+e4;
    %sumatoria en la dirección X
    TBx=e1+e3;
    %Resultante
    TB=sqrt (TBy*TBy+TBx*TBx);
    %PARA REALIZAR LAS GRÁFICAS
if YC==0 && YB==0 && YB==YC
    %caso 1
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')

```



```

        %cable deformado
        xxBC=xxBC+LAB;
        xx=[xxAB xxBC];
        yy=[yyAB yyBC];
        %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC<0 && YB>YC
    %caso 2
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YB<YC
    %caso 3
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC==0 && YB<YC
    %caso 4
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    yBC=yBC-YB;
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC==YB && YC<0
    %caso 5
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    yBC=yBC-YB;
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB>YC && YC<0
    %caso 6
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    yBC=yBC-YB;
    y=[yAB yBC];
    %subplot(221),
plot(x,y,'r')
    %cable deformado

```

```

xxBC=xxBC+LAB;
xx=[xxAB xxBC];
yyBC=yyBC-YB;
yy=[yyAB yyBC];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC>0 && YB<YC
%caso 7
%posición inicial
xBC=xBC+LAB;
x=[xAB xBC];
yBC=yBC-YB;
y=[yAB yBC];
%subplot(221),plot(x,y,'r')
%cable deformado
xxBC=xxBC+LAB;
xx=[xxAB xxBC];
yyBC=yyBC-YB;
yy=[yyAB yyBC];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC==0 && YB>YC
%caso 8
%posición inicial
xBC=xBC+LAB;
x=[xAB xBC];
yBC=yBC-YB;
y=[yAB yBC];
%subplot(221),plot(x,y,'r')
%cable deformado
xxBC=xxBC+LAB;
xx=[xxAB xxBC];
yyBC=yyBC-YB;
yy=[yyAB yyBC];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YB==YC
%caso 9
%posición inicial
xBC=xBC+LAB;
x=[xAB xBC];
yBC=yBC-YB;
y=[yAB yBC];
%subplot(221),plot(x,y,'r')
%cable deformado
xxBC=xxBC+LAB;
xx=[xxAB xxBC];
yyBC=yyBC-YB;
yy=[yyAB yyBC];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YB<YC
%caso 10
%posición inicial
xBC=xBC+LAB;
x=[xAB xBC];
yBC=yBC-YB;
y=[yAB yBC];
%subplot(221),plot(x,y,'r')
%cable deformado
xxBC=xxBC+LAB;
xx=[xxAB xxBC];
yyBC=yyBC-YB;
YY=[YYAB YYBC];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC<0 && YB>YC
%caso 11
%posición inicial
xBC=xBC+LAB;
x=[xAB xBC];

```

```

        yBC=yBC-YB;
        y=[yAB yBC];
        %subplot(221),plot(x,y,'r')
        %cable deformado
        xxBC=xxBC+LAB;
        xx=[xxAB xxBC];
        yyBC=yyBC-YB;
        yy=[yyAB yyBC];
        %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YB<YC
    %caso 12
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    yBC=yBC-YB;
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YB>YC
    %caso 13
    %posición inicial
    xBC=xBC+LAB;
    x=[xAB xBC];
    yBC=yBC-YB;
    y=[yAB yBC];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy=[yyAB yyBC];
    %subplot(221),plot(xx,yy,'r')
end
% fecha y hora
    str = datestr(now);
end

```

Figura 80. Código del experimento 3 – Función “CablesCargasPuntuales2”

Fuente: Autor

```

function [TA, TAx, TAy, TBAB, TBBC, TB, TBx, TBy, TCBC, TCCD, TC, TCx, TCy, TD, TDx,
TDy, TAB, TABmax, TABmin, hAB, LAB, SAB, TBC, TBCmax, TBCmin, hBC, LBC, SBC,
TCD, TCDmax, TCDmin, hCD, LCD, SCD, str, NT,
NWAB, WAB, DAB, h1AB, YB, NWBC, WBC, DBC, h1BC, YC, NWCD, WCD, DCD, h1CD, YD, x, y, xx, yy] =
CableCargasPuntuales3 (NT,
NWAB, WAB, DAB, h1AB, YB, NWBC, WBC, DBC, h1BC, YC, NWCD, WCD, DCD, h1CD, YD)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determinación de tensiones en cables que soportaban cuerpos suspendidos. %
%Se considera el caso de un número arbitrario N de cuerpos suspendidos de %
%un cable. Se supone que el peso del cable puede despreciarse comparado %
%con los pesos suspendidos, y que el cable es lo bastante flexible para %
%aproximar su forma con una serie de segmentos rectos. %
%
%      Caso 1: El Punto Inicial y Punto Final se encuentran a nivel %
%      Caso 1: El Punto Inicial y Punto Final se encuentran a desnivel YB %
%
%      Variables de entrada: %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DESARROLLO
%% IDENTIFICAR EL NÚMERO DE TRAMO
%% PARA EL TRAMO AB
NW=NWAB;
W=WAB;
D=DAB;
h1=h1AB;
[A, Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S, x, y, xx, yy] =
CableCargasDiscretas1 (NW, W, D, h1, YB);
TA=A;
TAx=Ax;
TAy=Ay;
TBAB=B;
TBx=Bx;
%e1= es igual a la variable TBx
e1=TBx;
TBy=By;
%e2= es igual a la variable TBy
e2=TBy;
TAB=T;
TABmax=Tmax;
TABmin=Tmin;
hAB=h;
LAB=L;
SAB=S;
% PARA GRAFICAS
xAB=x;
yAB=y;
xxAB=xx;
yyAB=yy;
%para reconocer el valor YB=YYBB
YYBB=YB;
YYCC=YC;
YYDD=YD;
%% PARA EL TRAMO BC
if YC==0 && YB==0 && YB==YC
    %caso 1
elseif YB==0 && YC<0 && YB>YC
    %caso 2
elseif YB==0 && YC<0 && YB<YC
    %caso 3
elseif YB<0 && YC==0 && YB<YC
    %caso 4
    YC=-YB;
elseif YB<0 && YC==YB && YC<0
    %caso 5
    YC=0;
elseif YB<0 && YB>YC && YC<0
    %caso 6

```

```

        YC=YC-YB;
elseif YB<0 && YC>0 && YB<YC
    %caso 7
    YC=-YB+YC;
elseif YB>0 && YC==0 && YB>YC
    %caso 8
    YC=-YB;
elseif YB>0 && YC>0 && YB==YC
    %caso 9
    YC=0;
elseif YB>0 && YC>0 && YB<YC
    %caso 10
    YC=YC-YB;
elseif YB>0 && YC<0 && YB>YC
    %caso 11
    YC=-YB+YC;
elseif YB<0 && YC<0 && YB<YC
    %caso 12
    YC=-(YB-YC);
elseif YB>0 && YC>0 && YB>YC
    %caso 13
    YC=(YB-YC);
end
NW=NWBC;
W=WBC;
D=DBC;
h1=h1BC;
YB=YC;
[A,Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S, x,y,xx,yy] =
CableCargasDiscretas1 (NW,W,D,h1,YB);
TBBC=A;
TBx=Ax;
%e3: es igual a la variable TBx
e3=TBx;
TBy=Ay;
%e4: es igual a la variable TBy
e4=TBy;
TCBC=B;
e5=Bx;
e6=By;
TBC=T;
TBCmax=Tmax;
TBCmin=Tmin;
hBC=h;
LBC=L;
SBC=S;
% PARA GRAFICAS
xBC=x;
yBC=y;
xxBC=xx;
yyBC=yy;
%%
YC=YYCC;
YB=YYBB;
%% REACCIÓN EN B
    %sumatoria en la dirección Y
    TBy=e2+e4;
    %sumatoria en la dirección X
    TBx=e1+e3;
    %Resultante
    TB=sqrt(TBy*TBy+TBx*TBx);
    %PARA REALIZAR LAS GRÁFICAS
if YC==0 && YB==0 && YB==YC
    %caso 1
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];

```

```

y1=[yAB yBC];
%subplot(221),plot(x1,y1,'r')
%cable deformado
xxBC=xxBC+LAB;
xx1=[xxAB xxBC];
yy1=[yyAB yyBC];
%subplot(221),plot(xx1,yy1,'r')
elseif YB==0 && YC<0 && YB>YC
%caso 2
%posición inicial
xBC=xB+LAB;
x1=[xAB xBC];
y1=[yAB yBC];
%subplot(221),plot(x1,y1,'r')
%cable deformado
xxBC=xxBC+LAB;
xx1=[xxAB xxBC];
yy1=[yyAB yyBC];
%subplot(221),plot(xx1,yy1,'r')
elseif YB==0 && YC>0 && YB<YC
%caso 3
%posición inicial
xBC=xB+LAB;
x1=[xAB xBC];
y1=[yAB yBC];
%subplot(221),plot(x,y,'r')
%cable deformado
xxBC=xxBC+LAB;
xx1=[xxAB xxBC];
yy1=[yyAB yyBC];
%subplot(221),plot(xx1,yy1,'r')
elseif YB<0 && YC==0 && YB<YC
%caso 4
%posición inicial
xBC=xB+LAB;
x1=[xAB xBC];
yBC=yBC-YB;
y1=[yAB yBC];
%subplot(221),plot(x1,y1,'r')
%cable deformado
xxBC=xxBC+LAB;
xx1=[xxAB xxBC];
yyBC=yyBC-YB;
yy1=[yyAB yyBC];
%subplot(221),plot(xx1,yy1,'r')
elseif YB<0 && YC==YB && YC<0
%caso 5
%posición inicial
xBC=xB+LAB;
x1=[xAB xBC];
yBC=yBC-YB;
y1=[yAB yBC];
%subplot(221),plot(x1,y1,'r')
%cable deformado
xxBC=xxBC+LAB;
xx1=[xxAB xxBC];
yyBC=yyBC-YB;
yy1=[yyAB yyBC];
%subplot(221),plot(xx1,yy1,'r')
elseif YB<0 && YB>YC && YC<0
%caso 6
%posición inicial
xBC=xB+LAB;
x1=[xAB xBC];
yBC=yBC-YB;
y1=[yAB yBC];
%subplot(221),plot(x1,y1,'r')

```

```

        %cable deformado
        xxBC=xxBC+LAB;
        xx1=[xxAB xxBC];
        yyBC=yyBC-YB;
        yy1=[yyAB yyBC];
        %subplot(221),plot(xx1,yy1,'r')
elseif YB<0 && YC>0 && YB<YC
    %caso 7
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
elseif YB>0 && YC==0 && YB>YC
    %caso 8
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
elseif YB>0 && YC>0 && YB==YC
    %caso 9
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
elseif YB>0 && YC>0 && YB<YC
    %caso 10
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
elseif YB>0 && YC<0 && YB>YC
    %caso 11
    %posición inicial
    xBC=xBC+LAB;

```

```

        x1=[xAB xBC];
        yBC=yBC-YB;
        y1=[yAB yBC];
        %subplot(221),plot(x1,y1,'r')
        %cable deformado
        xxBC=xxBC+LAB;
        xx1=[xxAB xxBC];
        yyBC=yyBC-YB;
        yy1=[yyAB yyBC];
        %subplot(221),plot(xx1,yy1,'r')
elseif YB<0 && YC<0 && YB<YC
    %caso 12
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
elseif YB>0 && YC>0 && YB>YC
    %caso 13
    %posición inicial
    xBC=xBC+LAB;
    x1=[xAB xBC];
    yBC=yBC-YB;
    y1=[yAB yBC];
    %subplot(221),plot(x1,y1,'r')
    %cable deformado
    xxBC=xxBC+LAB;
    xx1=[xxAB xxBC];
    yyBC=yyBC-YB;
    yy1=[yyAB yyBC];
    %subplot(221),plot(xx1,yy1,'r')
end
%% PARA EL TRAMO CD
if YB==0 && YC==0 && YD==0
    %CASO 1-1
elseif YB==0 && YC==0 && YD<0
    %CASO 2-1
elseif YB==0 && YC==0 && YD>0
    %CASO 3-1
elseif YB==0 && YC<0 && YD<0 && YD<YC
    %CASO 1-2
    YD=YD-YC;
elseif YB==0 && YC<0 && YD<0 && YD==YC
    %CASO 2-2
    YD=0;
elseif YB==0 && YC<0 && YD<0 && YD>YC
    %CASO 3-2
    YD=YD-YC;
elseif YB==0 && YC<0 && YD==0
    %CASO 4-2
    YD=-YC;
elseif YB==0 && YC<0 && YD>0
    %CASO 5-2
    YD=-YC+YD;
elseif YB==0 && YC>0 && YD<0
    %CASO 1-3
    YD=-YC+YD;
elseif YB==0 && YC>0 && YD==0
    %CASO 2-3
    YD=-YC;

```



```

elseif YB==0 && YC>0 && YD>0 && YD<YC
    %CASO 3-3
    YD=-YC+YD;
elseif YB==0 && YC>0 && YD>0 && YD==YC
    %CASO 4-3
    YD=0;
elseif YB==0 && YC>0 && YD>0 && YD>YC
    %CASO 5-3
    YD=-YC+YD;
elseif YB<0 && YC==0 && YD<0
    %CASO 1-4
elseif YB<0 && YC==0 && YD==0
    %CASO 2-4
elseif YB<0 && YC==0 && YD>0
    %CASO 3-4
elseif YB<0 && YB==YC && YD<YC
    %CASO 1-5
    YD=YD-YC;
elseif YB<0 && YB==YC && YD==YC
elseif YB<0 && YB==YC && YD==YC
    %CASO 2-5
    YD=0;
elseif YB<0 && YB==YC && YD<0 && YD>YC
    %CASO 3-5
    YD=YD-YC;
elseif YB<0 && YB==YC && YD==0
    %CASO 4-5
    YD=-YC;
elseif YB<0 && YB==YC && YD>0
    %CASO 5-5
    YD=YD-YC;
elseif YB<0 && YB>YC && YD<YC
    %CASO 1-6
    YD=YD-YC;
elseif YB<0 && YB>YC && YD==YC
    %CASO 2-6
    YD=0;
elseif YB<0 && YB>YC && YD>YC && YD<0
    %CASO 3-6
    YD=-YC+YD;
elseif YB<0 && YB>YC && YD==0
    %CASO 4-6
    YD=-YC;
elseif YB<0 && YB>YC && YD>0
    %CASO 5-6
    YD=YD-YC;
elseif YB<0 && YC>0 && YD<0
    %CASO 1-7
    YD=YD-YC;
elseif YB<0 && YC>0 && YD==0
    %CASO 2-7
    YD=-YC;
elseif YB<0 && YC>0 && YD==YC
    %CASO 3-7
    YD=0;
elseif YB<0 && YC>0 && YD>YC
    %CASO 4-7
    YD=YD-YC;
elseif YB>0 && YC==0 && YD<0
    %CASO 1-8
elseif YB>0 && YC==0 && YD==0
    %CASO 2-8
elseif YB>0 && YC==0 && YD>0
    %CASO 3-8
elseif YB>0 && YB==YC && YD<0
    %CASO 1-9
    YD=YC-YD;

```

```

elseif YB>0 && YB==YC && YD==0
    %CASO 2-9
    YD=-YC;
elseif YB>0 && YB==YC && YD<YC && YD>0
    %CASO 3-9
    YD=-YC+YD;
elseif YB>0 && YB==YC && YD==YC
    %CASO 4-9
    YD=0;
elseif YB>0 && YB==YC && YD>YC
    %CASO 5-9
    YD=YD-YC;
elseif YB>0 && YC>YB && YD<0
    %CASO 1-10
    %CASO 1-10
    YD=YD-YC;
elseif YB>0 && YC>YB && YD==0
    %CASO 2-10
    YD=-YC;
elseif YB>0 && YC>YB && YD>0 && YD<YC
    %CASO 3-10
    YD=YD-YC;
elseif YB>0 && YC>YB && YD==YC
    %CASO 4-10
    YD=0;
elseif YB>0 && YC>YB && YD>YC
    %CASO 5-10
    YD=YD-YC;
elseif YB>0 && YC<0 && YD<YC
    %CASO 1-11
    YD=YD-YC;
elseif YB>0 && YC<0 && YD<YC
    %CASO 2-11
    YD=0;
elseif YB>0 && YC<0 && YD<YC
    %CASO 3-11
    YD=-YC;
elseif YB>0 && YC<0 && YD<YC
    %CASO 4-11
    YD=YD-YC;
elseif YB<0 && YC<0 && YC>YB && YD<YC
    %CASO 1-12
    YD=YD-YC;
elseif YB<0 && YC<0 && YC>YB && YD==YC
    %CASO 2-12
    YD=0;
elseif YB<0 && YC<0 && YC>YB && YD==0
    %CASO 3-12
    YD=-YC;
elseif YB<0 && YC<0 && YC>YB && YD<0 && YD>YC
    %CASO 4-12
    YD=YD-YC;
elseif YB<0 && YC<0 && YC>YB && YD>0
    %CASO 5-12
    YD=YD-YC;
elseif YB>0 && YC>0 && YC<YB && YD<0
    %CASO 1-13
    YD=YD-YC;
elseif YB>0 && YC>0 && YC<YB && YD==0
    %CASO 2-13
    YD=-YC;
elseif YB>0 && YC>0 && YC<YB && YD>0 && YD<YC
    %CASO 3-13
    YD=YD-YC;
elseif YB>0 && YC>0 && YC<YB && YD==YC
    %CASO 4-13
    YD=0;

```

```

elseif YB>0 && YC>0 && YC<YB && YD>YC
    %CASO 5-13
    YD=YD-YC;
end
NW=NWCD;
W=WCD;
D=DCD;
h1=h1CD;
YB=YD;
[A,Ax, Ay, B, Bx, By, T, Tmax, Tmin, h, L, S, x,y,xx,yy] =
CableCargasDiscretas1 (NW,W,D,h1,YB);
TCCD=A;
e7=Ax;
e8=Ay;
TD=B;
TDx=Bx;
TDy=By;
TCD=T;
TCDmax=Tmax;
TCDmin=Tmin;
hCD=h;
LCD=L;
SCD=S;
% PARA GRAFICAS
xCD=x;
yCD=y;
xxCD=xx;
yyCD=yy;
%%
YC=YYCC;
YB=YYBB;
YD=YYDD;
%% REACCIÓN EN B
    %sumatoria en la dirección Y
    TCy=e6+e8;
    %sumatoria en la dirección X
    TCx=e5+e7;
    %Resultante
    TC=sqrt(TCy*TCy+TCx*TCx);
    %PARA REALIZAR LAS GRÁFICAS
if YB==0 && YC==0 && YD==0
    %CASO 1-1
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC==0 && YD<0
    %CASO 2-1
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC==0 && YD>0
    %CASO 3-1
    %posición inicial

```

```

        xCD=xCD+LAB+LBC;
        x=[x1 xCD];
        yyCD=yyCD-YC;
        yy=[yy1 yyCD];
        %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC<0 && YD>0
    %CASO 5-2
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    YY=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YD<0
    %CASO 1-3
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    YY=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YD==0
    %CASO 2-3
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    YY=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YD>0 && YD<YC
    %CASO 3-3
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YD>0 && YD==YC
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];

```

```

        %subplot(221),plot(x,y,'r')
        %cable deformado
        xxCD=xxCD+LAB+LBC;
        xx=[xx1 xxCD];
        yyCD=yyCD-YC;
        yy=[yy1 yyCD];
        %subplot(221),plot(xx,yy,'r')
elseif YB==0 && YC>0 && YD>0 && YD>YC
    %CASO 5-3
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC==0 && YD<0
    %CASO 1-4
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC==0 && YD==0
    %CASO 2-4
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC==0 && YD>0
    %CASO 3-4
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB==YC && YD<YC
    %CASO 1-5
    %posición inicial
    xCD=xCD+LAB+LBC;

```

```

x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB==YC && YD==YC
%CASO 2-5
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB==YC && YD<0 && YD>YC
%CASO 3-5
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB==YC && YD==0
%CASO 4-5
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YB>YC && YD>0
%CASO 5-6
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];

```

```

        yCD=yCD-YC;
        y=[y1 yCD];
        %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC>0 && YD<0
    %CASO 1-7
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC>0 && YD==0
    %CASO 2-7
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC>0 && YD==YC
    %CASO 3-7
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC>0 && YD>YC
    %CASO 4-7
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC==0 && YD<0
    %CASO 1-8
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];

```

```

yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC==0 && YD==0
%CASO 2-8
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC==0 && YD>0
%CASO 3-
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YB==YC && YD<0
%CASO 1-9
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YB==YC && YD==0
%CASO 2-9
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YB==YC && YD<YC && YD>0
%CASO 3-9

```



```

%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YB==YC && YD==YC
%CASO 4-9
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YB==YC && YD>YC
%CASO 5-9
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>YB && YD<0
%CASO 1-10
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>YB && YD==0
%CASO 2-10
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];

```

```

        %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>YB && YD>0 && YD<YC
    %CASO 3-10
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>YB && YD==YC
    %CASO 4-10
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>YB && YD>YC
    %CASO 5-10
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC<0 && YD<YC
    %CASO 1-11
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC<0 && YD>YC
    %CASO 2-11
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;

```

```

xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC<0 && YD<YC
%CASO 3-11
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC<0 && YD<YC
%CASO 4-11
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YC>YB && YD<YC
%CASO 1-12
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YC>YB && YD==YC
%CASO 2-12
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];
%subplot(221),plot(x,y,'r')
%cable deformado
xxCD=xxCD+LAB+LBC;
xx=[xx1 xxCD];
yyCD=yyCD-YC;
yy=[yy1 yyCD];
%subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YC>YB && YD==0
%CASO 3-12
%posición inicial
xCD=xCD+LAB+LBC;
x=[x1 xCD];
yCD=yCD-YC;
y=[y1 yCD];

```

```

        %subplot(221),plot(x,y,'r')
        %cable deformado
        xxCD=xxCD+LAB+LBC;
        xx=[xx1 xxCD];
        yyCD=yyCD-YC;
        yy=[yy1 yyCD];
        %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YC>YB && YD<0 && YD>YC
    %CASO 4-12
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB<0 && YC<0 && YC>YB && YD>0
    %CASO 5-12
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YC<YB && YD<0
    %CASO 1-13
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YC<YB && YD==0
    %CASO 2-13
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YC<YB && YD>0 && YD<YC
    %CASO 3-13
    %posición inicial
    xCD=xCD+LAB+LBC;

```

```

        x=[x1 xCD];
        yCD=yCD-YC;
        y=[y1 yCD];
        %subplot(221),plot(x,y,'r')
        %cable deformado
        xxCD=xxCD+LAB+LBC;
        xx=[xx1 xxCD];
        yyCD=yyCD-YC;
        yy=[yy1 yyCD];
        %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YC<YB && YD==YC
    %CASO 4-13
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
elseif YB>0 && YC>0 && YC<YB && YD>YC
    %CASO 5-13
    %posición inicial
    xCD=xCD+LAB+LBC;
    x=[x1 xCD];
    yCD=yCD-YC;
    y=[y1 yCD];
    %subplot(221),plot(x,y,'r')
    %cable deformado
    xxCD=xxCD+LAB+LBC;
    xx=[xx1 xxCD];
    yyCD=yyCD-YC;
    yy=[yy1 yyCD];
    %subplot(221),plot(xx,yy,'r')
end
% fecha y hora
    str = datestr(now);
end

```

Figura 81. Código del experimento 3 – Función “CablesCargasPuntuales3”

Fuente: Autor

```

function [Tmin] = SalidaCableCargasPuntuales1(A, Ax, Ay, B, Bx, By, T, Tmax,
Tmin, h, L, S, x,y,xx,yy,str,NW,W,D,h1,YB, ruta, nameFileUser,
nameFileExperimento,nameFileUserEnglish)
%%
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SÍSMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, ' DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, ' SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE CON CARGAS PUNTUALES\n');
fprintf(text_tiket, el punto inicial\n');
fprintf(text_tiket, ' y final, y sus componentes
correspondientes \n');
fprintf(text_tiket, [ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, 'CASO : T fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
|
|
|
| \n');
fprintf(text_tiket, [ ' Número de cargas | NW | #
| ' num2str(NW) '\n']);
s=NW;
for c = 1:s
if c==1
W1=W(1);
ramo 1: Determina las fuerzas reultantes en
fprintf(text_tiket, [ ' Carga 1 | W1 |
KN | ' num2str(W1) '\n']);
elseif c==2
W2=W(2);
fprintf(text_tiket, [ ' Carga 2 | W2 |
KN | ' num2str(W2) '\n']);
elseif c==3
W3=W(3);
fprintf(text_tiket, [ ' Carga 3 | W3 |

```

```

KN | ' num2str(W3) '\n'];
elseif c==4
    W4=W(4);
fprintf(text_tiket,[ '          Carga 4          | W4 |
KN | ' num2str(W4) '\n']);
elseif c==5
    W5=W(5);
fprintf(text_tiket,[ '          Carga 5          | W5 |
KN | ' num2str(W5) '\n']);
elseif c==6
    W6=W(6);
fprintf(text_tiket,[ '          Carga 6          | W6 |
KN | ' num2str(W6) '\n']);
elseif c==7
    W7=W(7);
fprintf(text_tiket,[ '          Carga 7          | W7 |
KN | ' num2str(W7) '\n']);
elseif c==8
    W8=W(8);
fprintf(text_tiket,[ '          Carga 8          | W8 |
KN | ' num2str(W8) '\n']);
elseif c==9
    W9=W(9);
fprintf(text_tiket,[ '          Carga 9          | W9 |
KN | ' num2str(W9) '\n']);
elseif c==10
    W10=W(10);
fprintf(text_tiket,[ '          Carga 10         | W10 |
KN | ' num2str(W10) '\n']);
end
end
s=NW+1;
for c = 1:s
    if c==1
        D1=D(1);
fprintf(text_tiket,[ '          Distancia 1        | D1 |
m | ' num2str(D1) '\n']);
elseif c==2
        D2=D(2);
fprintf(text_tiket,[ '          Distancia 2        | D2 |
m | ' num2str(D2) '\n']);
elseif c==3
        D3=D(3);
fprintf(text_tiket,[ '          Distancia 3        | D3 |
m | ' num2str(D3) '\n']);
elseif c==4
        D4=D(4);
fprintf(text_tiket,[ '          Distancia 4        | D4 |
m | ' num2str(D4) '\n']);
elseif c==5
        D5=D(5);
fprintf(text_tiket,[ m | ' num2str(D5) '\n']);
elseif c==6
        D6=D(6);
fprintf(text_tiket,[ '          Distancia 6        | D6 |
m | ' num2str(D6) '\n']);
elseif c==7
        D7=D(7);
fprintf(text_tiket,[ '          Distancia 7        | D7 |
'          Distancia 5          | D5 |
m | ' num2str(D7) '\n']);
elseif c==8
        D8=D(8);
fprintf(text_tiket,[ '          Distancia 8        | D8 |
m | ' num2str(D8) '\n']);
elseif c==9
        D9=D(9);

```

```

fprintf(text_tiket,[ '          Distancia 9          | D9 |
m | ' num2str(D9) '\n']);
    elseif c==10
        D10=D(10);
fprintf(text_tiket,[ '          Distancia 10         | D10 |
m | ' num2str(D10) '\n']);
    elseif c==11
        D11=D(11);
fprintf(text_tiket,[ '          Distancia 11         | D11 |
m | ' num2str(D11) '\n']);
    end
end
fprintf(text_tiket,[ '          Altura o flecha 1          | h1 | m
| ' num2str(h1) '\n']);
fprintf(text_tiket,[ '          Desnivel entre A y B          | YB | m
| ' num2str(YB) '\n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          DESCRIPTION          |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'          |          |          |          \n');
fprintf(text_tiket,[ ' Reacción en el punto incial A          | A | KN
| ' num2str(A) '\n']);
fprintf(text_tiket,[ ' Reacción en el eje x de A          | Ax | KN
| ' num2str(Ax) '\n']);
fprintf(text_tiket,[ ' Reacción en el eje y de A          | Ay | KN
| ' num2str(Ay) '\n']);
fprintf(text_tiket,[ ' Reacción en el punto incial B          | B | KN
| ' num2str(B) '\n']);
fprintf(text_tiket,[ ' Reacción en el eje x de B          | Bx | KN
| ' num2str(Bx) '\n']);
fprintf(text_tiket,[ ' Reacción en el eje y de B
| By | KN | ' num2str(By) '\n']);

s=NW+1;
v1=s;
for c = 1:s
    if c==1
        T1=T(1);
fprintf(text_tiket,[ ' Tensión Tramo A-1          | T1 |
KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=T(2);
fprintf(text_tiket,[ ' Tensión Tramo 1-B          | T2 |
KN | ' num2str(T2) '\n']);
        else
            T2=T(2);
fprintf(text_tiket,[ ' Tensión Tramo 1-2          | T2 |
KN | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=T(3);
fprintf(text_tiket,[ ' Tensión Tramo 2-B          | T3 |
KN | ' num2str(T3) '\n']);
        else
            T3=T(3);
fprintf(text_tiket,[ ' Tensión Tramo 2-3          | T3 |
KN | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1

```



