

UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE
INGENIERÍA CIVIL



TESIS

DISEÑO INTEGRAL DE ARMADURAS METALICAS USANDO
RHINOCEROS 3D:
MODELACION PARAMETRICA, SIMULACION
ESTRUCTURAL Y OPTIMIZACION

PARA OPTAR EL TITULO PROFESIONAL DE:
INGENIERO CIVIL

PRESENTADO POR EL BACHILLER:
JHOSEPH ALEJANDRO GONZALES RAMIREZ

ASESOR:
ING. MARCOS MENDOZA LINARES

CAJAMARCA – PERÚ

2018

AGRADECIMIENTO

- El primer lugar agradezco a Dios, mi padre, amigo y hermano; por los muchos ánimos, por ser la fortaleza de mi vida y permitirme lograr la meta de ser profesional.
- A toda mi familia por la confianza, comprensión, apoyo y motivación en mi etapa de estudiante y en el desarrollo de esta investigación.
- Agradecimiento a la Escuela Académico Profesional de Ingeniería Civil, Alma Máter donde me forme, por el conocimiento impartido.
- A mi asesor, el Ing. Marcos Mendoza Linares por su gran apoyo y asesoramiento durante el desarrollo de esta investigación.
- A todas las personas que de una u otra forma ayudaron para que esta investigación pueda realizarse, Dios les bendiga.

DEDICATORIA

Dedico esta tesis en primer lugar a Dios por su ayuda incondicional a lo largo de mi vida, por estar presente en los buenos y malos momentos, y haberme permitido llegar a esta etapa muy importante en mi formación académica. A mi mamá María por su gran apoyo durante mi carrera universitaria y por su gran ejemplo de trabajo y esfuerzo. A mi hermana Marisol por los ánimos y ayuda constante en mi formación como profesional. A mi papá Alejandro por las muchas motivaciones para superarse y ser mejor. A toda mi familia en general que son una gran bendición que Dios me ha dado.

CONTENIDO

AGRADECIMIENTO	2
DEDICATORIA.....	3
CONTENIDO	4
ÍNDICE DE FIGURAS.....	8
ÍNDICE DE TABLAS.....	17
RESUMEN.....	19
ABSTRACT.....	20
1 . CAPITULO I: INTRODUCCIÓN.....	21
1.1. PLANTEAMIENTO DEL PROBLEMA.....	21
1.2. PROBLEMA DE INVESTIGACIÓN	23
1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN	23
1.4. ALCANCES O DELIMITACIÓN DE LA INVESTIGACIÓN.....	23
1.5. LIMITACIONES	23
1.6. OBJETIVOS.....	24
1.6.1. Objetivo General	24
1.6.2. Objetivos Específicos.....	24
1.7. ORGANIZACIÓN DEL TRABAJO	24
2. CAPITULO II: MARCO TEÓRICO.....	25
2.1. ANTECEDENTES TEÓRICOS	25
2.2. BASES TEÓRICAS	27
2.2.1. Proceso de diseño estructural propuesto.....	27
2.2.2. Modelación paramétrica.....	29
2.2.2.1. Rhinoceros 3D: Herramienta de modelado formas arquitectónicas	30
2.2.2.2. Grasshopper: Complemento de Rhinoceros para el modelado generativo de formas arquitectónicas.....	31
2.2.2.3. Modelación paramétrica aplicada a armaduras metálicas	32
2.2.3. Análisis estructural	33
2.2.3.1. Simulación estructural.....	34
2.2.4. Optimización estructural con algoritmos evolutivos.....	35

2.2.4.1.	Optimización estructural	35
2.2.4.2.	Tipos de optimización estructural	36
2.2.4.3.	Técnicas de optimización estructural	39
2.2.4.3.1.	Técnicas metaheurísticas	39
2.2.4.3.2.	Computación Evolutiva	39
2.2.4.3.3.	Algoritmos Evolutivos	40
2.2.4.3.4.	Algoritmos Genéticos	41
2.2.4.4.	Complemento Galápagos de Grasshopper	43
2.2.5.	Sistematización con Grasshopper	52
2.2.5.1.	Nociones básicas de Grasshopper	54
2.2.6.	Verificación de deflexiones	57
2.2.7.	Diseño de elementos de armaduras metálicas usando las “Especificaciones para Construcciones de Acero Estructural” (AISC 360 -10) del AMERICAN INSTITUTE OF STEEL CONSTRUCTION (AISC, 2010)	58
2.2.7.1.	Diseño por resistencia usando el Diseño en Base a Factores de Carga y Resistencia (LRFD)	58
2.2.7.2.	Combinaciones de carga	58
2.2.7.3.	Clasificación de secciones para pandeo local	59
2.2.7.4.	Elementos rígidos	59
2.2.7.5.	Diseño de espesor de pared para HSS	60
2.2.7.6.	Determinación del área bruta y del área neta	60
2.2.7.7.	Diseño de miembros a tensión	62
2.2.7.7.1.	Limitaciones de esbeltez	62
2.2.7.7.2.	Resistencia a la tensión	62
2.2.7.7.3.	Área neta efectiva (A_e)	63
2.2.7.8.	Diseño de miembros a compresión	63
2.2.7.7.1.	Longitud efectiva	66
2.2.7.7.2.	Pandeo flexional de miembros sin elementos esbeltos	67
2.2.7.7.3.	Miembros con elementos esbeltos	68
2.2.7.7.3.1.	Elementos Esbeltos Rigidizados	69

2.2.8.	Aplicación del ciclo: Modelación Paramétrica, Simulación Estructural y Optimización	70
3.	CAPITULO III: MATERIALES Y MÉTODOS	71
3.1.	UBICACIÓN GEOGRÁFICA	71
3.2.	ÉPOCA DE LA INVESTIGACIÓN	71
3.3.	PROCEDIMIENTO	71
3.3.1.	Tipo de investigación	71
3.3.2.	Población de estudio	71
3.3.3.	Muestra	71
3.3.4.	Unidad de análisis	71
3.3.5.	Técnicas e instrumentos de recolección de datos	71
3.4.	TRATAMIENTO, ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS	72
3.4.1.	Desarrollo de scripts para realizar el diseño de armaduras metálicas según la AISC 360-10, N.T.E E 0.20 y N.T.E E 0.90	74
3.4.1.1.	Script N°1	74
3.4.1.2.	Script N°2	74
3.4.1.3.	Script N°3	76
3.4.1.4.	Script N°4	76
3.4.1.5.	Script N°5	77
3.4.1.6.	Script N°6	78
3.4.1.7.	Script N°7	79
3.4.1.8.	Script N°8	83
3.4.1.9.	Script N°9	87
3.4.1.10.	Script N°10	88
3.4.1.11.	Script N°11	88
3.4.2.	Ejemplo N°1	90
3.4.2.1.	Modelación paramétrica	90
3.4.2.2.	Simulación estructural	91
3.4.2.3.	Optimización	115
3.4.3.	Ejemplo N°2	138

3.4.3.1.	Modelación paramétrica.....	138
3.4.3.2.	Optimización.....	139
3.4.4.	Ejemplo N°3.....	161
3.4.4.1.	Modelación paramétrica.....	161
3.4.4.2.	Optimización.....	163
4.	CAPITULO IV: ANÁLISIS Y DISCUSIÓN DE RESULTADOS	196
4.1.	EJEMPLO N°1.....	196
4.2.	EJEMPLO N°2.....	201
4.3.	EJEMPLO N°3.....	208
4.4.	ANÁLISIS DEL “DISEÑO INTEGRAL DE ARMADURAS METÁLICAS USANDO RHINOCEROS 3D”	210
5.	. CAPITULO V: CONCLUSIONES Y RECOMENDACIONES.....	212
5.1.	CONCLUSIONES.....	212
5.2.	RECOMENDACIONES.....	212
	REFERENCIAS BIBLIOGRÁFICAS.....	213
	ANEXOS.....	217
	Anexo 1. Predimensionamiento de armaduras planas.....	217
	Anexo 2. Código en lenguaje de programación C Sharp (C#) del script N°1.....	217
	Anexo 3. Código en lenguaje de programación C Sharp (C#) del script N°2.....	217
	Anexo 4. Código en lenguaje de programación C Sharp (C#) del script N°3.....	218
	Anexo 5. Código en lenguaje de programación C Sharp (C#) del script N°4.....	218
	Anexo 6. Código en lenguaje de programación C Sharp (C#) del script N°5.....	219
	Anexo 7. Código en lenguaje de programación C Sharp (C#) del script N°6.....	221
	Anexo 8. Código en lenguaje de programación C Sharp (C#) del script N°7.....	221
	Anexo 9. Código en lenguaje de programación C Sharp (C#) del script N°8.....	236
	Anexo 10. Código en lenguaje de programación C Sharp (C#) del script N°9.....	256
	Anexo 11. Código en lenguaje de programación C Sharp (C#) del script N°10.....	257
	Anexo 12. Código en lenguaje de programación C Sharp (C#) del script N°11.....	257
	Anexo 13. Peso propio.....	258

ÍNDICE DE FIGURAS

Figura 2.1. Proceso de diseño estructural propuesto.....	27
Figura 2.2. Logo del software Rhinoceros 3D. Adaptado de (McNeel, Rhinoceros diseñar, modelar, presentar, analizar, realizar, 2017).....	30
Figura 2.3. Logo del Complemento Grasshopper. Adaptado de (Davidson, 2017).....	31
Figura 2.4. Armadura Howe parametrizada en función del número de segmentos (o ángulo entre diagonales y bridas) y peralte.	32
Figura 2.5. Código gráfico de armadura Howe mostrada en la figura 2.4.....	32
Figura 2.6. Logo del complemento “Karamba”. Adaptado de (Preisinger, 2017).....	33
Figura 2.7. Armadura Howe definida por parámetros (2.7a) de: Luz a cubrir, peralte de armadura plana y número de segmentos de armadura plana (o ángulo entre diagonales y bridas). Sliders o controles de magnitud deslizantes (2.7b) que cambian los valores de la luz a cubrir, el peralte y el número de segmentos con solo escribir una cantidad que se encuentre dentro de sus rangos en las cajas de texto o deslizándolas.	35
Figura 2.8. Los tres tipos de optimización estructural. Adaptado de (Richardson, 2010).....	36
Figura 2.9. Malla triangular de vigas antes (2.9a) y después (2.9b) de aplicar el componente "BESO for Beams". Adaptado de (Preisinger, 2016).	38
Figura 2.10. Logo de la herramienta Galápagos. Adaptado de (Rutten, 2017).	43
Figura 2.11. Trayectoria “A-B” a optimizar con Galápagos. Adaptado de (Tedeschi, 2014).	45
Figura 2.12. Puntos de interpolación para optimizar trayectoria “A-B” con Galápagos. Adaptado de (Tedeschi, 2014).....	45
Figura 2.13. Código gráfico de optimización de la trayectoria “A-B” mostrada en la figura 2.11. Adaptado de (Tedeschi, 2014).	46
Figura 2.14. Menú contextual de Galápagos. Adaptado de (Tedeschi, 2014).....	47
Figura 2.15. Componente Galápagos en pleno trabajo de optimización. Adaptado de (Tedeschi, 2014).	48
Figura 2.16. Uso de valores objetivo en la optimización con Galápagos. Adaptado de (Tedeschi, 2014).	49
Figura 2.17. Rangos de aptitud sin factores de ponderación de normalización. Adaptado de (Rutten, 2011).	51

Figura 2.18. Rangos de aptitud con factores de ponderación de normalización. Adaptado de (Rutten, 2011).	51
Figura 2.19. Logo del complemento Octopus. Adaptado de (Vier, 2012).....	52
Figura 2.20. Esta figura compara el proceso para dibujar una curva sinoidal en Python y en Grasshopper. Adaptado de (Akos y Parsons, 2015).....	53
Figura 2.21. Objetos en Grasshopper. Adaptado de (Payne y Issa, 2009).....	54
Figura 2.22. Partes de un componente en Grasshopper. Adaptado de (Payne y Issa, 2009).....	56
Figura 2.23. Herramientas de ayuda de un componente de Grasshopper. Adaptado de (Payne y Issa, 2009).....	56
Figura 3.1. Script “KL”.....	74
Figura 3.2. Script “LONGITUD”	75
Figura 3.3. Script “PP(kN)”.....	76
Figura 3.4. Script “CARGA DE SERVICIO”	77
Figura 3.5. Script “COMBINACION DE CARGAS”.....	78
Figura 3.6. Script “FILTRO1”.....	79
Figura 3.7. Script “AISC - TENSION - COMPRESION C”.....	83
Figura 3.8. Script “AISC - TENSION - COMPRESION R”.....	87
Figura 3.9. Script “FILTRO2”.....	87
Figura 3.10. Script “FILTRO3”.....	88
Figura 3.11. Script “PESO”	88
Figura 3.12. Código gráfico de la modelación paramétrica del ejemplo N°1.....	90
Figura 3.13. Código gráfico de modelación paramétrica y análisis estructural - ejemplo N°1....	91
Figura 3.14. Armadura plana tipo Howe con código gráfico del análisis estructural - ejemplo N°1.	91
Figura 3.15. Parámetros de control - simulación estructural - ejemplo N°1.....	92
Figura 3.16. Número de segmentos - Máxima deflexión - Carga de 3kN - ejemplo N°1.	93
Figura 3.17. Número de segmentos - Máxima deflexión - Carga de 4kN - ejemplo N°1.	94
Figura 3.18. Número de segmentos – Máxima deflexión – Carga de 5kN - ejemplo N°1.	95
Figura 3.19. Número de segmentos – Máxima deflexión – Carga de 6kN - ejemplo N°1.	96
Figura 3.20. Número de segmentos – Máxima deflexión – Carga de 7kN - ejemplo N°1.	97
Figura 3.21. Número de segmentos – Máxima deflexión – Carga de 8kN - ejemplo N°1.	98

Figura 3.22. Número de segmentos – Máxima deflexión – Carga de 9kN - ejemplo N°1.	99
Figura 3.23. Número de segmentos – Máxima deflexión – Carga de 10kN - ejemplo N°1.	100
Figura 3.24. Número de segmentos - Máxima deflexión - Peralte de 0.60m - ejemplo N°1.....	102
Figura 3.25. Número de segmentos - Máxima deflexión - Peralte de 0.65m - ejemplo N°1.....	103
Figura 3.26. Número de segmentos - Máxima deflexión - Peralte de 0.70m - ejemplo N°1.....	104
Figura 3.27. Número de segmentos - Máxima deflexión - Peralte de 0.75m - ejemplo N°1.....	105
Figura 3.28. Número de segmentos - Máxima deflexión - Peralte de 0.80m - ejemplo N°1.....	106
Figura 3.29. Número de segmentos - Máxima deflexión - Peralte de 0.85m - ejemplo N°1.....	107
Figura 3.30. Número de segmentos - Máxima deflexión - Peralte de 0.90m - ejemplo N°1.....	108
Figura 3.31. Número de segmentos - Máxima deflexión - Peralte de 0.95m - ejemplo N°1.....	109
Figura 3.32. Número de segmentos - Máxima deflexión - Peralte de 1.00m - ejemplo N°1.....	110
Figura 3.33. Disminución de máxima deflexión - Aumento acumulado de peralte - ejemplo N°1.	112
Figura 3.34. Peralte - Máxima deflexión - ejemplo N°1.	114
Figura 3.35. Código gráfico de modelación paramétrica y optimización - ejemplo N°1.	115
Figura 3.36. Ventana o entorno de “Octopus” - optimización geométrica - ejemplo N°1.	116
Figura 3.37. Solución óptima en los entornos de Rhino y “Octopus”, figuras 3.37a y 3.37b respectivamente - optimización geométrica - ejemplo N°1.....	117
Figura 3.38. Parámetros de solución óptima - optimización geométrica - ejemplo N°1.....	118
Figura 3.39. Peso de la armadura luego de la optimización geométrica - ejemplo N°1.....	118
Figura 3.40. Deflexión máxima de la armadura luego de la optimización geométrica - ejemplo N°1.	118
Figura 3.41. Ventana inicial de Galápagos - optimización de tamaño - ejemplo N°1.....	119
Figura 3.42. Optimización con el “Evolutionary Solver” de Galápagos - optimización de tamaño - ejemplo N°1.....	120
Figura 3.43. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización de tamaño - ejemplo N°1.	122
Figura 3.44. Resultados de diseño de la brida superior - optimización de tamaño - ejemplo N°1.	122
Figura 3.45. Resultados de diseño de la brida inferior - optimización de tamaño - ejemplo N°1.	123

Figura 3.46. Resultados de diseño de las montantes - optimización de tamaño - ejemplo N°1.	123
Figura 3.47. Resultados de diseño de las diagonales - izquierda - optimización de tamaño - ejemplo N°1.	124
Figura 3.48. Resultados de diseño de las diagonales - derecha - optimización de tamaño - ejemplo N°1.	124
Figura 3.49. Peso de la armadura luego de la optimización de tamaño - ejemplo N°1.	125
Figura 3.50. Deflexión máxima de la armadura luego de la optimización de tamaño - ejemplo N°1.	125
Figura 3.51. Implementaciones de código gráfico para realizar la optimización topológica - ejemplo N°1.	126
Figura 3.52. Implementaciones de código gráfico para eliminar las montantes de los extremos - ejemplo N°1.	127
Figura 3.53. Armadura luego de eliminar las montantes de los extremos - ejemplo N°1.	127
Figura 3.54. Componente “BESOBeam”, componente que realiza la optimización topológica - ejemplo N°1.	128
Figura 3.55. Armadura luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.	129
Figura 3.56. Deflexión máxima luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.	129
Figura 3.57. Porcentajes de utilización de los elementos luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.	130
Figura 3.58. Ratios de diseño de los elementos de la brida superior luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.	130
Figura 3.59. Resultados de diseño de la brida superior - armadura óptima final - ejemplo N°1.	131
Figura 3.60. Resultados de diseño de la brida inferior - armadura óptima final - ejemplo N°1.	132
Figura 3.61. Resultados de diseño de las montantes - armadura óptima final - ejemplo N°1.	132
Figura 3.62. Resultados de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°1.	133
Figura 3.63. Resultados de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°1.	133
Figura 3.64. Peso de armadura óptima final - ejemplo N°1.	134

Figura 3.65. Nodos de armadura óptima final - ejemplo N°1.....	134
Figura 3.66. Deflexión máxima de armadura óptima final - ejemplo N°1.	134
Figura 3.67. Elementos a tensión y/o compresión, con su respectivo número identificador en “SAP2000” - armadura óptima final - ejemplo N°1.	135
Figura 3.68. Deflexión máxima de armadura (en el extremo inferior de la montante central) obtenida con “SAP2000” - armadura óptima final - ejemplo N°1.....	136
Figura 3.69. Deflexión máxima de armadura (en el extremo superior de la montante central) obtenida con “SAP2000” - armadura óptima final - ejemplo N°1.....	136
Figura 3.70. Ratios de diseño de los elementos de la armadura óptima final obtenidos con “SAP2000”- ejemplo N°1.....	136
Figura 3.71. Código gráfico de la modelación paramétrica del ejemplo N°2.....	138
Figura 3.72. Código gráfico de modelación paramétrica y optimización - ejemplo N°2.	139
Figura 3.73. Función de optimización - optimización de tamaño - ejemplo N°2.....	140
Figura 3.74. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización de tamaño - ejemplo N°2.	141
Figura 3.75. Peso de la armadura luego de la optimización de tamaño - ejemplo N°2.	141
Figura 3.76. Nodos de armadura - optimización de tamaño - ejemplo N°2.	142
Figura 3.77. Deflexión máxima de la armadura luego de la optimización de tamaño - ejemplo N°2.....	142
Figura 3.78. Resultados de diseño de la brida superior - optimización de tamaño - ejemplo N°2.	143
Figura 3.79. Resultados de diseño de las montantes - optimización de tamaño - ejemplo N°2.	143
Figura 3.80. Resultados de diseño de la brida inferior - optimización de tamaño - ejemplo N°2.	144
Figura 3.81. Resultados de diseño de las diagonales - izquierda - optimización de tamaño - ejemplo N°2.	144
Figura 3.82. Resultados de diseño de las diagonales - derecha - optimización de tamaño - ejemplo N°2.	145
Figura 3.83. Elementos a tensión y/o compresión, con su respectivo número identificador en “SAP2000” - optimización de tamaño - ejemplo N°2.	145

Figura 3.84. Deflexión máxima de armadura obtenida con “SAP2000”- optimización de tamaño - ejemplo N°2.	146
Figura 3.85. Ratios de diseño de los elementos de la armadura obtenidos con “SAP2000”- optimización de tamaño - ejemplo N°2.	146
Figura 3.86. Función de optimización - optimización geométrica y de tamaño - ejemplo N°2.	149
Figura 3.87. Parámetros de control luego de optimización geométrica y de tamaño - ejemplo N°2.	150
Figura 3.88. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización geométrica y de tamaño - ejemplo N°2.....	150
Figura 3.89. Peso de la armadura luego de la optimización geométrica y de tamaño - ejemplo N°2.	151
Figura 3.90. Deflexión máxima de la armadura luego de la optimización geométrica y de tamaño - ejemplo N°2.....	151
Figura 3.91. Optimización topológica con “BESOBeam” - ejemplo N°2.....	152
Figura 3.92. Armadura luego de una optimización topológica con un “Target mass ratio” de 0.88 - ejemplo N°2.....	152
Figura 3.93. Resultados de diseño de la brida superior - optimización topológica - ejemplo N°2.	153
Figura 3.94. Resultados de diseño de las montantes - optimización topológica - ejemplo N°2.	154
Figura 3.95. Resultados de diseño de la brida inferior - optimización topológica - ejemplo N°2.	155
Figura 3.96. Resultados de diseño de las diagonales - izquierda - optimización topológica - ejemplo N°2.	156
Figura 3.97. Resultados de diseño de las diagonales - derecha - optimización topológica - ejemplo N°2.	157
Figura 3.98. Nodos de armadura óptima - optimización topológica - ejemplo N°2.....	157
Figura 3.99. Deflexión máxima de armadura óptima - optimización topológica - ejemplo N°2.	158
Figura 3.100. Elementos a tensión y/o compresión en “SAP2000” - optimización topológica - ejemplo N°2.	158

Figura 3.101. Deflexión máxima de armadura obtenida con “SAP2000”- optimización topológica - ejemplo N°2.....	158
Figura 3.102. Esquema de los ratios de diseño de los elementos de la armadura obtenidos con “SAP2000”- optimización topológica - ejemplo N°2.....	159
Figura 3.103. Cobertura metálica de bloque B1. Adaptado de (Quispe, 2016).....	161
Figura 3.104. Código gráfico de la modelación paramétrica del ejemplo N°3.....	162
Figura 3.105. Arco de viga de celosía definido en modelación paramétrica - ejemplo N°3.	162
Figura 3.106. Código gráfico de modelación paramétrica y optimización - ejemplo N°3.	163
Figura 3.107. Función de optimización de arco de viga de celosía - primera optimización geométrica y de tamaño - ejemplo N°3.....	166
Figura 3.108. Parámetros de control de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.....	166
Figura 3.109. Peso de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.....	167
Figura 3.110. Deflexión máxima de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.....	167
Figura 3.111. Longitudes de segmentos de la brida superior del arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.	168
Figura 3.112. Función de optimización - optimización geométrica de vigueta - ejemplo N°3..	169
Figura 3.113. Parámetros de control luego de la optimización geométrica de vigueta - ejemplo N°3.....	170
Figura 3.114. Parámetros de control - vigueta óptima final - ejemplo N°3.....	170
Figura 3.115. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - vigueta óptima final - ejemplo N°3.	170
Figura 3.116. Peso de vigueta óptima final - ejemplo N°3.....	171
Figura 3.117. Nodos de vigueta óptima final - ejemplo N°3.....	171
Figura 3.118. Deflexión máxima de vigueta óptima final - ejemplo N°3.....	171
Figura 3.119. Resultados de diseño de la brida superior - vigueta óptima final - ejemplo N°3.	172
Figura 3.120. Resultados de diseño de la brida inferior - vigueta óptima final - ejemplo N°3. .	172
Figura 3.121. Resultados de diseño de las diagonales - izquierda - vigueta óptima final - ejemplo N°3.....	173

Figura 3.122. Resultados de diseño de las diagonales - derecha - vigueta óptima final - ejemplo N°3.....	173
Figura 3.123. Función de optimización de arco de viga de celosía - segunda optimización geométrica y de tamaño - ejemplo N°3.....	174
Figura 3.124. Parámetros de control de arco de viga de celosía luego de la segunda optimización geométrica y de tamaño - ejemplo N°3.....	175
Figura 3.125. Optimización topológica con “BESOBeam” de arco de viga de celosía - ejemplo N°3.....	175
Figura 3.126. Diagonales - derecha de arco de viga de celosía eliminadas con la optimización topológica - ejemplo N°3.....	176
Figura 3.127. Parámetros de control - armadura óptima final - ejemplo N°3.	176
Figura 3.128. Optimización topológica con “BESOBeam” - peso - armadura óptima final - ejemplo N°3.	177
Figura 3.129. Nodos de armadura óptima final - ejemplo N°3.....	177
Figura 3.130. Deflexión máxima - armadura óptima final - ejemplo N°3.....	177
Figura 3.131. Sección de la brida superior - armadura óptima final - ejemplo N°3.....	178
Figura 3.132. Resistencia de diseño de los elementos de la brida superior - armadura óptima final - ejemplo N°3.....	179
Figura 3.133. Ratios de diseño de los elementos de la brida superior - armadura óptima final - ejemplo N°3.	180
Figura 3.134. Sección de montantes - armadura óptima final - ejemplo N°3.....	181
Figura 3.135. Resistencia de diseño de las montantes - armadura óptima final - ejemplo N°3.	182
Figura 3.136. Ratios de diseño de las montantes - armadura óptima final - ejemplo N°3.	183
Figura 3.137. Sección de la brida inferior - armadura óptima final - ejemplo N°3.....	184
Figura 3.138. Resistencia de diseño de los elementos de la brida inferior - armadura óptima final - ejemplo N°3.....	185
Figura 3.139. Ratios de diseño de los elementos de la brida inferior - armadura óptima final - ejemplo N°3.	186
Figura 3.140. Sección de las diagonales - izquierda - armadura óptima final - ejemplo N°3.	187
Figura 3.141. Resistencia de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°3.	187

Figura 3.142. Ratios de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°3.	188
Figura 3.143. Sección de las diagonales - derecha - armadura óptima final - ejemplo N°3.	189
Figura 3.144. Resistencia de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°3.	189
Figura 3.145. Ratios de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°3.	190
Figura 3.146. Sección de los enlaces superiores - armadura óptima final - ejemplo N°3.	191
Figura 3.147. Resistencia de diseño de los enlaces superiores - armadura óptima final - ejemplo N°3.	192
Figura 3.148. Ratios de diseño de los enlaces superiores - armadura óptima final - ejemplo N°3.	192
Figura 3.149. Sección de los enlaces inferiores - armadura óptima final - ejemplo N°3.	193
Figura 3.150. Resistencia de diseño de los enlaces inferiores - armadura óptima final - ejemplo N°3.	194
Figura 3.151. Ratios de diseño de los enlaces inferiores - armadura óptima final - ejemplo N°3.	194

ÍNDICE DE TABLAS

Tabla 2.1. Deflexiones máximas para elementos estructurales.	57
Tabla 2.2. Relaciones Ancho-Espesor: Elementos en Compresión Miembros Sujetos a Compresión Axial.	61
Tabla 2.3. Factores de Corte Diferido para Conexiones de Miembros a Tensión.	64
Tabla 2.4. Tabla de selección para la aplicación de las secciones del capítulo E de la guía AISC 360-2010.	65
Tabla 2.5. Valores aproximados de factor de longitud efectiva, K.....	66
Tabla 3.1. Datos de simulación estructural - Carga de 3kN - ejemplo N°1.....	93
Tabla 3.2. Datos de simulación estructural - Carga de 4kN - ejemplo N°1.....	94
Tabla 3.3. Datos de simulación estructural – Carga de 5kN - ejemplo N°1.....	95
Tabla 3.4. Datos de simulación estructural – Carga de 6kN - ejemplo N°1.....	96
Tabla 3.5. Datos de simulación estructural - Carga de 7kN - ejemplo N°1.....	97
Tabla 3.6. Datos de simulación estructural – Carga de 8kN - ejemplo N°1.....	98
Tabla 3.7. Datos de simulación estructural – Carga de 9kN - ejemplo N°1.....	99
Tabla 3.8. Datos de simulación estructural – Carga de 10kN - ejemplo N°1.....	100
Tabla 3.9. Datos de simulación estructural - Peralte de 0.60m - ejemplo N°1.....	101
Tabla 3.10. Datos de simulación estructural – Peralte de 0.65m - ejemplo N°1.....	103
Tabla 3.11. Datos de simulación estructural – Peralte de 0.70m - ejemplo N°1.....	104
Tabla 3.12. Datos de simulación estructural - Peralte de 0.75m - ejemplo N°1.....	105
Tabla 3.13. Datos de simulación estructural – Peralte de 0.80m - ejemplo N°1.....	106
Tabla 3.14. Datos de simulación estructural – Peralte de 0.85m - ejemplo N°1.....	107
Tabla 3.15. Datos de simulación estructural – Peralte de 0.90m - ejemplo N°1.....	108
Tabla 3.16. Datos de simulación estructural – Peralte de 0.95m - ejemplo N°1.....	109
Tabla 3.17. Datos de simulación estructural – Peralte de 1.00m - ejemplo N°1.....	110
Tabla 3.18. Disminución de la máxima deflexión por aumento de peralte - ejemplo N°1.....	111
Tabla 3.19. Peralte - Máxima deflexión - ejemplo N°1.....	113
Tabla 3.20. Resultados de diseño de armadura óptima final, obtenidos con “SAP2000”- ejemplo N°1.....	137

Tabla 3.21. Resultados de diseño de armadura óptima, obtenidos con “SAP2000” - optimización de tamaño - ejemplo N°2.	147
Tabla 3.22. Resultados de diseño de armadura óptima, obtenidos con “SAP2000”- optimización topológica - ejemplo N°2.	160
Tabla 4.1. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION R” y “SAP2000” - solución óptima - ejemplo N°1.....	198
Tabla 4.2. Deflexiones máximas obtenidas con “Karamba”- “AISC - TENSION - COMPRESION R” y “SAP2000” - solución óptima - ejemplo N°1.....	199
Tabla 4.3. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima – “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” - ejemplo N°2.	202
Tabla 4.4. Deflexiones máximas obtenidas con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima – “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”- ejemplo N°2.	203
Tabla 4.5. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°2.....	204
Tabla 4.6. Deflexiones máximas obtenidas con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°2.....	206
Tabla 4.7. Pesos de armaduras en cada proceso de diseño empleado - ejemplo N°2.....	206
Tabla 4.8. Secciones y longitudes totales de los elementos de las armaduras en cada proceso de diseño empleado - ejemplo N°2.....	206
Tabla 4.9. Secciones y longitudes totales de los elementos del arco de viga de celosía - ejemplo N°3.....	209
Tabla 4.10. Secciones y longitudes totales de los elementos de la vigueta - ejemplo N°3.....	209
Tabla 4.11. Deflexión máxima - solución óptima final - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°3.....	209

RESUMEN

En esta investigación se ha desarrollado un proceso de diseño estructural para armaduras metálicas usando el software Rhinoceros 3D, el cual tiene por nombre “Diseño integral de armaduras metálicas usando Rhinoceros 3D”, este cuenta con tres etapas innovadoras, la primera es la modelación paramétrica en la cual mediante una programación gráfica de algoritmos se parametriza una armadura en función a parámetros geométricos (peralte, ángulo de triangulación, flecha, etc.) de los cuales dependa su comportamiento, la segunda es la simulación estructural en la cual mediante controles deslizantes se cambian los parámetros definidos en el paso anterior de manera rápida y dinámica para simular comportamientos estructurales , y la tercera es la optimización, en la cual se optimiza el peso de una armadura aplicando tres tipos de optimización (optimización de tamaño, optimización geométrica y optimización topológica). Al aplicar el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” en un caso específico, se ha obtenido una armadura óptima con un peso menor en 28.35% con respecto al peso de una armadura óptima obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, lo cual demuestra la eficiencia del “Diseño integral de armaduras metálicas usando Rhinoceros 3D”.

Palabras clave: Armadura, modelación paramétrica, simulación estructural, optimización.

ABSTRACT

In this research, a structural design process for metal reinforcement has been developed using Rhinoceros 3D software, which is entitled "Integral design of metal reinforcement using Rhinoceros 3D", this has three innovative stages, the first is parametric modeling in which by means of a graphical programming of algorithms, an armature is parameterized according to geometric parameters (cant, angle of triangulation, arrow, etc.) on which its behavior depends, the second is the structural simulation in which sliding controls are changed the parameters defined in the previous step in a fast and dynamic way to simulate structural behaviors, and the third is the optimization, in which the weight of a reinforcement is optimized by applying three types of optimization (size optimization, geometric optimization and topological optimization) . By applying the "Integral design of metal reinforcement using Rhinoceros 3D" in a specific case, an optimal reinforcement with a weight of 28.35% less than the weight of an optimal reinforcement obtained with the "Traditional structural design process of proof and error using SAP2000 ", which demonstrates the efficiency of" Integral metal reinforcement design using Rhinoceros 3D ".

Keywords: Truss, parametric modeling, structural simulation, optimization.

CAPITULO I: INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El acero es uno de los materiales de construcción de gran demanda hoy en día puesto que nos brinda ventajas como: Alta resistencia, uniformidad, rapidez de montaje, etc. El sistema estructural que se suele utilizar en estructuras de acero o estructuras metálicas es el sistema articulado, donde solo hay esfuerzos a compresión y tracción, dentro de este sistema estructural las estructuras más comunes son las armaduras metálicas, con las cuales se ha trabajado en esta investigación.

Al igual que los otros tipos de estructuras, las armaduras metálicas siguen un proceso de diseño estructural. Zapata (1997) define el proceso de diseño estructural como una mezcla de arte y ciencia que suele combinar los sentimientos intuitivos del ingeniero con principios de la estática, dinámica, mecánica de los materiales y el análisis estructural, para poder producir una estructura segura; Reboredo (1999) menciona que el proceso de diseño estructural es el proceso en el cual partiendo de los datos propios del objeto a construir permite proyectar un sistema estructural completo, estable, permanente y factible. También se puede definir al proceso de diseño estructural como un conjunto de pasos que permiten establecer de manera garantizada los elementos estructurales de cualquier tipo de infraestructura.

El proceso de diseño estructural que comúnmente se usa para diseñar armaduras metálicas, según Zapata (1997) consta de las siguientes etapas: (1)Planeamiento, (2)configuración preliminar estructural, (3)determinación de las cargas, (4)selección preliminar de los miembros estructurales, (5)análisis estructural, (6)evaluación, (7)rediseños, (8)decisión y (9)elaboración de planos de diseño con las especificaciones de trabajos, cabe resaltar que este es un proceso de prueba y error y es el proceso de diseño estructural tradicional o típico que suele usarse.

Por otro lado el proceso de diseño estructural tradicional de prueba y error descrito en el párrafo anterior, es un proceso que tiene muchas deficiencias, a pesar de que se puede valer de softwares que facilitan el análisis estructural y diseño (SAP2000, Etabs, Cypecad, etc.), el proceso es lento y monótono, si en la etapa (6) de evaluación la resistencia, esa resistencia no es mayor que las fuerzas internas obtenidas en el análisis estructural se tiene que pasar a la etapa (7) de rediseño, lo

que implica a la vez regresar a la etapa (2) de selección preliminar de los miembros estructurales y este ciclo se repite hasta que la resistencia pueda ser lo suficientemente mayor que las fuerzas internas calculadas, de acuerdo al código de diseño utilizado, esto genera gran demanda de tiempo, requiere de experiencia para encontrar una solución al problema estructural en un solo intento.

Sánchez (2012) menciona que dos proyectistas diferentes llegan a dos soluciones diferentes perfectamente factibles desde el punto de vista de la resistencia, estabilidad estructural o del cumplimiento de normativas de diseño, sin embargo, estas estructuras seguramente tendrán costos y pesos diferentes. Bonelli y Gudiño (2016) afirman que es muy difícil saber cual es la solución mas adecuada desde el punto de vista económico o de ahorro de materiales, ya que se encuentran afectadas por una serie de decisiones subjetivas durante el proceso. El diseñador no puede asegurar que una propuesta seleccionada sea la óptima, ya que le es físicamente imposible evaluar el espectro completo de posibles soluciones (Velázquez y Santillán, 2013).

El proceso de diseño estructural tradicional de prueba y error a pesar de contar con softwares de análisis estructural y diseño (como SAP2000, Etabs, Cypecad, etc.), que facilitan el proceso de diseño estructural, no permite: Analizar las armaduras en función de determinados parámetros (distancia entre articulaciones, el número de elementos o miembros, los perfiles empleados, etc.) de una manera rápida y dinámica y así obtener resultados comparables cuando modificamos algunos de estos parámetros, encontrar valores de los parámetros que generan estructuras más óptimas (estructuras con el menor peso), establecer simplificaciones en el proceso de dibujo, optimizar el material de estructuras complejas, acelerar trabajos repetitivos y tediosos, analizar la influencia que tiene sobre la capacidad portante de una estructura las variaciones geométricas sobre un modelo base, explicar qué factores afectan al diseño y de qué modo, y generar claves que induzcan la elaboración de diseños más lógicos (Alfamege, 2015). Las carencias anteriormente descritas se resumen como la ausencia de modelación paramétrica, simulación de comportamientos estructurales y optimización; Es así que se requiere contar con un proceso de diseño estructural integral para armaduras metálicas que comprenda a estos procesos ausentes antes mencionados, para reemplazar al típico proceso de diseño estructural de prueba y error lento, monótono y con las deficiencias y carencias ya antes descritas.

Diseñar correctamente una estructura implica optimizarla, es decir aprovechar al máximo el material disponible y consecuentemente usar la menor cantidad de este y ese siempre debería ser el objetivo del proceso de diseño estructural pero habitualmente no se lleva a cabo en la práctica por razones diversas, sobre todo debido al ingente esfuerzo que demanda. No basta, por lo tanto, satisfacer sólo los requerimientos de resistencia, estabilidad y morfológicos del código o norma de diseño empleada y de determinada propuesta arquitectónica inicial (Maltagliatti, 2016).

1.2.PROBLEMA DE INVESTIGACIÓN

¿Cómo se puede desarrollar un proceso de diseño estructural integral para armaduras metálicas que permita modelarlas en función a parámetros de forma sistematizada, realizar simulaciones de comportamientos estructurales y optimizarlas?

1.3.JUSTIFICACIÓN DE LA INVESTIGACIÓN

Esta investigación se justifica por que se ha desarrollado un proceso de diseño estructural integral para armaduras metálicas que permite modelarlas en función a parámetros de forma sistematizada, realizar simulaciones de comportamientos estructurales y optimizarlas; lo que consecuentemente permite crear una gran variedad de soluciones a partir de una solución base y elegir la que mejor se adecue a la arquitectura propuesta, evaluar el comportamiento estructural en función a determinados parámetros de la armadura y optimizar el peso de la armadura; lo que posteriormente se traduce en ahorro de tiempo y dinero, ya que acelera en gran manera el proceso de diseño estructural y aprovecha al máximo los recursos disponibles, brindando así la mejor calidad de estructura posible.

1.4.ALCANCES O DELIMITACIÓN DE LA INVESTIGACIÓN

La aplicación de esta investigación va dirigida a armaduras planas y tridimensionales, conformadas por perfiles metálicos tubulares.

1.5.LIMITACIONES

No se han presentado limitaciones en el desarrollo de esta investigación.

1.6.OBJETIVOS

1.6.1. Objetivo General

Desarrollar un proceso de diseño estructural integral para armaduras metálicas que permita modelarlas en función a parámetros de forma sistematizada, realizar simulaciones de comportamientos estructurales y optimizarlas.

1.6.2. Objetivos Específicos

- Modelar paramétricamente de forma sistematizada armaduras metálicas usando el software Rhinoceros 3D y su complemento Grasshopper, mediante programación gráfica de algoritmos.
- Simular comportamientos estructurales realizando el análisis estructural de cada una de las armaduras metálicas obtenidas mediante variación de parámetros, de una manera rápida y dinámica, usando el complemento Karamba de Grasshopper, el cual realiza el análisis estructural mediante el método de elementos finitos.
- Optimizar el peso de las armaduras metálicas, las cuales deben estar previamente modeladas paramétricamente, usando los complementos Octopus y Galápagos de Grasshopper; lo que implica desarrollar scripts (componentes o herramientas personalizadas) en Grasshopper, para optimizar teniendo en cuenta los requisitos de diseño de las “Especificaciones para Construcciones de Acero Estructural” (AISC 360 -10) del AMERICAN INSTITUTE OF STEEL CONSTRUCTION (AISC, 2010).

1.7.ORGANIZACIÓN DEL TRABAJO

- CAPÍTULO I: INTRODUCCIÓN. Contiene el planteamiento del problema de investigación, el problema de investigación, la justificación de la investigación, los alcances o delimitación de la investigación, las limitaciones y los objetivos.
- CAPÍTULO II: MARCO TEÓRICO. Contiene los antecedentes teóricos y las bases teóricas de la investigación.
- CAPÍTULO III: MATERIALES Y MÉTODOS. Contiene el procedimiento, el tratamiento y análisis de datos y la presentación de resultados.
- CAPÍTULO IV: ANÁLISIS Y DISCUSIÓN DE RESULTADOS. Describe, explica y discute los resultados de la investigación.
- CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.

CAPITULO II: MARCO TEÓRICO

2.1.ANTECEDENTES TEÓRICOS

- Cristian Namoncura y Manuel Vásquez en el 2011 elaboraron una tesis titulada "OPTIMIZACION DE FORMAS ARQUITECTONICAS CON ALGORITMOS GENETICOS OTAG", en la cual desarrollaron el software "OTAG" fundamentado en algoritmos genéticos, teniendo como base el funcionamiento del software Rhinoceros 3D y sus complementos Grasshopper y Galápagos, para optimizar formas arquitectónicas e incorporar tempranamente aspectos funcionales constructivos al proceso de diseño de formas arquitectónicas.
- Jorge Alberto Loja Gonzales en el 2014 elaboró una tesis titulada "ALGORITMOS GENÉTICOS COMO HERRAMIENTAS EN EL PROCESO DE DISEÑO GENERATIVO" que consiste en la exploración de un nuevo proceso de diseño arquitectónico basado en la utilización de sistemas generativos y algoritmos genéticos como técnica de optimización y búsqueda avanzada, para lo cual se usó el software Rhinoceros 3D con sus complementos Grasshopper y Galápagos. Concluyó que las herramientas de diseño generativo proporcionan un medio constante de evaluación de propuestas y optimizan la etapa de diseño, y que los algoritmos genéticos son una fuerte fuente de resolución de problemas complejos y potentes herramientas de optimización.
- Rodrigo Jarquín Laguna en 2014 elaboró una tesis titulada "APLICACIÓN DE ALGORITMOS GENÉTICOS EN INGENIERÍA CIVIL", en la cual desarrolló 5 ejemplos para optimizar soluciones en proyectos carreteros, de agua potable y estructuras usando algoritmos genéticos. Concluyó que el uso de algoritmos genéticos como procedimiento de búsqueda de soluciones es muy eficiente y que la gran virtud que tienen los algoritmos genéticos es su alta adaptabilidad en la resolución de problemas de distintas áreas.
- Samuel Sánchez Caballero en 2012 elaboró una tesis titulada "OPTIMIZACIÓN ESTRUCTURAL Y TOPOLÓGICA DE ESTRUCTURAS MORFOLÓGICAMENTE NO DEFINIDAS MEDIANTE ALGORITMOS GENÉTICOS", con el objetivo de desarrollar un algoritmo eficiente y robusto que sea capaz de generar un conjunto de estructuras óptimas en base a las cargas que estas tengan que soportar y los apoyos que las sustenten, sin partir de ninguna forma, geometría, regla o modelo preconcebido para las mismas, sujetas a ciertas

restricciones de resistencia y rigidez. Concluyó que la definición previa de la forma, geometría, regla o modelo preconcebido en una estructura son restricciones del diseño en sí mismas y por lo tanto aquel algoritmo que no se encuentre sujeto a estas puede generar diseños necesariamente mejores, o al menos tan buenos como los existentes.

- Matías Bonelli en 2016 elaboró un artículo científico titulado "OPTIMIZACIÓN ESTRUCTURAL DE ARMADURAS PLANAS POR MEDIO DE ALGORITMOS GENÉTICOS", donde optimizó la armadura de una nave industrial existente con el Algoritmo genético CeFeLix, logrando una reducción cercana al 25% del peso de la armadura.
- Aitor Almaraz en 2015 elaboró una tesis titulada: "OPTIMIZACIÓN EVOLUTIVA DE ESTRUCTURAS PARAMÉTRICAS", donde exploró las posibilidades de implementar el diseño paramétrico como una herramienta que vincule decisiones proyectuales espaciales, estructurales y de costes, en términos económicos y energéticos, para analizar y determinar su viabilidad y rendimiento al ser incorporados en las primeras etapas del proyecto. A través de varios estudios de casos, analizó la validez de los resultados comparándolos con aquellos obtenidos mediante herramientas ampliamente estudiadas y avaladas empleadas en el cálculo estructural profesional. Concluyó que las herramientas de diseño paramétrico han traído una posibilidad innegable de explorar nuevas formas en que las estructuras pueden fomentar y reforzar activamente el concepto de diseño.
- Stuart Mackenzie Jones en 2012 elaboró una tesis titulada: "VISUALIZACIÓN DE SISTEMAS DE PAISAJE CON MODELADO PARAMÉTRICO" donde dio a conocer una visión general de las herramientas paramétricas y el pensamiento paramétrico, y describió cómo se aplican las herramientas paramétricas en el diseño del entorno construido y los sistemas de paisaje. Concluyó que las herramientas paramétricas ofrecen a los arquitectos paisajistas nuevas formas de sintetizar datos de diferentes disciplinas, a medida que se dispone de más datos sobre nuestro entorno construido, la capacidad de manipular y dar sentido a esos datos se vuelve cada vez más importante; las herramientas paramétricas son únicas en su capacidad para combinar interfaces de diseño visual con el poder de los scripts de computadora.

2.2.BASES TEÓRICAS

2.2.1. Proceso de diseño estructural propuesto

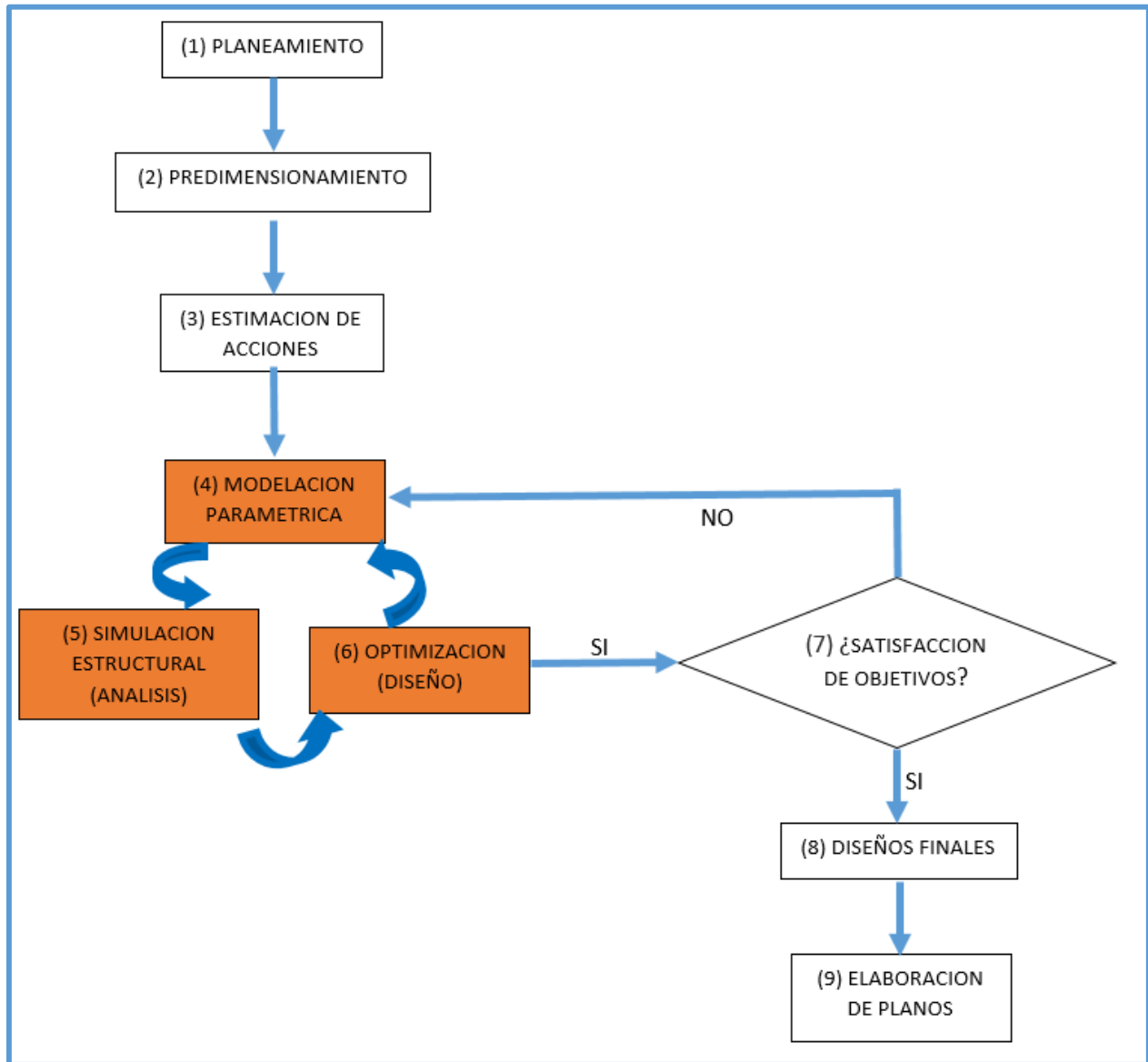


Figura 2.1. Proceso de diseño estructural propuesto.

A continuación se describe en que consiste cada una de las etapas del proceso de diseño estructural propuesto:

(1)Planeamiento: Etapa en la cual se crea un concepción inicial del proyecto a desarrollar, se identifican los problemas, deficiencias y fortalezas existentes.

(2)Predimensionamiento: Esta etapa está referida a la disposición de una armadura metálica en la estructura de la cual formara parte, por ejemplo, las armaduras metálicas en su gran mayoría se usan en coberturas metálicas, entonces se define el espaciamiento o ubicación de cada armadura (teniendo en cuenta las recomendaciones dadas por las diversas referencias bibliográficas afines) como parte de una cobertura metálica, determinando también así su ancho de influencia.

(3)Estimación de acciones: Se calculan las cargas que actuarán sobre la armadura de acuerdo a la norma o código de diseño usado.

(4)Modelación paramétrica: Se parametriza una forma o configuración estructural elegida para una armadura mediante una programación gráfica de algoritmos, es decir, en esta etapa se define una armadura en función a parámetros geométricos como luz a cubrir, peralte, número de segmentos de brida (Angulo de triangulación o Angulo entre montantes y bridas), flecha, etc. La modelación paramétrica es la base de la simulación estructural y la optimización ya que se evalúa comportamientos estructurales de una armadura cuando varían los parámetros que la definen (peralte, flecha, Angulo entre montantes y bridas, etc.) y luego se optimizan esos parámetros.

(5)Simulación estructural: Una vez realizada la modelación paramétrica de la armadura se procede a realizar el respectivo análisis estructural para calcular las fuerzas internas, deformaciones, deflexiones, etc. Se simula los comportamientos estructurales de manera rápida y dinámica realizando el análisis estructural de las distintas armaduras generadas mediante variación de parámetros.

(6)Optimización: Se optimiza el peso de la armadura aplicando tres tipos de optimización (optimización de tamaño, optimización geométrica y optimización topológica), mediante el complemento Karamba y herramientas fundamentadas en algoritmos genéticos (complemento Octopus y complemento Galápagos); y en simultaneo se la diseña, ya que la optimización se realizará teniendo en cuenta las condiciones de resistencia establecidas en la norma o código de diseño usado (en esta investigación se usó la AISC 360-10).

*Cabe mencionar que las etapas 4, 5 y 6 están fundamentadas en programación gráfica de algoritmos, consecuentemente están sistematizadas, pero no solo en forma individual sino también en forma conjunta, cada una de estas etapas y la sistematización se explica con mas detalle en las secciones consecutivas.

(7)Decisión: Se evalúa si el diseño satisface con los objetivos planteados, de ser el caso se pasa a la siguiente etapa, sino se regresa a la etapa (4).

(8)Diseños finales: Se diseña los componentes accesorios de la armadura: Placas base, pernos de anclaje, soldaduras, etc.

(9)Elaboración de planos: Definido el diseño de la armadura, se procede a elaborar los planos con sus respectivas especificaciones técnicas.

Las etapas 4, 5 y 6 son la novedad en este proceso de diseño estructural propuesto, es por eso que a continuación se las explica con más detalle.

2.2.2. Modelación paramétrica

La modelación es una representación o idealización de objetos reales o ficticios y por su parte la parametrización es definir un sistema en función a parámetros. Entonces modelación paramétrica vendría a ser la representación o modelación en función a parámetros.

Una ventaja que nos brinda Rhinoceros 3D y su complemento Grasshopper es realizar la modelación paramétrica referida a arquitectura, estructuras, industria, etc. En el ámbito de la arquitectura la modelación paramétrica mayormente se le conoce como diseño paramétrico puesto que encontrar la distribución del espacio y crear las distintas formas de una infraestructura (es decir se diseña el espacio creando un modelo en función a parámetros) es el objetivo de la arquitectura, es así que modelación paramétrica vendría a ser lo mismo que diseño paramétrico en el campo de la arquitectura.

La mayoría de bibliografía de modelación paramétrica con Rhinoceros 3D y su complemento Grasshopper está referida a la arquitectura y consecuentemente se encuentra con el nombre de diseño paramétrico (por lo ya indicado en el párrafo anterior), pero eso no impide que esta bibliografía pueda ser usada en otras ramas como las estructuras, ya que otra ventaja que nos brinda Rhinoceros 3D y su complemento Grasshopper es modelar estructuras como si se tratase de arquitectura.

2.2.2.1. Rhinoceros 3D: Herramienta de modelado formas arquitectónicas

Rhinoceros es un programa de modelado NURBS en 3D para Windows. Con Rhino puede modelar lo que sea, desde una válvula para el corazón hasta el casco de un barco, desde un ratón hasta un monstruo. Rhino le proporciona un ambiente de trabajo flexible, preciso y rápido. Se pueden modelar y renderizar objetos que antes solamente se podían crear mediante software y hardware mucho más caro. Rhino es fácil de aprender y de utilizar. Con Rhino podrá crear curvas de forma libre, superficies y sólidos. Tendrá total libertad para crear cualquier modelo a su gusto (McNeel, 2001).



Figura 2.2. Logo del software Rhinoceros 3D. Adaptado de (McNeel, Rhinoceros diseñar, modelar, presentar, analizar, realizar, 2017).

NURBS, B-splines racionales no uniformes, son representaciones matemáticas de geometría en 3D capaces de describir cualquier forma con precisión, desde simples líneas en 2D, círculos, arcos, o curvas hasta los más complejos sólidos o superficies orgánicas de formas libres en 3D (McNeel, 2001).

Gracias a su flexibilidad y precisión, se pueden utilizar modelos NURBS en cualquier proceso, desde la ilustración y animación hasta la fabricación (McNeel, 2001).

Rhinoceros posee su propio lenguaje interpretado de nombre Rhinoscript, el cual tiene su núcleo en visual script. Permite trabajar con variables, codificar algoritmos genéticos, usar las distintas funciones dentro del programa e intervenir con su interfaz, dando la posibilidad de crear una aplicación sobre esta herramienta y potenciar su funcionalidad. Este lenguaje ha tenido una explosión enorme dentro del área, tanto a nivel profesional como académico y cada vez se suman más problemas resueltos con este tipo de herramientas. En la red de internet se encuentran diversos recursos para el aprendizaje y desarrollo de códigos basados en Rhinoscript (Loja, 2014).

2.2.2.2. Grasshopper: Complemento de Rhinoceros para el modelado generativo de formas arquitectónicas

Grasshopper es un editor gráfico de algoritmos estrechamente integrado con las herramientas de modelado 3-D de Rhino. A diferencia de RhinoScript, Grasshopper no requiere conocimiento de programación o scripting, permite a los diseñadores construir generadores de formas desde lo simple hasta lo impresionante (Davidson, 2017).



*Figura 2.3. Logo del Complemento Grasshopper.
Adaptado de (Davidson, 2017).*

Grasshopper permite a los diseñadores definir las reglas de cómo un diseño responderá a cambios en las variables de diseño, así como explorar los diversos resultados de diferentes escenarios. Si bien Grasshopper se describe como una herramienta para explorar nuevas formas, hay una variedad de herramientas adicionales que mejoran la funcionalidad del programa. Las herramientas adicionales incluyen modelado de interacciones físicas, lectura de datos de sensores en tiempo real, importación de datos climáticos, computación evolutiva, así como automatizaciones celulares y simulaciones basadas en agentes (Mackenzie, 2012).

Grasshopper es una herramienta similar a Rhinoceros, pero tiene la particularidad de ser más amigable para el usuario, contiene diversos componentes con variadas funcionalidades y trabaja con parámetros en su programación de diversas funciones con múltiples objetivos. Permite la representación de formas arquitectónicas que pueden variar su apariencia, según los criterios seleccionados (Loja, 2014).

En Grasshopper, los parámetros contienen datos variables del algoritmo, los componentes contienen los diferentes pasos a seguir con las acciones que cumplan una función. La principal característica de Grasshopper es que posee un complemento llamado Galápagos que posibilita trabajar con algoritmos genéticos sobre una interfaz gráfica (Loja, 2014).