```

        T4=T(4);
fprintf(text_tiket,[ ' Tensión Tramo 3-B          | T4      |
KN | ' num2str(T4) '\n']);
        else
        T4=T(4);
fprintf(text_tiket,[ ' Tensión Tramo 3-4          | T4      |
KN | ' num2str(T4) '\n']);
        end
elseif c==5
        if c==v1
        T5=T(5);
fprintf(text_tiket,[ ' Tensión Tramo 4-B          | T5      |
KN | ' num2str(T5) '\n']);
        else
        T5=T(5);
fprintf(text_tiket,[ ' Tensión Tramo 4-5          | T5      |
KN | ' num2str(T5) '\n']);
        end
elseif c==6
        if c==v1
        T6=T(6);
fprintf(text_tiket,[ ' Tensión Tramo 5-B          | T6      |
KN | ' num2str(T6) '\n']);
        else
        T6=T(6);
fprintf(text_tiket,[ ' Tensión Tramo 5-6          | T6      |
KN | ' num2str(T6) '\n']);
        end
elseif c==7
        if c==v1
        T7=T(7);
fprintf(text_tiket,[ ' Tensión Tramo 6-B          | T7      |
KN | ' num2str(T7) '\n']);
        else
        T7=T(7);
fprintf(text_tiket,[ ' Tensión Tramo 6-7          | T7      |
KN | ' num2str(T7) '\n']);
        end
elseif c==8
        if c==v1
        T8=T(8);
fprintf(text_tiket,[ ' Tensión Tramo 7-B          | T8      |
KN | ' num2str(T8) '\n']);
        else
        T8=T(8);
fprintf(text_tiket,[ ' Tensión Tramo 7-8          | T8      |
KN | ' num2str(T8) '\n']);
        end
elseif c==9
        if c==v1
        T9=T(9);
fprintf(text_tiket,[ ' Tensión Tramo 8-B          | T9      |
KN | ' num2str(T9) '\n']);
        else
        T9=T(9);
fprintf(text_tiket,[ ' Tensión Tramo 8-9          | T9      |
KN | ' num2str(T9) '\n']);
        end
elseif c==10
        if c==v1
        T10=T(10);
fprintf(text_tiket,[ ' Tensión Tramo 9-10         | T10     |
KN | ' num2str(T10) '\n']);
        end
elseif c==11
        T11=T(11);
fprintf(text_tiket,[ ' Tensión Tramo 10-B         | T11     |

```

```

KN | ' num2str(T11) '\n']);
end
end
fprintf(text_tiket,[ ' Tensión máxima | Tmax |
N | ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ ' Tensión mínima | Tmin |
N | ' num2str(Tmin) '\n']);
s=NW;
for c = 1:s
    if c==1
        h1=h(1);
        fprintf(text_tiket,[ ' Flecha o altura 1 | h1 |
m | ' num2str(h1) '\n']);
    elseif c==2
        h2=h(2);
        fprintf(text_tiket,[ ' Flecha o altura 2 | h2 |
m | ' num2str(h2) '\n']);
    elseif c==3
        h3=h(3);
        fprintf(text_tiket,[ ' Flecha o altura 3 | h3 |
m | ' num2str(h3) '\n']);
    elseif c==4
        h4=h(4);
        fprintf(text_tiket,[ ' Flecha o altura 4 | h4 |
m | ' num2str(h4) '\n']);
    elseif c==5
        h5=h(5);
        fprintf(text_tiket,[ ' Flecha o altura 5 | h5 |
m | ' num2str(h5) '\n']);
    elseif c==6
        h6=h(6);
        fprintf(text_tiket,[ ' Flecha o altura 6 | h6 |
m | ' num2str(h6) '\n']);
    elseif c==7
        h7=h(7);
        fprintf(text_tiket,[ ' Flecha o altura 7 | h7 |
m | ' num2str(h7) '\n']);
    elseif c==8
        h8=h(8);
        fprintf(text_tiket,[ ' Flecha o altura 8 | h8 |
m | ' num2str(h8) '\n']);
    elseif c==9
        h9=h(9);
        fprintf(text_tiket,[ ' Flecha o altura 9 | h9 |
m | ' num2str(h9) '\n']);
    elseif c==10
        h10=h(10);
        fprintf(text_tiket,[ ' Flecha o altura 10 | h10 |
m | ' num2str(h10) '\n']);
    fprintf(text_tiket,[ ' Flecha o altura 8 | h8 |
end
end
fprintf(text_tiket,[ ' Vano entre A y B | L | m
| ' num2str(L) '\n']);
fprintf(text_tiket,[ ' Longitud del Cable | S | m
| ' num2str(S) '\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '_____);
fprintf(text_tiket,
'_____ \n'
);
fclose(text_tiket);
_____ \n'

text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'_____ \n'

```

```

);
fprintf(text_tiket,
'
');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, '
DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, '
SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, '
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, '
-----\n');
n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORY: CABLES \n');
fprintf(text_tiket, 'EXPERIMENT: CABLE WITH POINT LOADS\n');
fprintf(text_tiket, 'CASE : Stretch 1: Determine the resulting forces at
the initial and \n');
fprintf(text_tiket, '
final point, and their corresponding
components.\n');
fprintf(text_tiket, [ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Authors : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '
\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'
| | | \n');
fprintf(text_tiket, [ ' Number of loads | NW | #
| ' num2str(NW) '\n']);
s=NW;
for c = 1:s
    if c==1
        W1=W(1);
        fprintf(text_tiket, [ ' Load 1 | W1 |
KN | ' num2str(W1) '\n']);
    elseif c==2
        W2=W(2);
        fprintf(text_tiket, [ ' Load 2 | W2 |
KN | ' num2str(W2) '\n']);
    elseif c==3
        W3=W(3);
        fprintf(text_tiket, [ ' Load 3 | W3 |
KN | ' num2str(W3) '\n']);
    elseif c==4
        W4=W(4);
        fprintf(text_tiket, [ ' Load 4 | W4 |
KN | ' num2str(W4) '\n']);
    elseif c==5

```

```

        W5=W(5);
        fprintf(text_tiket,[ '          Load 5          | W5 |
KN | ' num2str(W5) '\n']);
        elseif c==6
            W6=W(6);
            fprintf(text_tiket,[ '          Load 6          | W6 |
KN | ' num2str(W6) '\n']);
        elseif c==7
            W7=W(7);
            fprintf(text_tiket,[ '          Load 7          | W7 |
KN | ' num2str(W7) '\n']);
        elseif c==8
            W8=W(8);
            fprintf(text_tiket,[ '          Load 8          | W8 |
KN | ' num2str(W8) '\n']);
        elseif c==9
            W9=W(9);
            fprintf(text_tiket,[ '          Load 9          | W9 |
KN | ' num2str(W9) '\n']);
        elseif c==10
            W10=W(10);
            fprintf(text_tiket,[ '          Load 10         | W10 |
KN | ' num2str(W10) '\n']);
        end
    end
    s=NW+1;
    for c = 1:s
        if c==1
            D1=D(1);
            fprintf(text_tiket,[ '          Distance 1        | D1 |
m  | ' num2str(D1) '\n']);
        elseif c==2
            D2=D(2);
            fprintf(text_tiket,[ '          Distance 2        | D2 |
m  | ' num2str(D2) '\n']);
        elseif c==3
            D3=D(3);
            fprintf(text m | ' num2str(D3) '\n']);
        elseif c==4
            D4=D(4);
            fprintf(text_tiket,[ '          Distance 4        | D4 |
m  | ' num2str(D4) '\n']);
        elseif c==5
            D5=D(5);
            fprintf(text_tiket,[ '          Distance 5        | D5 |
m  | ' num2str(D5) '\n']);
        elseif c==6
            D6=D(6);
            fprintf(text_tiket,[ '          Distance 6        | D6 |
m  | ' num2str(D6) '\n']);
        elseif c==7
            D7=D(7);
            fprintf(text_tiket,[ '          Distance 7        | D7 |
m  | ' num2str(D7) '\n']);
        elseif c==8
            D8=D(8);
            fprintf(text_tiket,[ '          Distance 8        | D8 |
t_tiket,[ '          Distance 3        | D3 |
m  | ' num2str(D8) '\n']);
        elseif c==9
            D9=D(9);
            fprintf(text_tiket,[ '          Distance 9        | D9 |
m  | ' num2str(D9) '\n']);
        elseif c==10
            D10=D(10);
            fprintf(text_tiket,[ '          Distance 10       | D10 |
m  | ' num2str(D10) '\n']);

```

```

elseif c==11
    D11=D(11);
    fprintf(text_tiket,[ '          Distance 11          | D11 |
m | ' num2str(D11) '\n']);
end
end
fprintf(text_tiket,[ '          Height or arrow 1          | h1 | m
| ' num2str(h1) '\n']);
fprintf(text_tiket,[ '          Difference between A and B    | YB | m
| ' num2str(YB) '\n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          DESCRIPTION          |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'          |          |          |          \n');
fprintf(text_tiket,[ ' Reaction at start point A          | A | KN
| ' num2str(A) '\n']);
fprintf(text_tiket,[ ' Reaction on the x-axis of A          | Ax | KN
| ' num2str(Ax) '\n']);
fprintf(text_tiket,[ ' Reaction on the y-axis of A          | Ay | KN
| ' num2str(Ay) '\n']);
fprintf(text_tiket,[ ' Reaction at start point B          | B | KN
| ' num2str(B) '\n']);
fprintf(text_tiket,[ ' Reaction on the x-axis of B          | Bx | KN
| ' num2str(Bx) '\n']);
fprintf(text_tiket,[ ' Reaction on the y-axis of B          | By | KN
| ' num2str(By) '\n']);
s=NW+1;
v1=s;
for c = 1:s
    if c==1
        T1=T(1);
        fprintf(text_tiket,[ ' Stretch tension A-1          | T1 |
KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=T(2);
            fprintf(text_tiket,[ ' Stretch tension 1-B          | T2 |
KN | ' num2str(T2) '\n']);
        else
            T2=T(2);
            fprintf(text_tiket,[ ' Stretch tension 1-2          | T2 |
KN | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=T(3);
            fprintf(text_tiket,[ ' Stretch tension 2-B          | T3 |
KN | ' num2str(T3) '\n']);
        else
            T3=T(3);
            fprintf(text_tiket,[ ' Stretch tension 2-3          | T3 |
KN | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1
            T4=T(4);
            fprintf(text_tiket,[ ' Stretch tension 3-B          | T4 |
KN | ' num2str(T4) '\n']);
        else
            T4=T(4);
            fprintf(text_tiket,[ ' Stretch tension 3-4          | T4 |

```

```

KN    | ' num2str(T4) '\n'];
      end
      elseif c==5
          if c==v1
              T5=T(5);
              fprintf(text_tiket,[ ' Stretch tension 4-B           | T5   |
KN    | ' num2str(T5) '\n']);
              else
                  T5=T(5);
                  fprintf(text_tiket,[ ' Stretch tension 4-5       | T5   |
KN    | ' num2str(T5) '\n']);
              end
          elseif c==6
              if c==v1
                  T6=T(6);
                  fprintf(text_tiket,[ ' Stretch tension 5-B       | T6   |
KN    | ' num2str(T6) '\n']);
              else
                  T6=T(6);
                  fprintf(text_tiket,[ ' Stretch tension 5-6       | T6   |
KN    | ' num2str(T6) '\n']);
              end
          elseif c==7
              if c==v1
                  T7=T(7);
                  fprintf(text_tiket,[ ' Stretch tension 6-B       | T7   |
KN    | ' num2str(T7) '\n']);
              else
                  T7=T(7);
                  fprintf(text_tiket,[ ' Stretch tension 6-7       | T7   |
KN    | ' num2str(T7) '\n']);
              end
          elseif c==8
              if c==v1
                  T8=T(8);
                  fprintf(text_tiket,[ ' Stretch tension 7-B       | T8   |
KN    | ' num2str(T8) '\n']);
              else
                  T8=T(8);
                  fprintf(text_tiket,[ ' Stretch tension 7-8       | T8   |
KN    | ' num2str(T8) '\n']);
              end
          elseif c==9
              if c==v1
                  T9=T(9);
                  fprintf(text_tiket,[ ' Stretch tension 8-B       | T9   |
KN    | ' num2str(T9) '\n']);
              else
                  T9=T(9);
                  fprintf(text_tiket,[ ' Stretch tension 8-9       | T9   |
KN    | ' num2str(T9) '\n']);
              end
          elseif c==10
              if c==v1
                  T10=T(10);
                  fprintf(text_tiket,[ ' Stretch tension 9-B      | T10  |
KN    | ' num2str(T10) '\n']);
              else
                  T10=T(10);
                  fprintf(text_tiket,[ ' Stretch tension 9-10      | T10  |
KN    | ' num2str(T10) '\n']);
              end
          elseif c==11
              T11=T(11);
              fprintf(text_tiket,[
' Stretch tension 10-B           | T11  | KN    | ' num2str(T11)
'\n']);

```

```

end
end
fprintf(text_tiket,[ ' Maximun Tension                                | Tmax |
N | ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ ' Minimun Tension                                | Tmin |
N | ' num2str(Tmin) '\n']);
s=NW;
for c = 1:s
    if c==1
        h1=h(1);
        fprintf(text_tiket,[ ' Aroow or height 1                        | h1   |
m | ' num2str(h1) '\n']);
    elseif c==2
        h2=h(2);
        fprintf(text_tiket,[ ' Aroow or height 2                        | h2   |
m | ' num2str(h2) '\n']);
    elseif c==3
        h3=h(3);
        fprintf(text_tiket,[ ' Aroow or height 3                        | h3   |
m | ' num2str(h3) '\n']);
    elseif c==4
        h4=h(4);
        fprintf(text_tiket,[ ' Aroow or height 4                        | h4   |
m | ' num2str(h4) '\n']);
    elseif c==5
        h5=h(5);
        fprintf(text_tiket,[ ' Aroow or height 5                        | h5   |
m | ' num2str(h5) '\n']);
    elseif c==6
        h6=h(6);
        fprintf(text_tiket,[ ' Aroow or height 6                        | h6   |
m | ' num2str(h6) '\n']);
    elseif c==7
        h7=h(7);
        fprintf(text_tiket,[ ' Aroow or height 7                        | h7   |
m | ' num2str(h7) '\n']);
    elseif c==8
        h8=h(8);
        fprintf(text_tiket,[ ' Aroow or height 8                        | h8   |
m | ' num2str(h8) '\n']);
    elseif c==9
        h9=h(9);
        fprintf(text_tiket,[ ' Aroow or height 9                        | h9   |
m | ' num2str(h9) '\n']);
    elseif c==10
        h10=h(10);
        fprintf(text_tiket,[ ' Aroow or height 10                       | h10  |
m | ' num2str(h10) '\n']);
    end
end
fprintf(text_tiket,[ ' Vain between A y B                                | L    | m
| ' num2str(L) '\n']);
fprintf(text_tiket,[ ' Cable Length                                | S    | m
| ' num2str(S) '\n']);
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'_____\n'
);
fprintf(text_tiket,
'_____\n'
);
fclose(text_tiket);
%%% DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento], 'wt');
text_tiket=fopen([ruta nameFileExperimento], 'a');
%%fprintf(text_tiket,[ 'DATOS DE SALIDA \n']);

```

```

fprintf(text_tiket,[ 'A ' num2str(A) '\n']);
fprintf(text_tiket,[ 'Ax ' num2str(Ax) '\n']);
fprintf(text_tiket,[ 'Ay ' num2str(Ay) '\n']);
fprintf(text_tiket,[ 'B ' num2str(B) '\n']);
fprintf(text_tiket,[ 'Bx ' num2str(Bx) '\n']);
fprintf(text_tiket,[ 'By ' num2str(By) '\n']);
fprintf(text_tiket,[ 'Tmax ' num2str(Tmax) '\n']);
fprintf(text_tiket,[ 'Tmin ' num2str(Tmin) '\n']);
fprintf(text_tiket,[ 'L ' num2str(L) '\n']);
fprintf(text_tiket,[ 'S ' num2str(S) '\n']);
%fprintf(text_tiket,[ 'Posición Inicial del cable \n']);
fprintf(text_tiket,['x' ' ']);
dlmwrite([ruta nameFileExperimento],x,'-append','roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fprintf(text_tiket,['y' ' ']);
dlmwrite([ruta nameFileExperimento],y,'-append','roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
%fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
%fprintf(text_tiket,[ 'La deformada del Cable \n']);
fprintf(text_tiket,[ 'xx' ' ']);
dlmwrite([ruta nameFileExperimento],(xx(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
fprintf(text_tiket,[ 'yy' ' ']);
dlmwrite([ruta nameFileExperimento],(yy(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
%fprintf(text_tiket,[ 'Distribuida-DEformada del Cable\n']);
T=T';
%Tensiones
fprintf(text_tiket,['T' ' ']);
dlmwrite([ruta nameFileExperimento],T,'-append','roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
h=h';
%Tensiones
fprintf(text_tiket,['h' ' ']);
dlmwrite([ruta nameFileExperimento],h,'-append','roffset', 0, 'delimiter',
' ', 'precision', '%.4f')
fclose(text_tiket);
end
'Posición Inicial del Cable - Carga
end

```

Figura 82. Código del experimento 3 – Función “SalidaCablesCargasPuntuales1”

Fuente: Autor


```

function [x,y] = SalidaCableCargasPuntuales2(TA, TAx, TAy, TBAB,TBBC, TB, TBx,
TBy,TC,TCx,TCy, TAB, TABmax, TABmin, hAB, LAB, SAB,TBC, TBCmax, TBCmin, hBC,
LBC, SBC,str,NT, NWAB,WAB,DAB,h1AB,YB,NWBC,WBC,DBC,h1BC,YC,x,y,xx,yy, ruta,
nameFileUser, nameFileExperimento, nameFileUserEnglish)
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SÍSMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, ' DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, ' SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE CON CARGAS PUNTUALES\n');
fprintf(text_tiket, 'CASO : Tramo 2: Determina las fuerzas resultantes
en el punto A, B, C\n');
fprintf(text_tiket, ' y D, y sus componentes
correspondientes \n');
fprintf(text_tiket,[ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n'); fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'| | | | \n');
%identificar el número de tramo
fprintf(text_tiket,[ ' Número de Tramos | NT | #
| ' num2str(NT) '\n']);
fprintf(text_tiket, ' \n');
% para el tramo AB datos de entrada
fprintf(text_tiket, ' Tramo AB\n');
fprintf(text_tiket, ' -----\n');
fprintf(text_tiket,[ ' Número de cargas | NW | #
| ' num2str(NWAB) '\n']);
s=NWAB;
for c = 1:s
if c==1
W1=WAB(1);
fprintf(text_tiket,[ ' Carga 1 del Tramo AB | W1 |
KN | ' num2str(W1) '\n']);
elseif c==2

```

	W2=WAB(2);	Carga 2 del TRamo AB	W2
KN	fprintf(text_tiket,[' num2str(W2) '\n']);		
	elseif c==3		
	W3=WAB(3);	Carga 3 del TRamo AB	W3
KN	fprintf(text_tiket,[' num2str(W3) '\n']);		
	elseif c==4		
	W4=WAB(4);	Carga 4 del TRamo AB	W4
KN	fprintf(text_tiket,[' num2str(W4) '\n']);		
	elseif c==5		
	W5=WAB(5);	Carga 5 del TRamo AB	W5
KN	fprintf(text_tiket,[' num2str(W5) '\n']);		
	elseif c==6		
	W6=WAB(6);	Carga 6 del TRamo AB	W6
KN	fprintf(text_tiket,[' num2str(W6) '\n']);		
	elseif c==7		
	W7=WAB(7);	Carga 7 del TRamo AB	W7
KN	fprintf(text_tiket,[' num2str(W7) '\n']);		
	elseif c==8		
	W8=WAB(8);	Carga 8 del TRamo AB	W8
KN	fprintf(text_tiket,[' num2str(W8) '\n']);		
	elseif c==9		
	W9=WAB(9);	Carga 9 del TRamo AB	W9
KN	fprintf(text_tiket,[' num2str(W9) '\n']);		
	elseif c==10		
	W10=WAB(10);	Carga 10 del Tramo	
AB	fprintf(text_tiket,[' W10 KN ' num2str(W10) '\n']);		
	end		
	end		
	s=NWAB+1;		
	for c = 1:s		
	if c==1		
	D1=DAB(1);	Distancia 1 del Tramo AB	D1
m	fprintf(text_tiket,[' num2str(D1) '\n']);		
	elseif c==2		
	D2=DAB(2);	Distancia 2 del Tramo AB	D2
m	fprintf(text_tiket,[' num2str(D2) '\n']);		
	elseif c==3		
	D3=DAB(3);	Distancia 3 del Tramo AB	D3
m	fprintf(text_tiket,[' num2str(D3) '\n']);		
	elseif c==4		
	D4=DAB(4);	Distancia 4 del Tramo AB	D4
m	fprintf(text_tiket,[' num2str(D4) '\n']);		
	elseif c==5		
	D5=DAB(5);	Distancia 5 del Tramo AB	D5
m	fprintf(text_tiket,[' num2str(D5) '\n']);		
	elseif c==6		
	D6=DAB(6);	Distancia 6 del Tramo AB	D6
m	fprintf(text_tiket,[' num2str(D6) '\n']);		
	elseif c==7		
	D7=DAB(7);	Distancia 7 del Tramo AB	D7
	fprintf(text_tiket,[' num2str(D7) '\n']);		

```

m      | ' num2str(D7) '\n']);
        elseif c==8
            D8=DAB(8);
            fprintf(text_tiket,[ '          Distancia 8 del Tramo AB          | D8      |
m      | ' num2str(D8) '\n']);
        elseif c==9
            D9=DAB(9);
            fprintf(text_tiket,[ '          Distancia 9 del Tramo AB          | D9      |
m      | ' num2str(D9) '\n']);
        elseif c==10
            D10=DAB(10);
            fprintf(text_tiket,[ '          Distancia 10 del Tramo AB         | D10     |
m      | ' num2str(D10) '\n']);
        elseif c==11
            D11=DAB(11);
            fprintf(text_tiket,[ '          Distancia 11 del Tramo AB        | D11     |
m      | ' num2str(D11) '\n']);
        end
    end
    fprintf(text_tiket,[ '          Altura o flecha 1 del Tramo AB | h1      | m
| ' num2str(h1AB) '\n']);
    fprintf(text_tiket,[ '          Desnivel entre A y B          | YB      | m
| ' num2str(YB) '\n']);
    % para el tramo BC datos de entrada
    fprintf(text_tiket, '          \n');
    fprintf(text_tiket, '          Tramo BC\n');
    fprintf(text_tiket, '          ----- \n');
    fprintf(text_tiket,[ '          Número de cargas          | NW      | #
| ' num2str(NWBC) '\n']);
    s=NWBC;
    for c = 1:s
        if c==1
            W1=WBC(1);
            fprintf(text_tiket,[ '          Carga 1 del Tramo BC          | W1      |
KN      | ' num2str(W1) '\n']);
        elseif c==2
            W2=WBC(2);
            fprintf(text_tiket,[ '          Carga 2 del TRamo BC          | W2      |
KN      | ' num2str(W2) '\n']);
        elseif c==3
            W3=WBC(3);
            fprintf(text_tiket,[ '          Carga 3 del TRamo BC          | W3      |
KN      | ' num2str(W3) '\n']);
        elseif c==4
            W4=WBC(4);
            fprintf(text_tiket,[ '          Carga 4 del TRamo BC          | W4      |
KN      | ' num2str(W4) '\n']);
        elseif c==5
            W5=WBC(5);
            fprintf(text_tiket,[ '          Carga 5 del TRamo BC          | W5      |
KN      | ' num2str(W5) '\n']);
        elseif c==6
            W6=WBC(6);
            fprintf(text_tike KN      | ' num2str(W6) '\n']);
        elseif c==7
            W7=WBC(7);
            fprintf(text_tiket,[ '          Carga 7 del TRamo BC          | W7      |
KN      | ' num2str(W7) '\n']);
        elseif c==8
            W8=WBC(8);
            fprintf(text_tiket,[ '          Carga 8 del TRamo BC          | W8      |
KN      | ' num2str(W8) '\n']);
        elseif c==9
            W9=WBC(9);
            fprintf(text_tiket,[ '          Carga 9 del TRamo BC          | W9      |
KN      | ' num2str(W9) '\n']);

```



```

elseif c==5
    if c==v1
        T5=TAB(5);
        fprintf(text_tiket,[ ' Tensión Tramo 4-B
| T5 | KN | ' num2str(T5) '\n']);
    else
        T5=TAB(5);
        fprintf(text_tiket,[ ' Tensión Tramo 4-5
| T5 | KN | ' num2str(T5) '\n']);
    end

elseif c==6
    if c==v1
        T6=TAB(6);
        fprintf(text_tiket,[ ' Tensión Tramo 5-B
| T6 | KN | ' num2str(T6) '\n']);
    else
        T6=TAB(6);
        fprintf(text_tiket,[ ' Tensión Tramo 5-6
| T6 | KN | ' num2str(T6) '\n']);
    end

elseif c==7
    if c==v1
        T7=TAB(7);
        fprintf(text_tiket,[ ' Tensión Tramo 6-B
| T7 | KN | ' num2str(T7) '\n']);
    else
        T7=TAB(7);
        fprintf(text_tiket,[ ' Tensión Tramo 6-7
| T7 | KN | ' num2str(T7) '\n']);
    end

elseif c==8
    if c==v1
        T8=TAB(8);
        fprintf(text_tiket,[ ' Tensión Tramo 7-B
| T8 | KN | ' num2str(T8) '\n']);
    else
        T8=TAB(8);
        fprintf(text_tiket,[ ' Tensión Tramo 7-8
| T8 | KN | ' num2str(T8) '\n']);
    end

elseif c==9
    if c==v1
        T9=TAB(9);
        fprintf(text_tiket,[ ' Tensión Tramo 8-B
| T9 | KN | ' num2str(T9) '\n']);
    else
        T9=TAB(9);
        fprintf(text_tiket,[ ' Tensión Tramo 8-9
| T9 | KN | ' num2str(T9) '\n']);
    end

elseif c==10
    if c==v1
        T10=TAB(10);
        fprintf(| T10 | KN | ' num2str(T10) '\n');
    else
        T10=TAB(10);
        fprintf(text_tiket,[ ' Tensión Tramo 9-10
| T10 | KN | ' num2str(T10) '\n']);
    end

elseif c==11
    T11=TAB(11);
text_tiket,[ ' Tensión Tramo 9-B
    fprintf(text_tiket,[ ' Tensión Tramo 10-B
| T11 | KN | ' num2str(T11) '\n']);
end
end

```