2.2.2.3. Modelación paramétrica aplicada a armaduras metálicas

Si se aplica la modelación paramétrica al diseño de las armaduras metálicas, entonces se puede decir que se puede definir una armadura en función a parámetros geométricos, por ejemplo, en caso de una armadura tipo Howe se puede definirla en función al número de segmentos en que se divide (o en función al ángulo que forman sus diagonales y bridas) y al peralte de esta.

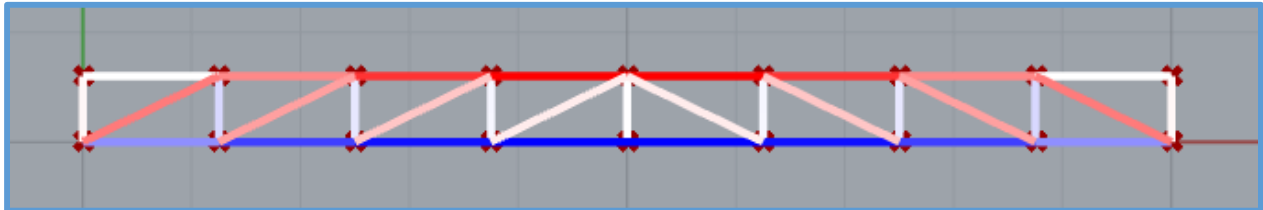


Figura 2.4. Armadura Howe parametrizada en función del número de segmentos (o ángulo entre diagonales y bridas) y peralte.

Para realizar la modelación paramétrica se ha usado Rhinoceros 3D y su complemento Grasshopper, mediante una programación gráfica de algoritmos en Grasshopper se realiza la modelación paramétrica de una armadura, todo lo programado en Grasshopper se visualiza en Rhinoceros 3D.

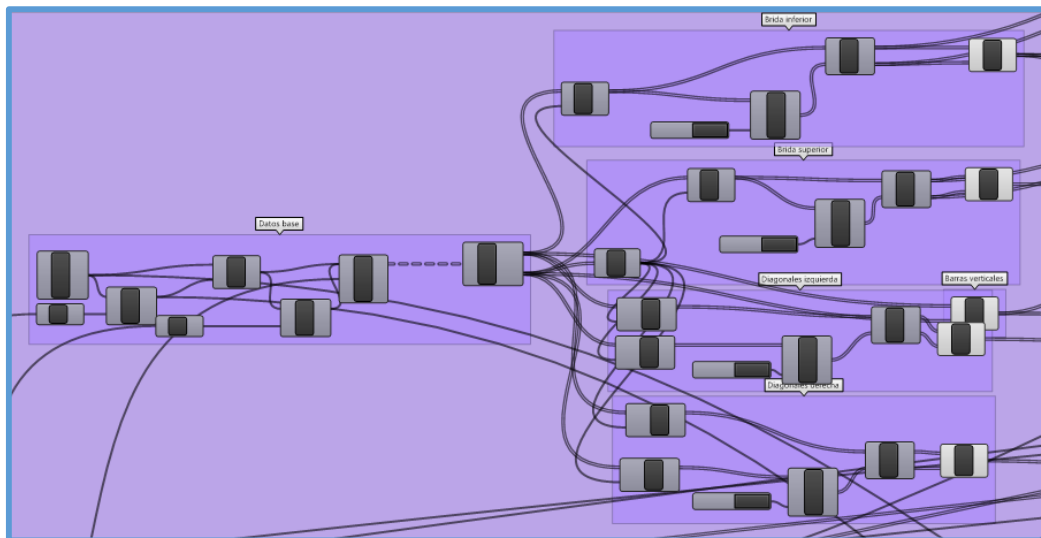


Figura 2.5. Código gráfico de armadura Howe mostrada en la figura 2.4.

La modelación paramétrica tiene muchas formas de aplicación debido a que se desarrolla con programación gráfica, entonces se puede adaptarla a las distintas necesidades o requerimientos que presente el usuario, así como a la capacidad creativa de este, es decir no existe patrones definidos para su aplicación. En esta investigación se ha desarrollado 3 ejemplos en los cuales se ha usado distintas formas de aplicación de la modelación paramétrica, las cuales pueden ser usadas por los

principiantes, como patrones básicos (guías o referencias), hasta que estos puedan generar sus propios modelos paramétricos.

La modelación paramétrica es la base de la simulación estructural y la optimización ya que se evalúa comportamientos estructurales de una armadura cuando varían los parámetros que la definen y se optimizan dichos parámetros.

En la modelación paramétrica se debe tener como resultado un modelo paramétrico en base a líneas rectas, si bien se puede apoyar de curvas NURBS, líneas rectas, curvas, etc.; para generar los modelos, es necesario tener líneas rectas en el resultado final del modelo paramétrico, puesto que Karamba trabaja con líneas rectas en las etapas de simulación estructural y optimización.

2.2.3. Análisis estructural

Entre el considerable número de plug-ins de Grasshopper, se encuentran varias herramientas de análisis estructural que se pueden implementar desde las primeras etapas del proyecto. Kangaroo y Karamba se destacarán específicamente debido a su alto rendimiento al calcular y mostrar los resultados de manera gráfica, de modo que el diseñador pueda ver los resultados al instante (Almaraz, 2015).



Figura 2.6. Logo del complemento “Karamba”. Adaptado de (Preisinger, 2017).

El complemento Karamba desarrollado por Clemens Preisinger en colaboración con los ingenieros de Bollinger + Grohmann ha sido desarrollado para predecir el comportamiento de las estructuras bajo cargas. Destinado a ser utilizado por los arquitectos en lugar de estar confinado únicamente a una configuración de ingeniería, habilita un flujo continuo de datos entre modelos estructurales y geométricos (Preisinger, 2013).

Karamba es un programa de elementos finitos (FE) para predecir comportamiento de estructuras bajo cargas externas. Aunque desarrollado en una empresa de ingeniería estructural, su enfoque principal no se encuentra en este campo. Está orientado al uso dentro de un entorno interactivo, entorno de diseño arquitectónico paramétrico. Uno de los objetivos era crear una herramienta rápida y liviana que facilitara un flujo de datos entre modelos estructurales y geométricos (Preisinger, 2013).

Karamba está completamente integrado en el entorno paramétrico de Grasshopper que es un complemento para la herramienta de modelado 3D Rhinoceros. Esto hace que sea fácil para combinar modelos geométricos parametrizados, cálculos de elementos finitos y algoritmos de optimización como Octopus o Galápagos (Preisinger, 2013).

Karamba genera un modelo que reacciona inmediatamente ante cualquier cambio en los parámetros de entrada, ayudando a comprender los mecanismos estructurales en el proceso de diseño (Almaraz, 2015).

La implementación de Karamba implica una respuesta dinámica del diseñador. Las consecuencias del diseño estructural se muestran instantáneamente (Almaraz, 2015).

El motor de programa permite realizar cálculos usando vigas y cáscaras, aplicando cualquier condición de carga y determinando, entre otros, materiales y secciones transversales. También incluye varios algoritmos de cálculo, como el análisis de primer orden y segundo orden o el algoritmo Beam Evolutive Structural Optimizers (BESO) (Preisinger, 2013).

2.2.3.1. Simulación estructural

El complemento Karamba integrado en Grasshopper, permite realizar el análisis estructural de las armaduras y también permite realizar simulaciones de comportamientos estructurales de las armaduras a medida que se cambian los parámetros que la definen, de una forma muy rápida y precisa. Con solo cambiar la magnitud en los “sliders” o controles de magnitud deslizantes (figura 2.7b) se puede modificar la armadura.

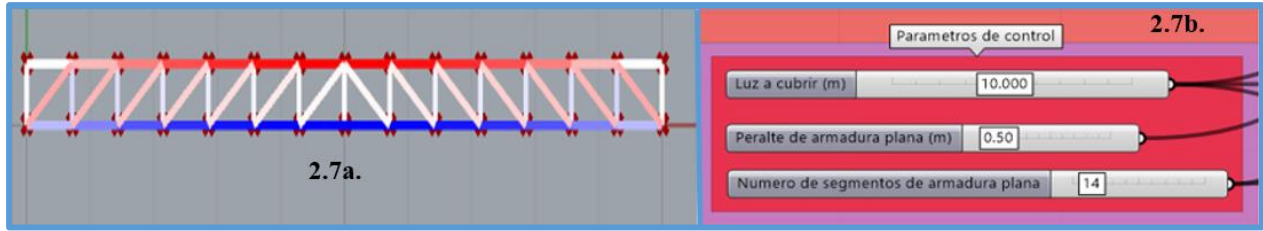


Figura 2.7. Armadura Howe definida por parámetros (2.7a) de: Luz a cubrir, peralte de armadura plana y número de segmentos de armadura plana (o ángulo entre diagonales y bridas). Sliders o controles de magnitud deslizantes (2.7b) que cambian los valores de la luz a cubrir, el peralte y el número de segmentos con solo escribir una cantidad que se encuentre dentro de sus rangos en las cajas de texto o deslizándolas.

2.2.4. Optimización estructural con algoritmos evolutivos

2.2.4.1. Optimización estructural

Christensen y Klarbring (2009) mencionan que un problema de diseño es formular y encontrar una respuesta a una pregunta apropiada (la función objetivo) dentro de los límites del espacio de diseño (las restricciones), cualquier tipo de problema de optimización se puede definir de la siguiente manera:

$$\begin{aligned} &\text{Minimizar: } f(x) \\ &\text{Sujeto a : } g_j(x) \leq 0, j = 1, \dots, p \\ &\quad \quad \quad h_k(x) \leq 0, k = 1, \dots, r \end{aligned}$$

Donde f es la función objetivo para minimizar al encontrar el mejor diseño variable (vector x) en el espacio de búsqueda o diseño, al mismo tiempo que se cumple con la desigualdad y las restricciones de igualdad g y h . La cantidad y el tipo de variables de diseño y las restricciones determinan la complejidad del problema de optimización. Las variables pueden ser discretas (como una lista finita de posibilidades) o continuas. El mínimo global es el valor más bajo de todas las soluciones factibles, mientras que un mínimo local es solo el valor más bajo en una región pequeña. En general, el objetivo es encontrar la solución óptima x_{opt} que conduce al mínimo global f_{min} del problema definido (Alegria, 2014).

El uso de técnicas de optimización estructural ganó interés debido a la necesidad para ahorrar materiales y reducir costos. Antes de la década de 1960 las estructuras se mejoraron por

consideraciones geométricas y matemáticas, experiencia y prueba y error. Después, el hardware computacional eficiente y poderoso estuvo disponible. Con estos, importantes desarrollos en técnicas numéricas ocurrieron favoreciendo métodos para la optimización. La optimización numérica se utilizó inicialmente en el campo de la ingeniería mecánica, pero hay un interés creciente en implementarla también en aplicaciones civiles y arquitectónicas. Para estas aplicaciones la optimización acelera el proceso de diseño debido a las incorporaciones de mejoras, ayuda en el descubrimiento de diseños innovadores y mejora de la producción masiva de estructuras repetitivas (Filomeno, Tysmans y Verwimp, 2013).

2.2.4.2. Tipos de optimización estructural

Christensen y Klarbring (2009) mencionan que los tres tipos principales de optimización estructural que se pueden aplicar en estructuras discretas (marcos espaciales) y en estructuras continuas (placas, coberturas) son:

- 1) Optimización de tamaño. La optimización del tamaño se refiere a las propiedades de la sección transversal de los elementos (figura 2.8a).
- 2) Optimización de forma (o geométrica). Determina la forma del límite o contorno del dominio estructural. Para estructuras discretas, esta es la posición de los nudos (figura 2.8b).
- 3) Optimización topológica. Determina el diseño de elementos, esto significa la existencia o ausencia de material en lugares específicos en el dominio estructural. En una formulación discreta de estructura de suelo, se refiere a la conectividad de los nodos (figura 2.8c).

La investigación presentada en esta tesis emplea los tres tipos de optimización anteriores.

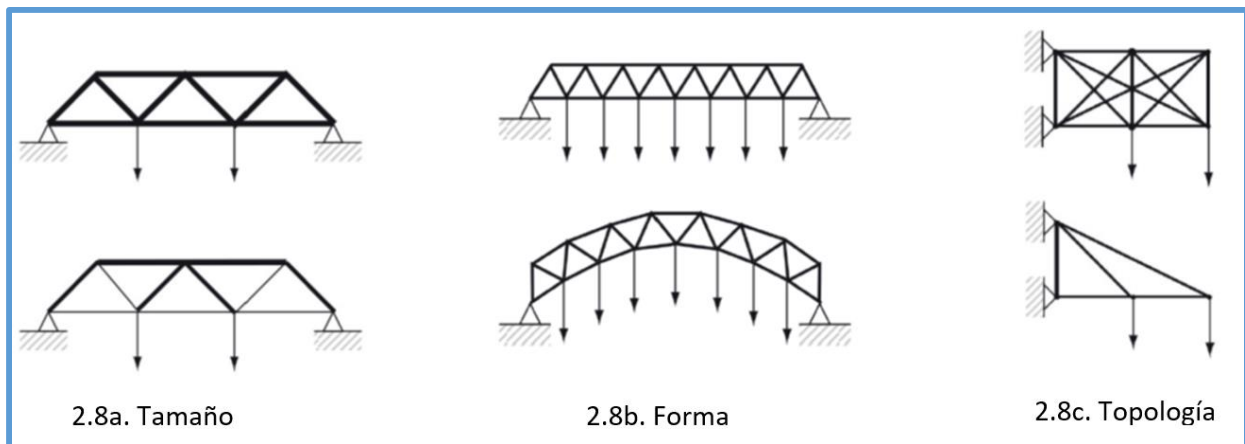


Figura 2.8. Los tres tipos de optimización estructural. Adaptado de (Richardson, 2010).

➤ **Optimización de tamaño**

Referida a la optimización de la sección transversal del elemento, en esta investigación se usó el complemento Galápagos para optimizar las secciones transversales de los elementos o miembros de una armadura, de tal manera que estas tengan un ratio de diseño según la AISC 360-10 de uno o lo más próximo a uno (acercándose a uno con valores menores a uno). Galápagos busca dentro de una lista de secciones transversales la sección que más se acomode para soportar las fuerzas internas según el diseño establecido en la AISC 360-10.

➤ **Optimización de forma (o geométrica)**

Tedeschi (2014) menciona que la optimización de forma u optimización geométrica es el proceso de calcular una geometría que minimiza la función de aptitud asignada dentro de las condiciones de contorno, menciona también que los elementos de una optimización geométrica son:

- Función aptitud. Comúnmente se define como un problema mínimo de investigación, varía de acuerdo a los requerimientos establecidos por el usuario.
- Definición de variable(s). Se define la(s) variable(s) a ser optimizada(s).
- Configuración de apoyos. Se establecen los tipos de apoyos de la estructura a optimizar.
- Definición de un dominio para la elaboración de una solución. Se define el dominio para la elaboración de una solución mediante la definición del rango de los distintos parámetros que generarán la solución óptima.
- En esta investigación se usó los complementos Octopus y/o Galápagos para realizar la optimización geométrica.

➤ **Optimización topológica con “Karamba”**

La optimización estructural evolutiva (ESO) constituye un método de optimización de topología que fue iniciado por Y.M. Xie y G.P. Steven. El principio subyacente es simple: se parte de un volumen dado formado por elementos estructurales en soportes predefinidos y con cargas preestablecidas que actúan sobre él (Preisinger, 2016).

El cálculo de la respuesta estructural mostrará que hay regiones que llevan más carga externa que otras. Ahora uno elimina un número de esos elementos de la estructura que están menos forzados y por lo tanto menos efectivos. De nuevo, la respuesta del modelo ahora diluido está determinada, los elementos infrutilizados se eliminan y así

sucesivamente. Este procedimiento iterativo se detiene cuando se alcanza un volumen objetivo o la cantidad de elementos estructurales restantes (Preisinger, 2016).

Con el objetivo de mejorar la capacidad de búsqueda del ESO original, la Optimización Estructural Evolutiva Bidireccional (BESO) tiene como objetivo agregar o eliminar elementos del modelo de elementos finitos de la estructura. En el ESO, debido a que los elementos ineficientes se eliminan por completo de la estructura, no hay información sobre los efectos de estos elementos en la función objetivo, en etapas posteriores de la optimización. La idea general de BESO es diseñar un esquema para restaurar los elementos eliminados, si es necesario. El enfoque BESO se puede ver como un desarrollo significativo que ha resultado de estudios en el ESO (Radman, 2013).

Todos los esquemas de BESO que se han introducido hasta ahora aplican la idea de estructura de suelo, en la cual sus elementos cubren todo el dominio de diseño, incluidas las regiones sólidas y vacías. El BESO enciende y apaga estos elementos, pero mantiene el registro de su información geométrica a través de todo el procedimiento de optimización (Radman, 2013).

Karamba cuenta con el componente "BESO for Beams" para realizar la Optimización Estructural Evolutiva Bidireccional (BESO), en la figura 2.9 se muestra dicho componente en pleno trabajo. En el lado izquierdo se puede ver la geometría inicial, que es una malla triangular derivada de una superficie. Existen dos casos de carga con cargas que actúan en el plano de la estructura en dirección horizontal y vertical, respectivamente. Tres nodos de esquina de la estructura se mantienen fijos. En el lado derecho se muestra la estructura optimizada reducida al 45% de su masa inicial en el transcurso de 20 iteraciones de diseño (Preisinger, 2016).

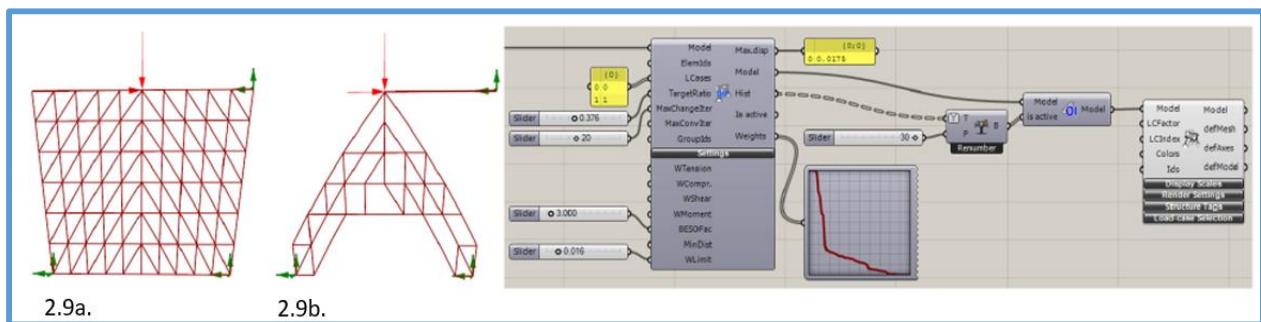


Figura 2.9. Malla triangular de vigas antes (2.9a) y después (2.9b) de aplicar el componente "BESO for Beams". Adaptado de (Preisinger, 2016).

2.2.4.3. Técnicas de optimización estructural

Sánchez (2012) menciona que como técnicas de optimización de estructuras tenemos a la programación matemática y las técnicas metaheurísticas, en esta investigación se trabajó en el campo de las técnicas metaheurísticas.

2.2.4.3.1. Técnicas metaheurísticas

Las técnicas metaheurísticas, a diferencia de las técnicas de optimización tradicionales, no siguen unos métodos o reglas preestablecidas de búsqueda. A pesar de no seguir ningún tipo de procedimiento deductivo son capaces de proporcionar soluciones buenas en un periodo de tiempo razonablemente corto. El principal inconveniente de este tipo técnicas es que no garantizan la localización del óptimo absoluto, por lo que para tener la certeza de haberlo obtenido (o al menos un punto muy próximo) debe ejecutarse el algoritmo varias veces. Estos métodos son de aplicación cuando no existe un algoritmo específico y determinístico para la resolución del problema que asegure la obtención del óptimo absoluto (Sánchez, 2012).

Sánchez (2012) menciona que dentro de las técnicas metaheurísticas que destacan en el campo de la optimización de estructuras tenemos al Recocido Simulado (SA) y la Computación Evolutiva(EC). En esta investigación se trabajó en el campo de la Computación Evolutiva.

2.2.4.3.2. Computación Evolutiva

La Computación Evolutiva (EC) es una rama de la inteligencia artificial, inspirada en los mecanismos de evolución biológica, que involucra problemas de optimización combinatoria. Está considerada como una técnica metaheurística y por lo tanto es adecuada para la resolución de problemas con espacios de búsqueda extensos, dispersos, no lineales y no convexos, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable (Sánchez, 2012).

La Computación Evolutiva es una rama de la computación emergente que engloba técnicas que simulan la evolución natural, y constituye un enfoque alternativo para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Parte de un hecho observado en la naturaleza: los organismos vivos poseen una destreza consumada en la resolución de los problemas que se les presentan, y obtienen sus habilidades, casi

sin proponérselo, a través del mecanismo de la evolución natural. La evolución se produce, en casi todos los organismos, como consecuencia de dos procesos primarios: la selección natural y la reproducción (cruce). La evolución natural es un proceso de cambio sobre una población reproductiva que contiene variedades de individuos con algunas características heredables y en donde algunas variedades difieren en su aptitud o éxito reproductivo (Mateos, 2004).

Sánchez (2012) menciona que existen 3 tipos de técnicas usadas en computación evolutiva: (1) Los Algoritmos Evolutivos, (2) La Optimización por enjambre de partículas (PSO) y (3) La Optimización por colonia de hormigas (ACO). En esta investigación se trabajó con los Algoritmos evolutivos.

2.2.4.3.3. Algoritmos Evolutivos

Este término es empleado para describir sistemas de resolución de problemas de optimización o búsqueda basados en el ordenador empleando modelos computacionales de algún mecanismo de evolución conocido como elemento clave en su diseño e implementación (Mateos, 2004).

Los algoritmos evolutivos trabajan con una población de individuos, que representan soluciones candidatas a un problema. Esta población se somete a ciertas transformaciones y después a un proceso de selección, que favorece a los mejores. Cada ciclo de transformación y selección constituye una generación, de forma que después de cierto número de generaciones se espera que el mejor individuo de la población esté cerca de la solución buscada. Los algoritmos evolutivos combinan la búsqueda aleatoria, dada por las transformaciones de la población, con una búsqueda dirigida dada por la selección (Mateos, 2004).

Mateos (2004) manifiesta que los principales componentes de los algoritmos evolutivos son:

- Población de individuos, que son una representación de posibles soluciones.
- Procedimiento de selección basado en la aptitud de los individuos para resolver el problema.
- Procedimiento de transformación para construir nuevos individuos a partir de los anteriores.

Sánchez (2012) menciona que los algoritmos evolutivos se clasifican en 4 tipos fundamentales: (1) Los Algoritmos Genéticos (GA), (2) La Programación Evolutiva (EP), (3) Las Estrategias

Evolutivas (ES) y (4) La Programación Genética (GP). En esta investigación se trabajó con los Algoritmos Genéticos.

2.2.4.3.4. Algoritmos Genéticos

Un Algoritmo Genético (AG) es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas. Dado un problema específico a resolver, la entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente (Marczyk, 2004).

Los algoritmos genéticos son algoritmos de búsqueda basados en los mecanismos de selección natural y genética natural. Combinan la supervivencia de los más compatibles entre las estructuras de cadenas, con una estructura de información ya aleatorizada, intercambiada para construir un algoritmo de búsqueda con algunas de las capacidades de innovación de la búsqueda humana. En cada generación, se crea un nuevo conjunto de criaturas artificiales (cadenas) utilizando partes del más fuerte de los antiguos, una parte nueva y ocasional se prueba por si acaso (Goldberg, 1989).

Aunque aleatorizados, los algoritmos genéticos no son simples caminos aleatorios. Explotan eficientemente la información histórica para especular sobre nuevos puntos de búsqueda con el rendimiento mejorado esperado (Goldberg, 1989).

Los algoritmos genéticos han sido desarrollados por John Holland, sus colegas y sus estudiantes en la universidad de Michigan. Los objetivos de su investigación han sido dos: (1) abstraer y explicar rigurosamente los procesos adaptativos de los sistemas naturales, y (2) diseñar un software de sistemas artificiales que conserve los importantes mecanismos de los sistemas naturales. Este enfoque ha llevado a importantes descubrimientos tanto en la ciencia de sistemas naturales como artificiales (Goldberg, 1989).

John H. Holland y sus colaboradores utilizaron inicialmente la codificación binaria, aunque en la actualidad también se ha extendido a la codificación con números reales. Emplea operadores

genéticos de selección, recombinación y mutación teniendo mayor relevancia los operadores de selección y recombinación (reproducción) (Sánchez, 2012).

El algoritmo genético trabaja en base a la evolución de los seres vivos contemplando diversas variables como mecanismos para llegar a generar individuos más adaptados al entorno que los rodea (Namoncuro y Vásquez, 2011).

Una de las características clave de los algoritmos genéticos es su proceso de adaptación, el cual no se realiza mediante el incremento de una estructura, sino que mantiene una población de estructuras, y a partir de ellas va generando nuevas. Esto permite explotar la información acumulada de un espacio de búsqueda desconocido, para guiar la próxima búsqueda a subespacios útiles. Es por eso, que los algoritmos genéticos necesitan de grandes tamaños de muestras para poder recabar la información necesaria, y así crear nuevos puntos de búsqueda (Jarquín, 2014).

Sánchez (2012) menciona que los Algoritmos Genéticos se basan en los siguientes principios de la Teoría de Selección Natural:

- Supervivencia del más apto.
- La evolución se genera durante la reproducción.
- Los hijos son generados mediante el cruce de los cromosomas de los padres.
- La mutación permite generar hijos diferentes de sus padres.

Según Goldberg (1989), los Algoritmos Genéticos son diferentes a otras técnicas porque:

- Utilizan funciones de aptitud en lugar de derivadas.
- Trabajan con un conjunto de parámetros codificados y no con los parámetros mismos.
- Su búsqueda parte de una población de puntos, en lugar de uno solo punto.
- Trabajan con reglas estocásticas (probabilísticas) de transición, no usan reglas deterministas.
- Sin restricción temporal, es matemáticamente posible obtener el máximo global.

2.2.4.4. Complemento Galápagos de Grasshopper

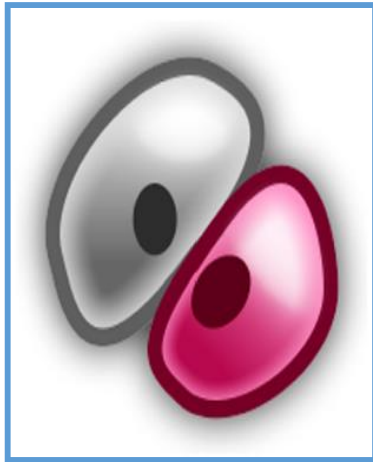


Figura 2.10. Logo de la herramienta Galápagos. Adaptado de (Rutten, 2017).

Galápagos es una herramienta o complemento que trabaja con algoritmos genéticos en el entorno de trabajo de Grasshopper.

Tedeschi (2014) afirma que Galápagos es un solucionador evolutivo (ES) heurístico que encontrará una solución a un problema expresado de una manera matemática: el problema debe ser descrito por una función objetivo que queremos minimizar o maximizar, o por un conjunto de variables definidas por un rango.

Por su parte Almaraz (2015) nos menciona que después de definir una “función de aptitud”, se le darán varios parámetros al solucionador evolutivo, Galápagos encontrará una solución que haga verdadera la declaración anterior ajustando los parámetros.

El trabajo que realiza Galápagos en palabras su creador David Rutten es: Primero trata de encontrar un terreno elevado prometedor, luego afina su posición para encontrar el pico más alto asociado con este terreno elevado. Los algoritmos evolutivos aplican los principios biológicos de mutación, selección y herencia. Ellos pueblan el paisaje con individuos virtuales y luego proceden a criar los más altos con la esperanza de que sus descendientes estén más cerca de una cumbre (Almaraz, 2015).

Lo que hace Galápagos es transformar o generar una figura con los parámetros y componentes, en base a la función objetivo va generando respuestas más óptimas o de mejor calidad (Namoncura y Vásquez, 2011).

Es necesario tener en cuenta que, en el trabajo con los solucionadores, la experiencia demuestra que cuando se trata de problemas complejos, a menudo la solución proporcionada por el solucionador no es exacta, sino cercana (Rutten, 2013).

Un algoritmo genético tiene su origen en la evolución de los seres vivos, contemplando diversas variables y mecanismos para generar individuos más adaptados al entorno que los rodea.

Galápagos adopta esta funcionalidad y la aplica en Grasshopper. Básicamente trabaja a partir de parámetros, los cuales va optimizando a medida que se acerca a un número de generaciones predeterminadas. Luego transforma y genera una figura formada por componentes de Grasshopper agregando la base del algoritmo genético con su función objetivo y los diferentes mecanismos de selección (Loja, 2014).

La forma en que se unen Galápagos y Grasshopper es muy sencilla, por ejemplo, basta con conectar Galápagos con algún(os) componente(s) de Grasshopper y una superficie visualizada en Rhinoceros será optimizada al hacer funcionar Galápagos. La particularidad que tiene Galápagos es que puede generar distintas formas de un objeto sobre Rhinoceros (Loja, 2014).

Como ya se mencionó Galápagos es fácil de usar, mayormente debido a la programación gráfica que ofrece de Grasshopper, es así que Rutten (2010) en la página web oficial de Grasshopper menciona: Es mi esperanza que Galápagos proporcione una plataforma genérica para la aplicación de Algoritmos Evolutivos para ser utilizados en una amplia variedad de problemas por los no programadores.