```

fprintf(text_tiket,[ '          Tensión máxima del Tramo AB      | Tmax | KN
| ' num2str(TABmax) '\n']);
fprintf(text_tiket,[ '          Tensión mínima del Tramo AB      | Tmin | KN
| ' num2str(TABmin) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        h1=hAB(1);
        fprintf(text_tiket,[ '          Flecha o altura 1 del
Tramo AB | h1      | m      | ' num2str(h1) '\n']);
    elseif c==2
        h2=hAB(2);
        fprintf(text_tiket,[ '          Flecha o altura 2 del Tramo
AB | h2      | m      | ' num2str(h2) '\n']);
    elseif c==3
        h3=hAB(3);
        fprintf(text_tiket,[ '          Flecha o altura 3 del Tramo
AB | h3      | m      | ' num2str(h3) '\n']);
    elseif c==4
        h4=hAB(4);
        fprintf(text_tiket,[ '          Flecha o altura 4 del Tramo
AB | h4      | m      | ' num2str(h4) '\n']);
    elseif c==5
        h5=hAB(5);
        fprintf(text_tiket,[ '          Flecha o altura 5 del Tramo
AB | h5      | m      | ' num2str(h5) '\n']);
    elseif c==6
        h6=hAB(6);
        fprintf(text_tiket,[ '          Flecha o altura 6 del Tramo
AB | h6      | m      | ' num2str(h6) '\n']);
    elseif c==7
        h7=hAB(7);
        fprintf(text_tiket,[ '          Flecha o altura 7 del Tramo
AB | h7      | m      | ' num2str(h7) '\n']);
    elseif c==8
        h8=hAB(8);
        fprintf(text_tiket,[ '          Flecha o altura 8 del Tramo
AB | h8      | m      | ' num2str(h8) '\n']);
    elseif c==9
        h9=hAB(9);
        fprintf(text_tiket,[ '          Flecha o altura 9 del Tramo
AB | h9      | m      | ' num2str(h9) '\n']);
    elseif c==10
        h10=hAB(10);
        fprintf(text_tiket,[ '          Flecha o altura 10 del Tramo
AB| h10     | m      | ' num2str(h10) '\n']);
    end
end
fprintf(text_tiket,[ '          Vano entre A y B          | L      | m
| ' num2str(LAB) '\n']);
fprintf(text_tiket,[ '          Longitud del Cable          | S      | m
| ' num2str(SAB) '\n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          Tramo BC\n');
fprintf(text_tiket, '          -----\n');
s=NWBC+1;
v1=s;
for c = 1:s
    if c==1
        T1=TBC(1);
        fprintf(text_tiket,[ '          Tensión Tramo B-1
| T1      | KN      | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TBC(2);
            fprintf(text_tiket,[ '          Tensión Tramo 1-C
| T2      | KN      | ' num2str(T2) '\n']);

```

```

else
    fprintf(text_tiket,[ '
| T2 | KN | ' num2str(T2) '\n']);
end
elseif c==3
    if c==v1
        T3=TBC(3);
        fprintf(text_tiket,[ '
| T3 | KN | ' num2str(T3) '\n']);
    else
        T3=TBC(3);
        fprintf(text_tiket,[ '
| T3 | KN | ' num2str(T3) '\n']);
    end
elseif c==4
    if c==v1
        T4=TBC(4);
        fprintf(text_tiket,[ '
| T4 | KN | ' num2str(T4) '\n']);
    else
        T4=TBC(4);
        fprintf(text_tiket,[ '
| T4 | KN | ' num2str(T4) '\n']);
    end
elseif c==5
    if c==v1
        T5=TBC(5);
        fprintf(text_tiket,[ '
| T5 | KN | ' num2str(T5) '\n']);
    else
        T5=TBC(5);
        fprintf(text_tiket,[ '
| T5 | KN | ' num2str(T5) '\n']);
    end
elseif c==6
    if c==v1
        T6=TBC(6);
        fprintf(text_tiket,[ '
| T6 | KN | ' num2str(T6) '\n']);
    else
        T6=TBC(6);
        fprintf(text_tiket,[ '
| T6 | KN | ' num2str(T6) '\n']);
    end
elseif c==7
    if c==v1
        T7=TBC(7);
        fprintf(text_tiket,[ '
| T7 | KN | ' num2str(T7) '\n']);
    else
        T7=TBC(7);
        fprintf(text_tiket,[ '
| T7 | KN | ' num2str(T7) '\n']);
    end
elseif c==8
    if c==v1
        T8=TBC(8);
        fprintf(text_tiket,[ '
| T8 | KN | ' num2str(T8) '\n']);
    else
        T8=TBC(8);
        fprintf(text_tiket,[ '
| T8 | KN | ' num2str(T8) '\n']);
    end
elseif c==9
    if c==v1
        T9=TBC(9);

```

Tensión Tramo 1-2

Tensión Tramo 2-C

Tensión Tramo 2-3

Tensión Tramo 3-C

Tensión Tramo 3-4

Tensión Tramo 4-C

Tensión Tramo 4-5

Tensión Tramo 5-C

Tensión Tramo 5-6

Tensión Tramo 6-C

Tensión Tramo 6-7

Tensión Tramo 7-C

Tensión Tramo 7-8


```

        fprintf(text_tiket,[ ' Tensión Tramo 8-C
| T9 | KN | ' num2str(T9) '\n']);
        else
            T9=TBC(9);
            fprintf(text_tiket,[ ' Tensión Tramo 8-9
| T9 | KN | ' num2str(T9) '\n']);
        end
        elseif c==10
            if c==v1
                T10=TBC(10);
                fprintf(text_tiket,[ ' Tensión Tramo 9-C
| T10 | KN | ' num2str(T10) '\n']);
            else
                T10=TBC(10);
                fprintf(text_tiket,[ ' Tensión Tramo 9-10
| T10 | KN | ' num2str(T10) '\n']);
            end
        elseif c==11
            T11=TBC(11);
            fprintf(text_tiket,[ ' Tensión Tramo 10-C
| T11 | KN | ' num2str(T11) '\n']);
        end
        end
        fprintf(text_tiket,[ ' Tensión máxima del Tramo BC | Tmax
| KN | ' num2str(TBCmax) '\n']);
        fprintf(text_tiket,[ ' Tensión mínima del Tramo BC | Tmin
| KN | ' num2str(TBCmin) '\n']);
        s=NWBC;
        for c = 1:s
            if c==1
                h1=hBC(1);
                fprintf(text_tiket,[ ' Flecha o altura 1 del Tramo
BC | h1 | m | ' num2str(h1) '\n']);
            elseif c==2
                h2=hBC(2);
                fprintf(text_tiket,[ ' Flecha o altura 2 del Tramo
BC | h2 | m | ' num2str(h2) '\n']);
            elseif c==3
                h3=hBC(3);
                fprintf(text_tiket,[ ' Flecha o altura 3 del Tramo
BC | h3 | m | ' num2str(h3) '\n']);
            elseif c==4
                h4=hBC(4);
                fprintf(text_tiket,[ ' Flecha o altura 4 del Tramo
BC | h4 | m | ' num2str(h4) '\n']);
            elseif c==5
                h5=hBC(5);
                fprintf(text_tiket,[ ' Flecha o altura 5 del Tramo
BC | h5 | m | ' num2str(h5) '\n']);
            elseif c==6
                h6=hBC(6);
                fprintf(text_tiket,[ ' Flecha o altura 6 del Tramo
BC | h6 | m | ' num2str(h6) '\n']);
            elseif c==7
                h7=hBC(7);
                fprintf(text_tiket,[ ' Flecha o altura 7 del Tramo
BC | h7 | m | ' num2str(h7) '\n']);
            elseif c==8
                h8=hBC(8);
                fprintf(text_tiket,[ ' Flecha o altura 8 del Tramo
BC | h8 | m | ' num2str(h8) '\n']);
            elseif c==9
                h9=hBC(9);
                fprintf(text_tiket,[ ' Flecha o altura 9 del Tramo
BC | h9 | m | ' num2str(h9) '\n']);
            elseif c==9
                h9=hBC(9);

```

```

        fprintf(text_tiket,[ '                               Flecha o altura 9 del Tramo
BC | h9      | m      | ' num2str(h9) '\n']);
        elseif c==10
            h10=hBC(10);
            fprintf(text_tiket,[ '                               Flecha o altura 10 del Tramo
BC| h10     | m      | ' num2str(h10) '\n']);
        end
    end
    fprintf(text_tiket,[ '                               Vano entre B y C                | L      | m
| ' num2str(LBC) '\n']);
    fprintf(text_tiket,[ '                               Longitud del Cable            | S      | m
| ' num2str(SBC) '\n']);
    fprintf(text_tiket, '\n');
    fprintf(text_tiket,
    '_____ \n'
);
    fprintf(text_tiket,
    '_____ \n'
);
    fclose(text_tiket);
    text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
    fprintf(text_tiket,
    '_____ \n'
);
    fprintf(text_tiket,
    '_____ \n'
);
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '                               VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '                               SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
    fprintf(text_tiket, '                               DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
    fprintf(text_tiket, '                               SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
    fprintf(text_tiket, '                               UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
    fprintf(text_tiket, '                               ----- \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, 'LABORATORY: CABLES \n');
    fprintf(text_tiket, 'EXPERIMENT: CABLE WITH POINT LOADS\n');
    fprintf(text_tiket, 'CASE          : Stretch 2: Determine the resulting forces at
point A, B, C, y D \n');
    fprintf(text_tiket, '                               and their corresponding components.\n');
    fprintf(text_tiket,[ 'FECHA          : ' num2str(str) '\n']);
    fprintf(text_tiket, 'Authors       : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
    fprintf(text_tiket,
    '_____ \n'
);
    fprintf(text_tiket,
    ' \n');
    fprintf(text_tiket, 'INPUT DATA \n');
    fprintf(text_tiket, '_____ \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, ' \n');
    fprintf(text_tiket, '                               DESCRIPTION
|SYMBOL| UNITY | MAGNITUDE\n');
    fprintf(text_tiket,
    '_____ | _____ | _____ \n');
    %identificar el número de tramo
    fprintf(text_tiket,[ ' Number of sections                | NT      | #
| ' num2str(NT) '\n']);

```

```

fprintf(text_tiket, '          \n');
% para el tramo AB datos de entrada
fprintf(text_tiket, '          Stretch AB\n');
fprintf(text_tiket, '          -----\n');
fprintf(text_tiket, [ '          Number of loads          | NW | #
| ' num2str(NWAB) '\n']];
s=NWAB;
for c = 1:s
    if c==1
        W1=WAB(1);
        fprintf(text_tiket, [ '          Load 1 of the AB section          | W1 |
KN | ' num2str(W1) '\n']];
        elseif c==2
            W2=WAB(2);
            fprintf(text_tiket, [ '          Load 2 of the AB section          | W2 |
KN | ' num2str(W2) '\n']];
            elseif c==3
                W3=WAB(3);
                fprintf(text_tiket, [ '          Load 3 of the AB section          | W3 |
KN | ' num2str(W3) '\n']];
                elseif c==4
                    W4=WAB(4);
                    fprintf(text_tiket, [ '          Load 4 of the AB section          | W4 |
KN | ' num2str(W4) '\n']];
                    elseif c==5
                        W5=WAB(5);
                        fprintf(text_tiket, [ '          Load 5 of the AB section          | W5 |
KN | ' num2str(W5) '\n']];
                        elseif c==6
                            W6=WAB(6);
                            fprintf(text_tiket, [ '          Load 6 of the AB section          | W6 |
KN | ' num2str(W6) '\n']];
                            elseif c==7
                                W7=WAB(7);
                                fprintf(text_tiket, [ '          Load 7 of the AB section          | W7 |
KN | ' num2str(W7) '\n']];
                                elseif c==8
                                    W8=WAB(8);
                                    fprintf(text_tiket, [ '          Load 8 of the AB section          | W8 |
KN | ' num2str(W8) '\n']];
                                    elseif c==9
                                        W9=WAB(9);
                                        fprintf(text_tiket, [ '          Load 9 of the AB section          | W9 |
KN | ' num2str(W9) '\n']];
                                        elseif c==10
                                            W10=WAB(10);
                                            fprintf(text_tiket, [ '          Load 10 of the AB section          | W10 |
KN | ' num2str(W10) '\n']];
                                            end
                                        end
                                        s=NWAB+1;
                                        for c = 1:s
                                            if c==1
                                                D1=DAB(1);
                                                fprintf(text_tiket, [ '          Distance 1 of the AB section | D1 |
m | ' num2str(D1) '\n']];
                                                elseif c==2
                                                    D2=DAB(2);
                                                    fprintf(text_tiket, [ '          Distance 2 of the AB section | D2 |
m | ' num2str(D2) '\n']];
                                                    elseif c==3
                                                        D3=DAB(3);
                                                        fprintf(text_tiket, [ '          Distance 3 of the AB section | D3 |
m | ' num2str(D3) '\n']];
                                                        elseif c==4
                                                            D4=DAB(4);

```

```

fprintf(text_tiket,[ '
m | ' num2str(D4) '\n']);
    elseif c==5
        D5=DAB(5);
fprintf(text_tiket,[ '
m | ' num2str(D5) '\n']);
    elseif c==6
        D6=DAB(6);
fprintf(text_tiket,[ '
m | ' num2str(D6) '\n']);
    elseif c==7
        D7=DAB(7);
fprintf(text_tiket,[ '
m | ' num2str(D7) '\n']);
    elseif c==8
        D8=DAB(8);
fprintf(text_tiket,[ '
m | ' num2str(D8) '\n']);
    elseif c==9
        D9=DAB(9);
fprintf(text_tiket,[ '
m | ' num2str(D9) '\n']);
    elseif c==10
        D10=DAB(10);
fprintf(text_tiket,[ '
m | ' num2str(D10) '\n']);
    elseif c==11
        D11=DAB(11);
fprintf(text_tiket,[ '
m | ' num2str(D11) '\n']);
    end
end
fprintf(text_tiket,[ '
| ' num2str(h1AB) '\n']);
fprintf(text_tiket,[ '
| ' num2str(YB) '\n']);
% para el tramo BC datos de entrada
fprintf(text_tiket, '
fprintf(text_tiket, '
fprintf(text_tiket, '
fprintf(text_tiket,[ '
| ' num2str(NWBC) '\n']);
    s=NWBC;
    for c = 1:s
        if c==1
            W1=WBC(1);
fprintf(text_tiket,[ '
KN | ' num2str(W1) '\n']);
            elseif c==2
                W2=WBC(2);
fprintf(text_tiket,[ '
KN | ' num2str(W2) '\n']);
            elseif c==3
                W3=WBC(3);
fprintf(text_tiket,[ '
KN | ' num2str(W3) '\n']);
            elseif c==4
                W4=WBC(4);
fprintf(text_tiket,[ '
KN | ' num2str(W4) '\n']);
            elseif c==5
                W5=WBC(5);
fprintf(text_tiket,[ '
KN | ' num2str(W5) '\n']);
            elseif c==6
                W6=WBC(6);
fprintf(text_tiket,[ '

```

	Distance 4 of the AB section	D4	
	Distance 5 of the AB section	D5	
	Distance 6 of the AB section	D6	
	Distance 7 of the AB section	D7	
	Distance 8 of the AB section	D8	
	Distance 9 of the AB section	D9	
	Distance 10 of the AB section	D10	
	Distance 11 of the AB section	D11	
	Height 1 of the AB section	h1	m
	Difference between A and B	YB	m
	Stretch BC		
	Number of loads	NW	#
	Load 1 of the BC section	W1	
	Load 2 of the BC section	W2	
	Load 3 of the BC section	W3	
	Load 4 of the BC section	W4	
	Load 5 of the BC section	W5	
	Load 6 of the BC section	W6	

KN	' num2str(W6) '\n']];			
	elseif c==7			
	W7=WBC(7);			
	fprintf(text_tiket,['	Load 7 of the BC section	W7	
KN	' num2str(W7) '\n']];			
	elseif c==8			
	W8=WBC(8);			
	fprintf(text_tiket,['	Load 8 of the BC section	W8	
KN	' num2str(W8) '\n']];			
	elseif c==9			
	W9=WBC(9);			
	fprintf(text_tiket,['	Load 9 of the BC section	W9	
KN	' num2str(W9) '\n']];			
	elseif c==10			
	W10=WBC(10);			
	fprintf(text_tiket,['	Load 10 of the BC section	W10	
KN	' num2str(W10) '\n']];			
	end			
	end			
	s=NWBC+1;			
	for c = 1:s			
	if c==1			
	D1=DBC(1);			
	fprintf(text_tiket,['	Distance 1 of the BC section	D1	
m	' num2str(D1) '\n']];			
	elseif c==2			
	D2=DBC(2);			
	fprintf(text_tiket,['	Distance 2 of the BC section	D2	
m	' num2str(D2) '\n']];			
	elseif c==3			
	D3=DBC(3);			
	fprintf(text_tiket,['	Distance 3 of the BC section	D3	
m	' num2str(D3) '\n']];			
	elseif c==4			
	D4=DBC(4);			
	fprintf(text_tiket,['	Distance 4 of the BC section	D4	
m	' num2str(D4) '\n']];			
	elseif c==5			
	D5=DBC(5);			
	fprintf(text_tiket,['	Distance 5 of the BC section	D5	
m	' num2str(D5) '\n']];			
	elseif c==6			
	D6=DBC(6);			
	fprintf(text_tiket,['	Distancia 6 of the BC section	D6	
m	' num2str(D6) '\n']];			
	elseif c==7			
	D7=DBC(7);			
	fprintf(text_tiket,['	Distance 7 of the BC section	D7	
m	' num2str(D7) '\n']];			
	elseif c==8			
	D8=DBC(8);			
	fprintf(text_tiket,['	Distance 8 of the BC section	D8	
m	' num2str(D8) '\n']];			
	elseif c==9			
	D9=DBC(9);			
	fprintf(text_tiket,['	Distance 9 of the BC section	D9	
m	' num2str(D9) '\n']];			
	elseif c==10			
	D10=DBC(10);			
	fprintf(text_tiket,['	Distance 10 of the BC section	D10	
m	' num2str(D10) '\n']];			
	elseif c==11			
	D11=DBC(11);			
	fprintf(text_tiket,['	Distance 11 of the BC section	D11	
m	' num2str(D11) '\n']];			
	end			
	end			

```

fprintf(text_tiket, '
| ' num2str(h1BC) '\n']);
fprintf(text_tiket, [ '
| ' num2str(YC) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
| MAGNITUDE\n');
fprintf(text_tiket,
'
| \n');
fprintf(text_tiket, [ '
| ' num2str(TA) '\n']);
fprintf(text_tiket, [ '
| ' num2str(TAx) '\n']);
fprintf(text_tiket, [ '
| ' num2str(TAy) '\n']);
fprintf(text_tiket, [ '
| ' num2str(TBAB) '\n']);
fprintf(text_tiket, [ '
| ' num2str(TBBC) '\n']);
fprintf(text_tiket, [ '
| ' num2str(TB) '\n']);
fprintf(text_tiket, [ '
| ' num2str(D4) '\n']);
elseif c==5
D5=DBC(5);
fprintf(text_tiket, [ '
| ' num2str(D5) '\n']);
elseif c==6
D6=DBC(6);
fprintf(text_tiket, [ '
| ' num2str(D6) '\n']);
elseif c==7
D7=DBC(7);
fprintf(text_tiket, [ '
| ' num2str(D7) '\n']);
elseif c==8
D8=DBC(8);
fprintf(text_tiket, [ '
| ' num2str(D8) '\n']);
elseif c==9
D9=DBC(9);
fprintf(text_tiket, [ '
| ' num2str(D9) '\n']);
elseif c==10
D10=DBC(10);
fprintf(text_tiket, [ '
| ' num2str(D10) '\n']);
elseif c==11
D11=DBC(11);
fprintf(text_tiket, [ '
| ' num2str(D11) '\n']);
end
end
fprintf(text_tiket, [ '
| ' num2str(h1BC) '\n']);
fprintf(text_tiket, [ '
| ' num2str(YC) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, ' \n');

```

```

fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
                                DESCRIPTION                                |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'-----|-----|-----|-----\n');
fprintf(text_tiket, [ '
                                Reaction at point A                                | A      | KN
| ' num2str(TA) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the x-axis of A                        | Ax     | KN
| ' num2str(TAx) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the y-axis of A                        | Ay     | KN
| ' num2str(TAy) '\n']);
fprintf(text_tiket, [ '
                                Reaction at point B section AB                    | B-AB   | KN
| ' num2str(TBAB) '\n']);
fprintf(text_tiket, [ '
                                Reaction at point B section BC                    | B-BC   | KN
| ' num2str(TBBC) '\n']);
fprintf(text_tiket, [ '
                                Reaction at point B                                | B      | KN
| ' num2str(TB) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the x-axis of B                        | Bx     | KN
| ' num2str(TBx) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the y-axis of B                        | By     | KN
| ' num2str(TBy) '\n']);
fprintf(text_tiket, [ '
                                Reaction at point C                                | C      | KN
| ' num2str(TC) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the x-axis of C                        | Cx     | KN
| ' num2str(TCx) '\n']);
fprintf(text_tiket, [ '
                                Reaction on the y-axis of C                        | Cy     | KN
| ' num2str(TCy) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '
                                Stretch AB\n');
fprintf(text_tiket, '
                                -----\n');
    s=NWAB+1;
    v1=s;
    for c = 1:s
        if c==1
            T1=TAB(1);
            fprintf(text_tiket, [ '
                                Stretch tension A-1
| T1   | KN   | ' num2str(T1) '\n']);
            elseif c==2
                if c==v1
                    T2=TAB(2);
                    fprintf(text_tiket, [ '
                                Stretch tension 1-B
| T2   | KN   | ' num2str(T2) '\n']);
                else
                    T2=TAB(2);
                    fprintf(text_tiket, [ '
                                Stretch tension 1-2
| T2   | KN   | ' num2str(T2) '\n']);
                end
            elseif c==3
                if c==v1
                    T3=TAB(3);
                    fprintf(text_tiket, [ '
                                Stretch tension 2-B
| T3   | KN   | ' num2str(T3) '\n']);
                else
                    T3=TAB(3);
                    fprintf(text_tiket, [ '
                                Stretch tension 2-3
| T3   | KN   | ' num2str(T3) '\n']);
                end
            elseif c==4
                if c==v1
                    T4=TAB(4);
                    fprintf(text_tiket, [ '
                                Stretch tension 3-B
| T4   | KN   | ' num2str(T4) '\n']);
                else
                    T4=TAB(4);
                    fprintf(text_tiket, [ '
                                Stretch tension 3-4
| T4   | KN   | ' num2str(T4) '\n']);

```

```

        end
        elseif c==5
            if c==v1
                T5=TAB(5);
                fprintf(text_tiket,[ '           Stretch tension 4-B
| T5      | KN      | ' num2str(T5) '\n']);
            else
                T5=TAB(5);
                fprintf(text_tiket,[ '           Stretch tension 4-5
| T5      | KN      | ' num2str(T5) '\n']);
            end
        elseif c==6

end
fprintf(text_tiket,[ '           Maximun stretch tension AB      | Tmax | KN
| ' num2str(TABmax) '\n']);
fprintf(text_tiket,[ '           Minimum stretch tension AB      | Tmin | KN
| ' num2str(TABmin) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        h1=hAB(1);
        fprintf(text_tiket,[ '           Arrow or height 1 of section
AB| h1     | m      | ' num2str(h1) '\n']);
    elseif c==2
        h2=hAB(2);
        fprintf(text_tiket,[ '           Arrow or height 2 of section
AB| h2     | m      | ' num2str(h2) '\n']);
    elseif c==3
        h3=hAB(3);
        fprintf(text_tiket,[ '           Arrow or height 3 of section
AB| h3     | m      | ' num2str(h3) '\n']);
    elseif c==4
        h4=hAB(4);
        fprintf(text_tiket,[ '           Arrow or height 4 of section
AB| h4     | m      | ' num2str(h4) '\n']);
    elseif c==5
        h5=hAB(5);
        fprintf(text_tiket,[ '           Arrow or height 5 of section
AB| h5     | m      | ' num2str(h5) '\n']);
    elseif c==6
        h6=hAB(6);
        fprintf(text_tiket,[ '           Arrow or height 6 of section
AB| h6     | m      | ' num2str(h6) '\n']);
    elseif c==7
        h7=hAB(7);
        fprintf(text_tiket,[ '           Arrow or height 7 of section
AB| h7     | m      | ' num2str(h7) '\n']);
    elseif c==8
        h8=hAB(8);
        fprintf(text_tiket,[ '           Arrow or height 8 del Tramo
AB| h8     | m      | ' num2str(h8) '\n']);
    elseif c==9
        h9=hAB(9);
        fprintf(text_tiket,[ '           Arrow or height 9 del Tramo
AB| h9     | m      | ' num2str(h9) '\n']);
    elseif c==10
        h10=hAB(10);
        fprintf(text_tiket,[ '           Arrow or height 10 del Tramo
AB| h10    | m      | ' num2str(h10) '\n']);
    end
end
fprintf(text_tiket,[ '           Vain between A and B           | L      | m
| ' num2str(LAB) '\n']);
fprintf(text_tiket,[ '           Cable length           | S      | m
| ' num2str(SAB) '\n']);
fprintf(text_tiket, '           \n');

```



```

fprintf(text_tiket, '          Stretch BC\n');
fprintf(text_tiket, '          -----\n');
s=NWBC+1;
v1=s;
for c = 1:s
    if c==1
        T1=TBC(1);

        fprintf(text_tiket,[ '          Stretch tension B-1
| T1 | KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TBC(2);
            fprintf(text_tiket,[ '          Stretch tension 1-C
| T2 | KN | ' num2str(T2) '\n']);
        else
            T2=TBC(2);
            fprintf(text_tiket,[ '          Stretch tension 1-2
| T2 | KN | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=TBC(3);
            fprintf(text_tiket,[ '          Stretch tension 2-C
| T3 | KN | ' num2str(T3) '\n']);
        else
            T3=TBC(3);
            fprintf(text_tiket,[ '          Stretch tension 2-3
| T3 | KN | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1
            T4=TBC(4);
            fprintf(text_tiket,[ '          Stretch tension 3-C
| T4 | KN | ' num2str(T4) '\n']);
        else
            T4=TBC(4);
            fprintf(text_tiket,[ '          Stretch tension 3-4
| T4 | KN | ' num2str(T4) '\n']);
        end
    elseif c==5
        if c==v1
            T5=TBC(5);
            fprintf(text_tiket,[ '          Stretch tension 4-C
| T5 | KN | ' num2str(T5) '\n']);
        else
            T5=TBC(5);
            fprintf(text_tiket,[ '          Stretch tension 4-5
| T5 | KN | ' num2str(T5) '\n']);
        end
    elseif c==6
        if c==v1
            T6=TBC(6);
            fprintf(text_tiket,[ '          Stretch tension 5-C
| T6 | KN | ' num2str(T6) '\n']);
        else
            T6=TBC(6);
            fprintf(text_tiket,[ '          Stretch tension 5-6
| T6 | KN | ' num2str(T6) '\n']);
        end
    elseif c==7
        if c==v1
            T7=TBC(7);
            fprintf(text_tiket,[ '          Stretch tension 6-C
| T7 | KN | ' num2str(T7) '\n']);
        else

```

```

        T7=TBC(7);
        fprintf(text_tiket,[ '           Stretch tension 6-7
| T7   | KN   | ' num2str(T7) '\n']);
    end
    elseif c==8
        if c==v1
            T8=TBC(8);
            fprintf(text_tiket,[ '           Stretch tension 7-C
| T8   | KN   | ' num2str(T8) '\n']);
        else
            T8=TBC(8);
            fprintf(text_tiket,[ '           Stretch tension 7-8
| T8   | KN   | ' num2str(T8) '\n']);
        end
    elseif c==9
        if c==v1
            T9=TBC(9);
            fprintf(text_tiket,[ '           Stretch tension 8-C
| T9   | KN   | ' num2str(T9) '\n']);
        else
            T9=TBC(9);
            fprintf(text_tiket,[ '           Stretch tension 8-9
| T9   | KN   | ' num2str(T9) '\n']);
        end
    elseif c==10
        if c==v1
            T10=TBC(10);
            fprintf(text_tiket,[ '           Stretch tension 9-C
| T10  | KN   | ' num2str(T10) '\n']);
        else
            T10=TBC(10);
            fprintf(text_tiket,[ '           Stretch tension 9-10
| T10  | KN   | ' num2str(T10) '\n']);
        end
    elseif c==11
        T11=TBC(11);
        fprintf(text_tiket,[ '           Stretch tension 10-C
| T11  | KN   | ' num2str(T11) '\n']);
    end
end
fprintf(text_tiket,[ '           Maximun stretch tension BC   | Tmax
| KN   | ' num2str(TBCmax) '\n']);
fprintf(text_tiket,[ '           Minimun stretch tension BC   | Tmin
| KN   | ' num2str(TBCmin) '\n']);
s=NWBC;
for c = 1:s
    if c==1
        h1=hBC(1);
        fprintf(text_tiket,[ '           Arrow or height 1 of section
BC| h1   | m   | ' num2str(h1) '\n']);
    elseif c==2
        h2=hBC(2);
        fprintf(text_tiket,[ '           Arrow or height 2 of section
BC| h2   | m   | ' num2str(h2) '\n']);
    elseif c==3
        h3=hBC(3);
        fprintf(text_tiket,[ '           Arrow or height
3 of section BC| h3   | m   | ' num2str(h3) '\n']);
    elseif c==4
        h4=hBC(4);
        fprintf(text_tiket,[ '           Arrow or height 4 of section
BC| h4   | m   | ' num2str(h4) '\n']);
    elseif c==5
        h5=hBC(5);
        fprintf(text_tiket,[ '           Arrow or height 5 of section
BC| h5   | m   | ' num2str(h5) '\n']);
    elseif c==6

```