Tedeschi (2014) manifiesta que los solucionadores evolutivos heurísticos se usan cuando los problemas de optimización tienen una gran cantidad de variables y no se puede encontrar una solución óptima mediante solucionadores exactos y que el solucionador evolutivo integrado en Grasshopper, Galápagos, es un solucionador heurístico, el cual presenta a través del siguiente ejemplo:

Dada una superficie de forma libre reparametrizada, y dos puntos fijos A y B, ubicados en los bordes opuestos de la superficie, el solucionador de Galápagos se puede usar para calcular el camino más corto en la superficie que conecta los dos puntos.

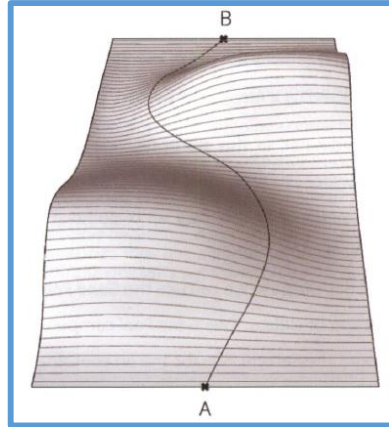


Figura 2.11. Trayectoria “A-B” a optimizar con Galápagos. Adaptado de (Tedeschi, 2014).

El primer paso en la definición de distancia más corta es describir una curva de ruta. La curva de trayectoria se define como una curva interpolada flexible que coincide con la superficie con los extremos A y B fijos. La geometría de la curva se controla mediante las coordenadas “uv” de un conjunto de puntos de interpolación P_i (P_0, P_1, P_n). Al cambiar las coordenadas de los puntos de interpolación, se producirán diferentes longitudes de curva. El solucionador heurístico se usa para encontrar las coordenadas de los puntos de interpolación que corresponden a la curva con la longitud mínima, entre todas las alternativas factibles.

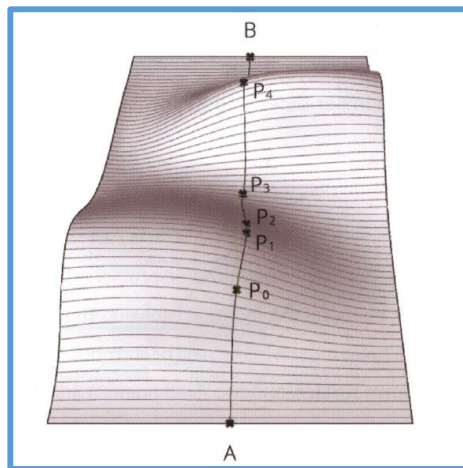


Figura 2.12. Puntos de interpolación para optimizar trayectoria “A-B” con Galápagos. Adaptado de (Tedeschi, 2014).

La curva de ruta se crea usando el componente Curva en la superficie (Curva > Spline). El componente “Curve on Surface” crea una curva interpolada en una entrada de superficie (S), a través de un conjunto arbitrario de puntos de entrada (uv) expresados en el LCS (sistema de

coordenadas local). En este caso, el componente “Curve on Surface” requiere tres conjuntos de coordenadas “uv”, que se recopilan mediante un componente Merge, cuyas entradas son:

D1: Son las coordenadas fijas del punto de inicio (A) convertidas al Sistema de Coordenadas Local a través del componente “Surface CP”.

D2: Son las coordenadas (uv) de un conjunto arbitrario de puntos (P) en la superficie. Para definir los puntos, se usa el componente Construir punto con coordenadas establecidas a través de controles deslizantes que varían entre 0 y 1. Para este propósito, podemos usar el conjunto de genes (Params> Util), un componente que integra una colección de controles deslizantes. Se utilizan dos componentes Gene Pool, el primero para las coordenadas (u) y el segundo para las coordenadas (v). La cantidad de controles deslizantes o Gene Count, así como el rango de los controles deslizantes y los dígitos significativos se pueden configurar haciendo doble clic en el componente Gene Pool. Cuanto mayor sea el número de controles deslizantes, mayor será la precisión de la curva resultante.

D3: Son las coordenadas fijas del punto final (B) convertidas al Sistema de Coordenadas Local a través del componente “Surface CP”.

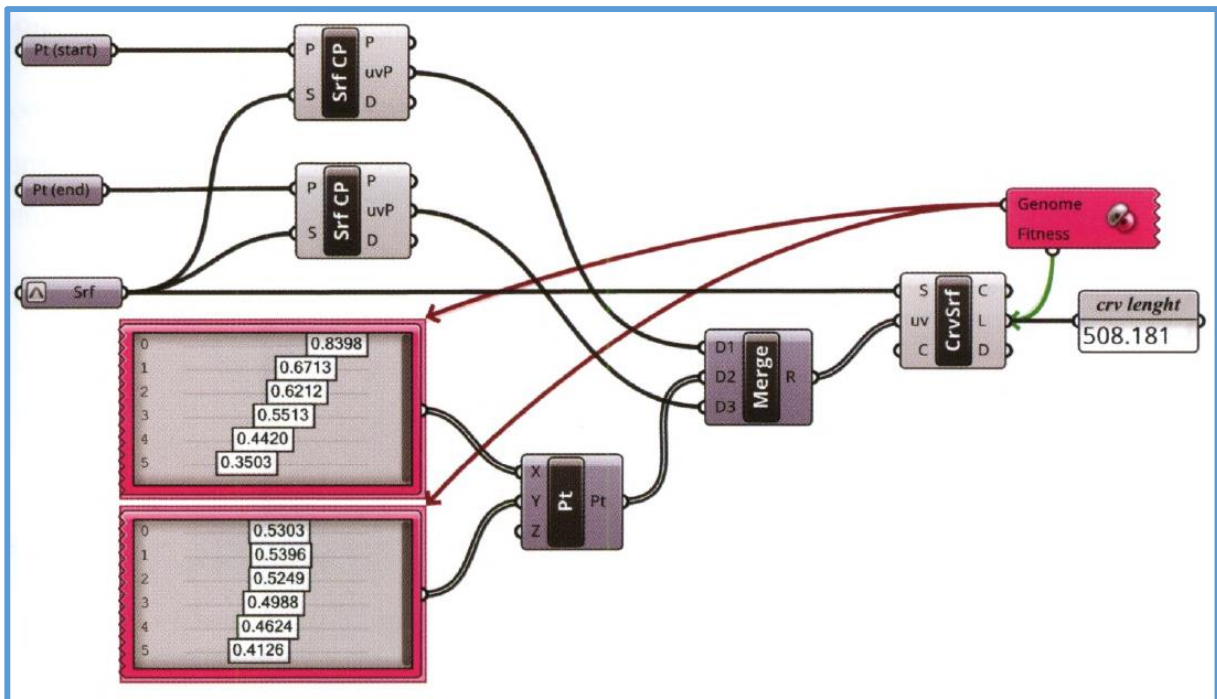


Figura 2.13. Código gráfico de optimización de la trayectoria “A-B” mostrada en la figura 2.11. Adaptado de (Tedeschi, 2014).

Como la curva en la salida de superficie (L) devuelve el valor de la longitud de la curva cuando las coordenadas (uv) cambian de posición, existen un elemento fijo y un elemento variable respectivamente, que se pueden usar para definir un procedimiento de optimización para un solucionador evolutivo. Para esta optimización se usa Galápagos (Params> Util). Galápagos tiene dos puertos para conexiones salientes: Genome que corresponde a Variables, y Fitness que corresponde al Objetivo. El puerto Genome se satisface con el conjunto arbitrario de controles deslizantes incrustados en los dos componentes Gene Pool. El puerto Fitness está conectado a la salida “L” del componente “Curve on Surface”.

Al hacer doble clic en el componente de Galápagos, se abre un menú contextual, donde se puede configurar o minimizar el estado físico.

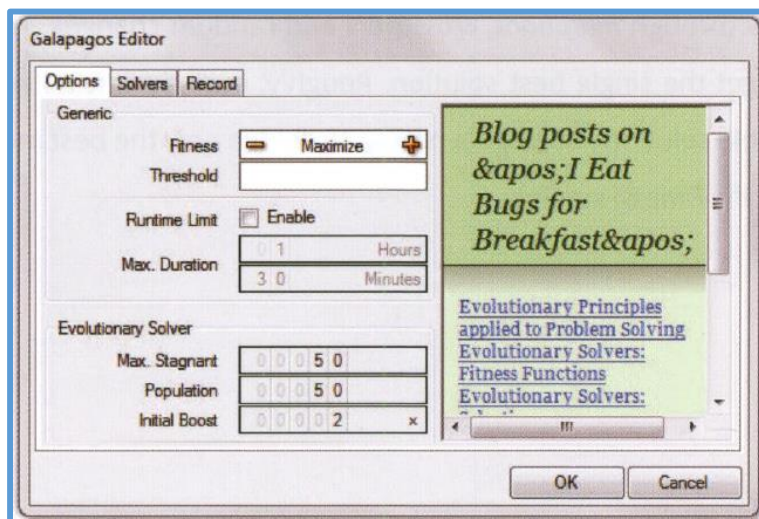


Figura 2.14. Menú contextual de Galápagos. Adaptado de (Tedeschi, 2014).

Dado que, se desea la longitud más corta, la opción de Aptitud está configurada para Minimizar. Para realizar la optimización, se abre la pestaña “Solvers” y se selecciona el botón “Start Solver”.



Figura 2.15. Componente Galápagos en pleno trabajo de optimización. Adaptado de (Tedeschi, 2014).

El proceso de optimización se ejecuta hasta que se encuentra un óptimo, que puede tomar minutos u horas dependiendo de la complejidad de la solución. Galápagos puede detenerse seleccionando el botón “Stop Solver” si el primer valor que se muestra en la ventana inferior derecha (V1 en la figura anterior) se estabiliza, tendiendo a un número específico. En este ejemplo, la optimización devuelve un conjunto de controles deslizantes que representan las coordenadas “uv” de los puntos de interpolación que corresponden a la curva de trayectoria mínima. La curva más corta es devuelta por la salida C del componente “Curve on Surface”.

Los solucionadores evolutivos aplican los principios de la evolución a la resolución de problemas, y en realidad se basan en la población de soluciones candidatas. Los candidatos intentan alcanzar mejores soluciones con mejores características genéticas a través de mutaciones, cruces y cambios aleatorios. Esto está en contraste con los solucionadores exactos que se dirigen a la mejor solución única. Aproximadamente, los solucionadores evolutivos realizan una selección natural entre posibles soluciones a través de un proceso en el que solo los mejores miembros de la población (de acuerdo con la “Aptitud” específica) sobreviven (Tedeschi, 2014).

➤ Valores objetivo

Los solucionadores tanto exactos como evolutivos operan minimizando o maximizando una función de Aptitud Física. En algunos casos, se desea un valor objetivo para la función Fitness; en estos casos, la diferencia de valor absoluto entre la función de Aptitud y el valor objetivo se minimiza y maximiza (Tedeschi, 2014).

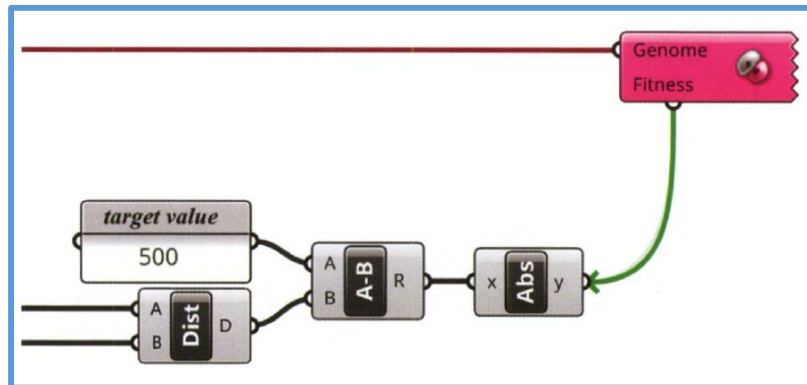


Figura 2.16. Uso de valores objetivo en la optimización con Galápagos. Adaptado de (Tedeschi, 2014).

En la definición anterior, la aptitud se define como la distancia entre dos puntos. En este caso, el valor objetivo es 500, el valor absoluto de la distancia menos el valor objetivo se minimiza utilizando la función de acondicionamiento físico para minimizar (Tedeschi, 2014).

Para entender mejor los valores objetivo a continuación se muestra un post publicado por Rutten (2011) en la página web de Grasshopper acerca de este tema, así en sus propias palabras tenemos:

Lo que debe apuntar a hacer es 'normalizar' todos sus componentes de acondicionamiento físico. Es decir, asegúrese de que la peor condición física posible y la mejor forma física posible para cada componente tengan el mismo valor.

Digamos que usted tiene sus tres componentes como se ha mencionado antes, A , B y C . A se debe minimizar, B se debe maximizar y C se debe optimizar a 15. Además, los valores posibles de A pueden variar entre 10 y 500, B puede variar entre 0,1 y 0,8 y C puede estar en cualquier lugar entre 5 y 60.

Entonces, la mejor aptitud posible será $\{A = 10, B = 0.8, C = 15\}$ y la peor aptitud posible será $\{A = 500, B = 0.1, C = 60\}$.

En forma de tabla:

$A \{min = 10; máx = 500; rango = 490; objetivo = 10\}$

$B \{min = 0.1; máx = 0.8; rango = 0.7; objetivo = 0,8\}$

$C \{min = 5; máx = 60; rango = 55; objetivo = 15\}$

El valor del rango es importante aquí porque te dice la 'fortaleza' de la variable en la aptitud total. En general, su objetivo es que todas las variables sean aproximadamente igual de fuertes. Lo que significa que nuestra función de acondicionamiento físico debe tener factores de ponderación, de modo que los componentes de la función de acondicionamiento físico sean todos $\{0.0, 1.0\}$. La función anterior (sin ponderación) se veía así:

$$f = -A + B - Abs(C - 15)$$

La nueva función podría verse así:

$$f = -((A-10)/490) + ((B-0.1)/0.7) - Abs((C-15)/55)$$

Las 'reglas' podrían resumirse aproximadamente de la siguiente manera, supongo:

- El signo delante de cada variable indica si queremos maximizar (+), minimizar (-) u optimizar (-) la variable.
- Si se va a optimizar una variable, entonces la aptitud se define como $Abs(x - c)$, donde x es la variable y c es el valor objetivo. (Es decir, la optimización equivale a la minimización de la *diferencia* entre la variable y el objetivo, por lo tanto, el símbolo menos).
- Las variables deben estar 'centradas' en cero (o en cualquier otro valor numérico constante, pero el cero es más fácil), por lo que restan el valor más bajo posible de la variable.
- Las variables necesitan ser normalizadas al dominio $(0 \sim 1)$ (o cualquier otro dominio constante, pero $0 \sim 1$ es más fácil), así que divide la variable centrada por el rango del dominio.

Suponiendo que la progresión de la aptitud es lineal (que no se da para nada), los rangos de aptitud antes de la normalización se ven así:

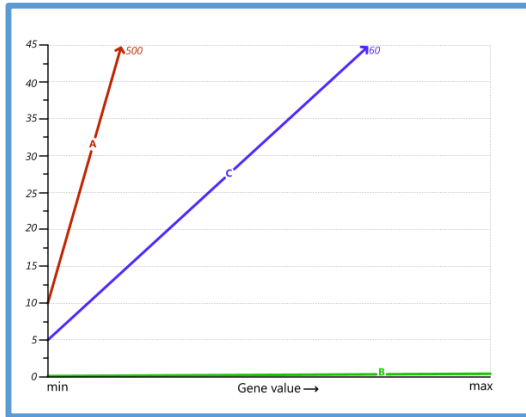


Figura 2.17. Rangos de aptitud sin factores de ponderación de normalización. Adaptado de (Rutten, 2011).

y los factores de ponderación de normalización los acercan uno al otro.

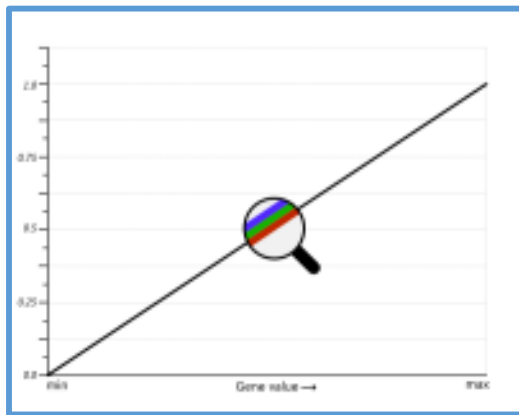


Figura 2.18. Rangos de aptitud con factores de ponderación de normalización. Adaptado de (Rutten, 2011).

2.2.4.5. Complemento Octopus de Grasshopper

Octopus es un complemento para aplicar principios evolutivos al diseño paramétrico y la resolución de problemas. Permite la búsqueda de muchos objetivos a la vez, produciendo una gama de soluciones optimizadas de compensación entre los extremos de cada objetivo (Vier, 2012).

Octopus es parte de una gama de herramientas desarrolladas en la Universidad de Artes Aplicadas de Viena y Bollinger + Grohmann Engineers, presenta múltiples valores de aptitud para la optimización. Se buscan las mejores soluciones de compromiso entre esos objetivos, produciendo un conjunto de posibles soluciones óptimas que, idealmente, abarcan desde un compromiso extremo hasta el otro (Vier, 2012).



Figura 2.19. Logo del complemento Octopus. Adaptado de (Vier, 2012).

Para más detalles de este complemento se recomienda revisar su manual el cual se encuentra en el sitio web de Rhino.

2.2.5. Sistematización con Grasshopper

Grasshopper es un entorno de programación visual dentro de Rhino. Su principal virtud es que nos permite diseñar algoritmos que operan con datos y geometría de manera intuitiva sin la necesidad de aprender ningún lenguaje de programación. Al funcionar dentro de Rhino, Grasshopper incorpora gran parte de sus comandos y funcionalidades, además, utiliza Rhino como entorno de visualización. De manera que podemos ver en tiempo real como afecta a nuestro modelo cualquier cambio que realicemos en Grasshopper (Vidal y Ramírez, 2011).

Grasshopper es una herramienta de diseño de algoritmos. Esto quiere decir que es capaz de encadenar una sucesión de comandos e instrucciones con entradas y salidas que son geometría y datos para producir un resultado. Esta manera de operar aporta gran flexibilidad de cara a manipular grandes cantidades de datos o datos variables en el espacio y el tiempo, pero condiciona enormemente nuestra manera de trabajar (Vidal y Ramírez, 2011).

Contrariamente a un problema de diseño, un algoritmo tiene que estar bien definido: a diferencia de los humanos, las computadoras no pueden adivinar en base a la experiencia y la intuición. Por lo tanto, cada paso debe estar completamente determinado sin ambigüedades por los pasos anteriores. Afortunadamente, siempre hay formas casi infinitas de cómo los algoritmos matemáticos y geométricos pueden describir un modelo (Almaraz, 2015).

El Visual Programming, o programación visual, es un paradigma de la programación dentro del que el usuario manipula elementos lógicos gráficamente en vez de textualmente. Algunos de los más conocidos lenguajes de programación textual tales como C#, Visual Basic, Processing - y los más cercanos a Rhino - Python y Rhinoscript, requieren de nosotros escribir código que se atiene a una sintaxis muy específica del lenguaje. En cambio, la programación visual nos permite conectar bloques funcionales a una secuencia de acciones donde la única "sintaxis" requerida es que los inputs de los bloques reciban datos del tipo adecuado, e idealmente, que esté organizada de acuerdo al resultado deseado. Esta característica de la programación visual evita la barrera de acceso comúnmente hallada al tratar de aprender un nuevo lenguaje, incluso uno hablado, y previsualiza la interfaz, lo cual para los diseñadores ubica a Grasshopper dentro de un territorio más familiarizado (Akos y Parsons, 2015).

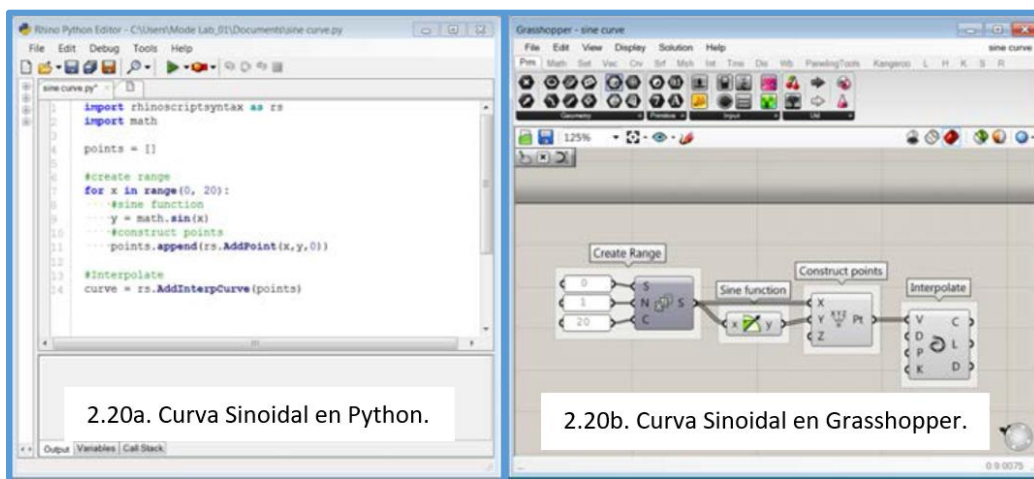


Figura 2.20. Esta figura compara el proceso para dibujar una curva sinoidal en Python y en Grasshopper. Adaptado de (Akos y Parsons, 2015).

La programación orientada a objetos es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos. Los objetos son entidades que tienen un determinado estado, comportamiento (método) e identidad (Vidal y Ramírez, 2011).

El estado está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos). El comportamiento está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, que operaciones se pueden realizar con él. La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante) (Vidal y Ramírez, 2011).

Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa del estado y el comportamiento (Vidal y Ramírez, 2011).

2.2.5.1. Nociones básicas de Grasshopper

Grasshopper está basado en la programación orientada a objetos. Sus algoritmos gráficos están compuestos de objetos diferentes relacionados o conectados entre sí (Cuevas, 2016).

Grasshopper consta de dos tipos principales de objetos: parámetros y componentes. Los parámetros contienen datos, lo que significa que almacenan las cosas. Los componentes contienen acciones, lo que significa que hacen cosas (Payne y Issa, 2009).

La forma más básica de comprender Grasshopper es recordar que usaremos los datos para definir las entradas de las acciones (lo que dará como resultado nuevos datos que podemos seguir usando) (Akos y Parsons, 2015).

Payne y Issa (2009) nos brindan la información que se muestra en adelante correspondiente a este ítem de nociones básicas de Grasshopper:

La siguiente figura muestra algunos de los objetos que probablemente encuentre en una definición de Grasshopper:

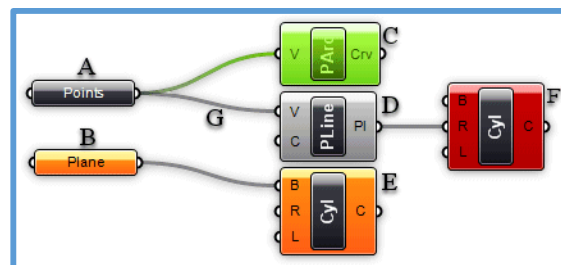


Figura 2.21. Objetos en Grasshopper. Adaptado de (Payne y Issa, 2009).

A) Un parámetro que contiene datos. Dado que no hay cables que salgan del lado izquierdo del objeto, este no hereda sus datos de otros lugares. Los parámetros que no contienen errores o advertencias son bloques negros con texto horizontal.

B) Un parámetro que no contiene datos. Cualquier objeto que no recoge los datos es considerado sospechoso en una definición explícita de la historia, ya que al parecer está perdiendo el tiempo y el dinero de todos. Por lo tanto, todos los parámetros (cuando son recién añadido) son de color naranja, para indicar que no contienen ningún dato y que por lo tanto no tiene efecto funcional sobre el resultado de la solución de la historia. Una vez que un parámetro hereda o define datos, adquiere un color negro.

C) Un componente seleccionado. Todos los objetos seleccionados son verde brillante.

D) Un componente regular.

E) Un componente que contiene advertencias. Dado que es probable que un componente contenga un número de parámetros de entrada y de salida, nunca está claro qué objeto concreto genera la alerta con sólo mirar el componente. Incluso puede haber múltiples fuentes de advertencias. Tendrá que usar el menú de contexto (véase más adelante) con el fin de localizar los problemas. Tenga en cuenta que las advertencias no necesariamente tienen que ser reparadas. Pueden ser completamente legítimas.

F) Un componente que contiene errores. Similares a las advertencias, no es posible ver donde es generado el error en un componente. Tendrá que utilizar el menú de contexto (véase más adelante). Tenga en cuenta que un componente que contiene advertencias y errores aparece de color rojo, el color del error tiene prioridad sobre el color de advertencia.

G) Una conexión. Las conexiones siempre aparecen entre los parámetros de entrada y salida. No hay límite de cuántas conexiones puede contener en particular cualquier parámetro, pero no se permite crear una configuración, con conexiones cíclicas o recursivas.

Tal recursividad es detectada y la solución completa está en cortocircuito cuando esto ocurre, resultando en un mensaje de error en el primer componente o parámetro que se ha detectado como recursivo. Para obtener más información sobre las conexiones, vea el capítulo sobre herencia de datos.

Un componente por lo general requiere de datos para llevar a cabo sus acciones, y usualmente viene con un resultado. Por eso, la mayoría de los componentes tienen un conjunto de parámetros

anidados, referido a sus parámetros de entrada y salida, respectivamente. Los parámetros de entrada están situados a lo largo del lado izquierdo y los parámetros de salida por el lado derecho:

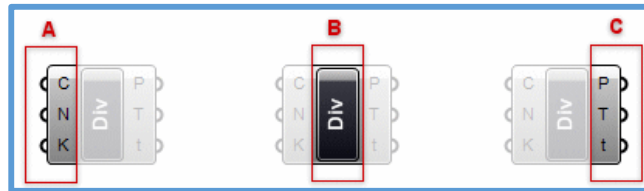


Figura 2.22. Partes de un componente en Grasshopper. Adaptado de (Payne y Issa, 2009).

A) Los tres parámetros de entrada de la división del componente. De forma predeterminada, los nombres de parámetros son siempre extremadamente cortos. Usted puede cambiar el nombre de cada parámetro que quiera.

B) El área de la división del componente (por lo general contiene el nombre del componente)

C) Los tres parámetros de salida de la división de componente.

Al pasar el ratón sobre las partes individuales de los componentes de un objeto, verá información sobre diferentes herramientas que indican el tipo particular de los (sub) objetos actualmente bajo el ratón. Las herramientas de ayuda son muy informativas, ya que dicen, tanto el tipo como los datos de los parámetros individuales:

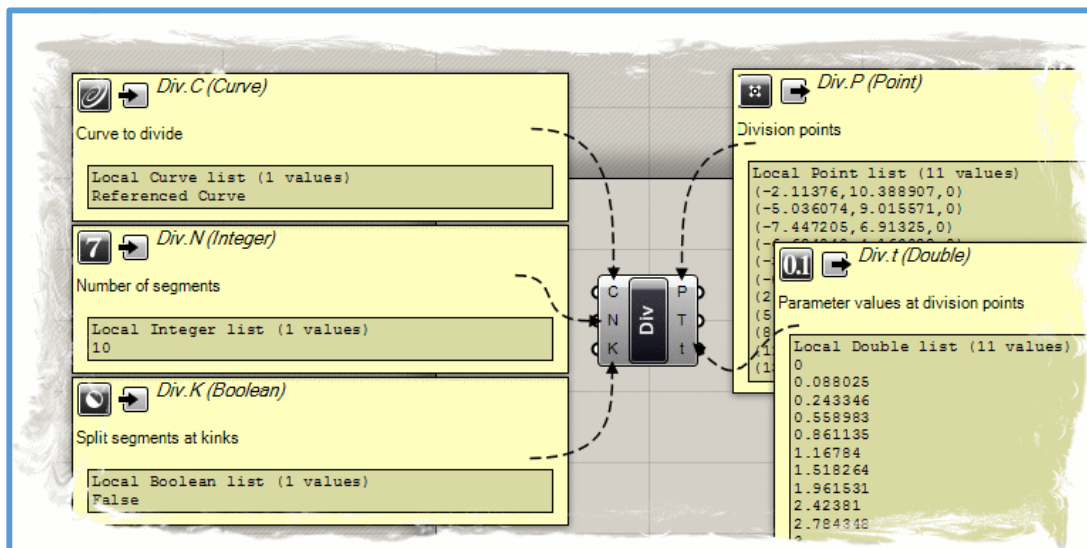


Figura 2.23. Herramientas de ayuda de un componente de Grasshopper. Adaptado de (Payne y Issa, 2009).