```

        h6=hBC(6);
        fprintf(text_tiket,[ '           Arrow or height 6 of section
BC| h6      | m           | ' num2str(h6) '\n']);
        elseif c==7
            h7=hBC(7);
            fprintf(text_tiket,[ '           Arrow or height 7 of section
BC| h7      | m           | ' num2str(h7) '\n']);
        elseif c==8
            h8=hBC(8);
            fprintf(text_tiket,[ '           Arrow or height 8 of section
BC| h8      | m           | ' num2str(h8) '\n']);
        elseif c==9
            h9=hBC(9);
            fprintf(text_tiket,[ '           Arrow or height 9 of section
BC| h9      | m           | ' num2str(h9) '\n']);
        elseif c==10
            h10=hBC(10);
            fprintf(text_tiket,[ '          Arrow or height 10 of section
BC| h10     | m           | ' num2str(h10) '\n']);
        end
    end
    fprintf(text_tiket,[ '           Vain between B y C           | L      | m
| ' num2str(LBC) '\n']);
    fprintf(text_tiket,[ '           Cable length           | S      | m
| ' num2str(SBC) '\n']);
    fprintf(text_tiket, '\n');
    fprintf(text_tiket,
    '_____ \n'
);
    fprintf(text_tiket,
    '_____ \n'
);
    fclose(text_tiket);
%% otras
%%% DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento],'wt');
text_tiket=fopen([ruta nameFileExperimento],'a');
%%fprintf(text_tiket,[ 'DATOS DE SALIDA \n']);
fprintf(text_tiket,[ 'A ' num2str(TA) '\n']);
fprintf(text_tiket,[ 'Ax ' num2str(TAx) '\n']);
fprintf(text_tiket,[ 'Ay ' num2str(TAy) '\n']);
fprintf(text_tiket,[ 'B ' num2str(TB) '\n']);
fprintf(text_tiket,[ 'Bx ' num2str(TBx) '\n']);
fprintf(text_tiket,[ 'By ' num2str(TBy) '\n']);
fprintf(text_tiket,[ 'TmaxAB ' num2str(TABmax) '\n']);
fprintf(text_tiket,[ 'TminAB ' num2str(TABmin) '\n']);
fprintf(text_tiket,[ 'LAB ' num2str(LAB) '\n']);
fprintf(text_tiket,[ 'SAB ' num2str(SAB) '\n']);
%fprintf(text_tiket,[ 'Posición Inicial del cable \n']);
fprintf(text_tiket,['x' ' ']);
dlmwrite([ruta nameFileExperimento],x,'-append','roffset', 0, 'delimiter',
' ', 'precision','%.4f')
fprintf(text_tiket,['y' ' ']);
dlmwrite([ruta nameFileExperimento],y,'-append','roffset', 0, 'delimiter',
' ', 'precision','%.4f')
%fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
%fprintf(text_tiket,[ 'La deformada del Cable \n']);
fprintf(text_tiket,['xx' ' ']);
dlmwrite([ruta nameFileExperimento],(xx(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
fprintf(text_tiket,['yy' ' ']);
dlmwrite([ruta nameFileExperimento],(yy(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
%fprintf(text_tiket,[ 'Posición Inicial del Cable - Carga Distribuida-
DEformada del Cable\n']);
TAB=TAB';
fprintf(text_tiket,['TAB' ' ']);

```

```

        dlmwrite([ruta nameFileExperimento],TAB,'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
        %flechas
        hAB=hAB';
        fprintf(text_tiket,['hAB' ' ']);
        dlmwrite([ruta nameFileExperimento],hAB,'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
        fprintf(text_tiket,['C ' num2str(TC) '\n']);
        fprintf(text_tiket,['Cx ' num2str(TCx) '\n']);
        fprintf(text_tiket,['Cy ' num2str(TCy) '\n']);
        fprintf(text_tiket,['TmaxBC ' num2str(TBCmax) '\n']);
        fprintf(text_tiket,['TminBC ' num2str(TBCmin) '\n']);
        fprintf(text_tiket,['LBC ' num2str(LBC) '\n']);
        fprintf(text_tiket,['SBC ' num2str(SBC) '\n']);
        fclose(text_tiket);
end

```

Figura 83. Código del experimento 3 – Función “SalidaCablesCargasPuntuales2”

Fuente: Autor

```

function [x,y] = SalidaCableCargasPuntuales3(TA, TAx, TAy, TBAB,TBBC, TB, TBx,
TBy,TCBC,TCCD,TC,TCx,TCy, TD, TDx, TDy, TAB, TABmax, TABmin, hAB, LAB, SAB,TBC,
TBCmax, TBCmin, hBC, LBC, SBC, TCD, TCDmax, TCDmin, hCD, LCD, SCD,str,NT,
NWAB,WAB,DAB,hLAB,YB,NWBC,WBC,DBC,hLBC,YC,NWCD,WCD,DCD,hLCD,YD,x,y,xx,yy, ruta,
nameFileUser, nameFileExperimento,nameFileUserEnglish)
%%
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SISMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, ' DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, ' SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, ' UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, ' -----
-----\n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: CABLE CON CARGAS PUNTUALES\n');
fprintf(text_tiket, 'CASO : Tramo 3: Determina las fuerzas resultantes
en el punto A, B, C,\n');
fprintf(text_tiket, ' D y E, y sus componentes
correspondientes \n');
fprintf(text_tiket,[ 'FECHA : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'| | | | \n');
%identificar el número de tramo
fprintf(text_tiket,[ ' Número de Tramos | NT | #
| ' num2str(NT) '\n']);
fprintf(text_tiket, ' \n');
% para el tramo AB datos de entrada
fprintf(text_tiket, ' Tramo AB\n');
fprintf(text_tiket, ' -----\n');
fprintf(text_tiket,[ ' Número de cargas | NW | #
| ' num2str(NWAB) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        W1=WAB(1);

```

KN	fprintf(text_tiket,[' ' num2str(W1) '\n']); elseif c==2 W2=WAB(2);	Carga 1 del Tramo AB	W1	
KN	fprintf(text_tiket,[' ' num2str(W2) '\n']); elseif c==3 W3=WAB(3);	Carga 2 del TRamo AB	W2	
KN	fprintf(text_tiket,[' ' num2str(W3) '\n']); elseif c==4 W4=WAB(4);	Carga 3 del TRamo AB	W3	
KN	fprintf(text_tiket,[' ' num2str(W4) '\n']); elseif c==5 W5=WAB(5);	Carga 4 del TRamo AB	W4	
KN	fprintf(text_tiket,[' ' num2str(W5) '\n']); elseif c==6 W6=WAB(6);	Carga 5 del TRamo AB	W5	
KN	fprintf(text_tiket,[' ' num2str(W6) '\n']); elseif c==7 W7=WAB(7);	Carga 6 del TRamo AB	W6	
KN	fprintf(text_tiket,[' ' num2str(W7) '\n']); elseif c==8 W8=WAB(8);	Carga 7 del TRamo AB	W7	
KN	fprintf(text_tiket,[' ' num2str(W8) '\n']); elseif c==9 W9=WAB(9);	Carga 8 del TRamo AB	W8	
KN	fprintf(text_tiket,[' ' num2str(W9) '\n']); elseif c==10 W10=WAB(10);	Carga 9 del TRamo AB	W9	
KN	fprintf(text_tiket,[' ' num2str(W10) '\n']); end end s=NWAB+1; for c = 1:s if c==1 D1=DAB(1);	Carga 10 del TRamo AB	W10	
m	fprintf(text_tiket,[' ' num2str(D1) '\n']); elseif c==2 D2=DAB(2);	Distancia 1 del Tramo AB	D1	
m	fprintf(text_tiket,[' ' num2str(D2) '\n']); elseif c==3 D3=DAB(3);	Distancia 2 del Tramo AB	D2	
m	fprintf(text_tiket,[' ' num2str(D3) '\n']); elseif c==4 D4=DAB(4);	Distancia 3 del Tramo AB	D3	
m	fprintf(text_tiket,[' ' num2str(D4) '\n']); elseif c==5 D5=DAB(5);	Distancia 4 del Tramo AB	D4	
m	fprintf(text_tiket,[' ' num2str(D5) '\n']); elseif c==6 D6=DAB(6);	Distancia 5 del Tramo AB	D5	
	fprintf(text_tiket,[' ' num2str(D6) '\n']);	Distancia 6 del Tramo		

```

AB      | D6      | m      | ' num2str(D6) '\n']];
        elseif c==7
            D7=DAB(7);
fprintf(text_tiket,[ '          Distancia 7 del Tramo AB      | D7      |
m      | ' num2str(D7) '\n']];
        elseif c==8
            D8=DAB(8);
fprintf(text_tiket,[ '          Distancia 8 del Tramo AB      | D8      |
m      | ' num2str(D8) '\n']];
        elseif c==9
            D9=DAB(9);
fprintf(text_tiket,[ '          Distancia 9 del Tramo AB      | D9      |
m      | ' num2str(D9) '\n']];
        elseif c==10
            D10=DAB(10);
fprintf(text_tiket,[ '          Distancia 10 del Tramo AB     | D10     |
m      | ' num2str(D10) '\n']];
        elseif c==11
            D11=DAB(11);
fprintf(text_tiket,[ '          Distancia 11 del Tramo AB     | D11     |
m      | ' num2str(D11) '\n']];
        end
    end
    fprintf(text_tiket,[ '          Altura o flecha 1 del Tramo AB | h1      | m
| ' num2str(h1AB) '\n']];
    fprintf(text_tiket,[ '          Desnivel entre A y B          | B        | m
| ' num2str(YB) '\n']];
    % para el tramo BC datos de entrada
    fprintf(text_tiket, '          \n');
    fprintf(text_tiket, '          Tramo BC\n');
    fprintf(text_tiket, '          ----- \n');
    fprintf(text_tiket,[ '          Número de cargas          | NW      | #
| ' num2str(NWBC) '\n']];
    s=NWBC;
    for c = 1:s
        if c==1
            W1=WBC(1);
fprintf(text_tiket,[ '          Carga 1 del Tramo BC          | W1      |
KN      | ' num2str(W1) '\n']];
        elseif c==2
            W2=WBC(2);
fprintf(text_tiket,[ '          Carga 2 del TRamo BC          | W2      |
KN      | ' num2str(W2) '\n']];
        elseif c==3
            W3=WBC(3);
fprintf(text_tiket,[ '          Carga 3 del TRamo BC          | W3      |
KN      | ' num2str(W3) '\n']];
        elseif c==4
            W4=WBC(4);
fprintf(text_tiket,[ '          Carga 4 del TRamo BC          | W4      |
KN      | ' num2str(W4) '\n']];
        elseif c==5
            W5=WBC(5);
fprintf(text_tiket,[ '          Carga 5 del TRamo BC          | W5      |
KN      | ' num2str(W5) '\n']];
        elseif c==6
            W6=WBC(6);
fprintf(text_tiket,[ '          Carga 6 del TRamo BC          | W6      |
KN      | ' num2str(W6) '\n']];
        elseif c==7
            W7=WBC(7);
fprintf(text_tiket,[ '          Carga 7 del TRamo BC          | W7      |
KN      | ' num2str(W7) '\n']];
        elseif c==8
            W8=WBC(8);
fprintf(text_tiket,[ '          Carga 8 del TRamo BC          | W8      |
KN      | ' num2str(W8) '\n']];

```

```

fprintf(text_tiket,[ '          Carga 9  del TRamo BC          | W9  | KN
| ' num2str(W9) '\n']);
    elseif c==10
        W10=WBC(10);
        fprintf(text_tiket,[ '          Carga 10 del TRamo BC          | W10 |
KN | ' num2str(W10) '\n']);
    end
    s=NWBC+1;
    for c = 1:s
        if c==1
            D1=DBC(1);
            fprintf(text_tiket,[ '          Distancia 1  del Tramo BC          | D1  |
m | ' num2str(D1) '\n']);
            elseif c==2
                D2=DBC(2);
                fprintf(text_tiket,[ '          Distancia 2  del Tramo BC          | D2  |
m | ' num2str(D2) '\n']);
            elseif c==3
                D3=DBC(3);
                fprintf(text_tiket,[ '          Distancia 3  del Tramo BC          | D3  |
m | ' num2str(D3) '\n']);
            elseif c==4
                D4=DBC(4);
                fprintf(text_tiket,[ '          Distancia 4  del Tramo BC          | D4  |
m | ' num2str(D4) '\n']);
            elseif c==5
                D5=DBC(5);
                fprintf(text_tiket,[ '          Distancia 5  del Tramo BC          | D5  |
m | ' num2str(D5) '\n']);
            elseif c==6
                D6=DBC(6);
                fprintf(text_tiket,[ '          Distancia 6  del Tramo BC          | D6  |
m | ' num2str(D6) '\n']);
            elseif c==7
                D7=DBC(7);
                fprintf(text_tiket,[ '          Distancia 7  del Tramo BC          | D7  |
m | ' num2str(D7) '\n']);
            elseif c==8
                D8=DBC(8);
                fprintf(text_tiket,[ '          Distancia 8  del Tramo BC          | D8  |
m | ' num2str(D8) '\n']);
            elseif c==9
                D9=DBC(9);
                fprintf(text_tiket,[ '          Distancia 9  del Tramo BC          | D9  |
m | ' num2str(D9) '\n']);
            elseif c==10
                D10=DBC(10);
                fprintf(text_tiket,[ '          Distancia 10 del Tramo BC          | D10 |
m | ' num2str(D10) '\n']);
            elseif c==11
                D11=DBC(11);
                fprintf(text_tiket,[ '          Distancia 11 del Tramo BC          | D11 |
m | ' num2str(D11) '\n']);
        end
    end
    fprintf(text_tiket,[ '          Altura o flecha 1 del Tramo BC | h1  | m
| ' num2str(h1BC) '\n']);
    fprintf(text_tiket,[ '          Desnivel entre A y C          | C  | m
| ' num2str(YC) '\n']);
    fprintf(text_tiket, '
\n');

    % para el tramo CD datos de entrada
    fprintf(text_tiket, '          \n');
    fprintf(text_tiket, '          Tramo CD\n');
    fprintf(text_tiket, '          ----- \n');

```


	fprintf(text_tiket,['Número de cargas NW #
' num2str(NWCD) '\n']];	
s=NWCD;	
for c = 1:s	
if c==1	
W1=WCD(1);	
fprintf(text_tiket,['Carga 1 del Tramo CD W1	
KN ' num2str(W1) '\n']];	
elseif c==2	
W2=WCD(2);	
fprintf(text_tiket,['Carga 2 del TRamo CD W2	
KN ' num2str(W2) '\n']];	
elseif c==3	
W3=WCD(3);	
fprintf(text_tiket,['Carga 3 del TRamo CD W3	
KN ' num2str(W3) '\n']];	
elseif c==4	
W4=WCD(4);	
fprintf(text_tiket,['Carga 4 del TRamo CD W4	
KN ' num2str(W4) '\n']];	
elseif c==5	
W5=WCD(5);	
fprintf(text_tiket,['Carga 5 del TRamo CD W5	
KN ' num2str(W5) '\n']];	
elseif c==6	
W6=WCD(6);	
fprintf(text_tiket,['Carga 6 del TRamo CD W6	
KN ' num2str(W6) '\n']];	
elseif c==7	
W7=WCD(7);	
fprintf(text_tiket,['Carga 7 del TRamo CD W7	
KN ' num2str(W7) '\n']];	
elseif c==8	
W8=WCD(8);	
fprintf(text_tiket,['Carga 8 del TRamo CD W8	
KN ' num2str(W8) '\n']];	
elseif c==9	
W9=WCD(9);	
fprintf(text_tiket,['Carga 9 del TRamo CD W9	
KN ' num2str(W9) '\n']];	
elseif c==10	
W10=WCD(10);	
fprintf(text_tiket,['Carga 10 del TRamo CD W10	
KN ' num2str(W10) '\n']];	
end	
end	
s=NWCD+1;	
for c = 1:s	
if c==1	
D1=DCD(1);	
fprintf(text_tiket,['Distancia 1 del Tramo CD D1	
m ' num2str(D1) '\n']];	
elseif c==2	
D2=DCD(2);	
fprintf(text_tiket,['Distancia 2 del Tramo CD D2	
m ' num2str(D2) '\n']];	
elseif c==3	
D3=DCD(3);	
fprintf(text_tiket,['Distancia 3 del Tramo CD D3	
m ' num2str(D3) '\n']];	
elseif c==4	
D4=DCD(4);	
fprintf(text_tiket,['Distancia 4 del Tramo CD D4	
m ' num2str(D4) '\n']];	
elseif c==5	
D5=DCD(5);	
fprintf(text_tiket,['Distancia 5 del Tramo CD D5	


```

fprintf(text_tiket,[ '          Reacción en el punto D          | D          |
KN | ' num2str(TD) '\n']);
fprintf(text_tiket,[ '          Reacción en el eje x de D          | Dx          |
KN | ' num2str(TDx) '\n']);
fprintf(text_tiket,[ '          Reacción en el eje y de D          | Dy          |
KN | ' num2str(TDy) '\n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          Tramo AB\n');
fprintf(text_tiket, '          -----\n');
s=NWAB+1;
v1=s;
for c = 1:s
    if c==1
        T1=TAB(1);
        fprintf(text_tiket,[ '          Tensión Tramo A-1
| T1 | KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TAB(2);
            fprintf(text_tiket,[ '          Tensión Tramo 1-B
| T2 | KN | ' num2str(T2) '\n']);
        else
            T2=TAB(2);
            fprintf(text_tiket,[ '          Tensión Tramo 1-2
| T2 | KN | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=TAB(3);
            fprintf(text_tiket,[ '          Tensión Tramo 2-B
| T3 | KN | ' num2str(T3) '\n']);
        else
            T3=TAB(3);
            fprintf(text_tiket,[ '          Tensión Tramo 2-3
| T3 | KN | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1
            T4=TAB(4);
            fprintf(text_tiket,[ '          Tensión Tramo 3-B
| T4 | KN | ' num2str(T4) '\n']);
        else
            T4=TAB(4);
            fprintf(text_tiket,[ '          Tensión Tramo 3-4
| T4 | KN | ' num2str(T4) '\n']);
        end
    elseif c==5
        if c==v1
            T5=TAB(5);
            fprintf(text_tiket,[ '          Tensión Tramo 4-B
| T5 | KN | ' num2str(T5) '\n']);
        else
            T5=TAB(5);
            fprintf(text_tiket,[ '          Tensión Tramo 4-5
| T5 | KN | ' num2str(T5) '\n']);
        end
    elseif c==6
        if c==v1
            T6=TAB(6);
            fprintf(text_tiket,[ '          Tensión Tramo 5-B
| T6 | KN | ' num2str(T6) '\n']);
        else
            T6=TAB(6);
            fprintf(text_tiket,[ '          Tensión Tramo 5-6
| T6 | KN | ' num2str(T6) '\n']);
        end
    end
end

```

```

elseif c==7
    if c==v1
        T7=TAB(7);
        fprintf(text_tiket,[ ' Tensión Tramo 6-B
| T7 | KN | ' num2str(T7) '\n']);
    else
        T7=TAB(7);
        fprintf(text_tiket,[ ' Tensión Tramo 6-7
| T7 | KN | ' num2str(T7) '\n']);
    end

elseif c==8
    if c==v1
        T8=TAB(8);
        fprintf(text_tiket,[ ' Tensión Tramo 7-B
| T8 | KN | ' num2str(T8) '\n']);
    else
        T8=TAB(8);
        fprintf(text_tiket,[ ' Tensión Tramo 7-8
| T8 | KN | ' num2str(T8) '\n']);
    end

elseif c==9
    if c==v1
        T9=TAB(9);
        fprintf(text_tiket,[ ' Tensión Tramo 8-B
| T9 | KN | ' num2str(T9) '\n']);
    else
        T9=TAB(9);
        fprintf(text_tiket,[ ' Tensión Tramo 8-9
| T9 | KN | ' num2str(T9) '\n']);
    end

elseif c==10
    if c==v1
        T10=TAB(10);
        fprintf(text_tiket,[ ' Tensión Tramo 9-B
| T10 | KN | ' num2str(T10) '\n']);
    else
        T10=TAB(10);
        fprintf(text_tiket,[ ' Tensión Tramo 9-10
| T10 | KN | ' num2str(T10) '\n']);
    end

elseif c==11
    T11=TAB(11);
    fprintf(text_tiket,[ ' Tensión Tramo 10-B
| T11 | KN | ' num2str(T11) '\n']);
end

end
fprintf(text_tiket,[ ' Tensión máxima del Tramo AB | Tmax
| KN | ' num2str(TABmax) '\n']);
fprintf(text_tiket,[ ' Tensión mínima del Tramo AB | Tmin
| KN | ' num2str(TABmin) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        h1=hAB(1);
        fprintf(text_tiket,[ ' Flecha o altura 1 del Tramo
AB | h1 | m | ' num2str(h1) '\n']);
    elseif c==2
        h2=hAB(2);
        fprintf(text_tiket,[ ' Flecha o altura 2 del Tramo
AB | h2 | m | ' num2str(h2) '\n']);
    elseif c==3
        h3=hAB(3);
        fprintf(text_tiket,[ ' Flecha o altura 3 del Tramo
AB | h3 | m | ' num2str(h3) '\n']);
    elseif c==4
        h4=hAB(4);

```

```

        fprintf(text_tiket,[ ' Flecha o altura 4 del Tramo
AB | h4 | m | ' num2str(h4) '\n']);
        elseif c==5
            h5=hAB(5);
            fprintf(text_tiket,[ ' Flecha o altura 5 del Tramo
AB | h5 | m | ' num2str(h5) '\n']);
        elseif c==6
            h6=hAB(6);
            fprintf(text_tiket,[ ' Flecha o altura 6 del Tramo
AB | h6 | m | ' num2str(h6) '\n']);
        elseif c==7
            h7=hAB(7);
            fprintf(text_tiket,[ ' Flecha o altura 7 del Tramo
AB | h7 | m | ' num2str(h7) '\n']);
        elseif c==8
            h8=hAB(8);
            fprintf(text_tiket,[ ' Flecha o altura 8 del Tramo
AB | h8 | m | ' num2str(h8) '\n']);
        elseif c==9
            h9=hAB(9);
            fprintf(text_tiket,[ ' Flecha o altura 9 del Tramo
AB | h9 | m | ' num2str(h9) '\n']);
        elseif c==10
            h10=hAB(10);
            fprintf(text_tiket,[ ' Flecha o altura 10 del Tramo
AB| h10 | m | ' num2str(h10) '\n']);
        end
        fprintf(text_tiket,[ ' Vano entre A y B | L
| m | ' num2str(LAB) '\n']);
        fprintf(text_tiket,[ ' Longitud del Cable
T10=TAB(10);
        fprintf(text_tiket,[ ' Tensión Tramo 9-10
| T10 | KN | ' num2str(T10) '\n']);
        end
        elseif c==11
            T11=TAB(11);
            fprintf(text_tiket,[ ' Tensión Tramo 10-B
| T11 | KN | ' num2str(T11) '\n']);
        end
        fprintf(text_tiket,[ ' Tensión máxima del Tramo AB | Tmax
| KN | ' num2str(TABmax) '\n']);
        fprintf(text_tiket,[ ' Tensión mínima del Tramo AB | Tmin
| KN | ' num2str(TABmin) '\n']);
        s=NWAB;
        for c = 1:s
            if c==1
                h1=hAB(1);
                fprintf(text_tiket,[ ' Flecha o altura 1 del Tramo
AB | h1 | m | ' num2str(h1) '\n']);
            elseif c==2
                h2=hAB(2);
                fprintf(text_tiket,[ ' Flecha o altura 2 del Tramo
AB | h2 | m | ' num2str(h2) '\n']);
            elseif c==3
                h3=hAB(3);
                fprintf(text_tiket,[ ' Flecha o altura 3 del Tramo
AB | h3 | m | ' num2str(h3) '\n']);
            elseif c==4
                h4=hAB(4);
                fprintf(text_tiket,[ ' Flecha o altura 4 del Tramo
AB | h4 | m | ' num2str(h4) '\n']);
            elseif c==5
                h5=hAB(5);
                fprintf(text_tiket,[ ' Flecha o altura 5 del Tramo

```

```

AB | h5 | m | ' num2str(h5) '\n']);
    elseif c==6
        h6=hAB(6);
        fprintf(text_tiket,[ ' Flecha o altura 6 del Tramo
AB | h6 | m | ' num2str(h6) '\n']);
    elseif c==7
        h7=hAB(7);
        fprintf(text_tiket,[ ' Flecha o altura 7 del Tramo
AB | h7 | m | ' num2str(h7) '\n']);
    elseif c==8
        h8=hAB(8);
        fprintf(text_tiket,[ ' Flecha o altura 8 del Tramo
AB | h8 | m | ' num2str(h8) '\n']);
    elseif c==9
        h9=hAB(9);
        fprintf(text_tiket,[ ' Flecha o altura 9 del Tramo
AB | h9 | m | ' num2str(h9) '\n']);
    elseif c==10
        h10=hAB(10);
        fprintf(text_tiket,[ ' Flecha o altura 10 del Tramo
AB| h10 | m | ' num2str(h10) '\n']);
    end
end

fprintf(text_tiket,[ ' Vano entre A y B | L
| m | ' num2str(LAB) '\n']);
fprintf(text_tiket,[ ' Longitud del Cable | S
| m | ' num2str(SAB) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' Tramo BC\n');
fprintf(text_tiket, ' ----- \n');
s=NWBC+1;
v1=s;
for c = 1:s
    if c==1
        T1=TBC(1);
        fprintf(text_tiket,[ ' Tensión Tramo B-1
| T1 | KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TBC(2);
            fprintf(text_tiket,[ ' Tensión Tramo 1-C
| T2 | KN | ' num2str(T2) '\n');
        else
            T2=TBC(2);
            fprintf(text_tiket,[ ' Tensión Tramo 1-2
| T2 | KN | ' num2str(T2) '\n');
        end
    elseif c==3
        if c==v1
            T3=TBC(3);
            fprintf(text_tiket,[ ' Tensión Tramo 2-C
| T3 | KN | ' num2str(T3) '\n');
        else
            T3=TBC(3);
            fprintf(text_tiket,[ ' Tensión Tramo 2-3
| T3 | KN | ' num2str(T3) '\n');
        end
    elseif c==4
        if c==v1
            T4=TBC(4);
            fprintf(text_tiket,[ ' Tensión Tramo 3-C
| T4 | KN | ' num2str(T4) '\n');
        else
            T4=TBC(4);
            fprintf(text_tiket,[ ' Tensión Tramo 3-4
| T4 | KN | ' num2str(T4) '\n');

```