2.2.6. Verificación de deflexiones

Para la verificación de deflexiones se ha considerado lo establecido en el artículo N°25 de la Norma Técnica de Edificaciones E 0.20. “Cargas” (N.T.E. E0.20, 2006), es así que se ha considerado lo siguiente:

Excepto en los casos expresamente cubiertos en las normas propias de los diversos materiales estructurales, la deflexión de cualquier elemento estructural no excederá los valores indicados en la tabla 2.3, excepto cuando soporte paneles de vidrio, en cuyo caso se considerara lo siguiente:

Excepto en los casos expresamente cubiertos en las Normas propias de los diversos materiales estructurales, la deflexión para carga viva más la parte correspondiente a las deflexiones diferidas, de elementos estructurales que soportan paneles de vidrio no excederá en ningún caso 20mm.

Tabla 2.1. *Deflexiones máximas para elementos estructurales.*

TIPO DE ELEMENTO	DEFLEXIÓN PRODUCIDA POR LA CARGA VIVA	DEFLEXIÓN PRODUCIDA POR LA CARGA VIVA MAS LAS FLECHAS DIFERIDAS
Pisos	L/360	L/240*
Techos	L/180	-
L: Luz del elemento. Para volados se tomara como L, el doble de la longitud del elemento.		
Deflexión diferida: Se establece en función de cada material de acuerdo a su Norma respectiva. La deflexión diferida se calculara para las cargas permanentes más la fracción de sobrecarga que actúa permanentemente.		
* No aplicable a estructuras metálicas.		

Nota. Adaptado de (N.T.E. E0.20, 2016).

En esta investigación a la deflexión máxima permisible definida en esta sección también se le ha denominado como deflexión límite.

2.2.7. Diseño de elementos de armaduras metálicas usando las “Especificaciones para Construcciones de Acero Estructural” (AISC 360 -10) del AMERICAN INSTITUTE OF STEEL CONSTRUCTION (AISC, 2010)

2.2.7.1. Diseño por resistencia usando el Diseño en Base a Factores de Carga y Resistencia (LRFD)

El diseño de acuerdo con las disposiciones de diseño en Base a Factores de Carga y Resistencia (LRFD) satisface los requisitos de esta Especificación cuando la resistencia de diseño de cada componente estructural es mayor o igual a la resistencia requerida determinada de acuerdo con las combinaciones de carga LRFD. El diseño se realizará de acuerdo con la ecuación 2.1:

$$R_u \leq \phi R_n \quad \text{Ecuación 2.1}$$

Donde:

R_u = resistencia requerida (usando combinaciones de carga LRFD).

R_n = resistencia nominal.

ϕ = factor de resistencia.

ϕR_n = resistencia de diseño.

2.2.7.2. Combinaciones de carga

Se ha considerado las combinaciones de carga (combinaciones de carga LRFD) presentes en la sección 1.4.1. de la Norma Técnica de Edificaciones E0.90 “ESTRUCTURAS METÁLICAS” (N.T.E. E0.90, 2006). Se ha considerado las siguientes cargas nominales:

D: Carga muerta debida al peso propio de los elementos y los efectos permanentes sobre la estructura.

L: Carga viva debida al mobiliario y ocupantes.

Lr: Carga viva en las azoteas.

W: Carga de viento.

S: Carga de nieve.

E: Carga de sismo de acuerdo a la Norma E.030 Diseño Sismoresistente.

R: Carga por lluvia o granizo.

La resistencia requerida de la estructura y sus elementos debe ser determinada para la adecuada combinación crítica de cargas factorizadas. El efecto crítico puede ocurrir cuando una o más cargas no estén actuando. Para la aplicación del método LRFD, las siguientes combinaciones deben ser investigadas:

$1,4D$	1
$1,2D + 1,6L + 0,5(L_r \text{ ó } S \text{ ó } R)$	2
$1,2D + 1,6(L_r \text{ ó } S \text{ ó } R) + (0,5L \text{ ó } 0,8W)$	3
$1,2D + 1,3W + 0,5L + 0,5(L_r \text{ ó } S \text{ ó } R)$	4
$1,2D \pm 1,0E + 0,5L + 0,2S$	5
$0,9D \pm (1,3W \text{ ó } 1,0E)$	6

En las combinaciones 3, 4 y 5 el factor de cargas para L debe ser considerado como 1,0 en el caso de estacionamientos, auditorios y todo lugar donde la carga viva sea mayor a 4800Pa.

2.2.7.3. Clasificación de secciones para pandeo local

Para compresión, las secciones se clasifican como secciones de elementos no esbeltos o secciones de elementos esbeltos. Para una sección de elementos no esbeltos, la relación ancho/espesor de sus elementos a compresión no deberán exceder λ_r de la tabla 2.2. Si la relación ancho/espesor de cualquier elemento a compresión excede λ_r , la sección es una sección de elementos esbeltos.

2.2.7.4. Elementos rígidos

Para elementos rígidos soportados a lo largo de dos bordes paralelos a la dirección de la fuerza de compresión, el ancho se tomará de la siguiente manera:

Para bridas de secciones estructurales huecas rectangulares (HSS), el ancho, b , es la distancia libre entre almas menos el radio de la esquina interior de cada lado. Para almas de secciones HSS rectangular, h es la distancia clara entre las bridas, menos el radio de la esquina interior de cada lado. Si no se conoce el radio de la esquina, b y h se tomarán como la dimensión exterior correspondiente menos tres veces el espesor de la pared de diseño. El espesor, t , se tomará como el espesor de la pared de diseño, según la sección 2.2.7.5.

2.2.7.5. Diseño de espesor de pared para HSS

El espesor de la pared de diseño, t , debe usarse en cálculos que involucren el espesor de la pared de secciones estructurales huecas (HSS). El espesor de la pared de diseño, t , debe tomarse igual a 0,93 veces el espesor de pared nominal para HSS de soldadura con resistencia eléctrica (ERW) e igual al espesor nominal para HSS de soldadura por arco sumergido (SAW).

En esta investigación se ha trabajado con secciones tubulares (Hss), circulares y rectangulares, por lo tanto, se ha considerado los casos 6 y 9 (encerrados en recuadro rojo) de la tabla 2.2, para verificar la relación ancho - espesor.

- Los tubos ASTM A500 HSS y ASTM A53 de grado B son producidos por un proceso ERW. Un proceso SAW se utiliza para secciones transversales que son más grandes que las permitidas por ASTM A500.

2.2.7.6. Determinación del área bruta y del área neta

Área Bruta: El área bruta de un miembro, A_g , es el área total de la sección transversal.

Área Neta: El área neta, A_n , de un miembro es la suma de los productos de los espesores por sus respectivos anchos netos, calculados como se indica a continuación:

Para calcular el área neta para tensión y corte, la perforación para alojar un conector deberá aumentarse 0,2 cm (2mm) respecto de la dimensión nominal de la perforación.

Para una cadena de perforaciones en cualquier línea diagonal o zigzag, el ancho neto se obtendrá deduciendo del ancho bruto la suma de las dimensiones de los diámetros de las perforaciones o ranuras según se indica en esta sección, de todas las perforaciones en una cadena, y agregando para cada cambio en zigzag en la cadena la cantidad $s^2/4g$, donde:

s = distancia longitudinal centro a centro de dos perforaciones consecutivas (paso), in. (mm)

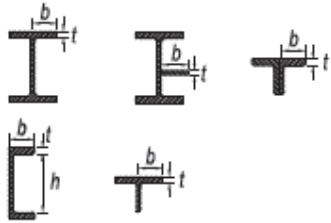
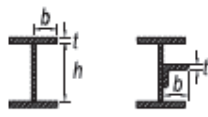
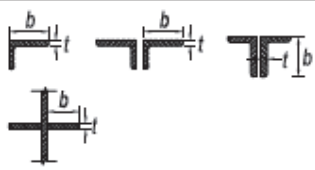

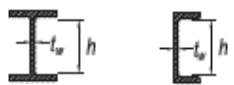



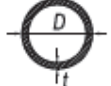
g = distancia transversal centro a centro (gramil) entre líneas de conectores, in. (mm)

Para ángulos, la separación de las perforaciones en alas opuestas adyacentes, será las sumas de las distancias medidas desde el respaldo del ángulo menos el espesor.

Para secciones tubulares ranuradas soldadas a una plancha gusset, el área neta, A_n , es el área de la sección transversal menos el producto del espesor y el ancho total del material que es removido para hacer la ranura.

No se considerará el metal de la soldadura en la determinación del área neta a través de soldaduras de tapón o ranura.

Tabla 2.2. Relaciones Ancho-Espesor: Elementos en Compresión Miembros Sujetos a Compresión Axial.

TABLE B4.1a Width-to-Thickness Ratios: Compression Elements Members Subject to Axial Compression				
Case	Description of Element	Width-to-Thickness Ratio	Limiting Width-to-Thickness Ratio λ_r (nonslender/slender)	Examples
Unstiffened Elements	1 Flanges of rolled I-shaped sections, plates projecting from rolled I-shaped sections; outstanding legs of pairs of angles connected with continuous contact, flanges of channels, and flanges of tees	b/t	$0.56 \sqrt{\frac{E}{F_y}}$	
	2 Flanges of built-up I-shaped sections and plates or angle legs projecting from built-up I-shaped sections	b/t	$0.64 \sqrt{\frac{k_c E}{F_y}}$ ^[a]	
	3 Legs of single angles, legs of double angles with separators, and all other unstiffened elements	b/t	$0.45 \sqrt{\frac{E}{F_y}}$	
	4 Stems of tees	d/t	$0.75 \sqrt{\frac{E}{F_y}}$	
Stiffened Elements	5 Webs of doubly-symmetric I-shaped sections and channels	h/t_w	$1.49 \sqrt{\frac{E}{F_y}}$	
	6 Walls of rectangular HSS and boxes of uniform thickness	b/t	$1.40 \sqrt{\frac{E}{F_y}}$	
	7 Flange cover plates and diaphragm plates between lines of fasteners or welds	b/t	$1.40 \sqrt{\frac{E}{F_y}}$	
	8 All other stiffened elements	b/t	$1.49 \sqrt{\frac{E}{F_y}}$	
	9 Round HSS	D/t	$0.11 \frac{E}{F_y}$	

Nota. Adaptado de (AISC, 2010).

Para secciones sin perforaciones el área neta, A_n , es igual al área bruta, A_g . En esta investigación se ha trabajado con secciones sin perforaciones conectadas completamente por soldadura, por lo tanto, el área neta es igual al área bruta.

2.2.7.7. Diseño de miembros a tensión

El diseño está aplicado a miembros sujetos a tensión axial causada por fuerzas estáticas que actúan a través de sus ejes centroidales.

2.2.7.7.1. Limitaciones de esbeltez

No hay límite máximo de esbeltez para los miembros en tensión. Para los miembros diseñados sobre la base de la tensión, la relación de esbeltez L/r preferiblemente no debe exceder 300. Esta sugerencia no se aplica a las varillas o colgadores en tensión.

2.2.7.7.2. Resistencia a la tensión

La resistencia de diseño en tensión, $\phi_t P_n$, o la resistencia admisible en tensión, P_n/Ω , de miembros tensionados debe ser el menor valor obtenido de acuerdo con los estados límites de fluencia en tensión calculado en la sección bruta y ruptura en tensión calculado en la sección neta.

(a) Para fluencia en tensión en la sección bruta:

$$\begin{aligned} P_n &= F_y A_g \\ \phi_t &= 0,90 \text{ (LRFD)} \quad \Omega_t = 1,67 \text{ (ASD)} \end{aligned} \quad \text{Ecuación 2.2}$$

(b) Para ruptura en tensión en la sección neta:

$$\begin{aligned} P_n &= F_u A_e \\ \phi_t &= 0,75 \text{ (LRFD)} \quad \Omega_t = 2,00 \text{ (ASD)} \end{aligned} \quad \text{Ecuación 2.3}$$

Donde:

A_e = Área neta efectiva (en in^2 o mm^2).

A_g = Área bruta del miembro (en in^2 o mm^2).

F_y = Tensión de fluencia mínima especificada (en ksi o MPa).

F_u = Tensión última mínima especificada (en ksi o MPa).

Cuando miembros sin perforaciones están completamente conectados por soldadura, el área neta efectiva usada en la Ecuación 2.3 debe ser la definida en Sección 2.2.7.7.3. Cuando se presentan perforaciones en el miembro con conexiones soldadas en sus extremos, o en conexiones soldadas mediante soldadura de ranura o tapón, debe usarse el área neta efectiva a través de las perforaciones en la Ecuación 2.3.

2.2.7.7.3. Área neta efectiva (A_e)

El área bruta, A_g , y el área neta, A_n , de elementos tensionados debe ser determinada de acuerdo con las disposiciones de la sección 2.2.7.6. El área neta efectiva de los miembros tensionados debe ser determinada de la siguiente forma:

$$A_e = A_n U \quad \text{Ecuación 2.4}$$

Donde U , el factor de corte diferido (shear lag), es determinado como se muestra en la Tabla 2.3. Dentro de esta investigación se ha trabajado con el factor de corte diferido (U), definido en el caso 1 (encerrado en recuadro rojo) de la tabla 2.3, donde nos menciona que se aplica a: Todos los miembros de tensión donde la carga de tensión se transmite directamente a cada uno de los elementos transversales mediante sujetadores o soldaduras (excepto en los casos 4, 5 y 6); dicho factor “ U ” tiene un valor de 1.

2.2.7.8. Diseño de miembros a compresión

La resistencia de diseño en compresión, $\phi_c P_n$, o la resistencia admisible en compresión, P_n/Ω_c , deben ser determinadas usando los siguientes factores:

$$\phi_c = 0,90 \text{ (LRFD)} \quad \Omega_c = 1,67 \text{ (ASD)} \quad \text{Ecuación 2.5}$$

La resistencia a la compresión nominal, P_n , será el valor más bajo obtenido en función de los estados límite aplicables de pandeo por flexión, pandeo por torsión y pandeo por flexo-torsión. Ver tabla 2.4.

Esta investigación ha desarrollado el diseño a compresión solo de perfiles tubulares o Hss (circulares y rectangulares), sin elementos esbeltos y con elementos esbeltos, es por eso que según la tabla 2.4 solo se ha considerado en el diseño los estados límite de pandeo por flexión (o pandeo flexionante) y pandeo local (recuadro rojo).




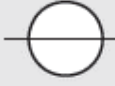


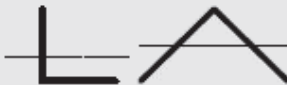

Tabla 2.3. Factores de Corte Diferido para Conexiones de Miembros a Tensión.

<p style="text-align: center;">TABLE D3.1 Shear Lag Factors for Connections to Tension Members</p>				
Case	Description of Element		Shear Lag Factor, U	Example
1	All tension members where the tension load is transmitted directly to each of the cross-sectional elements by fasteners or welds (except as in Cases 4, 5 and 6).		$U = 1.0$	
2	All tension members, except plates and HSS, where the tension load is transmitted to some but not all of the cross-sectional elements by fasteners or longitudinal welds or by longitudinal welds in combination with transverse welds. (Alternatively, for W, M, S and HP, Case 7 may be used. For angles, Case 8 may be used.)		$U = 1 - \bar{x}/l$	
3	All tension members where the tension load is transmitted only by transverse welds to some but not all of the cross-sectional elements.		$U = 1.0$ and $A_n =$ area of the directly connected elements	
4	Plates where the tension load is transmitted by longitudinal welds only.		$l \geq 2w \dots U = 1.0$ $2w > l \geq 1.5w \dots U = 0.87$ $1.5w > l \geq w \dots U = 0.75$	
5	Round HSS with a single concentric gusset plate		$l \geq 1.3D \dots U = 1.0$ $D \leq l < 1.3D \dots U = 1 - \bar{x}/l$ $\bar{x} = D/\pi$	
6	Rectangular HSS	with a single concentric gusset plate	$l \geq H \dots U = 1 - \bar{x}/l$ $\bar{x} = \frac{B^2 + 2BH}{4(B+H)}$	
		with two side gusset plates	$l \geq H \dots U = 1 - \bar{x}/l$ $\bar{x} = \frac{B^2}{4(B+H)}$	
7	W, M, S or HP Shapes or Tees cut from these shapes. (If U is calculated per Case 2, the larger value is permitted to be used.)	with flange connected with 3 or more fasteners per line in the direction of loading	$b_f \geq 2/3d \dots U = 0.90$ $b_f < 2/3d \dots U = 0.85$	
		with web connected with 4 or more fasteners per line in the direction of loading	$U = 0.70$	
8	Single and double angles (If U is calculated per Case 2, the larger value is permitted to be used.)	with 4 or more fasteners per line in the direction of loading	$U = 0.80$	
		with 3 fasteners per line in the direction of loading (With fewer than 3 fasteners per line in the direction of loading, use Case 2.)	$U = 0.60$	

l = length of connection, in. (mm); w = plate width, in. (mm); \bar{x} = eccentricity of connection, in. (mm); B = overall width of rectangular HSS member, measured 90° to the plane of the connection, in. (mm); H = overall height of rectangular HSS member, measured in the plane of the connection, in. (mm)

Nota. Adaptado de (AISC, 2010).

Tabla 2.4. Tabla de selección para la aplicación de las secciones del capítulo E de la guía AISC 360-2010.

TABLE USER NOTE E1.1 Selection Table for the Application of Chapter E Sections				
Cross Section	Without Slender Elements		With Slender Elements	
	Sections in Chapter E	Limit States	Sections in Chapter E	Limit States
	E3 E4	FB TB	E7	LB FB TB
	E3 E4	FB FTB	E7	LB FB FTB
	E3	FB	E7	LB FB
	E3	FB	E7	LB FB
	E3 E4	FB FTB	E7	LB FB FTB
	E6 E3 E4	FB FTB	E6 E7	LB FB FTB
	E5		E5	
	E3	FB	N/A	N/A
Unsymmetrical shapes other than single angles	E4	FTB	E7	LB FTB
FB = flexural buckling, TB = torsional buckling, FTB = flexural-torsional buckling, LB = local buckling				

Nota. Adaptado de (AISC, 2010).

2.2.7.7.1. Longitud efectiva

El factor de longitud efectiva "K" para calcular la esbeltez del miembro (KL/r) debe ser determinado de acuerdo a la tabla 2.5.

donde:

L = Longitud del miembro sin arriostramiento lateral (en in o mm).

r = Radio de giro (en in o mm).

Nota: Para miembros diseñados en base a compresión, la relación de esbeltez efectiva " KL/r " preferiblemente no debe exceder 200.

Tabla 2.5. Valores aproximados de factor de longitud efectiva, K.

<p align="center">TABLE C-A-7.1 Approximate Values of Effective Length Factor, K</p>						
<p>Buckled shape of column is shown by dashed line</p>	(a)	(b)	(c)	(d)	(e)	(f)
Theoretical K value	0.5	0.7	1.0	1.0	2.0	2.0
Recommended design value when ideal conditions are approximated	0.65	0.80	1.2	1.0	2.1	2.0
End condition code	<p> Rotation fixed and translation fixed Rotation free and translation fixed Rotation fixed and translation free Rotation free and translation free </p>					

Nota. Adaptado de (AISC, 2010).

Esta investigación ha sido aplicada a armaduras planas y tridimensionales, en las cuales cada uno de sus miembros se puede idealizar como simplemente apoyados, es por eso que se ha tomado el valor de $k = 1$ en el diseño de todos los miembros.

2.2.7.7.2. Pandeo flexional de miembros sin elementos esbeltos

Esta sección se aplica a los miembros a compresión de elementos no esbeltos como se define en la sección 2.2.7.3.

La resistencia a compresión nominal "Pn" debe ser determinada en función al estado límite de pandeo por flexión o pandeo flexionante según la siguiente ecuación:

$$P_n = F_{cr} A_g \quad \text{Ecuación 2.6}$$

El esfuerzo crítico "Fcr" se determina de la siguiente manera:

$$\begin{aligned} \text{(a) When } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}} \quad \text{(or } \frac{F_y}{F_e} \leq 2.25) \\ F_{cr} = \left[0.658 \frac{F_y}{F_e} \right] F_y \end{aligned} \quad \text{Ecuación 2.7}$$

$$\begin{aligned} \text{(b) When } \frac{KL}{r} > 4.71 \sqrt{\frac{E}{F_y}} \quad \text{(or } \frac{F_y}{F_e} > 2.25) \\ F_{cr} = 0.877 F_e \end{aligned} \quad \text{Ecuación 2.8}$$

Donde:

A_g = Área gruesa del miembro (en in² o mm²).

K = Factor de longitud efectiva.

r = Radio de giro (en in o mm).

L = Longitud del miembro sin arriostramiento lateral (en in o mm).

E = Modulo de elasticidad del material (en ksi o Mpa).

F_y = Esfuerzo de fluencia del material (en ksi o Mpa).

F_e = Esfuerzo de pandeo elástico, también se le conoce como el esfuerzo critico de Euler (en ksi o Mpa).

$$F_e = \frac{\pi^2 E}{\left(\frac{KL}{r} \right)^2} \quad \text{Ecuación 2.9}$$

2.2.7.7.3. Miembros con elementos esbeltos

Esta sección aplica para miembros solicitados en compresión con elementos esbeltos, como se define en la sección 2.2.7.3 para elementos en compresión uniforme. La resistencia de compresión nominal, P_n , debe ser determinada basada los estados límite de pandeo por flexión, pandeo por torsión y pandeo por flexo-torsión.

$$P_n = F_{cr} A_g \quad \text{Ecuación 2.10}$$

El esfuerzo crítico "Fcr" se determina de la siguiente manera:

$$\begin{aligned} \text{(a) When } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{QF_y}} \quad \left(\text{or } \frac{QF_y}{F_e} \leq 2.25 \right) \\ F_{cr} = Q \left[0.658 \frac{QF_y}{F_e} \right] F_y \end{aligned} \quad \text{Ecuación 2.11}$$

$$\begin{aligned} \text{(b) When } \frac{KL}{r} > 4.71 \sqrt{\frac{E}{QF_y}} \quad \left(\text{or } \frac{QF_y}{F_e} > 2.25 \right) \\ F_{cr} = 0.877 F_e \end{aligned} \quad \text{Ecuación 2.12}$$

Donde:

F_e = Esfuerzo de pandeo elástico, calculado usando la ecuación 2.9 para miembros con simetría doble, ecuación 2.9 para miembros con simetría simple.

Q = Factor de reducción neto que toma en cuenta todos los elementos esbeltos en compresión.

= 1 para miembros sin elementos esbeltos, como se define en la sección 2.2.7.3, para elementos en compresión uniforme.

= $Q_s * Q_a$ para miembros con secciones de elementos esbeltos, como se define en la sección 2.2.7.3, para elementos en compresión uniforme.

Nota: Para secciones conformadas solamente por elementos esbeltos no rigidizados, $Q = Q_s$ ($Q_a = 1$). Para secciones conformadas solamente por elementos esbeltos rigidizados, $Q = Q_a$ ($Q_s = 1$).

Para secciones conformadas por ambos elementos esbeltos no rigidizados y rigidizados, $Q = Q_s * Q_a$.

En esta investigación se ha trabajado solo con perfiles metálicos tubulares, en los cuales las secciones solo tienen elementos rigidizados, por lo tanto, se cumple que $Q = Q_a$ ($Q_s = 1$).

2.2.7.7.3.1. Elementos Esbeltos Rigidizados

El factor de reducción, Q_a , para elementos esbeltos rigidizados se define a continuación:

$$Q_a = \frac{A_e}{A_g} \quad \text{Ecuación 2.13}$$

Donde:

A_g = Área seccional total del miembro (en in² o mm²).

A_e = Suma de las áreas efectivas de las secciones basadas en el ancho efectivo reducido b_e (en in² o mm²).

El ancho efectivo reducido, b_e , se determina de la siguiente manera:

a) Para elementos esbeltos en compresión uniforme, con $\frac{b}{t} \geq 1.49\sqrt{\frac{E}{f}}$, excepto para alas de tubos cuadrados y rectangulares de espesor uniforme:

$$b_e = 1.92t\sqrt{\frac{E}{f}}\left[1 - \frac{0.34}{(b/t)}\sqrt{\frac{E}{f}}\right] \leq b \quad \text{Ecuación 2.14}$$

Donde:

f se toma como F_{cr} con F_{cr} calculada con $Q = 1$.

b) Para alas de elementos esbeltos de tubos cuadrados y rectangulares en compresión uniforme

con $\frac{b}{t} \geq 1.40\sqrt{\frac{E}{f}}$

$$b_e = 1.92t\sqrt{\frac{E}{f}}\left[1 - \frac{0.38}{(b/t)}\sqrt{\frac{E}{f}}\right] \leq b \quad \text{Ecuación 2.15}$$

Donde:

$f = P_n/A_e$

Nota: En vez de calcular $f = P_n/A_e$, que requiere iteración, f puede ser tomada igual a F_y . Esto resulta en una estimación levemente conservadora en la capacidad de la columna. En esta investigación se ha tomado f igual a F_y .

c) Para tubos circulares cargados axialmente, cuando $0.11\frac{E}{F_y} < \frac{D}{t} < 0.45\frac{E}{F_y}$

$$Q = Q_a = \frac{0.038E}{F_y(D/t)} + \frac{2}{3} \quad \text{Ecuación 2.16}$$

Donde:

D = Diámetro exterior (en in o mm).

t = Espesor de pared (en in o mm).

2.2.8. Aplicación del ciclo: Modelación Paramétrica, Simulación Estructural y Optimización

Las tres etapas fundamentales (Modelación Paramétrica, Simulación Estructural y Optimización) del proceso de diseño estructural propuesto en esta investigación tienen distintas formas de aplicación, no hay patrones definidos debido a que cada problema puede adoptar una solución según los requerimientos del usuario, es decir, cómo se tiene la ventaja de la programación gráfica, con ayuda de esta se puede desarrollar un modelo de acuerdo a las necesidades o requerimientos del usuario.

En la sección 3.4 se ha desarrollado 3 ejemplos en los cuales se ha aplicado el ciclo: Modelación Paramétrica, Simulación Estructural y Optimización de forma distinta para cada ejemplo según el código gráfico realizado, es decir cada ejemplo tiene un enfoque particular, dichos ejemplos constituyen una guía esencial para las personas que quieran aprender a usar el ciclo “Modelación Paramétrica, Simulación Estructural y Optimización” aplicado al diseño de armaduras metálicas con AISC 360-10.

CAPITULO III: MATERIALES Y MÉTODOS

3.1. UBICACIÓN GEOGRÁFICA

Esta investigación se ha desarrollado en Perú, en el departamento de Cajamarca, en la Provincia de Cajamarca y en el distrito de Cajamarca.

3.2. ÉPOCA DE LA INVESTIGACIÓN

Esta investigación se ha realizado en los años 2017-2018.

3.3. PROCEDIMIENTO

3.3.1. Tipo de investigación

- El tipo de investigación de acuerdo al nivel de investigación es descriptivo. Se ha desarrollado un proceso de diseño estructural para armaduras metálicas usando el software Rhinoceros 3D, describiéndolo de manera objetiva.
- El tipo de investigación de acuerdo al diseño de la investigación es documental. Se ha trabajado con información desarrollada por otros investigadores en fuentes documentales, para así poder desarrollar el proceso de diseño estructural para armaduras metálicas propuesto en esta investigación.

3.3.2. Población de estudio

La población está formada por las armaduras planas y tridimensionales constituidas por perfiles metálicos tubulares (circulares y/o rectangulares).

3.3.3. Muestra

La muestra está formada por dos armaduras planas y un arco de viga de celosía, los cuales serán diseñados con el proceso de diseño estructural propuesto en esta investigación.

3.3.4. Unidad de análisis

La unidad de análisis corresponde a una armadura metálica (plana o tridimensional).

3.3.5. Técnicas e instrumentos de recolección de datos

Se ha empleado las siguientes técnicas:

- Análisis documental. Se ha trabajado con información desarrollada por otros investigadores en fuentes documentales, se ha realizado un análisis de dicha información para así poder elaborar el proceso de diseño estructural para armaduras metálicas propuesto en esta investigación.

- Observación estructurada. Se ha observado los distintos comportamientos de las armaduras en función al cambio de los parámetros que las definen (peralte, número de segmentos, flecha, etc.).
- Análisis comparativo. Se ha comparado el proceso de diseño estructural tradicional de prueba y error con el proceso de diseño estructural propuesto en esta investigación, dando a conocer las múltiples ventajas de este último.

Se ha empleado los siguientes instrumentos:

- Software Rhinoceros 3D.
- Software SAP2000.
- Complemento Grasshopper.
- Complemento Galápagos.
- Complemento Octopus.
- Complemento Karamba.

3.4. TRATAMIENTO, ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS

Se ha desarrollado 3 ejemplos de diseño de armaduras metálicas (2 armaduras planas y una viga de celosía) con el proceso de diseño estructural para armaduras metálicas desarrollado en esta investigación, para conocer a fondo este proceso, realizar análisis comparativos con respecto al “Proceso de diseño tradicional de prueba y error usando SAP2000”, y evaluar los beneficios que brinda.

Grasshopper brinda el gran beneficio de crear scripts (componentes personalizados realizados con lenguaje de programación) y es así que para realizar el diseño de armaduras según la AISC 360-10 dentro de la etapa de optimización, se ha creado una secuencia de scripts utilizando el lenguaje de programación “C Sharp” (C#), esto también ha sido posible debido a que Rhinoceros 3D y Karamba, son software y complemento respectivamente, de código abierto, desarrollados con programación orientada a objetos, por lo tanto en el desarrollo de los scripts se hace uso de las clases ya definidas por estos. Rhinoceros 3D, contiene un manual para programar utilizando sus clases, el cual se encuentra en su página web, asimismo Karamba aparte del manual de usuario, cuenta con un manual para los interesados en programar utilizando las clases definidas dentro de él, el cual se puede descargar gratuitamente de su página web.

Karamba es un complemento desarrollado por Clemens Preisinger en Vienna (Austria), es así que diseña armaduras metálicas según el Eurocódigo, tiene combinaciones de carga según el Eurocódigo, en el cual la combinación más crítica vendría a ser la sumatoria de todas las cargas sin multiplicar a cada carga por algún factor (como se establece en las combinaciones de carga según el LRFD) , entonces los scripts desarrollados ayudan a implementar las normativas usadas en nuestro país (AISC 360-10, N.T.E E 0.20 y N.T.E E 0.90) en Karamba en lo correspondiente al diseño de armaduras metálicas.

Notas:

- **Se debe tener conocimientos básicos de Rhinoceros 3D, Grasshopper y Karamba para poder entender completamente el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” desarrollado en esta investigación.**
- **Se adjunta un cd que contiene los archivos en Grasshopper de cada uno de los ejemplos desarrollados en esta investigación, para mayores detalles y una mejor comprensión de estos ejemplos se debe revisar dichos archivos.**
- **Cabe indicar que la realización de simulaciones estructurales mediante variación de parámetros es necesaria para evaluar comportamientos de una armadura frente al cambio de los parámetros que la definen, se la realiza a criterio del usuario. Debido a los muchos comportamientos que se generan en una armadura al variar sus parámetros solo se ha realizado simulaciones estructurales para el ejemplo 1, para el resto de ejemplos se las puede realizar de manera similar, cabe recalcar que el objetivo de la simulación estructural es: “Saber que sucede con la armadura cuando modificamos sus parámetros, ¿cómo se comporta!, y así obtener los mejores parámetros”.**

3.4.1. Desarrollo de scripts para realizar el diseño de armaduras metálicas según la AISC 360-10, N.T.E E 0.20 y N.T.E E 0.90

3.4.1.1. Script N°1

El primer script tiene por nombre “KL”, el cual tiene la función de calcular la longitud de pandeo o longitud efectiva, tiene una entrada “L” en la cual se ingresan elementos de armadura obtenidos con el componente “RDLines”, lo que hace el script es tomar las longitudes de los elementos de armadura ingresados en este y multiplicarlos por su factor de longitud efectiva (para el caso de armaduras metálicas dicho factor es “1”, ya que cada elemento se considera como comprendido entre 2 apoyos fijos), obteniendo así las longitudes de pandeo o longitudes efectivas de dichos elementos. La salida “LE” del componente “KL” nos brinda las longitudes de pandeo o longitudes efectivas calculadas, esta salida se conecta a las entradas BklLenY y BklLenZ del componente “ModifyElem”, estableciendo así las longitudes de pandeo de los elementos dentro del modelo.

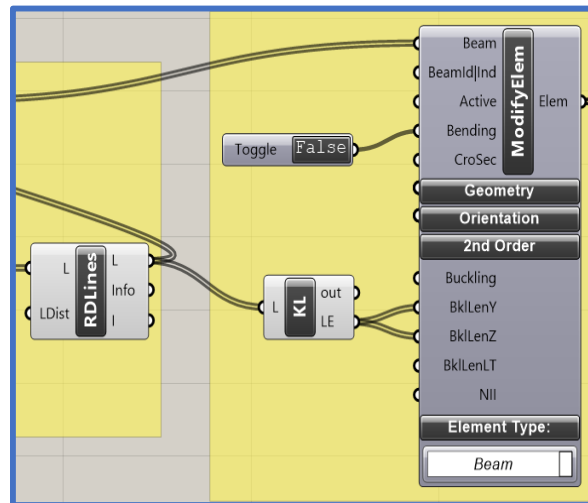


Figura 3.1. Script “KL”.

3.4.1.2. Script N°2

El segundo script tiene por nombre “LONGITUD”, tiene una entrada “Li” por la cual se ingresan todos los elementos de la brida superior de la armadura, brindados por la salida del componente “Line(Ln)”, el componente “LONGITUD” sirve como un filtro que convierte las longitudes de cada elementos o miembro de la armadura a valores tipo object y es así que su salida “Lt” nos brinda estos valores tipo object.

La longitud total de la brida superior de una armadura se obtiene sumando los valores que nos brinda la salida “Lt” del componente “LONGITUD”, lo cual se hace con el componente “Mass Addition (MA)”.

La creación del script “LONGITUD”, se justifica porque existe una ligera desventaja en cuanto al peso propio de una armadura, para poder usar las combinaciones de carga del LRFD y las cargas de servicio indicadas en la “N.T.E. E.020” se ha creado scripts en “Karmaba3D”, para calcular el peso propio como parte de la carga muerta, lo que se hace es, primero calcular el peso de la armadura con el componente “Assemble”, luego este peso se divide entre la longitud total de la brida superior con el script “PP(kN)”, obteniendo así una carga distribuida por peso propio (Karamba nos permite aplicar cargas distribuidas en armaduras, ya que las convierte a cargas nodales al indicar al componente LineToBeam o al componente “ModifyElem” falso en la opción Bending, es decir, le estamos indicando que dichos elementos no tendrán flexión, entonces el programa calcula las cargas en los nodos a partir de la carga distribuida, multiplicando dicha carga distribuida por su longitud de influencia), la cual se aplica a la armadura como una carga distribuida externa mediante el script “COMBINACIÓN DE CARGAS” y/o el script “CARGA DE SERVICIO” los cuales realizan las combinaciones de carga del LRFD y calculan la carga de servicio indicada en la “N.T.E. E.020” respectivamente. El análisis estructural con la carga distribuida por peso propio aplicada en la brida superior como si fuese una carga distribuida externa nos brinda resultados muy aproximados a los obtenidos al realizar el análisis estructural considerando solo el peso propio (ver anexo 13).

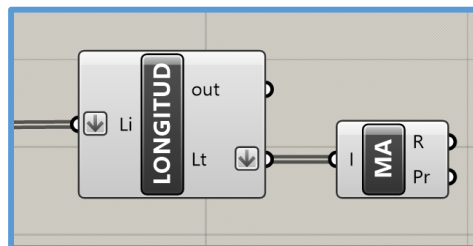


Figura 3.2. Script “LONGITUD”.

3.4.1.3.Script N°3

El tercer script tiene por nombre “PP(kN)”, este script tiene dos entradas, la primera es la entrada “W”, donde se ingresa el peso propio de la armadura (en kg) obtenido con el componente “Assemble” y una segunda entrada “L” donde se ingresa la longitud total de la brida superior de la armadura (en m) obtenida con el componente “Mass Addition(MA)”, lo que hace es obtener una carga distribuida por peso propio al dividir el peso propio de la armadura entre la longitud de la brida superior, luego la convierte de kg/m a kN/m (debido a que Karamba trabaja con unidades del Sistema Internacional) y finalmente le asigna un signo de acuerdo a su sentido de aplicación, como es en dirección de la gravedad el signo es negativo, la salida “Pp” del script “PP(kN)” es quien brinda la carga distribuida por peso propio en kN/m, la cual se debe ingresar dentro de la carga muerta, en los scripts “CARGA DE SERVICIO” y “COMBINACION DE CARGAS”.

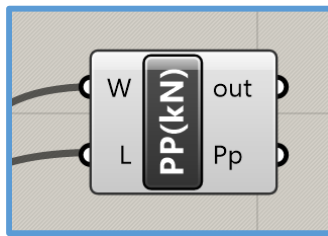


Figura 3.3. Script “PP(kN)”.

3.4.1.4.Script N°4

El cuarto script tiene por nombre “CARGA DE SERVICIO”, lo que hace es obtener la carga de servicio para el cálculo de deflexiones según lo establecido en la N.T.E. E.020, tiene 3 entradas la primera es “nameid” donde se ingresa el nombre de los elementos o miembros a los cuales se les aplicara las cargas de servicio, la segunda “D”, donde se ingresa la carga muerta en kN/m y la tercera “Lr” donde se ingresa la carga viva en azoteas, lo que hace el script es sumar estas 2 cargas, dicha suma es la carga por servicio para evaluar deflexiones. La salida Cs del componente “CARGA DE SERVICIO”, es quien nos brinda la carga de servicio en kN/m, con su respectiva dirección (signo negativo ya que actúa en dirección de la gravedad).

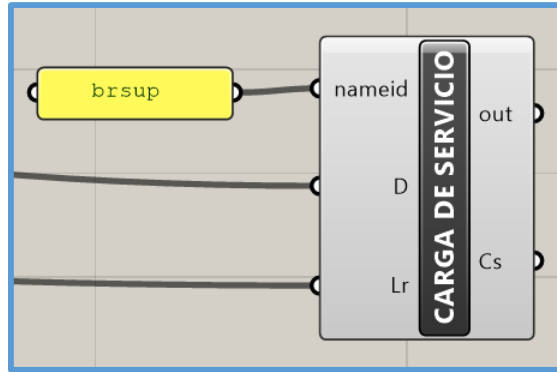


Figura 3.4. Script “CARGA DE SERVICIO”.

Nota: Cabe indicar que los 3 ejemplos desarrollados en esta investigación corresponden a armaduras metálicas usadas en coberturas, consecuentemente se calcula la carga de servicio para ese caso y con esa carga de servicio se evalúan las deflexiones respectivas según lo indica la N.T.E E.0.20, el componente “CARGA DE SERVICIO” aplica a ese caso. Para armaduras con otros usos, y que requieran distintas cargas de servicio para evaluar sus deflexiones respectivas se tendrá que hacer las modificaciones e implementaciones al script “CARGA DE SERVICIO”, para lo cual se requiere conocimientos de programación en C Sharp (C#).

3.4.1.5.Script N°5

El quinto script tiene por nombre “COMBINACION DE CARGAS”, lo que hace es obtener la combinación de carga que tenga mayor carga o carga ultima, ya que con esa combinación se diseñaran los elementos de la armadura. Este script se basa en las combinaciones de carga del “LRFD”, indicadas en la N.T.E E.090, se cuenta con 8 entradas que corresponden a: “nameid” – nombre de los elementos o miembros a los cuales se les aplicara la combinación de cargas de mayor valor, “D” - Carga muerta, “L” - carga viva, “Lr” - carga viva en azoteas, “W” - carga de viento, “S” - carga de nieve, “E” - carga de sismo y “R” - carga de lluvia, en las cuales se ingresa su respectivo valor de carga como carga distribuida en kN/m (como ya se indicó en párrafos anteriores, Karamba convierte la carga distribuida a cargas nodales al indicar a los componentes “LineToBeam” y “ModifyElem” que los elementos no tendrán flexión, ya que al hacer esto le indicamos que los elementos con los cuales estamos trabajando corresponden a una armadura, en esta investigación trabajaremos las cargas como cargas distribuidas ya que se cuenta con esta

ventaja), teniendo en cuenta el signo, el cual indica su dirección (para cargas en dirección de la gravedad se usa signo negativo). La salida “Cu” del componente “COMBINACION DE CARGAS” es la que nos brinda la carga ultima o la combinación de cargas de mayor valor, la cual también esta como carga distribuida en kN/m y con su respectivo signo de dirección de carga (negativo porque va en dirección de la gravedad).

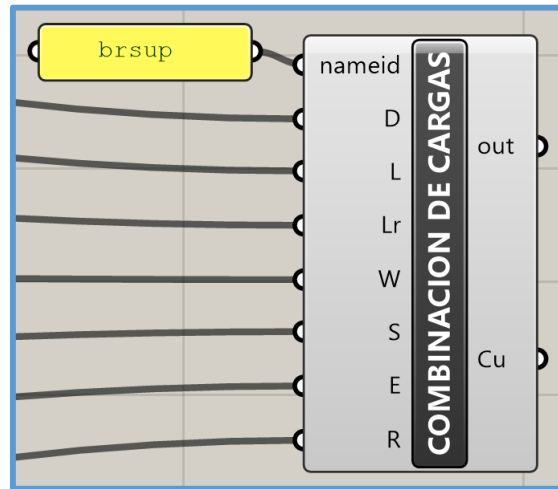


Figura 3.5. Script “COMBINACION DE CARGAS”.

3.4.1.6. Script N°6

El sexto script tiene por nombre “FILTRO1”, este script tiene dos entradas, la primera es la entrada “DC” que es la deflexión máxima (en dirección de la gravedad) calculada al aplicar la carga de servicio y la segunda entrada es la entrada “DL” que es la deflexión limite en dirección de la gravedad, calculada en función de la luz que cubre la armadura (de acuerdo a lo establecido en la N.T.E E0.20). Lo que hace el script es, si la deflexión máxima es mayor que la deflexión limite, a la deflexión máxima se le asigna un valor de 1000, que es brindado por la salida “DCO” del script, pero si la deflexión máxima es menor que la deflexión limite, la deflexión máxima conserva su valor el cual también es brindado por la salida “DCO”, esto se realiza para que al momento de usar Octopus o Galápagos y optimizar, estos tomen valores de deflexión máxima menores o iguales a la deflexión limite.

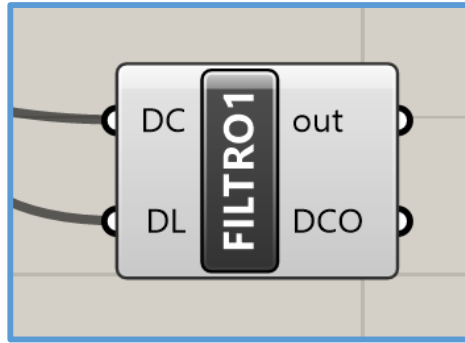


Figura 3.6. Script “FILTRO1”.

3.4.1.7. Script N°7

El séptimo script tiene por nombre “AISC - TENSION - COMPRESION C”. El script diseña elementos o miembros (de una armadura) con secciones de perfiles tubulares (Hss) circulares a tensión y/o compresión según la AISC 360-10.

➤ **Dentro del diseño de miembros a tensión se tiene las siguientes consideraciones:**

- El script trabaja con secciones sin perforaciones conectadas completamente por soldadura, por lo tanto, el área neta será igual al área bruta ($A_n = A_g$).
- El script verifica que la relación L / r no exceda 300, para el caso de elementos o miembros a tensión. Si en caso se excediera 300, entonces el script no diseña los elementos e indica que se reduzca las longitudes mayores de estos y así pueda se cumplir con dicha condición.
- El área neta efectiva (A_e) se define en la sección 2.2.7.7.3 , donde nos indica que es el producto del área neta(A_n) por el factor de corte diferido, el cual se determina según la tabla 2.3 , en esta investigación solo se está trabajando con el caso 1 de la tabla 2.3 (el cual menciona que el caso se aplica a todos los miembros a tensión donde la carga de tensión se transmite directamente a cada uno de los elementos de la sección mediante sujetadores o soldaduras), donde el factor de corte diferido tiene un valor de 1, por lo tanto se cumple que : $A_e = A_n = A_g$.
- Karamba no tiene alguna propiedad o método que nos brinde el valor de F_u , el cual es necesario para el diseño de los elementos o miembros a tensión, entonces se cuenta con la entrada F_u en la cual se ingresara su respectivo valor en psi (lb/in²).

➤ **Dentro del diseño de miembros a compresión se tiene las siguientes consideraciones:**

- El script calcula la relación ancho/espesor de las secciones utilizadas según el caso 9 de la tabla 2.2 (relación ancho espesor para secciones tubulares circulares), y la compara con su λ_r (relación ancho/espesor limite), determinando así, si las secciones son de elementos no esbeltos o de elementos esbeltos, si son de elementos no esbeltos se realiza el diseño de los elementos por pandeo global y si son de elementos esbeltos se realiza el diseño de los elementos por pandeo local.
- El script verifica que la relación KL / r no exceda 200, para el caso de elementos o miembros a compresión. Si en caso se excediera 200, entonces el script no diseña los elementos e indica que se reduzca las longitudes mayores de los estos y así pueda se cumplir con dicha condición.

➤ **Entradas del Script “AISC - TENSION - COMPRESION C”**

- Soldadura: Se escribe en un componente “panel” los procesos de soldadura “ERW” o “SAW”, con los cuales se elabora los perfiles de secciones tubulares (Hss) circulares (según sea el caso), los cuales están indicados en la AISC 360-10 y se conecta a la entrada “Soldadura”. Si se escribe un proceso de soldadura distinto a los dos antes mencionados, las salidas del script (exceptuando la salida “Ratio”) muestran los siguiente: “Indicar un proceso de soldadura usado para la elaboración de perfiles tubulares(HSS), especificado en la AISC 360-10” y la salida “Ratio” arroja todos sus valores como cero.
- Secciones: Se ingresa la sección transversal (sección tubular circular) usada para los elementos, de las distintas formas indicadas en el manual de Karamba. Cabe mencionar que una de las formas que indica dicho manual, es que se puede hacer una lista con secciones personalizadas y trabajar con ellas.
- Elementos: Se conecta la salida “Elem” del componente “ModifyElem” a la entrada “Elementos”, ingresando así los elementos que se quieren diseñar”. Si se conectan otras salidas tipo “Elem” que no provengan del componente “ModifyElem”, se realizara un diseño incorrecto puesto que con en el componente “ModifyElem”, modificamos las longitudes de pandeo de los elementos para que estén de acuerdo a la AISC 360-10.

- Material: Se ingresa el material (solo acero) de los elementos, de las distintas formas indicadas en el manual de Karamba. Cabe mencionar que una de las formas que indica dicho manual, es que se puede hacer una lista con materiales personalizados y trabajar con estos.
- Fu: Se ingresa el Fu o tensión ultima del material en psi (lb/in2).
- Fuerzas: Se conecta la salida “N” del componente “B-Res-Force” a la entrada “Fuerzas”, ingresando así las fuerzas actuantes en los elementos.

Nota:

Para elementos que no presentan fuerzas actuantes, se los considerara en el diseño como elementos o miembros a compresión en los cuales su ratio de diseño será cero.

➤ **Salidas del Script “AISC - TENSION - COMPRESION C”**

- Ranchoespesor: Esta salida para el caso de elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor de la sección con la cual se está trabajando en el diseño, para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión (ya que la verificación de la relación ancho/espesor solo aplica a elementos o miembros a compresión).
- RanchoespesorL: Esta salida para el caso de elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor límite (λ_r), con la que se clasifica la sección con la cual se está trabajando como sección de elementos no esbeltos o sección de elementos esbeltos y para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión.
- RanchoespesorLEE: Esta salida para el caso elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor límite para secciones de elementos esbeltos (hasta dicho valor la sección se considera como de elementos esbeltos), si se supera ese valor lo que se debe de hacer es cambiar de sección puesto que la AISC 360-10 no considera el diseño de secciones que superen ese valor de relación ancho/espesor. Para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión.

- **EsbeltezSección:** Esta salida nos indica si la sección es de elementos no esbeltos o de elementos esbeltos para el caso de elementos o miembros a compresión, para el caso de elementos o miembros a tensión nos indica que el elemento o miembro se encuentra a tensión (ya que el verificar si una sección es de elementos no esbeltos o de elementos esbeltos solo aplica a elementos o miembros a compresión).
- **REsbeltezEfectiva:** Nos brinda el valor de esbeltez efectiva de cada uno de los elementos o miembros con los cuales estamos trabajando. Nos brinda el valor de KL/r para elementos o miembros a compresión y L/r para elementos o miembros a tensión.
- **CREsbeltezEfectiva:** Esta salida nos indica la condición de la esbeltez efectiva de los elementos o miembros, para el caso de elementos o miembros a tensión nos indica que la esbeltez efectiva es adecuada si el valor L/r es menor igual a 300, si L/r es mayor a 300 nos indica que debemos reducir las longitudes mayores de los elementos, así lograremos cumplir con L/r menor igual a 300. Para el caso de elementos o miembros a compresión nos indica que la esbeltez efectiva es adecuada si el valor KL/r es menor igual a 200, si KL/r es mayor a 200 nos indica que debemos reducir las longitudes mayores de los elementos, así lograremos cumplir con KL/r menor igual a 200.
- **PandeoElemento:** Esta salida para el caso elementos o miembros a compresión nos indica el tipo de pandeo (inelástico o elástico) que se produce en estos. Para elementos o miembros a tensión nos indica que dichos elementos o miembros se encuentran a tensión (ya que el pandeo solo aplica a elementos o miembros a compresión).
- **RDiseño:** Esta salida nos brinda los valores de la resistencia de diseño (en psi) de cada uno de los elementos o miembros (a tensión y/o compresión) con los cuales se está trabajando.
- **Ratio:** Esta salida nos brinda el valor de los ratios de diseño de cada uno de los elementos o miembros (a tensión y/o compresión) con los cuales se está trabajando.
- **DiseñoElemento:** Esta salida nos indica si el diseño es correcto (cuando el ratio de diseño es menor o igual a 1) o si el diseño es incorrecto (cuando el ratio de diseño es mayor a 1).



Figura 3.7. Script “AISC - TENSION - COMPRESION C”.

3.4.1.8. Script N°8

El octavo script tiene por nombre “AISC - TENSION - COMPRESION R”. El script diseña elementos o miembros (de una armadura) con secciones de perfiles tubulares (Hss) rectangulares a tensión y/o compresión según la AISC 360-10.

➤ **Dentro del diseño de miembros a tensión se tiene las siguientes consideraciones:**

- El script trabaja con secciones sin perforaciones conectadas completamente por soldadura, por lo tanto, el área neta será igual al área bruta ($A_n = A_g$).
- El script verifica que la relación L / r no exceda 300, para el caso de elementos o miembros a tensión. Si en caso se excediera 300, entonces el script no diseña los elementos e indica que se reduzca las longitudes mayores de estos y así pueda se cumplir con dicha condición.
- El área neta efectiva (A_e) se define en la sección 2.2.7.7.3, donde nos indica que es el producto del área neta (A_n) por el factor de corte diferido, el cual se determina según la tabla 2.3, en esta investigación solo se está trabajando con el caso 1 de la tabla 2.3 (el cual menciona que el caso se aplica a todos los miembros a tensión donde la carga de tensión se transmite directamente a cada uno de los elementos de la sección mediante sujetadores o soldaduras), donde el factor de corte diferido tiene un valor de 1, por lo tanto se cumple que : $A_e = A_n = A_g$.

- Karamba no tiene alguna propiedad o método que nos brinde el valor de F_u , el cual es necesario para el diseño de los elementos o miembros a tensión, entonces se cuenta con la entrada F_u en la cual se ingresara su respectivo valor en psi (lb/in²).
- **Dentro del diseño de miembros a compresión se tiene las siguientes consideraciones:**
- El script calcula la relación ancho/espesor de las secciones utilizadas según el caso 6 de la tabla 2.2 (relación ancho espesor para secciones tubulares rectangulares), y la compara con su λ_r (relación ancho/espesor limite), determinando así, si las secciones son de elementos no esbeltos o de elementos esbeltos, si son de elementos no esbeltos se realiza el diseño de los elementos por pandeo global y si son de elementos esbeltos se realiza el diseño de los elementos por pandeo local.
 - El script verifica que la relación KL / r no exceda 200, para el caso de elementos o miembros a compresión. Si en caso se excediera 200, entonces el script no diseña los elementos e indica que se reduzca las longitudes mayores de estos y así pueda se cumplir con dicha condición.
- **Entradas del Script “AISC - TENSION - COMPRESION R”**
- Soldadura: Se escribe en un componente “panel” los procesos de soldadura “ERW” o “SAW”, con los cuales se elabora los perfiles de secciones tubulares (Hss) rectangulares (según sea el caso), los cuales están indicados en la AISC 360-10 y se conecta a la entrada “Soldadura”. Si se escribe un proceso de soldadura distinto a los dos antes mencionados, las salidas del script (exceptuando la salida “Ratio”) muestran los siguiente: “Indicar un proceso de soldadura usado para la elaboración de perfiles tubulares(HSS), especificado en la AISC 360-10” y la salida “Ratio” arroja todos sus valores como cero.
 - Secciones: Se ingresa la sección transversal (sección tubular rectangular) usada para los elementos, de las distintas formas indicadas en el manual de Karamba. Cabe mencionar que una de las formas que indica dicho manual, es que se puede hacer una lista con secciones personalizadas y trabajar con ellas.

- Elementos: Se conecta la salida “Elem” del componente “ModifyElem” a la entrada “Elementos”, ingresando así los elementos que se quieren diseñar”. Si se conectan otras salidas tipo “Elem” que no provengan del componente “ModifyElem”, se realizara un diseño incorrecto puesto que con en el componente “ModifyElem”, modificamos las longitudes de pandeo de los elementos para que estén de acuerdo a la AISC 360-10.
- Material: Se ingresa el material (solo acero) de los elementos, de las distintas formas indicadas en el manual de Karamba. Cabe mencionar que una de las formas que indica dicho manual, es que se puede hacer una lista con materiales personalizados y trabajar con estos.
- Fu: Se ingresa el Fu o tensión ultima del material en psi (lb/in²).
- Fuerzas: Se conecta la salida “N” del componente “B-Res-Force” a la entrada “Fuerzas”, ingresando así las fuerzas actuantes en los elementos.

Nota:

Para elementos que no presentan fuerzas actuantes, se los considerara en el diseño como elementos o miembros a compresión en los cuales su ratio de diseño será cero.

➤ **Salidas del Script “AISC - TENSION - COMPRESION R”**

- RanchoespesorA: Esta salida para el caso de elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor evaluando el ancho de la sección con la cual se está trabajando en el diseño, para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión (ya que la verificación de la relación ancho/espesor solo aplica a elementos o miembros a compresión).
- RanchoespesorP: Esta salida para el caso de elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor evaluando el peralte de la sección con la cual se está trabajando en el diseño, para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión (ya que la verificación de la relación ancho/espesor solo aplica a elementos o miembros a compresión).
- RanchoespesorL: Esta salida para el caso de elementos o miembros a compresión nos brinda el valor de la relación ancho/espesor límite (λ_r), con la que se clasifica la sección con la cual se está trabajando como sección de elementos no esbeltos o sección de

elementos esbeltos y para el caso de elementos o miembros a tensión solo indicara que dichos elementos o miembros se encuentran a tensión.

- **EsbeltezSección:** Esta salida nos indica si la sección es de elementos no esbeltos o de elementos esbeltos (sección de ancho esbelto, sección de peralte esbelto o sección de ancho y peralte esbeltos, según se de uno de estos casos) para el caso de elementos o miembros a compresión, para el caso de elementos o miembros a tensión nos indica que el elemento o miembro se encuentra a tensión (ya que el verificar si una sección es de elementos no esbeltos o de elementos esbeltos solo aplica a elementos o miembros a compresión).
- **REsbeltezEfectiva:** Nos brinda el valor de esbeltez efectiva de cada uno de los elementos o miembros con los cuales estamos trabajando. Nos brinda el valor de KL/r para elementos o miembros a compresión y L/r para elementos o miembros a tensión.
- **CREsbeltezEfectiva:** Esta salida nos indica la condición de la esbeltez efectiva de los elementos o miembros, para el caso de elementos o miembros a tensión nos indica que la esbeltez efectiva es adecuada si el valor L/r es menor igual a 300, si L/r es mayor a 300 nos indica que debemos reducir las longitudes mayores de los elementos, así lograremos cumplir con L/r menor igual a 300. Para el caso de elementos o miembros a compresión nos indica que la esbeltez efectiva es adecuada si el valor KL/r es menor igual a 200, si KL/r es mayor a 200 nos indica que debemos reducir las longitudes mayores de los elementos, así lograremos cumplir con KL/r menor igual a 200.
- **PandeoElemento:** Esta salida para el caso elementos o miembros a compresión nos indica el tipo de pandeo (inelástico o elástico) que se produce en estos. Para elementos o miembros a tensión nos indica que dichos elementos o miembros se encuentran a tensión (ya que el pandeo solo aplica a elementos o miembros a compresión).
- **RDiseño:** Esta salida nos brinda los valores de la resistencia de diseño (en psi) de cada uno de los elementos o miembros (a tensión y/o compresión) con los cuales se está trabajando.
- **Ratio:** Esta salida nos brinda el valor de los ratios de diseño de cada uno de los elementos o miembros (a tensión y/o compresión) con los cuales se está trabajando.
- **DiseñoElemento:** Esta salida nos indica si el diseño es correcto (cuando el ratio de diseño es menor o igual a 1) o si el diseño es incorrecto (cuando el ratio de diseño es mayor a 1).



Figura 3.8. Script “AISC - TENSION - COMPRESION R”.

3.4.1.9. Script N°9

El noveno script tiene por nombre “FILTRO2”, este script tiene una entrada que es la entrada “Ratio” donde se ingresan los ratios de diseño demanda/capacidad de los elementos o miembros de armadura, brindados por los scripts “AISC - TENSION - COMPRESION C” y/o “AISC - TENSION - COMPRESION R”, lo que hace el script es, si existen ratios mayores a 1, entonces a esos ratios le asigna un valor de 1000, el cual es brindado por la salida “NRatio” del script, pero si los ratios son menores o iguales a 1, estos conservan su valor y también son brindados por la salida “NRatio”; esto es necesario para realizar la optimización de secciones transversales con Galápagos y así este no tome valores de ratio mayores a 1.