```

end
elseif c==5
    if c==v1
        T5=TBC(5);
        fprintf(text_tiket,[ '          Tensión Tramo 4-C
| T5    | KN    | ' num2str(T5) '\n']);
    else
        T5=TBC(5);
        fprintf(text_tiket,[ '          Tensión Tramo 4-5
| T5    | KN    | ' num2str(T5) '\n']);
    end
elseif c==6
    if c==v1
        T6=TBC(6);
        fprintf(text_tiket,[ '          Tensión Tramo 5-C
| T6    | KN    | ' num2str(T6) '\n']);
    else
        T6=TBC(6);
        fprintf(text_tiket,[ '          Tensión Tramo 5-6
| T6    | KN    | ' num2str(T6) '\n']);
    end
elseif c==7
    if c==v1
        T7=TBC(7);
        fprintf(text_tiket,[ '          Tensión Tramo 6-C
| T7    | KN    | ' num2str(T7) '\n']);
    else
        T7=TBC(7);
        fprintf(text_tiket,[ '          Tensión Tramo 6-7
| T7    | KN    | ' num2str(T7) '\n']);
    end
elseif c==8
    if c==v1
        T8=TBC(8);
        fprintf(text_tiket,[ '          Tensión Tramo 7-C
| T8    | KN    | ' num2str(T8) '\n']);
    else
        T8=TBC(8);
        fprintf(text_tiket,[ '          Tensión Tramo 7-8
| T8    | KN    | ' num2str(T8) '\n']);
    end
elseif c==9
    if c==v1
        T9=TBC(9);
        fprintf(text_tiket,[ '          Tensión Tramo 8-C
| T9    | KN    | ' num2str(T9) '\n']);
    else
        T9=TBC(9);
        fprintf(text_tiket,[ '          Tensión Tramo 8-9
| T9    | KN    | ' num2str(T9) '\n']);
    end
elseif c==10
    if c==v1
        T10=TBC(10);
        fprintf(text_tiket,[ '          Tensión Tramo 9-C
| T10   | KN    | ' num2str(T10) '\n']);
    else
        T10=TBC(10);
        fprintf(text_tiket,[ '          Tensión Tramo 9-10
| T10   | KN    | ' num2str(T10) '\n']);
    end
elseif c==11
    T11=TBC(11);
    fprintf(text_tiket,[ '          Tensión Tramo 10-C
| T11   | KN    | ' num2str(T11) '\n']);
end
end

```

```

end
fprintf(text_tiket,[ ' Tensión máxima del Tramo BC | Tmax
| KN | ' num2str(TBCmax) '\n']);
fprintf(text_tiket,[ ' Tensión mínima del Tramo BC | Tmin
| KN | ' num2str(TBCmin) '\n']);
s=NWBC;
for c = 1:s
if c==1
h1=hBC(1);
fprintf(text_tiket,[ ' Flecha o altura 1 del Tramo
BC | h1 | m | ' num2str(h1) '\n']);
elseif c==2
h2=hBC(2);
fprintf(text_tiket,[ ' Flecha o altura 2 del Tramo
BC | h2 | m | ' num2str(h2) '\n']);
elseif c==3
h3=hBC(3);
fprintf(text_tiket,[ ' Flecha o altura 3 del Tramo
BC | h3 | m | ' num2str(h3) '\n']);
elseif c==4
h4=hBC(4);
fprintf(text_tiket,[ ' Flecha o altura 4 del Tramo
BC | h4 | m | ' num2str(h4) '\n']);
elseif c==5
h5=hBC(5);
fprintf(text_tiket,[ ' Flecha o altura 5 del Tramo
BC | h5 | m | ' num2str(h5) '\n']);
elseif c==6
h6=hBC(6);
fprintf(text_tiket,[ ' Flecha o altura 6 del Tramo
BC | h6 | m | ' num2str(h6) '\n']);
elseif c==7
h7=hBC(7);
fprintf(text_tiket,[ ' Flecha o altura 7 del Tramo
BC | h7 | m | ' num2str(h7) '\n']);
elseif c==8
h8=hBC(8);
fprintf(text_tiket,[ ' Flecha o altura 8 del Tramo
BC | h8 | m | ' num2str(h8) '\n']);
elseif c==9
h9=hBC(9);
fprintf(text_tiket,[ ' Flecha o altura 9 del Tramo
BC | h9 | m | ' num2str(h9) '\n']);
elseif c==10
h10=hBC(10);
fprintf(text_tiket,[ ' Flecha o altura 10 del Tramo
BC | h10 | m | ' num2str(h10) '\n']);
end
end
fprintf(text_tiket,[ ' Vano entre B y C | L
| m | ' num2str(LBC) '\n']);
fprintf(text_tiket,[ ' Longitud del Cable | S
| m | ' num2str(SBC) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' Tramo CD\n');
fprintf(text_tiket, ' ----- \n');
s=NWCD+1;
v1=s;
for c = 1:s
if c==1
T1=TCD(1);
fprintf(text_tiket,[ ' Tensión Tramo C-1
| T1 | KN | ' num2str(T1) '\n']);
elseif c==2
if c==v1
T2=TCD(2);
fprintf(text_tiket,[ ' Tensión Tramo 1-D

```



```

| T2 | KN | ' num2str(T2) '\n']);
      else
        T2=TCD(2);
        fprintf(text_tiket,[ '          Tensión Tramo 1-2
| T2 | KN | ' num2str(T2) '\n']);
      end
      elseif c==3
        if c==v1
          T3=TCD(3);
          fprintf(text_tiket,[ '          Tensión Tramo 2-D
| T3 | KN | ' num2str(T3) '\n']);
        else
          T3=TCD(3);
          fprintf(text_tiket,[ '          Tensión Tramo 2-3
| T3 | KN | ' num2str(T3) '\n']);
        end
      elseif c==4
        if c==v1
          T4=TCD(4);
          fprintf(text_tiket,[ '          Tensión Tramo 3-D
| T4 | KN | ' num2str(T4) '\n']);
        else
          T4=TCD(4);
          fprintf(text_tiket,[ '          Tensión Tramo 3-4
| T4 | KN | ' num2str(T4) '\n']);
        end
      elseif c==5
        if c==v1
          T5=TCD(5);
          fprintf(text_tiket,[ '          Tensión Tramo 4-D
| T5 | KN | ' num2str(T5) '\n']);
        else
          T5=TCD(5);
          fprintf(text_tiket,[ '          Tensión Tramo 4-5
| T5 | KN | ' num2str(T5) '\n']);
        end
      elseif c==6
        if c==v1
          T6=TCD(6);
          fprintf(text_tiket,[ '          Tensión Tramo 5-D
| T6 | KN | ' num2str(T6) '\n']);
        else
          T6=TCD(6);
          fprintf(text_tiket,[ '          Tensión Tramo 5-6
| T6 | KN | ' num2str(T6) '\n']);
        end
      elseif c==7
        if c==v1
          T7=TCD(7);
          fprintf(text_tiket,[ '          Tensión Tramo 6-D
| T7 | KN | ' num2str(T7) '\n']);
        else
          T7=TCD(7);
          fprintf(text_tiket,[ '          Tensión Tramo 6-7
| T7 | KN | ' num2str(T7) '\n']);
        end
      elseif c==8
        if c==v1
          T8=TCD(8);
          fprintf(text_tiket,[ '          Tensión Tramo 7-D
| T8 | KN | ' num2str(T8) '\n']);
        else
          T8=TCD(8);
          fprintf(text_tiket,[ '          Tensión Tramo 7-8
| T8 | KN | ' num2str(T8) '\n']);
        end
      elseif c==9

```

```

        if c==v1
            T9=TCD(9);
            fprintf(text_tiket,[ '          Tensión Tramo 8-D
| T9   | KN   | ' num2str(T9) '\n']);
        else
            T9=TCD(9);
            fprintf(text_tiket,[ '          Tensión Tramo 8-9
| T9   | KN   | ' num2str(T9) '\n']);
        end
        elseif c==10
            if c==v1
                T10=TCD(10);
                fprintf(text_tiket,[ '          Tensión Tramo 9-D
| T10  | KN   | ' num2str(T10) '\n']);
            else
                T10=TCD(10);
                fprintf(text_tiket,[ '          Tensión Tramo 9-10
| T10  | KN   | ' num2str(T10) '\n']);
            end
        elseif c==11
            T11=TCD(11);
            fprintf(text_tiket,[ '          Tensión Tramo 10-D
| T11  | KN   | ' num2str(T11) '\n']);
        end
    end
    fprintf(text_tiket,[ '          Tensión máxima del Tramo CD   | Tmax
| KN   | ' num2str(TCDmax) '\n']);
    fprintf(text_tiket,[ '          Tensión mínima del Tramo CD   | Tmin
| KN   | ' num2str(TCDmin) '\n']);
    s=NWCD;
    for c = 1:s
        if c==1
            h1=hCD(1);
            fprintf(text_tiket,[ '          Flecha o altura 1 del Tramo
CD | h1 | m   | ' num2str(h1) '\n']);
        elseif c==2
            h2=hCD(2);
            fprintf(text_tiket,[ '          Flecha o altura 2 del Tramo
CD | h2 | m   | ' num2str(h2) '\n']);
        elseif c==3
            h3=hCD(3);
            fprintf(text_tiket,[ '          Flecha o altura 3 del Tramo
CD | h3 | m   | ' num2str(h3) '\n']);
        elseif c==4
            h4=hCD(4);
            fprintf(text_tiket,[ '          Flecha o altura 4 del Tramo
CD | h4 | m   | ' num2str(h4) '\n']);
        elseif c==5
            h5=hCD(5);
            fprintf(text_tiket,[ '          Flecha o altura 5 del Tramo
CD | h5 | m   | ' num2str(h5) '\n']);
        elseif c==6
            h6=hCD(6);
            fprintf(text_tiket,[ '          Flecha o altura 6 del Tramo
CD | h6 | m   | ' num2str(h6) '\n']);
        elseif c==7
            h7=hCD(7);
            fprintf(text_tiket,[ '          Flecha o altura 7 del Tramo
CD | h7 | m   | ' num2str(h7) '\n']);
        elseif c==8
            h8=hCD(8);
            fprintf(text_tiket,[ '          Flecha o altura 8 del Tramo
CD | h8 | m   | ' num2str(h8) '\n']);
        elseif c==9
            h9=hCD(9);
            fprintf(text_tiket,[ '          Flecha o altura 9 del Tramo
CD | h9 | m   | ' num2str(h9) '\n']);

```



```

%% identificar el número de tramo
fprintf(text_tiket,[ ' Number of sections           | NT   | #
| ' num2str(NT) '\n']);
fprintf(text_tiket, '           \n');
% para el tramo AB datos de entrada
fprintf(text_tiket, '           Stretch AB\n');
fprintf(text_tiket, '           -----\n');
fprintf(text_tiket,[ '           Number of loads           | NW   | #
| ' num2str(NWAB) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        W1=WAB(1);
        fprintf(text_tiket,[ '           Load 1   of the AB section           | W1   |
KN | ' num2str(W1) '\n']);
        elseif c==2
            W2=WAB(2);
            fprintf(text_tiket,[ '           Load 2   of the AB section           | W2   |
KN | ' num2str(W2) '\n']);
            elseif c==3
                W3=WAB(3);
                fprintf(text_tiket,[ '           Load 3   of the AB section           | W3   |
KN | ' num2str(W3) '\n']);
                elseif c==4
                    W4=WAB(4);
                    fprintf(text_tiket,[ '           Load 4   of the AB section           | W4   |
KN | ' num2str(W4) '\n']);
                    elseif c==5
                        W5=WAB(5);
                        fprintf(text_tiket,[ '           Load 5   of the AB section           | W5   |
KN | ' num2str(W5) '\n']);
                        elseif c==6
                            W6=WAB(6);
                            fprintf(text_tiket,[ '           Load 6   of the AB section           | W6   |
KN | ' num2str(W6) '\n']);
                            elseif c==7
                                W7=WAB(7);
                                fprintf(text_tiket,[ '           Load 7   of the AB section           | W7   |
KN | ' num2str(W7) '\n']);
                                elseif c==8
                                    W8=WAB(8);
                                    fprintf(text_tiket,[ '           Load 8   of the AB section           | W8   |
KN | ' num2str(W8) '\n']);
                                    elseif c==9
                                        W9=WAB(9);
                                        fprintf(text_tiket,[ '           Load 9   of the AB section           | W9   |
KN | ' num2str(W9) '\n']);
                                        elseif c==10
                                            W10=WAB(10);
                                            fprintf(text_tiket,[ '           Load 10 of the AB section           | W10  |
KN | ' num2str(W10) '\n']);
                                            end
                                        end
                                s=NWAB+1;
                                for c = 1:s
                                    if c==1
                                        D1=DAB(1);
                                        fprintf(text_tiket,[ '           Distance 1   of the AB section           | D1   |
m | ' num2str(D1) '\n']);
                                        elseif c==2
                                            D2=DAB(2);
                                            fprintf(text_tiket,[ '           Distance 2 of the AB section           | D2   |
m | ' num2str(D2) '\n']);
                                            elseif c==3
                                                D3=DAB(3);
                                                fprintf(text_tiket,[ '           Distance 3 of the AB section           | D3   |
m | ' num2str(D3) '\n']);

```

```

elseif c==4
    D4=DAB(4);
m   fprintf(text_tiket,[ '           Distance 4 of the AB section | D4 |
    | ' num2str(D4) '\n']);
    elseif c==5
        D5=DAB(5);
m   fprintf(text_tiket,[ '           Distance 5 of the AB section | D5 |
    | ' num2str(D5) '\n']);
    elseif c==6
        D6=DAB(6);
m   fprintf(text_tiket,[ '           Distance 6 of the AB section | D6 |
    | ' num2str(D6) '\n']);
    elseif c==7
        D7=DAB(7);
m   fprintf(text_tiket,[ '           Distance 7 of the AB section | D7 |
    | ' num2str(D7) '\n']);
    elseif c==8
        D8=DAB(8);
m   fprintf(text_tiket,[ '           Distance 8 of the AB section | D8 |
    | ' num2str(D8) '\n']);
    elseif c==9
        D9=DAB(9);
m   fprintf(text_tiket,[ '           Distance 9 of the AB section | D9 |
    | ' num2str(D9) '\n']);
    elseif c==10
        D10=DAB(10);
m   fprintf(text_tiket,[ '           Distance 10 of the AB section | D10 |
    | ' num2str(D10) '\n']);
    elseif c==11
        D11=DAB(11);
m   fprintf(text_tiket,[ '           Distance 11 of the AB section | D11 |
    | ' num2str(D11) '\n']);
    end
    fprintf(text_tiket,[ '           Height 1 of the AB section | h1
| m   | ' num2str(h1AB) '\n']);
    fprintf(text_tiket,[ '           Difference between A and B | YB | m
| ' num2str(YB) '\n']);
    % para el tramo BC datos de entrada
    fprintf(text_tiket, '           \n');
    fprintf(text_tiket, '           Stretch BC\n');
    fprintf(text_tiket, '           ----- \n');
    fprintf(text_tiket,[ '           Number of loads | NW | #
| ' num2str(NWBC) '\n']);
    s=NWBC;
    for c = 1:s
        if c==1
            W1=WBC(1);
KN   fprintf(text_tiket,[ '           Load 1 of the BC section | W1 |
    | ' num2str(W1) '\n']);
            elseif c==2
                W2=WBC(2);
KN   fprintf(text_tiket,[ '           Load 2 of the BC section | W2 |
    | ' num2str(W2) '\n']);
            elseif c==3
                W3=WBC(3);
KN   fprintf(text_tiket,[ '           Load 3 of the BC section | W3 |
    | ' num2str(W3) '\n']);
            elseif c==4
                W4=WBC(4);
KN   fprintf(text_tiket,[ '           Load 4 of the BC section | W4 |
    | ' num2str(W4) '\n']);
            elseif c==5
                W5=WBC(5);
KN   fprintf(text_tiket,[ '           Load 5 of the BC section | W5 |
    | ' num2str(W5) '\n']);
            elseif c==6

```

	W6=WBC(6);	Load 6 of the BC section	W6	
KN	fprintf(text_tiket,[' num2str(W6) '\n']);			
	elseif c==7			
	W7=WBC(7);	Load 7 of the BC section	W7	
KN	fprintf(text_tiket,[' num2str(W7) '\n']);			
	elseif c==8			
	W8=WBC(8);	Load 8 of the BC section	W8	
KN	fprintf(text_tiket,[' num2str(W8) '\n']);			
	elseif c==9			
	W9=WBC(9);	Load 9 of the BC section	W9	
KN	fprintf(text_tiket,[' num2str(W9) '\n']);			
	elseif c==10			
	W10=WBC(10);	Load 10 of the BC section	W10	
KN	fprintf(text_tiket,[' num2str(W10) '\n']);			
	end			
	end			
	s=NWBC+1;			
	for c = 1:s			
	if c==1			
	D1=DBC(1);	Distance 1 of the BC section	D1	
m	fprintf(text_tiket,[' num2str(D1) '\n']);			
	elseif c==2			
	D2=DBC(2);	Distance 2 of the BC section	D2	
m	fprintf(text_tiket,[' num2str(D2) '\n']);			
	elseif c==3			
	D3=DBC(3);	Distance 3 of the BC section	D3	
m	fprintf(text_tiket,[' num2str(D3) '\n']);			
	elseif c==4			
	D4=DBC(4);	Distance 4 of the BC section	D4	
m	fprintf(text_tiket,[' num2str(D4) '\n']);			
	elseif c==5			
	D5=DBC(5);	Distance 5 of the BC section	D5	
m	fprintf(text_tiket,[' num2str(D5) '\n']);			
	elseif c==6			
	D6=DBC(6);	Distancia 6 of the BC section	D6	
m	fprintf(text_tiket,[' num2str(D6) '\n']);			
	elseif c==7			
	D7=DBC(7);	Distance 7 of the BC section	D7	
m	fprintf(text_tiket,[' num2str(D7) '\n']);			
	elseif c==8			
	D8=DBC(8);	Distance 8 of the BC section	D8	
m	fprintf(text_tiket,[' num2str(D8) '\n']);			
	elseif c==9			
	D9=DBC(9);	Distance 9 of the BC section	D9	
m	fprintf(text_tiket,[' num2str(D9) '\n']);			
	elseif c==10			
	D10=DBC(10);	Distance 10 of the BC section	D10	
m	fprintf(text_tiket,[' num2str(D10) '\n']);			
	elseif c==11			
	D11=DBC(11);	Distance 11 of the BC section	D11	
m	fprintf(text_tiket,[' num2str(D11) '\n']);			

```

        end
    end
    fprintf(text_tiket,[ '          Height 1 of the BC section      | h1      | m
| ' num2str(h1BC) '\n']);
    fprintf(text_tiket,[ '          Difference between A y C      | YC      | m
| ' num2str(YC) '\n']);
    fprintf(text_tiket, '
\n');
    % para el tramo CD datos de entrada
    fprintf(text_tiket, '          \n');
    fprintf(text_tiket, '          Tramo CD\n');
    fprintf(text_tiket, '          ----- \n');
    fprintf(text_tiket,[ '          Number of loads                      | NW      | #
| ' num2str(NWCD) '\n']);
    s=NWCD;
    for c = 1:s
        if c==1
            W1=WCD(1);
            fprintf(text_tiket,[ '          Load 1 of the CD section      | W1      |
KN | ' num2str(W1) '\n']);
            elseif c==2
                W2=WCD(2);
                fprintf(text_tiket,[ '          Load 2 of the CD section      | W2      |
KN | ' num2str(W2) '\n']);
            elseif c==3
                W3=WCD(3);
                fprintf(text_tiket,[ '          Load 3 of the CD section      | W3      |
KN | ' num2str(W3) '\n']);
            elseif c==4
                W4=WCD(4);
                fprintf(text_tiket,[ '          Load 4 of the CD section      | W4      |
KN | ' num2str(W4) '\n']);
            elseif c==5
                W5=WCD(5);
                fprintf(text_tiket,[ '          Load 5 of the CD section      | W5      |
KN | ' num2str(W5) '\n']);
            elseif c==6
                W6=WCD(6);
                fprintf(text_tiket,[ '          Load 6 of the CD section      | W6      |
KN | ' num2str(W6) '\n']);
            elseif c==7
                W7=WCD(7);
                fprintf(text_tiket,[ '          Load 7 of the CD section      | W7      |
KN | ' num2str(W7) '\n']);
            elseif c==8
                W8=WCD(8);
                fprintf(text_tiket,[ '          Load 8 of the CD section      | W8      |
KN | ' num2str(W8) '\n']);
            elseif c==9
                W9=WCD(9);
                fprintf(text_tiket,[ '          Load 9 of the CD section      | W9      |
KN | ' num2str(W9) '\n']);
            elseif c==10
                W10=WCD(10);
                fprintf(text_tiket,[ '          Load 10 of the CD section     | W10     |
KN | ' num2str(W10) '\n']);
            end
        end
        s=NWCD+1;
        for c = 1:s
            if c==1
                D1=DCD(1);
                fprintf(text_tiket,[ '          Distance 1 of the CD section   | D1      |
m | ' num2str(D1) '\n']);
            elseif c==2
                D2=DCD(2);
                fprintf(text_tiket,[ '          Distance 2 of the CD section   | D2      |

```

```
m      | ' num2str(D2) '\n']);
        elseif c==3
            D3=DCD(3);
fprintf(text_tiket,[ '                               Distance 3 of the CD section   | D3   |
m      | ' num2str(D3) '\n']);
        elseif c==4
            D4=DCD(4);
fprintf(text_tiket,[ '                               Distance 4 of the CD section   | D4   |
m      | ' num2str(D4) '\n']);
        elseif c==5
            D5=DCD(5);
fprintf(text_tiket,[ '                               Distance 5 of the CD section   | D5   |
m      | ' num2str(D5) '\n']);
elseif c==6
            D6=DCD(6);
fprintf(text_tiket,[ '                               Distance 6 of the CD section   | D6   |
m      | ' num2str(D6) '\n']);
        elseif c==7
            D7=DCD(7);
fprintf(text_tiket,[ '                               Distance 7 of the CD section   | D7   |
m      | ' num2str(D7) '\n']);
        elseif c==8
            D8=DCD(8);
fprintf(text_tiket,[ '                               Distance 8 of the CD section   | D8   |
m      | ' num2str(D8) '\n']);
        elseif c==9
            D9=DCD(9);
fprintf(text_tiket,[ '                               Distance 9 of the CD section   | D9   |
m      | ' num2str(D9) '\n']);
        elseif c==10
            D10=DCD(10);
fprintf(text_tiket,[ '                              Distance 10 of the CD section  | D10  |
m      | ' num2str(D10) '\n']);
        elseif c==11
            D11=DCD(11);
fprintf(text_tiket,[ '                              Distance 11 of the CD section  | D11  |
m      | ' num2str(D11) '\n']);
    end
end
fprintf(text_tiket,[ '                               Height 1 of the CD section   | h1   | m
| ' num2str(h1CD) '\n']);
fprintf(text_tiket,[ '                               Difference between C y D     | YD   | m
| ' num2str(YD) '\n']);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '_____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '                                DESCRIPTION                        |SYMBOL| UNITY
| MAGNITUDE\n');
fprintf(text_tiket,
'_____|_____|_____|_____ \n');
fprintf(text_tiket,[ '                                Reaction at point A                | A     | KN
| ' num2str(TA) '\n']);
fprintf(text_tiket,[ '                                Reaction on the x-axis of A         | Ax    | KN
| ' num2str(TAx) '\n']);
fprintf(text_tiket,[ '                                Reaction on the y-axis of A         | Ay    | KN
| ' num2str(TAy) '\n']);
fprintf(text_tiket,[ '                                Reaction at point B section AB      | B-AB  | KN
| ' num2str(TBAB) '\n']);
fprintf(text_tiket,[ '                                Reaction at point B section BC      | B-BC  | KN
| ' num2str(TBBC) '\n']);
fprintf(text_tiket,[ '                                Reaction at point B                  | B     | KN
| ' num2str(TB) '\n']);
fprintf(text_tiket,[ '                                Reaction on the x-axis of B          | Bx    | KN
| ' num2str(TBx) '\n');
```



```

fprintf(text_tiket,[ '          Reaction on the y-axis of B      | By      | KN
| ' num2str(TBy) '\n']);
fprintf(text_tiket,[ '          Reaction at point B section BC | C-BC | KN
| ' num2str(TCBC) '\n']);
fprintf(text_tiket,[ '          Reaction at point B section CD | C-CD
| KN      | ' num2str(TCCD) '\n']);
fprintf(text_tiket,[ '          Reaction at point C              | C      | KN
| ' num2str(TC) '\n']);
fprintf(text_tiket,[ '          Reaction on the x-axis of C      | Cx      | KN
| ' num2str(TCx) '\n']);
fprintf(text_tiket,[ '          Reaction on the y-axis of C      | Cy      | KN
| ' num2str(TCy) '\n']);
fprintf(text_tiket,[ '          Reaction at point D              | D      | KN
| ' num2str(TD) '\n']);
fprintf(text_tiket,[ '          Reaction on the x-axis of D      | Dx      | KN
| ' num2str(TDx) '\n']);
fprintf(text_tiket,[ '          Reaction on the y-axis of D      | Dy      | KN
| ' num2str(TDy) '\n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          Stretch AB\n');
fprintf(text_tiket, '          -----\n');
s=NWAB+1;
v1=s;
for c = 1:s
    if c==1
        T1=TAB(1);
        fprintf(text_tiket,[ '          Stretch tension A-1
| T1      | KN      | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TAB(2);
            fprintf(text_tiket,[ '          Stretch tension 1-B
| T2      | KN      | ' num2str(T2) '\n']);
        else
            T2=TAB(2);
            fprintf(text_tiket,[ '          Stretch tension 1-2
| T2      | KN      | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=TAB(3);
            fprintf(text_tiket,[ '          Stretch tension 2-B
| T3      | KN      | ' num2str(T3) '\n']);
        else
            T3=TAB(3);
            fprintf(text_tiket,[ '          Stretch tension 2-3
| T3      | KN      | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1
            T4=TAB(4);
            fprintf(text_tiket,[ '          Stretch tension 3-B
| T4      | KN      | ' num2str(T4) '\n']);
        else
            T4=TAB(4);
            fprintf(text_tiket,[ '          Stretch tension 3-4
| T4      | KN      | ' num2str(T4) '\n']);
        end
    elseif c==5
        if c==v1
            T5=TAB(5);
            fprintf(text_tiket,[ '          Stretch tension 4-B
| T5      | KN      | ' num2str(T5) '\n']);
        else
            T5=TAB(5);
            fprintf(text_tiket,[ '          Stretch tension 4-5
| T5      | KN      | ' num2str(T5) '\n']);

```

```

end
    elseif c==6
        if c==v1
            T6=TAB(6);
            fprintf(text_tiket,[ '           Stretch tension 5-B
| T6   | KN   | ' num2str(T6) '\n']);
        else
            T6=TAB(6);
            fprintf(text_tiket,[ '           Stretch tension 5-6
| T6   | KN   | ' num2str(T6) '\n']);
        end
    elseif c==7
        if c==v1
            T7=TAB(7);
            fprintf(text_tiket,[ '           Stretch tension 6-B
| T7   | KN   | ' num2str(T7) '\n']);
        else
            T7=TAB(7);
            fprintf(text_tiket,[ '           Stretch tension 6-7
| T7   | KN   | ' num2str(T7) '\n']);
        end
    elseif c==8
        if c==v1
            T8=TAB(8);
            fprintf(text_tiket,[ '           Stretch tension 7-B
| T8   | KN   | ' num2str(T8) '\n']);
        else
            T8=TAB(8);
            fprintf(text_tiket,[ '           Stretch tension 7-8
| T8   | KN   | ' num2str(T8) '\n']);
        end
    elseif c==9
        if c==v1
            T9=TAB(9);
            fprintf(text_tiket,[ '           Stretch tension 8-B
| T9   | KN   | ' num2str(T9) '\n']);
        else
            T9=TAB(9);
            fprintf(text_tiket,[ '           Stretch tension 8-9
| T9   | KN   | ' num2str(T9) '\n']);
        end
    elseif c==10
        if c==v1
            T10=TAB(10);
            fprintf(text_tiket,[ '           Stretch tension 9-B
| T10  | KN   | ' num2str(T10) '\n']);
        else
            T10=TAB(10);
            fprintf(text_tiket,[ '           Stretch tension 9-10
| T10  | KN   | ' num2str(T10) '\n']);
        end
    elseif c==11
        T11=TAB(11);
        fprintf(text_tiket,[ '           Stretch
tension 10-B | T11 | KN   | ' num2str(T11) '\n']);
    end
end
fprintf(text_tiket,[ '           Maximun stretch tension AB   | Tmax | KN
| ' num2str(TABmax) '\n']);
fprintf(text_tiket,[ '           Minimum stretch tension AB   | Tmin | KN
| ' num2str(TABmin) '\n']);
s=NWAB;
for c = 1:s
    if c==1
        h1=hAB(1);
        fprintf(text_tiket,[ '           Arrow or height 1 of section

```