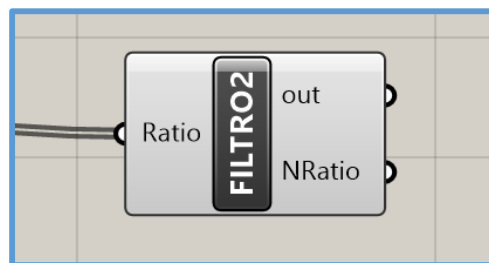


Figura 3.9. Script “FILTRO2”.

3.4.1.10. Script N°10

El décimo script tiene por nombre “FILTRO3”, este script tiene una entrada que es la entrada “Ratio” donde se ingresan los ratios de diseño demanda/capacidad de los elementos o miembros de armadura, brindados por el script “FILTRO2”, lo que hace el script es, si existe algún ratio igual a 1000, entonces a todos los ratios restantes se les asigna un valor de 1000, los cuales son brindados por la salida “NRatio” del script, pero si todos los ratios son distintos a 1000, entonces conservan su valor y también son brindados por la salida “NRatio”; esto es necesario para realizar la optimización de secciones transversales con Galápagos y así todos los ratios de diseño de los elementos o miembros adopten valores menores o iguales a 1.

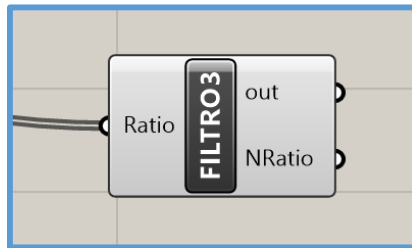


Figura 3.10. Script “FILTRO3”.

3.4.1.11. Script N°11

El undécimo script tiene por nombre “PESO”, este script tiene una entrada que es la entrada “Model” donde se ingresa el modelo de la armadura luego de la optimización topológica (brindado por el componente “BESOBBeam”), lo que hace el script es obtener el peso de la armadura (en kg) luego de la optimización topología y lo brinda a través de la salida “P”.



Figura 3.11. Script “PESO”.

Notas:

- En las entradas de los scripts anteriores solo se deben conectar las salidas ya indicadas, cualquier otro tipo de conexión puede hacer que el script se torne rojo (indicando que existen errores) y arroje todas sus salidas como “null” o que trabaje de manera errónea.
- Cuando no se conecta ningún dato a las entradas de los scripts estos se tornarán rojo indicando que el script presenta errores, es por eso que se debe conectar todos los datos respectivos, si no se usare algún dato de entrada, como puede ser el caso del script 5, en cual puede que no se utilice todos los tipos de carga, se debe colocar un valor de cero en las entradas de los tipos de carga que no se usaran y no se debe dejarlas vacías.
- En los scripts 4 y 5 las cargas a considerar tienen una orientación local, es por eso que los metrados de cargas se deben realizar con una orientación local de las cargas respecto a los elementos (sobre los cuales actuaran).
- La salida “out” en cada uno de los scripts anteriores es la salida por defecto de los scripts personalizados en Grasshopper la cual brinda información sobre errores presentados en el código en Lenguaje de programación C Sharp (C#) del script, si no hay errores pues no brinda ningún dato.
- Los scripts 7 y 8 solo trabajan con elementos o miembros de armadura con secciones transversales de espesor constante.
- Los códigos en lenguaje de programación C Sharp (C#) de cada uno de los scripts anteriores se encuentran en los anexos.

3.4.2. Ejemplo N°1

En este ejemplo se ha diseñado una armadura plana tipo Howe mediante el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” desarrollado en esta investigación.

3.4.2.1. Modelación paramétrica

➤ Se han definido los siguientes parámetros para la armadura plana tipo Howe.

- Luz a cubrir. Tiene un rango de 1m a 20m.
- Peralte de armadura plana. Tiene un rango de 0.1m a 5m.
- Número de segmentos de armadura plana. Tiene un rango de 2 a 80.

Se ha trabajado con una luz a cubrir de 10m.

➤ Se puede apreciar el código gráfico de la modelación paramétrica del ejemplo N°1 en la siguiente figura:

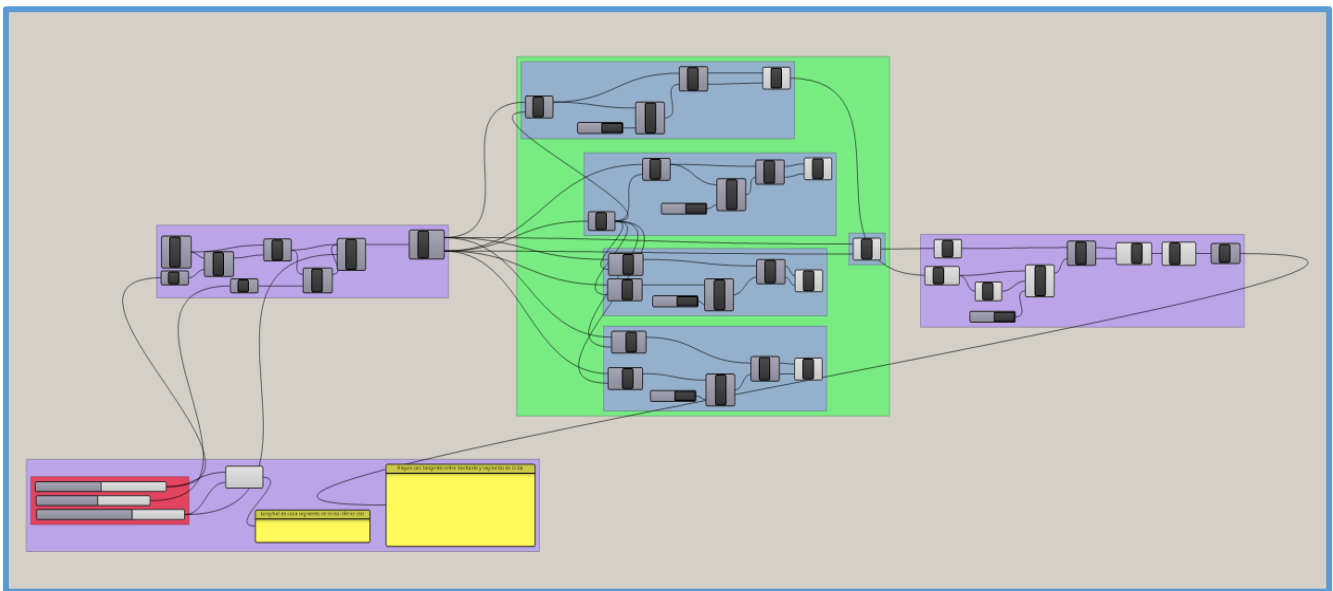


Figura 3.12. Código gráfico de la modelación paramétrica del ejemplo N°1.

3.4.2.2. Simulación estructural

Concluida la modelación paramétrica, se definió el código gráfico del análisis estructural y posteriormente se realizó simulaciones estructurales, así tenemos el código gráfico de la siguiente figura:

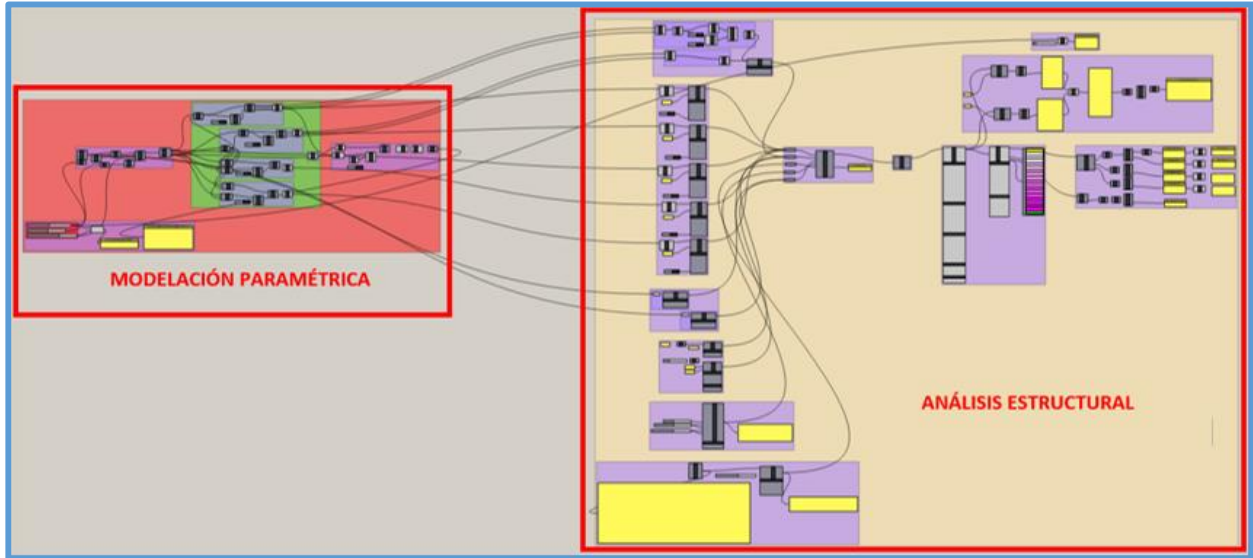



Figura 3.13. Código gráfico de modelación paramétrica y análisis estructural - ejemplo N°1.

En la siguiente figura se muestra la armadura plana tipo Howe, luego de definido el código gráfico del análisis estructural.



Figura 3.14. Armadura plana tipo Howe con código gráfico del análisis estructural - ejemplo N°1.

Una vez definido el código gráfico del análisis estructural, se procedió a realizar las simulaciones estructurales, mediante variación de parámetros. En este ejemplo se realizó 2 tipos de simulaciones estructurales, en la primera, con un peralte de armadura constante de 0.65m, se evaluó el comportamiento de la armadura debido a la variación de su número de segmentos (o número de segmentos de brida) con respecto a la variación de la carga. En la segunda para una carga constante de 5kN se evaluó el comportamiento de la armadura debido a la variación de su número de

segmentos (o número de segmentos de brida) con respecto a la variación del peralte. Los comportamientos más óptimos se encuentran sombreados de color anaranjado oscuro ().

Se usó los parámetros de control definidos en el panel de control de la modelación paramétrica. La luz a cubrir se mantuvo constante, los parámetros que variaron fueron el peralte de armadura y el número de segmentos de armadura.

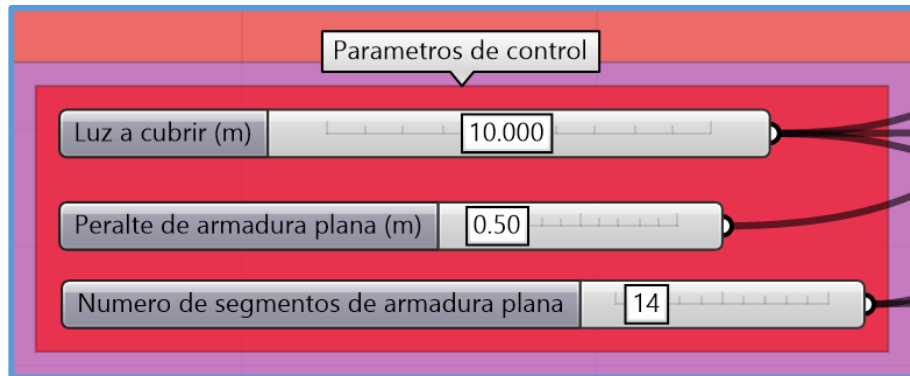


Figura 3.15. Parámetros de control - simulación estructural - ejemplo N°1.

- Variación del número de segmentos de armadura con respecto a la variación de la carga (peralte constante de 0.65m).

1. Carga de 3.00 kN/m.

Tabla 3.1. Datos de simulación estructural - Carga de 3kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0135580 m	58.36 kN	62.26 kN	23.70	1.25 m
10	33.02 °	254.24 kN	0.0132730 m	60.00 kN	62.50 kN	23.80	1.00 m
12	37.95 °	267.62 kN	0.0131660 m	61.01 kN	62.76 kN	23.90	0.83 m
14	42.30 °	281.53 kN	0.0131450 m	61.74 kN	63.03 kN	24.00	0.71 m
16	46.12 °	295.89 kN	0.0131700 m	62.31 kN	63.30 kN	24.10	0.63 m
18	49.48 °	310.62 kN	0.0132250 m	62.80 kN	63.58 kN	24.20	0.56 m
20	52.43 °	325.65 kN	0.0132990 m	63.23 kN	63.87 kN	24.30	0.50 m
22	55.03 °	340.93 kN	0.0133860 m	63.64 kN	64.17 kN	24.40	0.45 m
24	57.34 °	356.41 kN	0.0134800 m	64.02 kN	64.47 kN	24.50	0.42 m
26	59.39 °	372.07 kN	0.0135840 m	64.38 kN	64.77 kN	24.60	0.38 m
28	61.21 °	387.88 kN	0.0136920 m	64.74 kN	65.07 kN	24.80	0.36 m
30	62.85 °	403.81 kN	0.0138040 m	65.09 kN	65.38 kN	24.90	0.33 m
32	64.32 °	419.84 kN	0.0139200 m	65.43 kN	65.68 kN	25.00	0.31 m
34	65.65 °	435.97 kN	0.0140400 m	65.77 kN	65.99 kN	25.10	0.29 m
36	66.86 °	452.17 kN	0.0141620 m	66.10 kN	66.31 kN	25.20	0.28 m
38	67.96 °	468.43 kN	0.0142850 m	66.43 kN	66.62 kN	25.30	0.26 m
40	68.96 °	484.76 kN	0.0144120 m	66.77 kN	66.93 kN	25.50	0.25 m

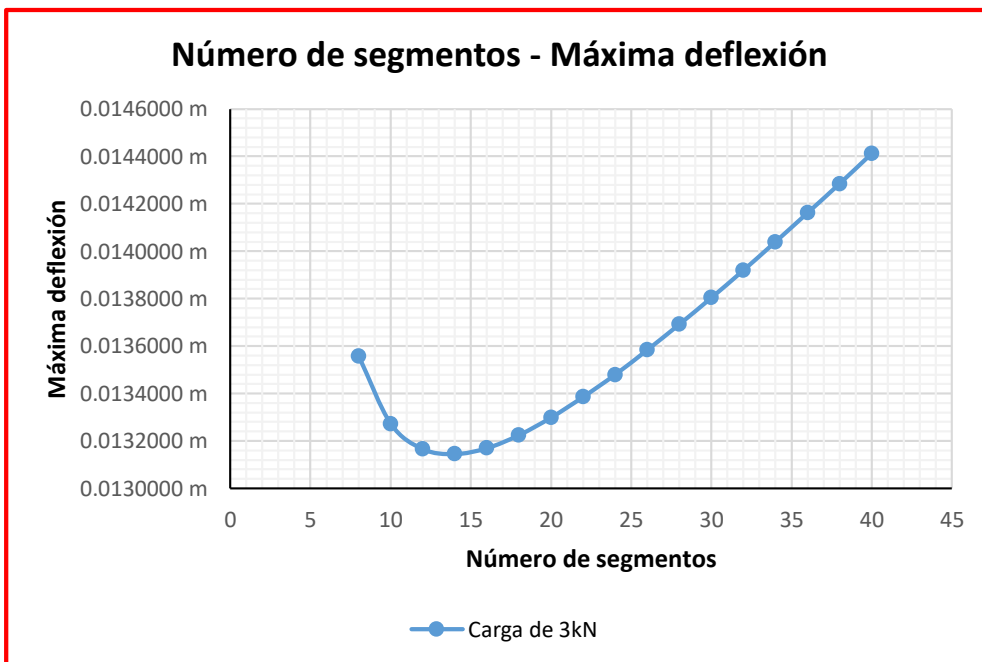


Figura 3.16. Número de segmentos - Máxima deflexión - Carga de 3kN - ejemplo N°1.

2. Carga de 4.00 kN/m.

Tabla 3.2. Datos de simulación estructural - Carga de 4kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0177450 m	76.39 kN	81.49 kN	31.00	1.25 m
10	33.02 °	254.24 kN	0.0173560 m	78.46 kN	81.73 kN	31.10	1.00 m
12	37.95 °	267.62 kN	0.0172000 m	79.71 kN	81.99 kN	31.20	0.83 m
14	42.30 °	281.53 kN	0.0171550 m	80.58 kN	82.26 kN	31.30	0.71 m
16	46.12 °	295.89 kN	0.0171700 m	81.24 kN	82.53 kN	31.40	0.63 m
18	49.48 °	310.62 kN	0.0172240 m	81.79 kN	82.82 kN	31.50	0.56 m
20	52.43 °	325.65 kN	0.0173020 m	82.27 kN	83.10 kN	31.60	0.50 m
22	55.03 °	340.93 kN	0.0173960 m	82.71 kN	83.40 kN	31.70	0.45 m
24	57.34 °	356.41 kN	0.0175010 m	83.11 kN	83.70 kN	31.80	0.42 m
26	59.39 °	372.07 kN	0.0176160 m	83.50 kN	84.00 kN	32.00	0.38 m
28	61.21 °	387.88 kN	0.0177370 m	83.87 kN	84.30 kN	32.10	0.36 m
30	62.85 °	403.81 kN	0.0178640 m	84.23 kN	84.61 kN	32.20	0.33 m
32	64.32 °	419.84 kN	0.0179940 m	84.58 kN	84.92 kN	32.30	0.31 m
34	65.65 °	435.97 kN	0.0181300 m	84.93 kN	85.23 kN	32.40	0.29 m
36	66.86 °	452.17 kN	0.0182680 m	85.27 kN	85.54 kN	32.50	0.28 m
38	67.96 °	468.43 kN	0.0184080 m	85.61 kN	85.85 kN	32.70	0.26 m
40	68.96 °	484.76 kN	0.0185510 m	85.95 kN	86.16 kN	32.80	0.25 m

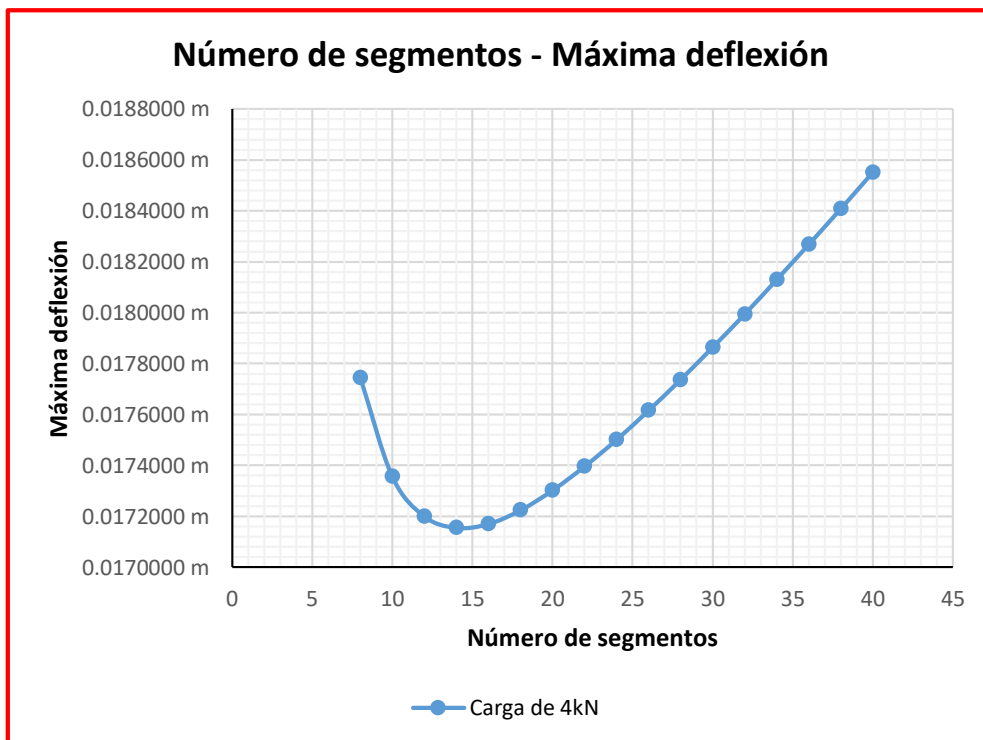


Figura 3.17. Número de segmentos - Máxima deflexión - Carga de 4kN - ejemplo N°1.

3. Carga de 5.00 kN/m.

Tabla 3.3. Datos de simulación estructural – Carga de 5kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0219320 m	94.42 kN	100.72 kN	38.30	1.25 m
10	33.02 °	254.24 kN	0.0214390 m	96.92 kN	100.96 kN	38.40	1.00 m
12	37.95 °	267.62 kN	0.0212330 m	98.41 kN	101.22 kN	38.50	0.83 m
14	42.30 °	281.53 kN	0.0211650 m	99.42 kN	101.49 kN	38.60	0.71 m
16	46.12 °	295.89 kN	0.0211710 m	100.17 kN	101.76 kN	38.70	0.63 m
18	49.48 °	310.62 kN	0.0212230 m	100.79 kN	102.05 kN	38.80	0.56 m
20	52.43 °	325.65 kN	0.0213050 m	101.31 kN	102.33 kN	38.90	0.50 m
22	55.03 °	340.93 kN	0.0214070 m	101.78 kN	102.63 kN	39.00	0.45 m
24	57.34 °	356.41 kN	0.0215210 m	102.21 kN	102.93 kN	39.20	0.42 m
26	59.39 °	372.07 kN	0.0216480 m	102.62 kN	103.23 kN	39.30	0.38 m
28	61.21 °	387.88 kN	0.0217830 m	103.00 kN	103.53 kN	39.40	0.36 m
30	62.85 °	403.81 kN	0.0219230 m	103.38 kN	103.84 kN	39.50	0.33 m
32	64.32 °	419.84 kN	0.0220690 m	103.74 kN	104.15 kN	39.60	0.31 m
34	65.65 °	435.97 kN	0.0222200 m	104.10 kN	104.46 kN	39.70	0.29 m
36	66.86 °	452.17 kN	0.0223740 m	104.44 kN	104.77 kN	39.90	0.28 m
38	67.96 °	468.43 kN	0.0225300 m	104.79 kN	105.08 kN	40.00	0.26 m
40	68.96 °	484.76 kN	0.0226900 m	105.13 kN	105.39 kN	40.10	0.25 m

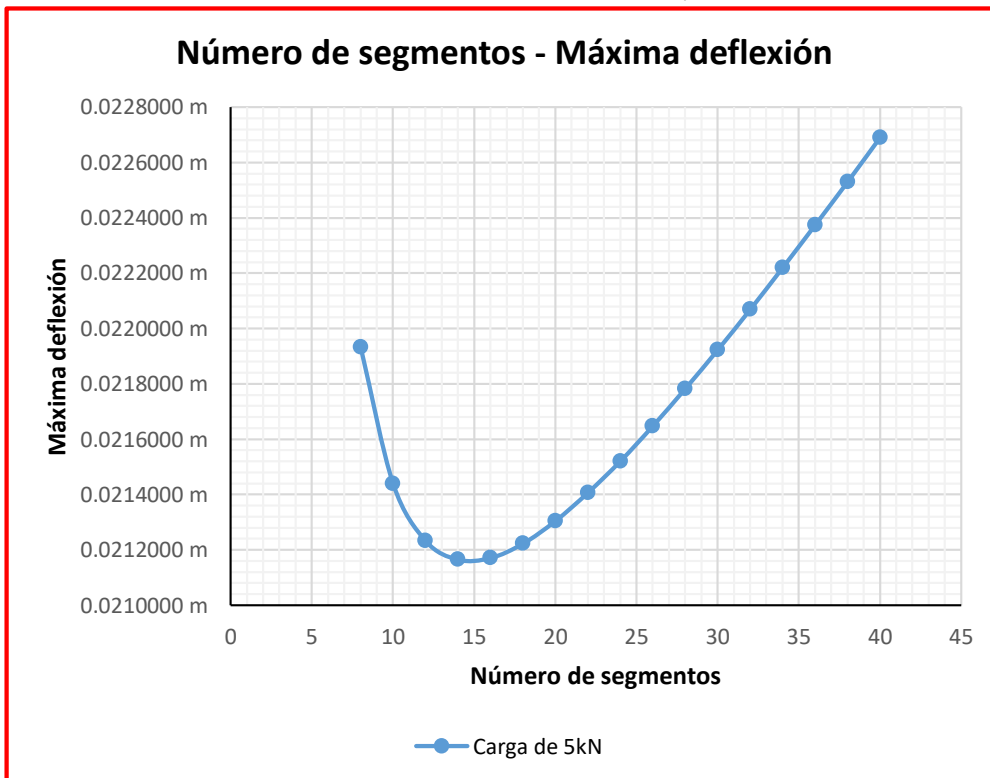


Figura 3.18. Número de segmentos – Máxima deflexión – Carga de 5kN - ejemplo N°1.

4. Carga de 6.00 kN/m.

Tabla 3.4. Datos de simulación estructural – Carga de 6kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0261190 m	112.45 kN	119.95 kN	45.60	1.25 m
10	33.02 °	254.24 kN	0.0252230 m	115.38 kN	120.19 kN	45.70	1.00 m
12	37.95 °	267.62 kN	0.0252670 m	117.10 kN	120.45 kN	45.80	0.83 m
14	42.30 °	281.53 kN	0.0251750 m	118.25 kN	120.72 kN	45.90	0.71 m
16	46.12 °	295.89 kN	0.0251710 m	119.10 kN	120.99 kN	46.00	0.63 m
18	49.48 °	310.62 kN	0.0252220 m	119.78 kN	121.28 kN	46.10	0.56 m
20	52.43 °	325.65 kN	0.0253080 m	120.35 kN	121.57 kN	46.20	0.50 m
22	55.03 °	340.93 kN	0.0254170 m	120.85 kN	121.86 kN	46.40	0.45 m
24	57.34 °	356.41 kN	0.0255420 m	121.31 kN	122.16 kN	46.50	0.42 m
26	59.39 °	372.07 kN	0.0256800 m	121.73 kN	122.46 kN	46.60	0.38 m
28	61.21 °	387.88 kN	0.0258280 m	122.14 kN	122.76 kN	46.70	0.36 m
30	62.85 °	403.81 kN	0.0259830 m	122.52 kN	123.07 kN	46.80	0.33 m
32	64.32 °	419.84 kN	0.0261430 m	122.90 kN	123.38 kN	46.90	0.31 m
34	65.65 °	435.97 kN	0.0263090 m	123.26 kN	123.69 kN	47.10	0.29 m
36	66.86 °	452.17 kN	0.0264800 m	123.62 kN	124.00 kN	47.20	0.28 m
38	67.96 °	468.43 kN	0.0266530 m	123.97 kN	124.31 kN	47.30	0.26 m
40	68.96 °	484.76 kN	0.0268300 m	124.31 kN	124.63 kN	47.40	0.25 m

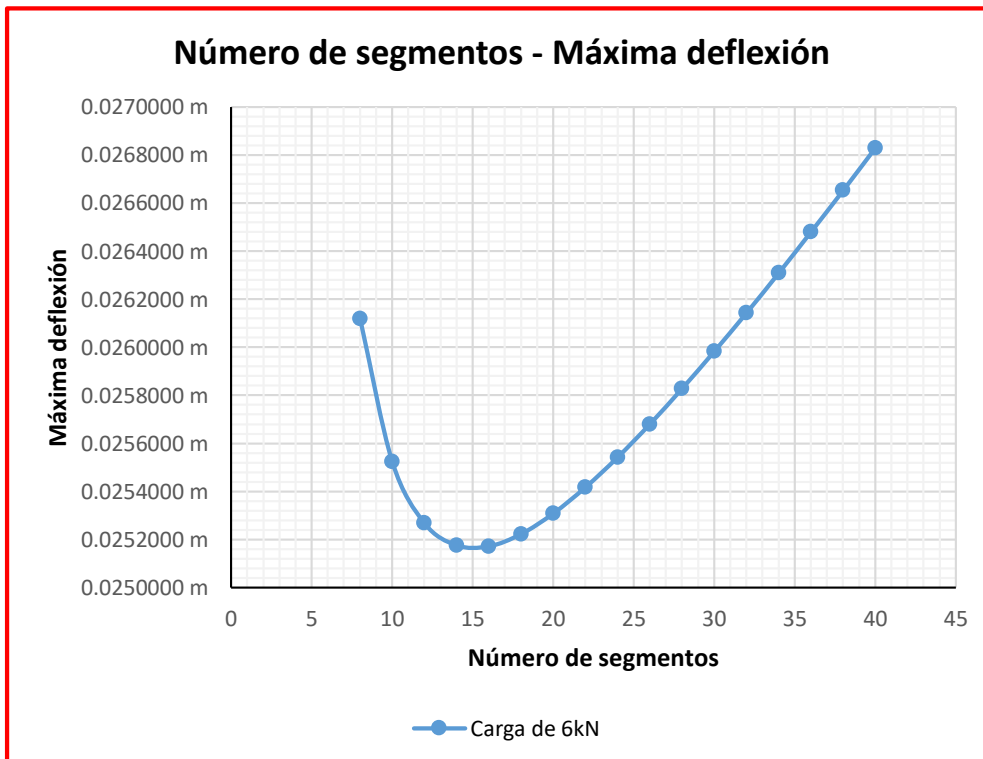


Figura 3.19. Número de segmentos – Máxima deflexión – Carga de 6kN - ejemplo N°1.