```

AB| h1      | m      | ' num2str(h1) '\n']);
      elseif c==2
          h2=hAB(2);
          fprintf(text_tiket,[ '
AB| h2      | m      | ' num2str(h2) '\n']);
      elseif c==3
          h3=hAB(3);
          fprintf(text_tiket,[ '
AB| h3      | m      | ' num2str(h3) '\n']);
      elseif c==4
          h4=hAB(4);
          fprintf(text_tiket,[ '
AB| h4      | m      | ' num2str(h4) '\n']);
      elseif c==5
          h5=hAB(5);
          fprintf(text_tiket,[ '
AB| h5      | m      | ' num2str(h5) '\n']);
      elseif c==6
          h6=hAB(6);
          fprintf(text_tiket,[ '
AB| h6      | m      | ' num2str(h6) '\n']);
      elseif c==7
          h7=hAB(7);
          fprintf(text_tiket,[ '
AB| h7      | m      | ' num2str(h7) '\n']);
      elseif c==8
          h8=hAB(8);
          fprintf(text_tiket,[ '
AB| h8      | m      | ' num2str(h8) '\n']);
      elseif c==9
          h9=hAB(9);
          fprintf(text_tiket,[ '
AB| h9      | m      | ' num2str(h9) '\n']);
      elseif c==10
          h10=hAB(10);
          fprintf(text_tiket,[ '
AB| h10     | m      | ' num2str(h10) '\n']);
      end
  end
  fprintf(text_tiket,[ '
      | ' num2str(LAB) '\n']);
  fprintf(text_tiket,[ '
      | ' num2str(SAB) '\n']);
  fprintf(text_tiket, '
  fprintf(text_tiket, '
  fprintf(text_tiket, '
  s=NWBC+1;
  v1=s;
  for c = 1:s
      if c==1
          T1=TBC(1);
          fprintf(text_tiket,[ '
| T1      | KN      | ' num2str(T1) '\n']);
      elseif c==2
          if c==v1
              T2=TBC(2);
              fprintf(text_tiket,[ '
| T2      | KN      | ' num2str(T2) '\n']);
          else
              T2=TBC(2);
              fprintf(text_tiket,[ '
| T2      | KN      | ' num2str(T2) '\n']);
          end
      elseif c==3
          if c==v1
              T3=TBC(3);
              fprintf(text_tiket,[ '

```

Arrow or height 2 of section
 Arrow or height 3 of section
 Arrow or height 4 of section
 Arrow or height 5 of section
 Arrow or height 6 of section
 Arrow or height 7 of section
 Arrow or height 8 del Tramo
 Arrow or height 9 del Tramo
 Arrow or height 10 del Tramo
 Vain between A and B | L | m
 Cable length | S | m
 Tramo BC\n')
 -----\n')
 Stretch tension B-1
 Stretch tension 1-C
 Stretch tension 1-2
 Stretch tension 2-C

```

| T3 | KN | ' num2str(T3) '\n'];
      else
        T3=TBC(3);
        fprintf(text_tiket,[ '           Stretch tension 2-3
| T3 | KN | ' num2str(T3) '\n'];
      end
      elseif c==4
        if c==v1
          T4=TBC(4);
          fprintf(text_tiket,[ '           Stretch tension 3-C
| T4 | KN | ' num2str(T4) '\n'];
        else
          T4=TBC(4);
          fprintf(text_tiket,[ '           Stretch tension 3-4
| T4 | KN | ' num2str(T4) '\n'];
        end
      elseif c==5
        if c==v1
          T5=TBC(5);
          fprintf(text_tiket,[ '           Stretch tension 4-C
| T5 | KN | ' num2str(T5) '\n'];
        else
          T5=TBC(5);
          fprintf(text_tiket,[ '           Stretch tension 4-5
| T5 | KN | ' num2str(T5) '\n'];
        end
      elseif c==6
        if c==v1
          T6=TBC(6);
          fprintf(text_tiket,[ '           Stretch tension 5-C
| T6 | KN | ' num2str(T6) '\n'];
        else
          T6=TBC(6);
          fprintf(text_tiket,[ '           Stretch tension 5-6
| T6 | KN | ' num2str(T6) '\n'];
        end
      elseif c==7
        if c==v1
          T7=TBC(7);
          fprintf(text_tiket,[ '           Stretch tension 6-C
| T7 | KN | ' num2str(T7) '\n'];
      else
        T7=TBC(7);
        fprintf(text_tiket,[ '           Stretch tension 6-7
| T7 | KN | ' num2str(T7) '\n'];
      end

      elseif c==8
        if c==v1
          T8=TBC(8);
          fprintf(text_tiket,[ '           Stretch tension 7-C
| T8 | KN | ' num2str(T8) '\n'];
        else
          T8=TBC(8);
          fprintf(text_tiket,[ '           Stretch tension 7-8
| T8 | KN | ' num2str(T8) '\n'];
        end
      elseif c==9
        if c==v1
          T9=TBC(9);
          fprintf(text_tiket,[ '           Stretch tension 8-C
| T9 | KN | ' num2str(T9) '\n'];
        else
          T9=TBC(9);
          fprintf(text_tiket,[ '           Stretch tension 8-9
| T9 | KN | ' num2str(T9) '\n'];

```

```

end

elseif c==10
    if c==v1
        T10=TBC(10);
        fprintf(text_tiket,[ '          Stretch tension 9-C
| T10 | KN | ' num2str(T10) '\n']);
    else
        T10=TBC(10);
        fprintf(text_tiket,[ '          Stretch tension 9-10
| T10 | KN | ' num2str(T10) '\n']);
    end
end
elseif c==11
    T11=TBC(11);
    fprintf(text_tiket,[ '          Stretch tension 10-C
| T11 | KN | ' num2str(T11) '\n']);
end
end
fprintf(text_tiket,[ '          Maximun stretch tension BC | Tmax
| KN | ' num2str(TBCmax) '\n']);
fprintf(text_tiket,[ '          Minimun stretch tension BC | Tmin
| KN | ' num2str(TBCmin) '\n']);
s=NWBC;
for c = 1:s
    if c==1
        h1=hBC(1);
        fprintf(text_tiket,[ '          Arrow or height 1 of section
BC| h1 | m | ' num2str(h1) '\n']);
    elseif c==2
        h2=hBC(2);
        fprintf(text_tiket,[ '          Arrow or height 2 of section
BC| h2 | m | ' num2str(h2) '\n']);
    elseif c==3
        h3=hBC(3);
        fprintf(text_tiket,[ '          Arrow or height 3 of section BC| h3 | m
| ' num2str(h3) '\n']);
    elseif c==4
        h4=hBC(4);
        fprintf(text_tiket,[ '          Arrow or height 4 of section
BC| h4 | m | ' num2str(h4) '\n']);
    elseif c==5
        h5=hBC(5);
        fprintf(text_tiket,[ '          Arrow or height 5 of section
BC| h5 | m | ' num2str(h5) '\n']);
    elseif c==6
        h6=hBC(6);
        fprintf(text_tiket,[ '          Arrow or height 6 of section
BC| h6 | m | ' num2str(h6) '\n']);
    elseif c==7
        h7=hBC(7);
        fprintf(text_tiket,[ '          Arrow or height 7 of section
BC| h7 | m | ' num2str(h7) '\n']);
    elseif c==8
        h8=hBC(8);
        fprintf(text_tiket,[ '          Arrow or height 8 of section
BC| h8 | m | ' num2str(h8) '\n']);
    elseif c==9
        h9=hBC(9);
        fprintf(text_tiket,[ '          Arrow or height 9 of section
BC| h9 | m | ' num2str(h9) '\n']);
    elseif c==10
        h10=hBC(10);
        fprintf(text_tiket,[ '          Arrow or height 10 of section
BC| h10 | m | ' num2str(h10) '\n']);
    end
end
end
fprintf(text_tiket,[ '          Vain between B y C | L | m

```

```

| ' num2str(LBC) '\n'];
fprintf(text_tiket,[ ' Cable length | S | m
| ' num2str(SBC) '\n']);

fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' Stretch CD\n');
fprintf(text_tiket, ' -----\n');
s=NWCD+1;
v1=s;
for c = 1:s
    if c==1
        T1=TCD(1);
        fprintf(text_tiket,[ ' Stretch tension C-1
| T1 | KN | ' num2str(T1) '\n']);
    elseif c==2
        if c==v1
            T2=TCD(2);
            fprintf(text_tiket,[ ' Stretch tension 1-D
| T2 | KN | ' num2str(T2) '\n']);
        else
            T2=TCD(2);
            fprintf(text_tiket,[ ' Stretch tension 1-2
| T2 | KN | ' num2str(T2) '\n']);
        end
    elseif c==3
        if c==v1
            T3=TCD(3);
            fprintf(text_tiket,[ ' Stretch tension 2-D
| T3 | KN | ' num2str(T3) '\n']);
        else
            T3=TCD(3);
            fprintf(text_tiket,[ ' Stretch tension 2-3
| T3 | KN | ' num2str(T3) '\n']);
        end
    elseif c==4
        if c==v1
            T4=TCD(4);
            fprintf(text_tiket,[ ' Stretch tension 3-D
| T4 | KN | ' num2str(T4) '\n']);
        else
            T4=TCD(4);
            fprintf(text_tiket,[ ' Stretch tension 3-4
| T4 | KN | ' num2str(T4) '\n']);
        end
    elseif c==5
        if c==v1
            T5=TCD(5);
            fprintf(text_tiket,[ ' Stretch tension 4-D
| T5 | KN | ' num2str(T5) '\n']);
        else
            T5=TCD(5);
            fprintf(text_tiket,[ ' Stretch tension 4-5
| T5 | KN | ' num2str(T5) '\n']);
        end
    elseif c==6
        if c==v1
            T6=TCD(6);
            fprintf(text_tiket,[ ' Stretch tension 5-D
| T6 | KN | ' num2str(T6) '\n']);
        else
            T6=TCD(6);
            fprintf(text_tiket,[ ' Stretch tension 5-6
| T6 | KN | ' num2str(T6) '\n']);
        end
    elseif c==7

```

```

        if c==v1
            T7=TCD(7);
            fprintf(text_tiket,[ '           Stretch tension 6-D
| T7   | KN   | ' num2str(T7) '\n']);
        else
            T7=TCD(7);
            fprintf(text_tiket,[ '           Stretch tension 6-7
| T7   | KN   | ' num2str(T7) '\n']);
        end

        elseif c==8
            if c==v1
                T8=TCD(8);
                fprintf(text_tiket,[ '           Stretch tension 7-D
| T8   | KN   | ' num2str(T8) '\n']);
            else
                T8=TCD(8);
                fprintf(text_tiket,[ '           Stretch tension
7-8      | T8   | KN   | ' num2str(T8) '\n']);
            end

        elseif c==9
            if c==v1
                T9=TCD(9);
                fprintf(text_tiket,[ '           Stretch tension 8-D
| T9   | KN   | ' num2str(T9) '\n']);
            else
                T9=TCD(9);
                fprintf(text_tiket,[ '           Stretch tension 8-9
| T9   | KN   | ' num2str(T9) '\n']);
            end

        elseif c==10
            if c==v1
                T10=TCD(10);
                fprintf(text_tiket,[ '           Stretch tension 9-D
| T10  | KN   | ' num2str(T10) '\n']);
            else
                T10=TCD(10);
                fprintf(text_tiket,[ '           Stretch tension 9-10
| T10  | KN   | ' num2str(T10) '\n']);
            end
        elseif c==11
            T11=TCD(11);
            fprintf(text_tiket,[ '           Stretch tension 10-D
| T11  | KN   | ' num2str(T11) '\n']);
        end
        fprintf(text_tiket,[ '           Maximun stretch tension  CD   |
Tmax | KN   | ' num2str(TCDmax) '\n']);
        fprintf(text_tiket,[ '           Minimun stretch tension  CD   |
Tmin | KN   | ' num2str(TCDmin) '\n']);
        s=NWCD;
        for c = 1:s
            if c==1
                h1=hCD(1);
                fprintf(text_tiket,[ '           Arrow or height 1 of section
CD   | h1   | m   | ' num2str(h1) '\n']);
            elseif c==2
                h2=hCD(2);
                fprintf(text_tiket,[ '           Arrow or height 2 of section
CD   | h2   | m   | ' num2str(h2) '\n']);
            elseif c==3
                h3=hCD(3);
                fprintf(text_tiket,[ '           Arrow or height 3 of section
CD   | h3   | m   | ' num2str(h3) '\n']);
            elseif c==4

```

```

        h4=hCD(4);
        fprintf(text_tiket,[ '           Arrow or height 4 of section
CD | h4 | m | ' num2str(h4) '\n']);
        elseif c==5
            h5=hCD(5);
            fprintf(text_tiket,[ '           Arrow or height 5 of section
CD | h5 | m | ' num2str(h5) '\n']);
        elseif c==6
            h6=hCD(6);
            fprintf(text_tiket,[ '           Arrow or height 6 of section
CD | h6 | m | ' num2str(h6) '\n']);
            elseif c==7
                h7=hCD(7);
                fprintf(text_tiket,[ '           Arrow or height 7 of section
CD | h7 | m | ' num2str(h7) '\n']);
            elseif c==8
                h8=hCD(8);
                fprintf(text_tiket,[ '           Arrow or height 8 of section
CD | h8 | m | ' num2str(h8) '\n']);
            elseif c==9
                h9=hCD(9);
                fprintf(text_tiket,[ '           Arrow or height 9 of section
CD | h9 | m | ' num2str(h9) '\n']);
            elseif c==10
                h10=hCD(10);
                fprintf(text_tiket,[ '           Arrow or height 10 of section
CD | h10 | m | ' num2str(h10) '\n']);
            end
        end
        fprintf(text_tiket,[ '           Vain between C y D | L
| m | ' num2str(LCD) '\n']);
        fprintf(text_tiket,[ '           Cable length | S
| m | ' num2str(SCD) '\n']);
        fprintf(text_tiket, '\n');
        fprintf(text_tiket,
'_____ \n'
);
        fprintf(text_tiket,
'_____ \n'
);
    fclose(text_tiket);

    %% otras
    %% _DATOS DE SALIDA PARA LA WEB
    text_tiket=fopen([ruta nameFileExperimento],'wt');
    text_tiket=fopen([ruta nameFileExperimento],'a');
    fprintf(text_tiket,[ 'DATOS DE SALIDA \n']);
    fprintf(text_tiket,[ 'A ' num2str(TA) '\n']);
    fprintf(text_tiket,[ 'Ax ' num2str(TAx) '\n']);
    fprintf(text_tiket,[ 'Ay ' num2str(TAy) '\n']);
    fprintf(text_tiket,[ 'B ' num2str(TB) '\n']);
    fprintf(text_tiket,[ 'Bx ' num2str(TBx) '\n']);
    fprintf(text_tiket,[ 'By ' num2str(TBy) '\n']);
    fprintf(text_tiket,[ 'TmaxAB ' num2str(TABmax) '\n']);
    fprintf(text_tiket,[ 'TminAB ' num2str(TABmin) '\n']);
    fprintf(text_tiket,[ 'LAB ' num2str(LAB) '\n']);
    fprintf(text_tiket,[ 'SAB ' num2str(SAB) '\n']);
    fprintf(text_tiket,[ 'Posición Inicial del cable \n']);
    fprintf(text_tiket,[ 'x' ' ']);
    dlmwrite([ruta nameFileExperimento],x,'-append','roffset', 0,
'delimiter', ' ', 'precision','%4f')
    fprintf(text_tiket,['y' ' ']);
    dlmwrite([ruta nameFileExperimento],y,'-append','roffset', 0, 'delimiter',
' ', 'precision','%4f')
    fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
    fprintf(text_tiket,[ 'La deformada del Cable \n']);
    fprintf(text_tiket,[ 'xx' ' ']);

```



```

        dlmwrite([ruta nameFileExperimento], (xx(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
        fprintf(text_tiket, [ 'yy' ' ']);
        dlmwrite([ruta nameFileExperimento], (yy(1,:)), '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
        %fprintf(text_tiket, [ 'Posición Inicial del Cable - Carga Distribuida-
DEformada del Cable\n']);
        %Tensiones
        TAB=TAB';
        fprintf(text_tiket, ['TAB' ' ']);
        dlmwrite([ruta nameFileExperimento], TAB, '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
        %flechas
        hAB=hAB';
        fprintf(text_tiket, ['hAB' ' ']);
        dlmwrite([ruta nameFileExperimento], hAB, '-append', 'roffset', 0,
'delimiter', ' ', 'precision', '%.4f')
        fprintf(text_tiket, [ 'C ' num2str(TC) '\n']);
        fprintf(text_tiket, [ 'Cx ' num2str(TCx) '\n']);
        fprintf(text_tiket, [ 'Cy ' num2str(TCy) '\n']);
        fprintf(text_tiket, [ 'TmaxBC ' num2str(TBCmax) '\n']);
        fprintf(text_tiket, [ 'TminBC ' num2str(TBCmin) '\n']);
        fprintf(text_tiket, [ 'LBC ' num2str(LBC) '\n']);
        fprintf(text_tiket, [ 'SBC ' num2str(SBC) '\n']);
        fprintf(text_tiket, [ 'D ' num2str(TD) '\n']);
        fprintf(text_tiket, [ 'Dx ' num2str(TDx) '\n']);
        fprintf(text_tiket, [ 'Dy ' num2str(TDy) '\n']);
        fprintf(text_tiket, [ 'TmaxCD ' num2str(TCDmax) '\n']);
        fprintf(text_tiket, [ 'TminCD ' num2str(TCDmin) '\n']);
        fprintf(text_tiket, [ 'LCD ' num2str(LCD) '\n']);
        fprintf(text_tiket, [ 'SCD ' num2str(SCD) '\n']);
        fclose(text_tiket);
end

```

Figura 84. Código del experimento 3 – Función “SalidaCablesCargasPuntuales3”

Fuente: Autor

A4) Códigos del experimento 4 – Cable Catenaria Elástica.

[illegible]

Figura 85. Código del experimento 4 – Función “LeerCatElas2”

Fuente: Autor

```

function [r7,r13,r15,XG,ZG,RCR,T,str]= CatElas2 (NTra)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CATENARIA ELÁSTICA
%
%La presente herramienta plantea la resolución de un problema clásico de la
%teoría de cables, conocido como PROBLEMA DE LA CATENARIA ELÁSTICA. el
%enfoque utilizado es el propuesto por IRVINE (1992)
%
%Datos de Entrada:
% l : Distancia Horizontal entre Punto Inicial y Punto Final (m)
% h : Diferencia de nivel entre los apoyos (m)
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m)
% Ao: Es el Área en la geometría original del cable (m^2)
% E : es el módulo de Young del cable (KN/m^2)
% W : es el peso total del cable (KN)
%
%Datos de Salida:
% Ho: es la Reacción Horizontal (KN)
% Vo: es la Reacción vertical (KN)
% T : Tensiones a lo largo del cable (KN)
% x : Coordenada x de la geometría original (KN)
% Z : Coordenada Z de la geometría original (KN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%TRAMO 1
a1=importdata(['Tramos ',num2str(1),'.mat'],'Tram01');
NFP=a1(3,1);
r8=NFP*2;
r9=7+r8;
r7=ones(r9,1);
r13=ones(12,1);
r15=ones(3,1);
k=0;
k4=0;
l1=0;
hh=0;
for r = 1:NTra
    rt=r;
    %DATOS DE ENTRADA
    %
    a1=importdata(['Tramos ',num2str(r),'.mat'],['Tramo',num2str(r)]);
    l=a1(1,1);
    h=a1(1,2);
    Lo=a1(1,3);
    Ao=a1(2,1);
    E=a1(2,2);
    Wo=a1(2,3);
    NFP=a1(3,1);
    FP=a1(4,1:NFP);
    SF=a1(5,1:NFP);
    %ANALISIS => FUNCIONES PARA UN TRAMO
    %
    %paso1:convertir la carga distribuida de KN/m a KN
    W=(Wo*Lo);
    %paso2: Determinar Ho (Tensión Horizontal) y Vo (Tensión Vertical); KN
    [XH,OV,BW,YN,SN]= Matriz3 (l,h,Lo,Ao,E,W,NFP,FP,SF);
    %paso3: Determinar las Tensiones a lo largo del cable
    [T,OS,O,F,Hf,Ho,Vo,Vf,Tmax,Tmin,Tp,YN,SNN,YNN]= TensionCE
    (W,XH,OV,YN,SN,NFP);
    %paso4: Determinar las coordenadas x de la geometría original
    [X,TTT]= CoorxCE (XH,BW,OV,OS,NFP,Lo,YN,SNN,YNN);
    %paso5: Determinar las coordenadas z de la geometría deformada
    [Z,S]= CoorzCE (Lo,OS,TTT,BW,OV,XH,SN,YN,NFP,SNN);
    A=[S,X,Z,T];
    B=round(S*100)/100;
    %paso6: Resultados del punto más bajo
    [Tp,Sp,Xp,Zp]= PuntoMasBajoC (A,T,Tp);
    %PASO7: Resultados

```

```

SF=[0,SF];
SF=SF';
FP=[0,FP];
%Paso8: RESULTADOS CON RESPESTO A LA Cargas #1... #N
[RC]= CARGA1 (B,A,SF,NFP,FP);
%Paso9
T=T';
%RESULTADOS
%
FP=FP';
SF(1,:)=[];
FP(1,:)=[];
%Matriz de Datos de Entrada
r1=[1;h];
r3=[Lo;Ao;E;Wo];
r4=NFP;
u=NFP*2;
r5=ones(u,1);
for v=1:u
    if v<NFP
        vv=v;
        v1=FP(vv);
        r5(v)=v1;
    elseif v==NFP
        vv=v;
        v1=FP(vv);
        r5(v)=v1;
    else
        vv=v-NFP;
        v1=SF(vv);
        r5(v)=v1;
    end
    r5(v)=v1;
end
r6=[r1;r3;r4;r5];
k1=size(r6);
k1=k1(1);
k2=k1-k;
if r>1
    if k2>0
        k3=zeros(k2,k4);
        r7=[r7;k3];
    else
        k2=abs(k2);
        k3=zeros(k2,1);
        r6=[r6;k3];
    end
end
r7=[r7,r6]; %Matriz de datos de entrada
k=size(r7);
k4=k(2);
k=k(1);
%Matriz de Datos de Salida
r10=[Ho;Vo;O;Hf; Vf;F;Tmax;Tmin];
r11=[Tp;Sp;Xp;Zp];
r12=[r10;r11];
r13=[r13 r12]; %Matriz de datos de salida
%Matriz de Reacciones
if r==1
    w1=Ho;
    w2=Vo;
    w3=O;
    w4=Hf;
    w5=Vf;
    w6=F ;
    r14=[w1 w4; w2 w5; w3 w6];
    XG=X;

```

```

        ZG=-Z;
        TG=T;
        hhh=-h;
        RCR=RC;
    else
        v=r+1;
        r15(:,v)=[];
        w1=Ho+w4;
        w2=Vo+w5;
        w3=sqrt(w1^2+w2^2);
        w4=Hf;
        w5=Vf;
        w6=F ;
        r14=[w1 w4; w2 w5; w3 w6];
        X=X+l1;
        XG=[XG;X];
        Z=-Z+hhh;
        ZG=[ZG;Z];
        hhh=hhh-h;
        TG=[TG T];
        RCR=[RCR;RC];
    end
    r15=[r15 r14];
    l1=l+l1;
    hh=h+hh;
end
r7(:,1)=[];
r13(:,1)=[];
r15(:,1)=[];
KKKK=[XG ZG]
XG=XG';
ZG=ZG';
plot(XG,ZG)
    % fecha y hora
    str = datestr(now);
end

```

Figura 86. Código del experimento 4 – Función “CatElas2”
Fuente: Autor

```

function [] = SalidaCatElas2(NTra,r7,r13,r15,XG,ZG,RCR,T,str,ruta,
nameFileUser, nameFileExperimento,nameFileUserEnglish)
%%
text_tiket=fopen([ruta nameFileUser],'wt');
fprintf(text_tiket,
'
');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA
(VLEE) \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, '          GRUPO DE INVESTIGACIÓN DE INGENIERÍA SÍSMICA Y
SISMOLOGIA (GRISS-UTPL)\n');
fprintf(text_tiket, '          DEPARTAMENTO DE GEOLOGÍA Y MINAS, E INGENIERÍA
CIVIL \n');
fprintf(text_tiket, '          SECCIÓN DE ESTRUCTURAS, TRANSPORTE Y
CONSTRUCCIÓN\n');
fprintf(text_tiket, '          UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, '          -----
-----\n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'LABORATORIO: CABLES \n');
fprintf(text_tiket, 'EXPERIMENTO: Catenaria Elástica\n');
fprintf(text_tiket, ['FECHA      : ' num2str(str) '\n']);
fprintf(text_tiket, 'Autores      : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
c=0;
for t=1:NTra
fprintf(text_tiket, [ '          TRAMO ' num2str(t) ' \n']);
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          DESCRIPTION          |SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'          |          |          |          \n');
);
fprintf(text_tiket, [ 'Distancia Horizontal entre los puntos(O) - (F) | 1 |
m | ' num2str(r7(1,t)) '\n']);
fprintf(text_tiket, [ 'Diferencia de nivel entre los puntos (O) - (F) | h |
m | ' num2str(r7(2,t)) '\n']);
fprintf(text_tiket, 'Propiedades del Cable\n');
fprintf(text_tiket, [ 'Longitud inicial entre los puntos (O)-(F) | Lo |
m | ' num2str(r7(3,t)) '\n']);
fprintf(text_tiket, [ 'Área de la geometría original | Ao |
m^2 | ' num2str(r7(4,t)) '\n']);
fprintf(text_tiket, [ 'Módulo de young | E |
KN/m^2 | ' num2str(r7(5,t)) '\n']);
fprintf(text_tiket, [ 'Peso por unidad de longitud | Wo |
KN/m | ' num2str(r7(6,t)) '\n']);

```

```

fprintf(text_tiket, 'Cargas\n');
fprintf(text_tiket, [ '          Número de cargas                                | NFP |
#          | ' num2str(r7(7,t)) '\n']];
b=r7(7,t);
for r= 1:b
    u=7+r;
    D=r7(u,t);
    fprintf(text_tiket, [ '          Carga ' num2str(r) '
| FP | KN | ' num2str(D) '\n']);
end
for r= 1:b
    u=7+r7(7,t)+r;
    E=r7(u,t);
    fprintf(text_tiket, [ '          Posición de la carga ' num2str(r) ' a lo largo
- cable | SF | m | ' num2str(E) '\n']);
end
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          \n');
fprintf(text_tiket, '          DESCRIPTION                                | SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
'          |          |          |          \n'
);
fprintf(text_tiket, [ 'Reacción horizontal del punto      (O)                | Ho    |
KN          | ' num2str(r13(1,t)) '\n']);
fprintf(text_tiket, [ 'Reacción vertical del punto        (O)                | Vo    |
KN          | ' num2str(r13(2,t)) '\n']);
fprintf(text_tiket, [ 'Reacción en el punto      (O)                | O     |
KN          | ' num2str(r13(3,t)) '\n']);
fprintf(text_tiket, [ 'Reacción horizontal del punto      (F)                | Hf    |
KN          | ' num2str(r13(4,t)) '\n']);
fprintf(text_tiket, [ 'Reacción vertical del punto        (F)                | Vf    |
KN          | ' num2str(r13(5,t)) '\n']);
fprintf(text_tiket, [ 'Reacción en el punto      (F)                | F     |
KN          | ' num2str(r13(6,t)) '\n']);
fprintf(text_tiket, [ 'Tensión máxima del cable                                | Tmax  |
KN          | ' num2str(r13(7,t)) '\n']);
fprintf(text_tiket, [ 'Tensión mínima del cable                                | Tmin  |
KN          | ' num2str(r13(8,t)) '\n']);
fprintf(text_tiket, 'Resultados del punto más bajo\n');
fprintf(text_tiket, [ '          Tensión                                | Tp    |
KN          | ' num2str(r13(9,t)) '\n']);
fprintf(text_tiket, [ '          Abscisa a lo largo del cable                | Sp    |
m          | ' num2str(r13(10,t)) '\n']);
fprintf(text_tiket, [ '          Coordenada en el eje x                                | Xp    |
m          | ' num2str(r13(11,t)) '\n']);
fprintf(text_tiket, [ '          Coordenada en el eje z                                | Zp    |
m          | ' num2str(r13(12,t)) '\n']);
fprintf(text_tiket, 'Resultados donde se localiza la carga\n');
fprintf(text_tiket, '
\n');
fprintf(text_tiket, '          Carga |          S          |          X(s)          |          Z(s)          |
Tensión\n');
fprintf(text_tiket, '          [KN] |          [m]          |          [m]          |          [m]          |
[KN/m] \n');
fprintf(text_tiket, '
|          |          |          |          \n');
bb=b+1;
for r= 1:bb
    d=r+c;
    F=RCR(d,:);
    fprintf(text_tiket, [ '          ' num2str(F) '          \n' ]]);
end

```

```

c=d;
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'
');
);
fprintf(text_tiket,
'
');
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
end
    fprintf(text_tiket, '
                                REACCIONES EN LOS PUNTOS
\n');
    fprintf(text_tiket, '\n');
    t=NTra+1;
    for r=1:t
        fprintf(text_tiket, [ '
                                Punto      ' num2str(r) '\n']);
        fprintf(text_tiket, '
                                \n');
        fprintf(text_tiket, [ 'Reacción - proyección X      | KN | ' num2str(r15(1,r))
\n']);
        fprintf(text_tiket, [ 'Reacción - proyección Y      | KN | ' num2str(r15(2,r))
\n']);
        fprintf(text_tiket, [ 'Reacción                      | KN | ' num2str(r15(3,r))
\n']);
    end
    fprintf(text_tiket, '\n');
    fprintf(text_tiket,
'
\n');
);
fprintf(text_tiket,
'
\n');
);
fclose(text_tiket);
%%
text_tiket=fopen([ruta nameFileUserEnglish], 'wt');
fprintf(text_tiket,
'
\n');
);
fprintf(text_tiket,
'
\n');
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
                                VIRTUAL LABORATORY FOR EARTHQUAKE ENGINEERING
(VLEE) \n');
fprintf(text_tiket, '\n');
fprintf(text_tiket, '
                                SEISMIC ENGINEERING AND SEISMOLOGY RESEARCH GROUP
(GRISS-UTPL) \n');
fprintf(text_tiket, '
                                DEPARTAMENT OF GEOLOGY AND MINES AND CIVIL
ENGINEERING \n');
fprintf(text_tiket, '
                                SECTION OF STRUCTURES, TRANSPORT AND
CONSTRUCTION \n');
fprintf(text_tiket, '
                                UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
(UTPL) \n');
fprintf(text_tiket, '
                                -----
\n');
fprintf(text_tiket, '
\n');
fprintf(text_tiket, '
\n');
fprintf(text_tiket, 'LABORATORY:      CABLES \n');
fprintf(text_tiket, 'EXPERIMENT:      CATENARY ELASTIC \n');
fprintf(text_tiket, [ 'DATE          :      ' num2str(str) '\n']);
fprintf(text_tiket, 'AUTHORS       : Msc. Duque,E., Ing. Quiñonez,S. Pélaez,D.,
\n');
fprintf(text_tiket,
'
\n');
);
fprintf(text_tiket,

```



```

' _____ \n'
);
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
c=0;
for t=1:NTra
fprintf(text_tiket,[ ' _____ Stretch ' num2str(t) '
\n']);
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'INPUT DATA \n');
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
' _____ | _____ | _____ \n'
);
fprintf(text_tiket,[ 'Horizontal distance between points (O) and (F) | l |
m | ' num2str(r7(1,t)) '\n']);
fprintf(text_tiket,[ 'Difference in level between points (O) and (F) | h |
m | ' num2str(r7(2,t)) '\n']);
fprintf(text_tiket, 'Cable Properties\n');
fprintf(text_tiket,[ ' Initial length between points (O) and (F) | Lo |
m | ' num2str(r7(3,t)) '\n']);
fprintf(text_tiket,[ ' Area in the original geometry | Ao |
m^2 | ' num2str(r7(4,t)) '\n']);
fprintf(text_tiket,[ ' Young Module | E |
KN/m^2 | ' num2str(r7(5,t)) '\n']);
fprintf(text_tiket,[ ' Weight per unit length | Wo |
KN/m | ' num2str(r7(6,t)) '\n']);
fprintf(text_tiket, 'Loads\n');
fprintf(text_tiket,[ ' Number of loads | NFP |
# | ' num2str(r7(7,t)) '\n']);
b=r7(7,t);
for r= 1:b
u=7+r;
D=r7(u,t);
fprintf(text_tiket,[ ' Load ' num2str(r) '
| FP | KN | ' num2str(D) '\n']);
end
for r= 1:b
u=7+r7(7,t)+r;
E=r7(u,t);
fprintf(text_tiket,[ ' Position of the load ' num2str(r) ' on the
Length-Cable| SF | m | ' num2str(E) '\n']);
end
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, 'ANALYSIS OUPUT \n');
fprintf(text_tiket, ' _____ \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' \n');
fprintf(text_tiket, ' DESCRIPTION |SYMBOL|
UNITY | MAGNITUDE\n');
fprintf(text_tiket,
' _____ | _____ | _____ \n'
);
fprintf(text_tiket,[ 'Horizontal reaction at the point (O) | Ho |
KN | ' num2str(r13(1,t)) '\n']);
fprintf(text_tiket,[ 'Vertical reaction at the point (O) | Vo |
KN | ' num2str(r13(2,t)) '\n']);
fprintf(text_tiket,[ 'Reaction at the point (O) | O |
KN | ' num2str(r13(3,t)) '\n']);
fprintf(text_tiket,[ 'Horizontal reaction at the point (F) | Hf |

```

```

KN      | ' num2str(r13(4,t)) '\n']);
fprintf(text_tiket,[ 'Vertical reaction at the point (F)          | Vf      |
KN      | ' num2str(r13(5,t)) '\n']);
fprintf(text_tiket,[ 'Reaction at the point (F)                  | F        |
KN      | ' num2str(r13(6,t)) '\n']);
fprintf(text_tiket,[ 'Maximun cable tension                      | Tmax     |
KN      | ' num2str(r13(7,t)) '\n']);
fprintf(text_tiket,[ 'Minimun cable tension                      | Tmin     |
KN      | ' num2str(r13(8,t)) '\n']);
fprintf(text_tiket, 'Lowest point results\n');
fprintf(text_tiket,[ '      Tension                                | Tp       |
KN      | ' num2str(r13(9,t)) '\n']);
fprintf(text_tiket,[ '      Abscissa along the cable                          | Sp       |
m       | ' num2str(r13(10,t)) '\n']);
fprintf(text_tiket,[ '      Coordinate on the X axis                          | Xp       |
m       | ' num2str(r13(11,t)) '\n']);
fprintf(text_tiket,[ '      Coordinate on the Z axis                          | Zp       |
m       | ' num2str(r13(12,t)) '\n']);
fprintf(text_tiket, 'Results of the point where the load is located\n');
fprintf(text_tiket, '
\n');
fprintf(text_tiket, '      Load      |      S      |      X(s)      |      Z(s)      |
Tension\n');
fprintf(text_tiket, '      [KN]      |      [m]      |      [m]      |      [m]      |
[KN/m] \n');
fprintf(text_tiket, '
_____|_____|_____|_____ \n');
bb=b+1;
for r= 1:bb
    d=r+c;
    F=RCR(d,:);

    fprintf(text_tiket,[ '      ' num2str(F) ' \n' ]);
end
c=d;
fprintf(text_tiket, '\n');
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket,
'_____ \n'
);
fprintf(text_tiket, '\n');
fprintf(text_tiket, '\n');
end
    fprintf(text_tiket, '
REACCIONES EN LOS PUNTOS
\n');
    fprintf(text_tiket, '\n');

t=NTra+1;
for r=1:t

    fprintf(text_tiket,[ '      Punto      ' num2str(r) '\n']);
    fprintf(text_tiket, '_____ \n');
    fprintf(text_tiket,[ 'Reaction in projection X      | KN | ' num2str(r15(1,r))
\n']);
    fprintf(text_tiket,[ 'Reaction in projection Y      | KN | ' num2str(r15(2,r))
\n']);
    fprintf(text_tiket,[ 'Reaction                        | KN | ' num2str(r15(3,r))
\n']);
end
    fprintf(text_tiket, '\n');
    fprintf(text_tiket,
'_____ \n'
);
    fprintf(text_tiket,
'_____ \n'
);

```

```

fclose(text_tiket);
%%%_DATOS DE SALIDA PARA LA WEB
text_tiket=fopen([ruta nameFileExperimento],'wt');
text_tiket=fopen([ruta nameFileExperimento],'a');
%%fprintf(text_tiket,[ 'DATOS DE SALIDA \n']);
for r=1:NTra
    fprintf(text_tiket,[ 'Tramo ' num2str(r) ' \n' ]);
    fprintf(text_tiket,[ 'Ho ' num2str(r13(1,r)) '\n']);
    fprintf(text_tiket,[ 'Vo ' num2str(r13(2,r)) '\n']);
    fprintf(text_tiket,[ 'O ' num2str(r13(3,r)) '\n']);
    fprintf(text_tiket,[ 'Hf ' num2str(r13(4,r)) '\n']);
    fprintf(text_tiket,[ 'Vf ' num2str(r13(5,r)) '\n']);
    fprintf(text_tiket,[ 'F ' num2str(r13(6,r)) '\n']);
    fprintf(text_tiket,[ 'Tmax ' num2str(r13(7,r)) '\n']);
    fprintf(text_tiket,[ 'Tmin ' num2str(r13(8,r)) '\n']);
    fprintf(text_tiket,[ 'Tp ' num2str(r13(9,r)) '\n']);
    fprintf(text_tiket,[ 'Sp ' num2str(r13(10,r)) '\n']);
    fprintf(text_tiket,[ 'Xp ' num2str(r13(11,r)) '\n']);
    fprintf(text_tiket,[ 'Zp ' num2str(r13(12,r)) '\n']);
end
fprintf(text_tiket, 'Para grafica ' '\n' );
fprintf(text_tiket,[ 'CARGA DISTRIBUIDA \n']);
fprintf(text_tiket,[ 'La deformada del Cable \n']);
fprintf(text_tiket,[ 'XG' ' ']);
dlmwrite([ruta nameFileExperimento],(XG(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
fprintf(text_tiket,[ 'ZG' ' ']);
dlmwrite([ruta nameFileExperimento],(ZG(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
fprintf(text_tiket,[ 'T' ' ']);
dlmwrite([ruta nameFileExperimento],(T(1,:)),'-append','roffset', 0,
'delimiter', ' ', 'precision','%.4f')
fprintf(text_tiket,[ 'Posición Inicial del Cable - Carga Distribuida-
DEformada del Cable\n']);
fclose(text_tiket);
end

```

Figura 87. Código del experimento 4– Función “SalidaCatElas2”

Fuente: Autor

```

function [XH,OV,BW,YN,SN]= Matriz3 (l,h,Lo,Ao,E,W,NFP,FP,SF)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLANTEAMIENTO DE LA SOLUCIÓN NÚMERICA
%
%El desarrollo de la solución se presenta tres fases, inicialmente se hace %
%el planteamiento del método de Newton-Raphson, adaptado al problema de la%
%catenaria del cable, después se describe la solución explícita del sist. %
%lineal de ecuaciones encontrado en la formulación del método iterativo, y%
%finalmente se establecen los criterios de convergencia necesarios para %
%determinar el fin de este proceso.
%
%Datos de Entrada:
% l : Distancia Horizontal entre Punto Inicial y Punto Final (m) %
% h : Diferencia de nivel entre los apoyos (m) %
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m) %
% Ao: Es el Área en la geometría original del cable (m^2) %
% E : es el módulo de Young del cable (KN/m^2) %
% W : es el peso total del cable (KN) %
%Datos de Salida:
% Ho: es la Reacción Horizontal (KN) %
% Vo: es la Reacción vertical (KN) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLANTEAMIENTO DEL MÉTODO DE NEWTON-RAPHSON, ADAPTADO AL PROBLEMA DE LA
%CATENARIA
%pasol: determinar X0 - valores iniciales
[Ho,Vo]= Matriz1 (l,Lo,W);
%pasol2: magnitudes adimensionales
[SN,Sh,RL,XH,OV,YN,BW,YYN]= MAGADIMEN (l,h,Lo,Ao,E,W,FP,SF,Ho,Vo);
X0=[XH;OV];
%SOLUCIÓN EXPLÍCITA DEL SISTEMA DE ECUACIONES LINEALES DEL ALGORITMO DE
%NEWTON-RAPHSON
%pasol2: determinar X1 - valores de la variación de Ho y Vo
[VarHo,VarVo,b,d]= Matriz2 (NFP,SN,Sh,RL,XH,OV,YN,BW,YYN);
X1=[VarHo;VarVo];
%CRITERIOS DE CONVERGENCIA DE LA SOLUCIÓN
%pasol3: determinar X
X=X0+X1;
%pasol4:criterios de convergencia
[ex]= tolX (X0,X);
ex1=ex(1);
ex2=ex(2);
[ef]= tolF (b,RL,d,Sh);
ef1=ef(1);
ef2=ef(2);
%MÉTODO ITERATIVO
chile ex1>ef1
X0=X;
XH=X0(1);
OV=X0(2);
[VarHo,VarVo,b,d]= Matriz2 (NFP,SN,Sh,RL,XH,OV,YN,BW,YYN);
X1=[VarHo;VarVo];
X=X0+X1;
[ex]= tolX (X0,X);
ex1=ex(1);
ex2=ex(2);
[ef]= tolF (b,RL,d,Sh);
ef1=ef(1);
ef2=ef(2);
end
while ex2>ef2
X0=X;
XH=X0(1);
OV=X0(2);
[VarHo,VarVo,b,d]= Matriz2 (NFP,SN,Sh,RL,XH,OV,YN,BW,YYN);
X1=[VarHo;VarVo];

```

```

        X=X0+X1;
        [ex]= tolx (X0,X);
        ex2=ex(2);
        [ef]= tolf (b,RL,d,Sh);
        ef2=ef(2);
    end
    X0=X;
    XH=X0(1);
    OV=X0(2);
end

```

Figura 88. Código del experimento 4 – Función “Matriz3”
Fuente: Autor

```

function [VarHo,VarVo,b,d]= Matriz2 (NFP,SN,Sh,RL,XH,OV,YN,BW,YYN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Para resolver el planteamiento, la solución explícita del sistema de %
%ecuaciones lineales del algoritmo de Newton-Raphson. %
% %
%A continuación, se presenta el desarrollo para las expresiones de VarHo y%
%varVo siguiendo el procedimiento planteado anteriormente %
% %
%Datos de Entrada: %
% l : Distancia Horizontal entre Punto Inicial y Punto Final (m) %
% h : Diferencia de nivel entre los apoyos (m) %
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m) %
% Ao: Es el Área en la geometría original del cable (m^2) %
% E : es el módulo de Young del cable (KN/m^2) %
% W : es el peso total del cable (KN) %
% Ho: es la Reacción Horizontal (KN) %
% Vo: es la Reacción vertical (KN) %
% %
%Datos de Salida %
% VarHo: es la Variación de Ho %
% VarVo: es la variación de Vo %
% b : es la variable con respecto f(Ho,Vo) %
% d : es la variable con respecto g(Ho,Vo) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Segunda Formulación
%% Determinación de VarVo
%Variable a => corresponde dg/dx
%a1:primer componente
a1=1/sqrt((XH)^2+(YYN-OV+1)^2);
%a2:segundo componente
a2=1/sqrt((XH)^2+(OV)^2);
%a3:tercer componente
a33=(XH^2);
drows=NFP+1;
dcol=1;
a3=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    a333=(SNi-OV+YNi)^2;
    a3(r)=1/sqrt(a33+a333);
end
a3=sum(a3);
%a4:cuarto componente
drows=NFP+1;
dcol=1;
a4=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        a4(r)=1/sqrt(a33+a444);
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        a4(r)=1/sqrt(a33+a444);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        a4(r)=1/sqrt(a33+a444);
    end
end
end

```

```

        a444=( (SNi-OV+YNi)^2);
        a4(r)=1/sqrt(a33+a444);
    end
end
a4=sum(a4);
%a5:quinto componente
a5=(a3-a4-a1+a2)*XH;
%a: es la variable dg/dH
a=a5;
%Variable b => corresponde f
%b1:primer componente
b1=(BW);
%b2:segundo componente
b2=(asinh(OV/XH));
%b3:tercer componente
b3=(asinh((OV-YYN-1)/XH));
%b4:cuarto componente
drows=NFP+1;
dcol=1;
b44=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    b444=( (OV-SNi-YNi)/XH);
    b44(r)=asinh(b444);
end
b4=sum(b44);
%b5:quinto componente
drows=NFP+1;
dcol=1;
b55=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        b555=( (OV-SNi-YNi)/XH);
        b55(r)=asinh(b555);
    elseif r==2
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        b555=( (OV-SNi-YNi)/XH);
        b55(r)=asinh(b555);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        b555=( (OV-SNi-YNi)/XH);
        b55(r)=asinh(b555);
    end
end
b5=sum(b55);
%b6:sexto componente
b6=(b4-b5);
%b7:quinto componente
b7=(XH*(b1+b2-b3+b6));
%b: es la variable f(Ho,Vo)
b=(b7);
%Variable c => corresponde df/dH
%c1:primer componente
c1=(BW);
%c2:segundo componente
c2=(asinh((YYN-OV+1)/XH));
%c3:tercer componente
drows=NFP+1;
dcol=1;
c33=ones(drows,dcol);

```

```

for r = 1:drows
    if r==1
        YNi=0;
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c33(r)=asinh(c333);
    elseif r==2
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c33(r)=asinh(c333);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c33(r)=asinh(c333);
    end
end
c3=sum(c33);
%c4:cuarto componente
drows=NFP+1;
dcol=1;
c44=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    c444=( (YNi-OV+SNi)/XH);
    c44(r)=asinh(c444);
end
c4=sum(c44);
%c5:quinto componente
c5=(c3-c4);
%c6:sexto componente
drows=NFP+1;
dcol=1;
c66=ones(drows,dcol);
for r = 1:drows
    if r==1
        YNi=0;
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c66(r)=(c333/sqrt(1+(c333)^2));
    elseif r==2
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c66(r)=(c333/sqrt(1+(c333)^2));
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        c333=( (YNi-OV+SNi)/XH);
        c66(r)=(c333/sqrt(1+(c333)^2));
    end
end
c6=sum(c66);
%c7:séptimo componente
drows=NFP+1;
dcol=1;
c77=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    c444=( (YNi-OV+SNi)/XH);

```



```

c77(r)=(c444/sqrt(1+(c444)^2));
end
c7=sum(c77);
%c8:octavo componente
c8=(c6-c7);
%c9:noveno componente
c9=((YYN-OV+1)/XH)/sqrt(1+((YYN-OV+1)/XH)^2);
%c10:decimo componente
c10=(OV/XH)/sqrt(1+(OV/XH)^2);
%c11:onceavo componente
c11=asinh(OV/XH);
%c:variable de df/dh
c=(c1+c2+c11+c5-(c8+c9+c10));
%Variable d => corresponde g
%d1:primer componente
d1=(BW*(OV-1/2));
%d2:segundo componente
d2=sqrt((XH)^2+(OV)^2);
%d3:tercer componente
d3=sqrt((XH^2)+((OV-YYN-1)^2));
%d4:cuarto componente
drows=NFP+1;
dcol=1;
d44=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    d444=BW*YNi*(SNi-1);
    d44(r)=d444;
end
d4=sum(d44);
%d5:quinto componente
drows=NFP+1;
dcol=1;
d55=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    d555=sqrt((XH^2)+(OV-YNi-SNi)^2);
    d55(r)=d555;
end
d5=sum(d55);
%d6:sexto componente
drows=NFP+1;
dcol=1;
d66=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        d666=(sqrt((XH^2)+(OV-YNi-SNi)^2));
        d66(r)=d666;
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        d666=(sqrt((XH^2)+(OV-YNi-SNi)^2));
        d66(r)=d666;
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        d666=(sqrt((XH^2)+(OV-YNi-SNi)^2));
        d66(r)=d666;
    end
    rr=r-1;
    YNi=YN(rr);

```

```

        SNi=SN(r);
        d666=(sqrt((XH^2)+(OV-YNi-SNi)^2));
        d66(r)=d666;
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        d666=(sqrt((XH^2)+(OV-YNi-SNi)^2));
        d66(r)=d666;
    end
end
d6=sum(d66);
    %d7:septimo componente
    d7=(d4+d5-d6);
    %d:variable de g(Ho,Vo)
d=(d1+d2-d3+d7);
%Variable e => corresponde df/dV
    %e1:primer componente
    e1=(1/(sqrt(1+((YYN-OV+SNi)/XH)^2)));
    %e2:segundo componente
    e2=(1/(sqrt(1+(OV/XH)^2)));
    %e3:tercer componente
    drows=NFP+1;
    dcol=1;
    e66=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        e666=(1/(sqrt(1+((YNi-OV+SNi)/XH)^2)));
        e66(r)=e666;
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        e666=(1/(sqrt(1+((YNi-OV+SNi)/XH)^2)));
        e66(r)=e666;
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        e666=(1/(sqrt(1+((YNi-OV+SNi)/XH)^2)));
        e66(r)=e666;
    end
end
e3=sum(e66);
    %e4:tercer componente
    drows=NFP+1;
    dcol=1;
    e44=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    e444=(1/(sqrt(1+((YNi-OV+SNi)/XH)^2)));
    e44(r)=e444;
end
e4=sum(e44);
    %e:variable de df/dV
e=(-(e1-e2+e3-e4));
%Variable f => corresponde dg/dV
    %f1:primer componente
    f1=BW;
    %f2:segundo componente
    drows=NFP+1;
    dcol=1;
    e444=(1/(sqrt(1+((YNi-OV+SNi)/XH)^2)));
    e44(r)=e444;

```

```

end
e4=sum(e44);
    %e:variable de df/dv
e=(-(e1-e2+e3-e4));
%Variable f => corresponde dg/dV
%f1:primer componente
    f1=BW;
    %f2:segundo componente
        drows=NFP+1;
        dcol=1;
        f2=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f2(r)=OV/sqrt(a33+a444);
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f2(r)=OV/sqrt(a33+a444);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f2(r)=OV/sqrt(a33+a444);
    end
end
f2=sum(f2);
    %f3:tercer componente
    f33=(YYN-OV+1);
    f3=(f33/sqrt((XH)^2+(f33)^2));
    %f4:cuarto componente
        drows=NFP+1;
        dcol=1;
        f4=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f4(r)=YNi/sqrt(a33+a444);
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f4(r)=YNi/sqrt(a33+a444);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=((SNi-OV+YNi)^2);
        f4(r)=YNi/sqrt(a33+a444);
    end
end
f4=sum(f4);
    %f5:quinto componente
    a33=(XH^2);
    drows=NFP+1;
    dcol=1;
    f5=ones(drows,dcol);
for r = 1:drows
    YNi=YN(rr);

```

```

        SNi=SN(r);
        a444=( (SNi-OV+YNi)^2);
        f4(r)=YNi/sqrt(a33+a444);
    end
end
f4=sum(f4);
%f5:quinto componente
a33=(XH^2);
drows=NFP+1;
dcol=1;
f5=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    a333=(SNi-OV+YNi)^2;
    f5(r)=SNi/sqrt(a33+a333);
end
f5=sum(f5);
%f6:sexto componente
a33=(XH^2);
drows=NFP+1;
dcol=1;
f6=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    a333=(SNi-OV+YNi)^2;
    f6(r)=YNi/sqrt(a33+a333);
end
f6=sum(f6);
%f7:septimo componente
a33=(XH^2);
drows=NFP+1;
dcol=1;
f7=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    a333=(SNi-OV+YNi)^2;
    f7(r)=OV/sqrt(a33+a333);
end
f7=sum(f7);
%f8:octavo componente
f8=OV/sqrt((XH)^2+(OV)^2);
%f9:noveno componente
drows=NFP+1;
dcol=1;
f9=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        a444=( (SNi-OV+YNi)^2);
        f9(r)=SNi/sqrt(a33+a444);
    elseif r==2;
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=( (SNi-OV+YNi)^2);
        f9(r)=SNi/sqrt(a33+a444);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        a444=( (SNi-OV+YNi)^2);
        f9(r)=SNi/sqrt(a33+a444);
    end
end

```

```

end
f9=sum(f9);
%fl0:decimo componente
f10=f1-f2+f3+f4-f5-f6+f7+f8+f9;
%f: variable de dg/dv
f=(f10);
%VarVo
VarVo=(a*b-a*RL-c*d+c*Sh)/(-e*a+c*f);
%% Determinación de VarHo
VarHo=(-b+RL-e*VarVo)/c;
end

```

Figura 89. Código del experimento 4 – Función “Matriz2”
Fuente: Autor

```

function [Ho,Vo]= Matriz1 (l,Lo,W)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Para el planteamiento del método de Newton-Raphson, adaptado al problema %
%de la catenaria Para el planteamiento del método de Nexton-Raphsom,      %
%adaptado al problema de la catenaria.                                     %
%                                                                           %
%El primer paso consiste en definir el vector de incógnita X,             %
%                                                                           %
% Ho: es la Reacción Horizontal      (KN)                                %
% Vo: es la Reacción vertical        (KN)                                %
%                                                                           %
%Seguidamente se toman valores de inicio de acuerdo a la versión del      %
%problema de catenaria planteada por Irvine en 1992. Esta versión propone %
%valores iniciales los siguientes:                                         %
%                                                                           %
%Datos de Entrada:                                                         %
% l : Distancia Horizontal entre Punto Inicial y Punto Final             (m) %
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final      (m) %
% W : es el peso total del cable                                           (KN) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Valor inicial para la Reacción Horizontal
Ho=( (W/(2*sqrt(6))) * (1/Lo)*sqrt(1/(Lo-1))) );
%Valor inicial para la Reacción Vertical
Vo=W/2;
end

```

Figura 90. Código del experimento 4 – Función “Matriz1”
Fuente: Autor

```

function [SN,Sh,RL,XH,OV,YN,BW,YYN]= MAGADIMEN (l,h,Lo,Ao,E,W,FP,SF,Ho,Vo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Con vistas a la exposición concisa de la solución y el conjunto de %
%ecuaciones para la determinación de los parámetros desconocidos, %
%introducimos ahora las siguientes magnitudes adimensionales %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Coordenada de carga Lagrangiana (n=1,2...,N)
SN=SF/Lo;
SN=[0,SN];
%Desplazamiento vertical relativo de los puntos extremos
Sh=h/Lo;
%Relación de aspecto del cable
RL=l/Lo;
%Reacción horizontal en soportes Po y P(N+1)
XH=Ho/W;
%Reacción vertical en apoyo Po
OV=Vo/W;
%Aplicaron cargas verticales concentradas (n=0,1...,N)
YN=FP/W;
YN=[0,YN];
%Factor de flexibilidad
BW=W/(E*Ao);
%Aplicación de cargas verticales concentradas - suma total de las cargas
YYN=(sum(FP))/W;
end

```

Figura 91. Código del experimento 4 – Función “MAGADIMEN”
Fuente: Autor

```

function [T,OS,O,F,Hf,Ho,Vo,Vf,Tmax,Tmin,Tp,YN,SNN,YNN]= TensionCE
(W,XH,OV,YN,SN,NFP)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tensión
%
%Determinar la Tensión T(s) del cable
%
%Datos de Entrada:
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m)
% W : es el peso total del cable (KN)
% Ho: es la Reacción Horizontal (KN)
% Vo: es la Reacción vertical (KN)
%
%Datos de Salida:
% T : Tensiones a lo largo del cable (KN)
% S : Coordenada del cable de la geometría original (m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DETERMINAR LA MATRIZ CORRESPONDIENTE A LA COORDENADA S
OS=0:0.01:1;
OS=[OS,SN];
OS=sort(OS);
OS=OS';
OS(1,:)=[];
drows=NFP+1;
dcol=1;
YNN=ones(drows,dcol);
for r = 1:drows
    if r==1
        YN2=YN(r);
        YNN(r)=YN2;
    else
        YN1=YN(r);
        YN2=YN1+YN2;
        YNN(r)=YN2;
    end
end
SNN=SN;
%DETERMINAR LA TENSIÓN DEL CABLE
SNN=SNN';
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
TTTU=ones(drows,dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=YNN(d);
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP,TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=YNN(d);
    end
end

```

```

        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTU=vertcat(TTTP,TTT);
    end
end
TTTU(1,:)=[];
TTT=TTTU;
TT=ones(drows,dcol);
for r = 1:drows
    SS=OS(r);
    TTTT=TTT(r);
    T1=(XH)^2;
    T2=((OV-TTTT-SS).^2);
    TT(r)=((T1+T2)^(1/2));
end
T=TT*W;
Ho=XH*W;
Vo=OV*W;
%DETERMINAR LAS REACCIONES EN EL PUNTO INICIAL Y PUNTO FINAL
%PUNTO INICIAL
%O=Reacción en el punto Inicial
O=T(1);
%PUNTO FINAL
F=T(u);
Hf=-Ho;
Vf=sqrt(F^2-Hf^2);
%DETERMINAR LA TENSIÓN MÁXIMA Y MINIMA A LO LARGO DEL CABLE
Tmax=max(T);
Tmin=min(T);
%A PARTIR DE LA TENSIÓN MÍNIMA : RESULTADOS DEL PUNTO MÁS BAJO
Tp=Tmin;
end

```

Figura 92. Código del experimento 1 – Función “TensionCE”

Fuente: Autor


```

function [RC]= CARGA1 (B,A,SF,NFP,FP)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CATENARIA ELÁSTICA
%
%La presente herramienta plantea la resolución de un problema clásico de la
%teoría de cables, conocido como PROBLEMA DE LA CATENARIA ELÁSTICA. el
%enfoque utilizado es el propuesto por IRVINE (1992)
%
%Datos de Entrada:
% l : Distancia Horizontal entre Punto Inicial y Punto Final (m)
% h : Diferencia de nivel entre los apoyos (m)
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m)
% Ao: Es el Área en la geometría original del cable (m^2)
% E : es el módulo de Young del cable (KN/m^2)
% W : es el peso total del cable (KN)
%
%Datos de Salida:
% Ho: es la Reacción Horizontal (KN)
% Vo: es la Reacción vertical (KN)
% T : Tensiones a lo largo del cable (KN)
% x : Coordenada x de la geometría original (KN)
% Z : Coordenada Z de la geometría original (KN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DETERMINAR LA COORDENADA S REFERENTE A LA CARGA #1
%DETERMINAR LA TENSION DEL CABLE EN LA CARGA #1
SF=SF';
FP=FP';
b=NFP+1;
drows=b;
dcol=4;
z61=ones(drows,dcol);
for r = 1:b
    SNNN=SF(r);
    [kk]=find(B(:,1) == SNNN);
    kkk=kk(1);
    TTTP=A(kkk,:);
    z61(r,:)=TTTP;
end
RC=z61;
RC=[FP,RC];
end

```

Figura 93. Código del experimento 4 – Función “Carga 1”
Fuente: Autor

```

function [X,TTT]= CoorxCE (XH,BW,OV,OS,NFP,Lo,YN,SNN,YNN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%COORDENADA X DE LA GEOMETRIA ORIGINAL
%
%Datos de Entrada:
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m) %
% W : es el peso total del cable (KN) %
% Ho: es la Reacción Horizontal (KN) %
% Vo: es la Reacción vertical (KN) %
% Ao: Es el Área en la geometría original del cable (m^2) %
% E : es el módulo de Young del cable (KN/m^2) %
% S : Coordenada del cable de la geometría original (m) %
%
%Datos de Salida:
% x : Coordenada x de la geometría horizontal (KN) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u=length(OS);
drows=u;
dcol=1;
x1=ones(drows,dcol);
for r=1:drows
    xx1=OS(r);
    xx2=BW*xx1;
    x1(r)=xx2;
end
x2=asinh(OV/XH);
drows=NFP+1;
dcol=1;
x5=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SNN(r);
    x5(r)=(asinh((OV-YNi-SNi)/XH));
end
drows=NFP+1;
dcol=1;
x55=ones(drows,dcol);
for r = 1:drows
    x551=x5(r);
    if r==1
        x555=x551;
        x5555=x555;
        x55(r)=x5555;
    else
        x555=x551;
        x5555=x555+x5555;
        x55(r)=x5555;
    end
end
x55(r)=x5555;
end
x5=x55;
x6=ones(drows,dcol);
for r = 1:drows
    if r==1
        YNi=0;
        SNi=SNN(r);
        x6(r)=(asinh((OV-YNi-SNi)/XH));
    elseif r==2
        rr=r-1;
        YNi=YN(rr);
        SNi=SNN(r);
        x6(r)=(asinh((OV-YNi-SNi)/XH));
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SNN(r);
        x6(r)=(asinh((OV-YNi-SNi)/XH));
    end
end

```

```

else
    rr=r-1;
    YNi=YN(rr);
    SNi=SNN(r);
    x6(r)=(asinh((OV-YNi-SNi)/XH));
end
x6(r)=(asinh((OV-YNi-SNi)/XH));
end
drows=NFP+1;
dcol=1;
x66=ones(drows,dcol);
for r = 1:drows
    x661=x6(r);
    if r==1
        x666=x661;
        x6666=x666;
        x66(r)=x6666;
    else
        x666=x661;
        x6666=x666+x6666;
        x66(r)=x6666;
    end
    x66(r)=x6666;
end
x6=x66;
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
x4=ones(drows,dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=(x5(d)-x6(d));
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP,TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=(x5(d)-x6(d));
        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        x4=vertcat(TTTP,TTT);
    end
end
x4(1,:)=[];
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;

```

```

TTTU=ones(drows, dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=YNN(d);
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4, dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP, TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=YNN(d);
        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4, dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP, TTT);
    end
end
TTTU(1, :)=[];
TTT=TTTU;
u=length(OS);
drows=u;
dcol=1;
x3=ones(drows, dcol);
for r=1:drows
    x7=TTT(r);
    x8=OS(r);
    x9=asinh((OV-x7-x8)/XH);
    x3(r)=x9;
end
u=length(OS);
drows=u;
dcol=1;
x11=ones(drows, dcol);
for r=1:drows
    xx1=x1(r);
    xx3=x3(r);
    xx4=x4(r);
    x10=(xx1+x2-xx3+xx4)*XH;
    x11(r)=x10;
end
XXX=x11;
X=XXX*Lo;
end

```

Figura 94. Código del experimento 4 – Función “CoorxCE”
Fuente: Autor

```

function [X]= GraficasCE (X,Z)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GRÁFICAS
%
%Datos de Entrada:
% T : Tensiones a lo largo del cable (KN)
% x : Coordenada x de la geometría original (KN)
% Z : Coordenada Z de la geometría original (KN)
%
%Datos de Salida:
% x : Coordenada x de la geometría horizontal (KN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y1 = -Z;
%y2 = -T;
figure
plot(X,y1)
%,X,y2,'--'
end

```

Figura 95. Código del experimento 4– Función “GraficaCE”
Fuente: Autor

```

function [Z,S]= CoorzCE (Lo,OS,TTT,BW,OV,XH,SN,YN,NFP,SNN)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%COORDENADA X DE LA GEOMETRIA ORIGINAL
%
%Datos de Entrada:
% Lo: Longitud inicial del cable entre Punto Inicial y Punto Final (m) %
% W : es el peso total del cable (KN) %
% Ho: es la Reacción Horizontal (KN) %
% Vo: es la Reacción vertical (KN) %
% Ao: Es el Área en la geometría original del cable (m^2) %
% E : es el módulo de Young del cable (KN/m^2) %
% S : Coordenada del cable de la geometría original (m) %
%
%Datos de Salida:
% z : Coordenada z de la geometría horizontal (KN) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%primera variable
OS=OS';
u=length(OS);
drows=u;
dcol=1;
z3=ones(drows,dcol);
for r=1:drows
    ZZ=OS(r);
    z1=BW*ZZ;
    z2=(OV-ZZ/2);
    z33=z1*z2;
    z3(r)=z33;
end
%segunda variable
z4=sqrt((OV)^2+(XH)^2);
%tercera variable
drows=u;
dcol=1;
z5=ones(drows,dcol);
for r = 1:drows
    z55=TTT(r);
    z66=OS(r);
    z5(r)=sqrt((XH)^2+(OV-z55-z66)^2);
end
%cuarta variable
%cuarta variable: primera
drows=NFP+1;
dcol=1;
z6=ones(drows,dcol);
for r = 1:drows
    if r==1;
        YNi=0;
        SNi=SN(r);
        z6(r)=sqrt(XH^2+(OV-YNi-SNi)^2);
    else
        rr=r-1;
        YNi=YN(rr);
        SNi=SN(r);
        z6(r)=sqrt(XH^2+(OV-YNi-SNi)^2);
    end
end
drows=NFP+1;
dcol=1;
z66=ones(drows,dcol);
for r = 1:drows
    z661=z6(r);
    if r==1
        z666=z661;
        z6666=z666;
        z66(r)=z6666;
    else

```

```

        z666=z661;
        z6666=z666+z6666;
        z66(r)=z6666;
    end
    z66(r)=z6666;
end
z6=z66;
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
z61=ones(drows,dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=(z6(d));
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP,TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=(z6(d));
        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        z61=vertcat(TTTP,TTT);
    end
end
z61(1,:)=[];
z6=z61;
%cuarta variable: segunda
drows=NFP+1;
dcol=1;
z7=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    SNi=SN(r);
    z7(r)=sqrt((XH)^2+(OV-YNi-SNi)^2);
end
drows=NFP+1;
dcol=1;
z77=ones(drows,dcol);
for r = 1:drows
    z771=z7(r);
    if r==1
        z777=z771;
        z7777=z777;
        z77(r)=z7777;
    else
        z777=z771;
        z7777=z777+z7777;
    end
end

```

```

        z77(r)=z7777;
    end
    z77(r)=z7777;
end
z7=z77;
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
z71=ones(drows,dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=(z7(d));
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP,TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=(z7(d));
        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        z71=vertcat(TTTP,TTT);
    end
end
z71(1,:)=[];
z7=z71;
%cuarta variable: tercera
drows=NFP+1;
dcol=1;
z8=ones(drows,dcol);
for r = 1:drows
    SNi=SN(r);
    YNi=YN(r);
    z8(r)=(BW*YNi*SNi);
end
x66=ones(drows,dcol);
for r = 1:drows
    x661=z8(r);
    if r==1
        x666=x661;
        x6666=x666;
        x66(r)=x6666;
    else
        x666=x661;
        x6666=x666+x6666;
        x66(r)=x6666;
    end
    x66(r)=x6666;
end
end

```



```

z8=x66;
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
z108=ones(drows,dcol);
for r = 1:b
    c=r;
    if r==1
        SNNN=SNN(c);
        [b2]=find(OS==SNNN);
        TTTP=0;
    elseif r>1 && r<b;
        d=c-1;
        SNNN=SNN(c);
        YNNN=(z8(d));
        [b3]=find(OS==SNNN);
        b3=b3(1);
        b4=b3-b2;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        TTTP=vertcat(TTTP,TTT);
        b2=b3;
    else
        d=c-1;
        YNNN=(z8(d));
        [b3]=find(OS==1);
        b3=b3(1);
        b4=b3-b2+1;
        TTT=ones(b4,dcol);
        for rr=1:b4
            TTT(rr)=YNNN;
        end
        z108=vertcat(TTTP,TTT);
    end
end
z108(1,:)=[];
drows=NFP+1;
dcol=1;
z1010=ones(drows,dcol);
for r = 1:drows
    YNi=YN(r);
    z1010(r)=(BW*YNi);
end
x66=ones(drows,dcol);
for r = 1:drows
    x661=z1010(r);
    if r==1
        x666=x661;
        x6666=x666;
        x66(r)=x6666;
    else
        x666=x661;
        x6666=x666+x6666;
        x66(r)=x6666;
    end
    x66(r)=x6666;
end
z1010=x66;
b=NFP+2;
u=length(OS);
drows=u;
dcol=1;
z10=ones(drows,dcol);
for r = 1:b

```

```

c=r;
if r==1
    SNNN=SNN(c);
    [b2]=find(OS==SNNN);
    TTTP=0;
elseif r>1 && r<b;
    d=c-1;
    SNNN=SNN(c);
    YNNN=(z1010(d));
    [b3]=find(OS==SNNN);
    b3=b3(1);
    b4=b3-b2;
    TTT=ones(b4,dcoll);
    for rr=1:b4
        TTT(rr)=YNNN;
    end
    TTTP=vertcat(TTTP,TTT);
    b2=b3;

else
    d=c-1;
    YNNN=(z1010(d));
    [b3]=find(OS==1);
    b3=b3(1);
    b4=b3-b2+1;
    TTT=ones(b4,dcoll);
    for rr=1:b4
        TTT(rr)=YNNN;
    end
    z10=vertcat(TTTP,TTT);
end
end
z10(1,:)=[];
OS=OS';
u=length(OS);
drows=u;
dcoll=1;
z11=ones(drows,dcoll);
for r=1:drows
    ZZ=OS(r);
    zz1=z108(r);
    zz2=z10(r);
    zz3=zz1-zz2*ZZ;
    z11(r)=zz3;
end
z13=z3+z4-z5-z6+z7+z11;
Z=z13*Lo;
S=OS*Lo;
end

```

Figura 96. Código del experimento 4 – Función “CoorzCE”
Fuente: Autor

```

function [ef]= tolf (b,RL,d,Sh)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Criterios de convergencia de la solución
%
%El segundo criterio de convergencia se aplica a la función de ecuaciones
%no lineales, donde la tolerancia Tol f debe escogerse con el mismo
%criterio mencionado para la tol x
%
%Datos de Entrada:
% l : Distancia Horizontal entre Punto Inicial y Punto Final (m)
% h : Diferencia de nivel entre los apoyos (m)
% b : es la variable con respecto f(Ho,Vo)
% d : es la variable con respecto g(Ho,Vo)
%
%Datos de Salida
% ef: es la función par ala tolerancia f
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ef=[b-RL;d-Sh];
end

```

Figura 97. Código del experimento 4 – Función “Tolf”
Fuente: Autor

```

function [ex]= tal (X0,X)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Criterios de convergencia de la solución
%
%El primer criterio sobre el error relativo del vector solución
%
%Datos de Entrada:
% X0: Es el vector de incógnitas Ho y Vo cuando es k (0 o anterior)
%
% X1: Es el vector de incógnitas Ho y Vo cuando es k+1 (1 o siguiente)
%Datos de Salida
% ex: es la función para la tolerancia x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ex= (X0-X)/X;
ex(:,1)=[];
end

```

Figura 98. Código del experimento 4 – Función “Tolx”
Fuente: Autor

B. Plataforma virtual de los experimentos

Experimento 1 – Cable Parabólico

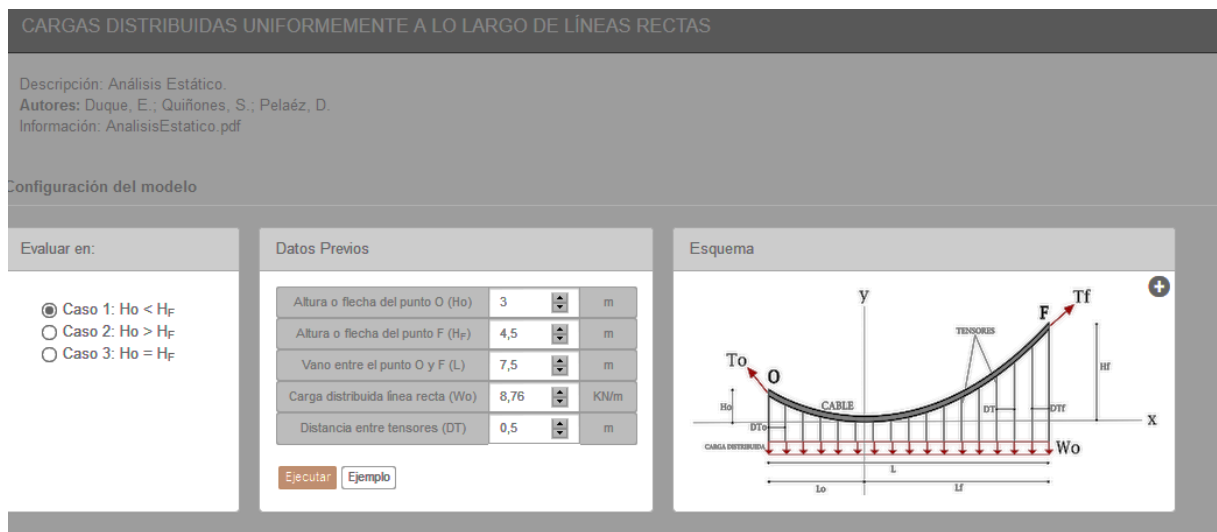


Figura 99. Interfaz gráfica del experimento 1 – configuración de datos de entrada con respecto al caso 1.

Fuente: Autor

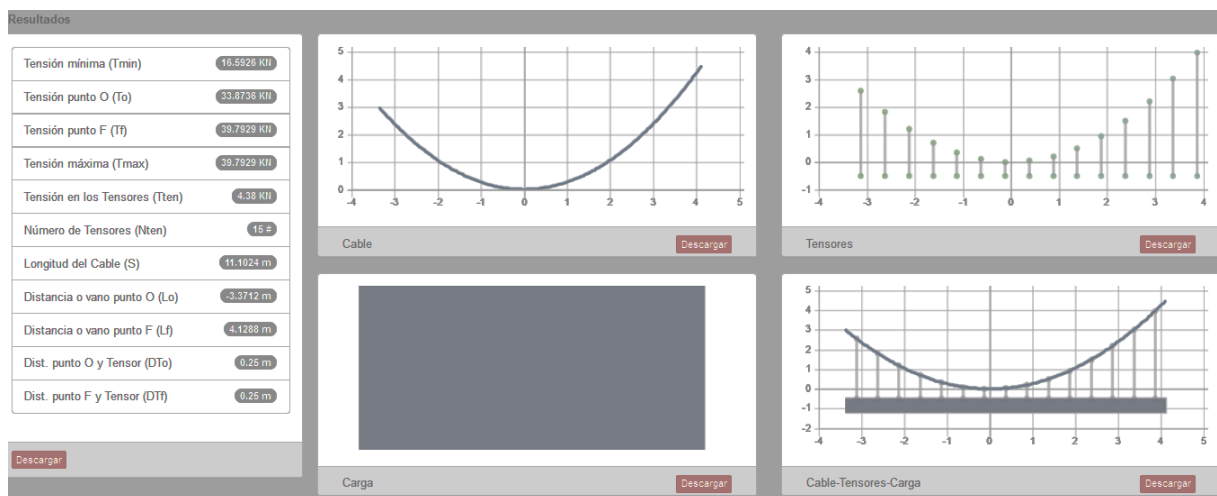


Figura 100. Interfaz gráfica del experimento 1 – configuración de los resultados de análisis con respecto al caso 1 y graficas.

Fuente: Autor

"UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA"
TITULACIÓN DE INGENIERÍA CIVIL
LABORATORIO VIRTUAL DE Ingeniería Sísmica (VLEE)

LABORATORIO: CABLES - STRUCTURES
EXPERIMENTO: Análisis Estático de Cables con Cargas distribuidas uniformemente a lo largo de líneas rectas
CASO : 1 : Ho<Hf : La flecha en el punto O es menor a la flecha del punto F
FECHA : 14-Sep-2017 15:25:45

DATOS DE ENTRADA

Detalle	Símbolo	Unidades	Magnitud
Altura o flecha del punto O	Ho	m	3
Altura o flecha del punto F	Hf	m	4.5
Vano entre el punto O y F	L	m	7.5
Carga distribuida línea recta	Wo	KN	8.76
Distancia entre tensores	Wo	m	0.5

DATOS DE SALIDA

Detalle	Símbolo	Unidades	Magnitud
TensiOn mínima	Tmin	KN	16.5926
TensiOn punto O	To	KN	33.8736
TensiOn punto F	Tf	KN	39.7929
TensiOn máxima	Tmax	KN	39.7929
TensiOn en los Tensores	Tten	KN	4.38
NUmero de Tensores	Nten	#	15
Longitud del Cable	S	m	11.1024
Distancia o vano punto O	Lo	m	-3.3712
Distancia o vano punto F	Lf	m	4.1288
Dist. punto O y Tensor	DTo	m	0.25
Dist. punto F y Tensor	DTf	m	0.25

Figura 101. Archivo de Texto de los resultados generados por el análisis del experimento 1.

Fuente: Autor

Experimento 2 – Cable Catenaria

CATENARIA

Descripción: Catenaria....
 Autores: Duque, E.; Quiñones, S.; Peláez, D.
 Información: Catenaria.pdf

Configuración del modelo

Datos de entrada

Tipo:

☒ Flecha
☐ Longitud

Datos:

Esfuerzo de ruptura (ER)	900	N
Factor de seguridad (FS)	1	#
Carga distribuida (W_0)	3,058	kg/m
Flecha o altura - (O) al eje H_o	90	m
Flecha o altura - (F) al eje H_f	100	m
Cota (Hm)	15	m
Cota (b)	70	m

Ejecutar Ejemplo Catalogo Tabla de factores

Esquema

Figura 102. Interfaz gráfica del experimento 2 – configuración de datos de entrada con respecto al caso 1 Flecha.

Fuente: Autor

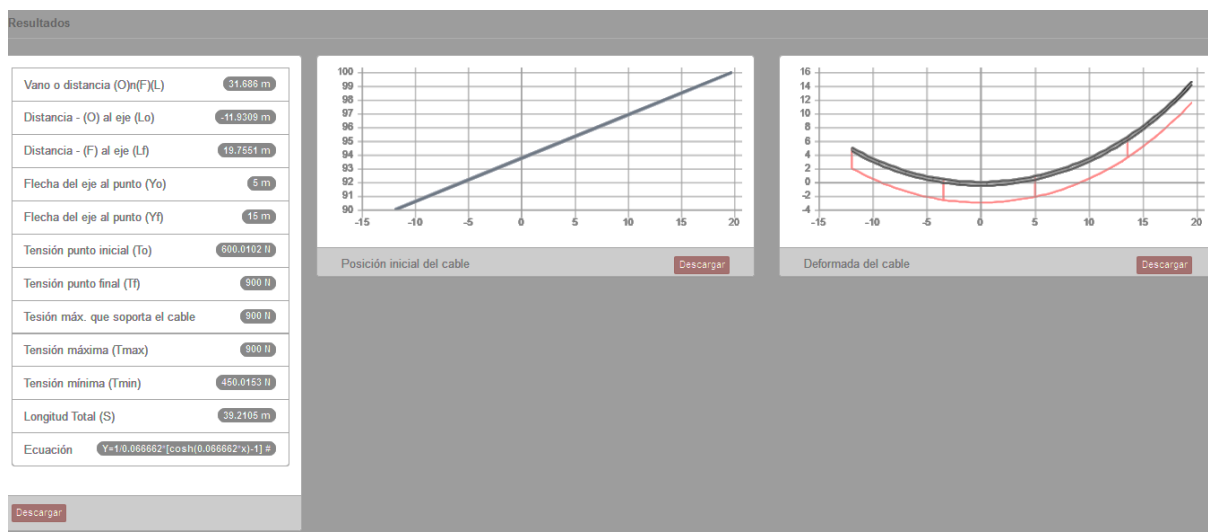


Figura 103. Interfaz gráfica del experimento 1 – configuración de los resultados de análisis con respecto al caso 1 y graficas.

Fuente: Autor

Figura 104. Archivo de Texto de los resultados generados por el análisis del experimento 2.

Fuente: Autor

Experimento 3 – Cable con Cargas Puntuales



Descripción: Cargas Puntuales

Autores: Duque, E.; Quiñones, S.; Peláez, D.

Información: CargasPuntuales.pdf

Si experimenta algún problema o si necesita ayuda para ejecutar este programa, póngase en contacto con epduque@utpl.edu.ec

ESQUEMA

INPUT

Número de tramos: [Ejemplo](#)

TRAMO AB TRAMO BC TRAMO CD

Número de cargas:

	1	2	3	4	5	6	7	8	9	10	11
WAB	10	5	20	10	20	25					
DAB	4	4	5	4	7	2	6				
HLAB	5										
YB	-30										

Figura 105. Interfaz gráfica de los datos de entrada para el tramo 1 del experimento 3.

Fuente: Autor



Figura 106. Interfaz gráfica de los resultados generados por el análisis del experimento 3. Grafica cuando el cable está en posición inicial. Resultados de las reacciones en los apoyos.

Fuente: Autor

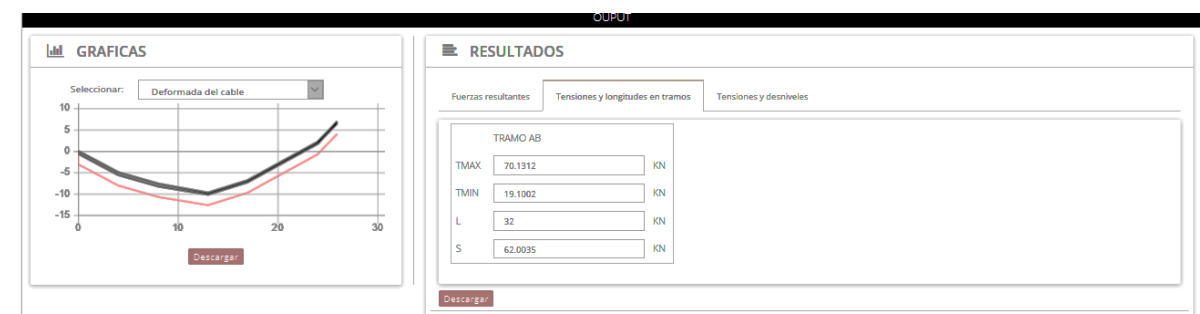


Figura 107. Interfaz gráfica de los resultados generados por el análisis del experimento 3. Grafica con respecto a la deformada del cable. Resultados de las tensiones en el tramo correspondiente.

Fuente: Autor

"UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA"
TITULACIÓN DE INGENIERÍA CIVIL
LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA (VLEE)

LABORATORIO: CABLES - STRUCTURES
EXPERIMENTO: Análisis Estático de Cables con Cargas Puntuales
CASO : Tramo || Determina las fuerzas reultantes en el punto inicial y final, y sus componentes correspondientes
FECHA : 14-Sep-2017 15:44:44

DATOS DE ENTRADA

Detalle	Símbolo	Unidades	Magnitud
Numero de cargas	NW	#	6
Carga 1	W1	KN	10
Carga 2	W2	KN	5
Carga 3	W3	KN	20
Carga 4	W4	KN	10
Carga 5	W5	KN	20
Carga 6	W6	KN	25
Distancia 1	D1	m	4
Distancia 2	D2	m	4
Distancia 3	D3	m	5
Distancia 4	D4	m	4
Distancia 5	D5	m	7
Distancia 6	D6	m	2
Distancia 7	D7	m	6
Altura o flecha 1	h1	m	5
Desnivel entre A y B	YB	m	-30

DATOS DE SALIDA

Detalle	Símbolo	Unidades	Magnitud
Reacción en el punto incial A	A	KN	28.3567
Reacción en el eje x de A	Ax	KN	-17.7143
Reacción en el eje y de A	Ay	KN	22.1429
Reacción en el punto incial B	B	KN	70.1312
Reacción en el eje x de B	Bx	KN	17.7143
Reacción en el eje y de B	By	KN	67.8571
Tensión Tramo A-1	T1	KN	28.3567
Tensión Tramo 1-2	T2	KN	21.4766
Tensión Tramo 2-3	T3	KN	19.1002
Tensión Tramo 3-4	T4	KN	21.8884
Tensión Tramo 4-5	T5	KN	28.9179
Tensión Tramo 5-6	T6	KN	46.3738
Tensión Tramo 6-B	T7	KN	70.1312
Tensión máxima	Tmax	N	70.1312
Tensión mínima	Tmin	N	19.1002
Flecha o altura 1	h1	m	5
Flecha o altura 2	h2	m	7.7419
Flecha o altura 3	h3	m	9.7581
Flecha o altura 4	h4	m	6.8548
Flecha o altura 5	h5	m	-2.1774
Flecha o altura 6	h6	m	-7.0161
Vano entre A y B	L	m	32
Longitud del Cable	S	m	62.0035

Figura 108. Archivo de texto de los resultados generados por el análisis del experimento 3.

Fuente: Autor

C. Modelo del Texto guía de prácticas de los experimentos

D. Modelo del Manual de los experimentos