5. Carga de 7.00 kN/m.

Tabla 3.5. Datos de simulación estructural - Carga de 7kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0303070 m	130.48 kN	139.18 kN	52.90	1.25 m
10	33.02 °	254.24 kN	0.0296060 m	133.85 kN	139.42 kN	53.00	1.00 m
12	37.95 °	267.62 kN	0.0293010 m	135.80 kN	139.68 kN	53.10	0.83 m
14	42.30 °	281.53 kN	0.0291850 m	137.09 kN	139.95 kN	53.20	0.71 m
16	46.12 °	295.89 kN	0.0291720 m	138.03 kN	140.22 kN	53.30	0.63 m
18	49.48 °	310.62 kN	0.0292210 m	138.77 kN	140.51 kN	53.50	0.56 m
20	52.43 °	325.65 kN	0.0293110 m	139.39 kN	140.80 kN	53.60	0.50 m
22	55.03 °	340.93 kN	0.0294280 m	139.92 kN	141.09 kN	53.70	0.45 m
24	57.34 °	356.41 kN	0.0295620 m	140.41 kN	141.39 kN	53.80	0.42 m
26	59.39 °	372.07 kN	0.0297130 m	140.85 kN	141.69 kN	53.90	0.38 m
28	61.21 °	387.88 kN	0.0298730 m	141.27 kN	141.99 kN	54.00	0.36 m
30	62.85 °	403.81 kN	0.0300420 m	141.67 kN	142.30 kN	54.10	0.33 m
32	64.32 °	419.84 kN	0.0302180 m	142.05 kN	142.61 kN	54.20	0.31 m
34	65.65 °	435.97 kN	0.0303990 m	142.42 kN	142.92 kN	54.40	0.29 m
36	66.86 °	452.17 kN	0.0305860 m	142.79 kN	143.23 kN	54.50	0.28 m
38	67.96 °	468.43 kN	0.0307750 m	143.14 kN	143.54 kN	54.60	0.26 m
40	68.96 °	484.76 kN	0.0309690 m	143.50 kN	143.86 kN	54.70	0.25 m

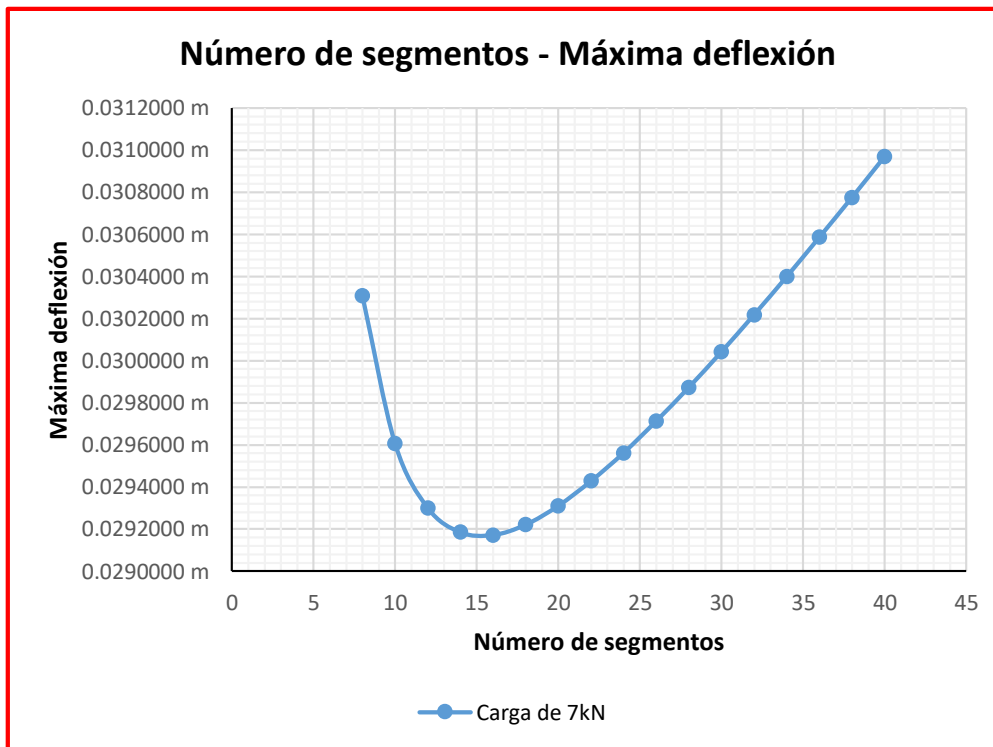


Figura 3.20. Número de segmentos – Máxima deflexión – Carga de 7kN - ejemplo N°1.

6. Carga de 8.00 kN/m.

Tabla 3.6. Datos de simulación estructural – Carga de 8kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0344940 m	148.51 kN	158.41 kN	60.30	1.25 m
10	33.02 °	254.24 kN	0.0336890 m	152.31 kN	158.65 kN	60.40	1.00 m
12	37.95 °	267.62 kN	0.0333340 m	154.50 kN	158.91 kN	60.50	0.83 m
14	42.30 °	281.53 kN	0.0331950 m	155.93 kN	159.18 kN	60.60	0.71 m
16	46.12 °	295.89 kN	0.0331720 m	156.96 kN	159.46 kN	60.70	0.63 m
18	49.48 °	310.62 kN	0.0332210 m	157.77 kN	159.74 kN	60.80	0.56 m
20	52.43 °	325.65 kN	0.0333140 m	158.43 kN	160.03 kN	60.90	0.50 m
22	55.03 °	340.93 kN	0.0334380 m	159.00 kN	160.32 kN	61.00	0.45 m
24	57.34 °	356.41 kN	0.0335830 m	159.50 kN	160.62 kN	61.10	0.42 m
26	59.39 °	372.07 kN	0.0337450 m	159.97 kN	160.92 kN	61.20	0.38 m
28	61.21 °	387.88 kN	0.0339190 m	160.40 kN	161.22 kN	61.30	0.36 m
30	62.85 °	403.81 kN	0.0341020 m	160.81 kN	161.53 kN	61.40	0.33 m
32	64.32 °	419.84 kN	0.0342920 m	161.21 kN	161.84 kN	61.60	0.31 m
34	65.65 °	435.97 kN	0.0344890 m	161.59 kN	162.15 kN	61.70	0.29 m
36	66.86 °	452.17 kN	0.0346920 m	161.96 kN	162.46 kN	61.80	0.28 m
38	67.96 °	468.43 kN	0.0348980 m	162.32 kN	162.77 kN	61.90	0.26 m
40	68.96 °	484.76 kN	0.0351080 m	162.68 kN	163.09 kN	62.00	0.25 m

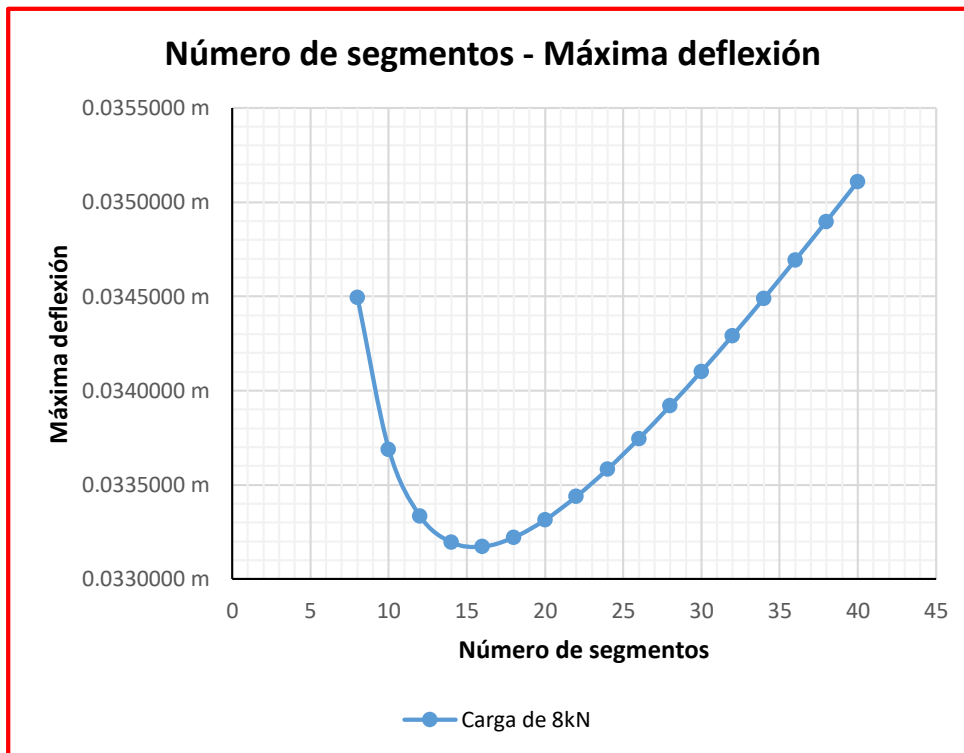


Figura 3.21. Número de segmentos – Máxima deflexión – Carga de 8kN - ejemplo N°1.

7. Carga de 9.00 kN/m.

Tabla 3.7. Datos de simulación estructural – Carga de 9kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0386810 m	166.54 kN	177.64 kN	67.60	1.25 m
10	33.02 °	254.24 kN	0.0377720 m	170.77 kN	177.88 kN	67.70	1.00 m
12	37.95 °	267.62 kN	0.0373680 m	173.19 kN	178.14 kN	67.80	0.83 m
14	42.30 °	281.53 kN	0.0372050 m	174.77 kN	178.41 kN	67.90	0.71 m
16	46.12 °	295.89 kN	0.0371720 m	175.89 kN	178.69 kN	68.00	0.63 m
18	49.48 °	310.62 kN	0.0372200 m	176.76 kN	178.97 kN	68.10	0.56 m
20	52.43 °	325.65 kN	0.0373170 m	177.47 kN	179.26 kN	68.20	0.50 m
22	55.03 °	340.93 kN	0.0374490 m	178.07 kN	179.55 kN	68.30	0.45 m
24	57.34 °	356.41 kN	0.0376030 m	178.60 kN	179.85 kN	68.40	0.42 m
26	59.39 °	372.07 kN	0.0377770 m	179.08 kN	180.15 kN	68.50	0.38 m
28	61.21 °	387.88 kN	0.0379640 m	179.53 kN	180.45 kN	68.60	0.36 m
30	62.85 °	403.81 kN	0.0381610 m	179.96 kN	180.76 kN	68.80	0.33 m
32	64.32 °	419.84 kN	0.0383660 m	180.36 kN	181.07 kN	68.90	0.31 m
34	65.65 °	435.97 kN	0.0385790 m	180.75 kN	181.38 kN	69.00	0.29 m
36	66.86 °	452.17 kN	0.0387980 m	181.13 kN	181.69 kN	69.10	0.28 m
38	67.96 °	468.43 kN	0.0390200 m	181.50 kN	182.00 kN	69.20	0.26 m
40	68.96 °	484.76 kN	0.0392480 m	181.86 kN	182.32 kN	69.40	0.25 m

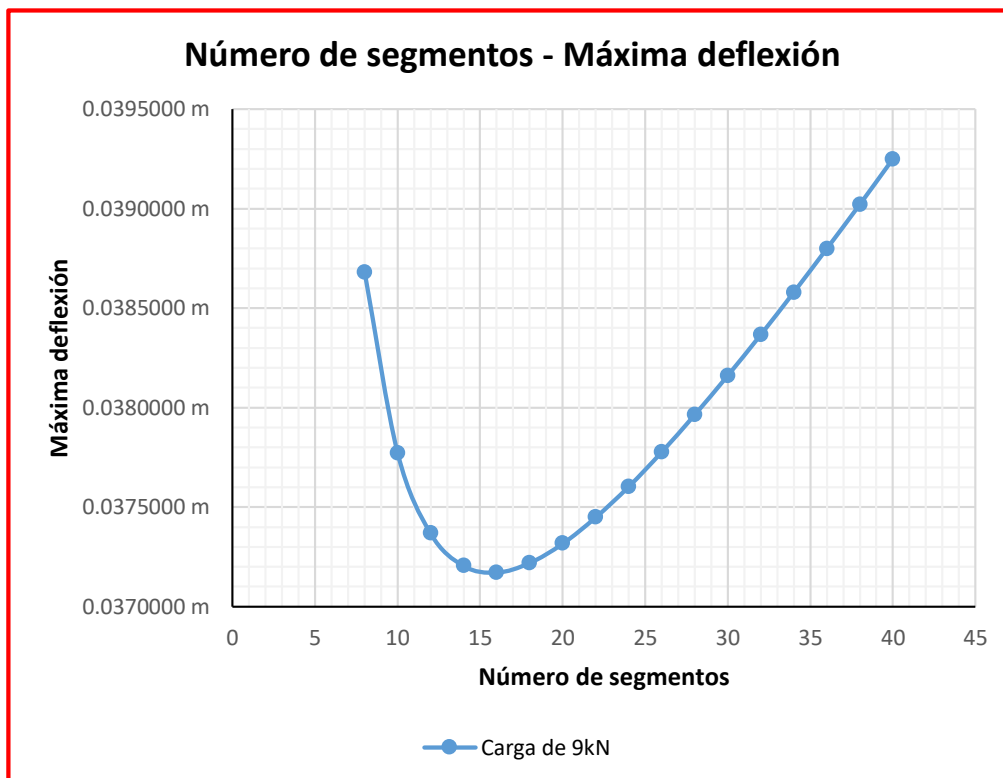


Figura 3.22. Número de segmentos – Máxima deflexión – Carga de 9kN - ejemplo N°1.

8. Carga de 10.00 kN/m.

Tabla 3.8. Datos de simulación estructural – Carga de 10kN - ejemplo N°1.

Luz = L = 10.00 m Peralte = P = 0.65 m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0428680 m	184.57 kN	196.87 kN	74.90	1.25 m
10	33.02 °	254.24 kN	0.0418560 m	189.23 kN	197.12 kN	75.00	1.00 m
12	37.95 °	267.62 kN	0.0414020 m	191.89 kN	197.37 kN	75.10	0.83 m
14	42.30 °	281.53 kN	0.0412150 m	193.61 kN	197.64 kN	75.20	0.71 m
16	46.12 °	295.89 kN	0.0411730 m	194.82 kN	197.92 kN	75.30	0.63 m
18	49.48 °	310.62 kN	0.0412190 m	195.75 kN	198.20 kN	75.40	0.56 m
20	52.43 °	325.65 kN	0.0413200 m	196.50 kN	198.49 kN	75.50	0.50 m
22	55.03 °	340.93 kN	0.0414590 m	197.14 kN	198.78 kN	75.60	0.45 m
24	57.34 °	356.41 kN	0.0416240 m	197.70 kN	199.08 kN	75.70	0.42 m
26	59.39 °	372.07 kN	0.0418090 m	198.20 kN	199.38 kN	75.80	0.38 m
28	61.21 °	387.88 kN	0.0420090 m	198.67 kN	199.69 kN	76.00	0.36 m
30	62.85 °	403.81 kN	0.0422200 m	199.10 kN	199.99 kN	76.10	0.33 m
32	64.32 °	419.84 kN	0.0424410 m	199.52 kN	200.30 kN	76.20	0.31 m
34	65.65 °	435.97 kN	0.0426690 m	199.92 kN	200.61 kN	76.30	0.29 m
36	66.86 °	452.17 kN	0.0429040 m	200.30 kN	200.92 kN	76.40	0.28 m
38	67.96 °	468.43 kN	0.0431430 m	200.68 kN	201.23 kN	76.60	0.26 m
40	68.96 °	484.76 kN	0.0433870 m	201.04 kN	201.55 kN	76.70	0.25 m

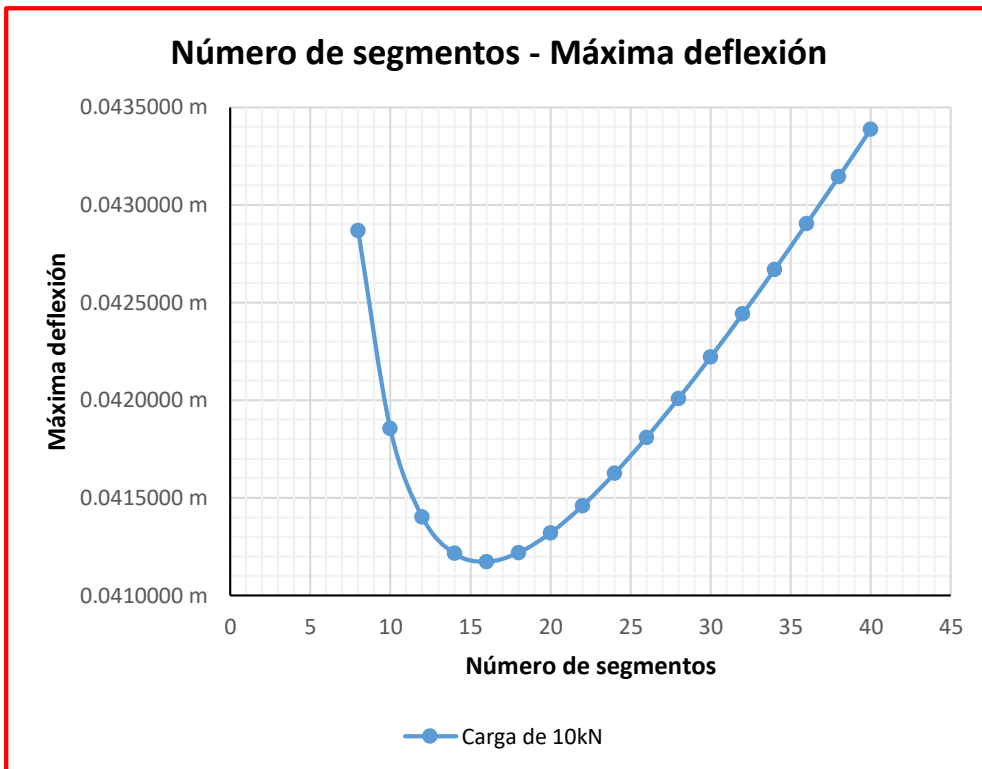


Figura 3.23. Número de segmentos – Máxima deflexión – Carga de 10kN - ejemplo N°1.

- **Variación del número de segmentos de armadura con respecto a la variación del peralte (carga constante de 5kN/m).**

1. Peralte de 0.60m.

Tabla 3.9. *Datos de simulación estructural - Peralte de 0.60m - ejemplo N°1.*

Luz = L = 10.00 m Carga = W = 5.00 kN/m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	25.64 °	237.43 kN	0.0255760 m	102.22 kN	109.03 kN	41.50	1.25 m
10	30.96 °	248.94 kN	0.0249750 m	104.90 kN	109.27 kN	41.60	1.00 m
12	35.75 °	261.05 kN	0.0247070 m	106.48 kN	109.52 kN	41.70	0.83 m
14	40.03 °	273.65 kN	0.0245980 m	107.55 kN	109.79 kN	41.80	0.71 m
16	43.83 °	286.68 kN	0.0245780 m	108.34 kN	110.06 kN	41.90	0.63 m
18	47.20 °	300.06 kN	0.0246100 m	108.97 kN	110.34 kN	42.00	0.56 m
20	50.19 °	313.74 kN	0.0246750 m	109.52 kN	110.62 kN	42.10	0.50 m
22	52.85 °	327.66 kN	0.0247640 m	109.99 kN	110.91 kN	42.20	0.45 m
24	55.22 °	341.78 kN	0.0248680 m	110.43 kN	111.21 kN	42.30	0.42 m
26	57.34 °	356.09 kN	0.0249850 m	110.84 kN	111.50 kN	42.40	0.38 m
28	59.24 °	370.54 kN	0.0251100 m	111.23 kN	111.80 kN	42.50	0.36 m
30	60.95 °	385.11 kN	0.0252430 m	111.61 kN	112.11 kN	42.60	0.33 m
32	62.49 °	399.80 kN	0.0253810 m	111.98 kN	112.41 kN	42.80	0.31 m
34	63.89 °	414.57 kN	0.0255250 m	112.33 kN	112.72 kN	42.90	0.29 m
36	65.16 °	429.43 kN	0.0256720 m	112.68 kN	113.03 kN	43.00	0.28 m
38	66.32 °	444.36 kN	0.0258220 m	113.03 kN	113.34 kN	43.10	0.26 m
40	67.38 °	459.34 kN	0.0259750 m	113.37 kN	113.65 kN	43.20	0.25 m
42	68.36 °	474.38 kN	0.0261310 m	113.71 kN	113.97 kN	43.40	0.24 m
44	69.25 °	489.469105	0.026289	114.04649	114.282612	43.50	0.23 m

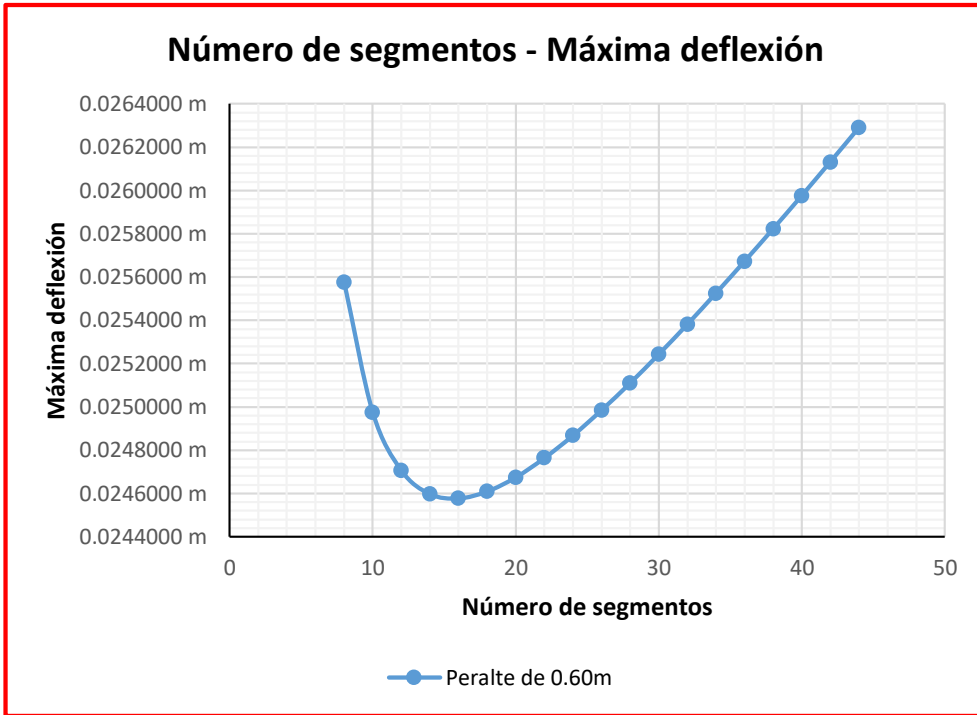


Figura 3.24. Número de segmentos - Máxima deflexión - Peralte de 0.60m - ejemplo N°1.

2. Peralte de 0.65m.

Tabla 3.10. Datos de simulación estructural – Peralte de 0.65m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN/m

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	27.47 °	241.52 kN	0.0219320 m	94.42 kN	100.72 kN	38.30	1.25 m
10	33.02 °	254.24 kN	0.0214390 m	96.92 kN	100.96 kN	38.40	1.00 m
12	37.95 °	267.62 kN	0.0212330 m	98.41 kN	101.22 kN	38.50	0.83 m
14	42.30 °	281.53 kN	0.0211650 m	99.42 kN	101.49 kN	38.60	0.71 m
16	46.12 °	295.89 kN	0.0211710 m	100.17 kN	101.76 kN	38.70	0.63 m
18	49.48 °	310.62 kN	0.0212230 m	100.79 kN	102.05 kN	38.80	0.56 m
20	52.43 °	325.65 kN	0.0213050 m	101.31 kN	102.33 kN	38.90	0.50 m
22	55.03 °	340.93 kN	0.0214070 m	101.78 kN	102.63 kN	39.00	0.45 m
24	57.34 °	356.41 kN	0.0215210 m	102.21 kN	102.93 kN	39.20	0.42 m
26	59.39 °	372.07 kN	0.0216480 m	102.62 kN	103.23 kN	39.30	0.38 m
28	61.21 °	387.88 kN	0.0217830 m	103.00 kN	103.53 kN	39.40	0.36 m
30	62.85 °	403.81 kN	0.0219230 m	103.38 kN	103.84 kN	39.50	0.33 m
32	64.32 °	419.84 kN	0.0220690 m	103.74 kN	104.15 kN	39.60	0.31 m
34	65.65 °	435.97 kN	0.0222200 m	104.10 kN	104.46 kN	39.70	0.29 m
36	66.86 °	452.17 kN	0.0223740 m	104.44 kN	104.77 kN	39.90	0.28 m
38	67.96 °	468.43 kN	0.0225300 m	104.79 kN	105.08 kN	40.00	0.26 m
40	68.96 °	484.76 kN	0.0226900 m	105.13 kN	105.39 kN	40.10	0.25 m

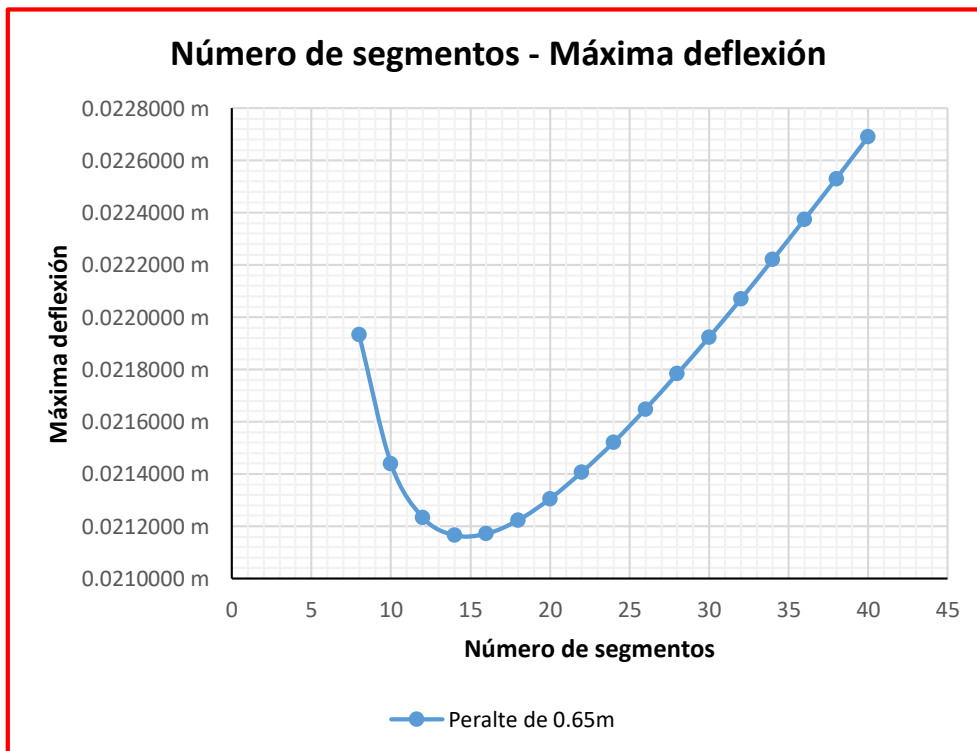


Figura 3.25. Número de segmentos - Máxima deflexión - Peralte de 0.65m - ejemplo N°1.

3. Peralte de 0.70m.

Tabla 3.11. Datos de simulación estructural – Peralte de 0.70m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	29.25 °	245.68 kN	0.0190420 m	87.74 kN	93.59 kN	35.60	1.25 m
10	34.99 °	259.64 kN	0.0186390 m	90.09 kN	93.84 kN	35.70	1.00 m
12	40.03 °	274.30 kN	0.0184830 m	91.49 kN	94.10 kN	35.80	0.83 m
14	44.42 °	289.54 kN	0.0184460 m	92.45 kN	94.37 kN	35.90	0.71 m
16	48.24 °	305.24 kN	0.0184770 m	93.18 kN	94.66 kN	36.00	0.63 m
18	51.56 °	321.32 kN	0.0185460 m	93.77 kN	94.94 kN	36.10	0.56 m
20	54.46 °	337.71 kN	0.0186420 m	94.28 kN	95.23 kN	36.20	0.50 m
22	57.00 °	354.34 kN	0.0187560 m	94.74 kN	95.53 kN	36.30	0.45 m
24	59.24 °	371.19 kN	0.0188810 m	95.17 kN	95.83 kN	36.50	0.42 m
26	61.21 °	388.20 kN	0.0190170 m	95.57 kN	96.14 kN	36.60	0.38 m
28	62.97 °	405.36 kN	0.0191600 m	95.95 kN	96.44 kN	36.70	0.36 m
30	64.54 °	422.64 kN	0.0193090 m	96.32 kN	96.75 kN	36.80	0.33 m
32	65.94 °	440.02 kN	0.0194620 m	96.68 kN	97.06 kN	36.90	0.31 m
34	67.21 °	457.49 kN	0.0196200 m	97.04 kN	97.37 kN	37.00	0.29 m
36	68.36 °	475.03 kN	0.0197820 m	97.39 kN	97.69 kN	37.20	0.28 m
38	69.40 °	492.64 kN	0.0199450 m	97.73 kN	98.00 kN	37.30	0.26 m

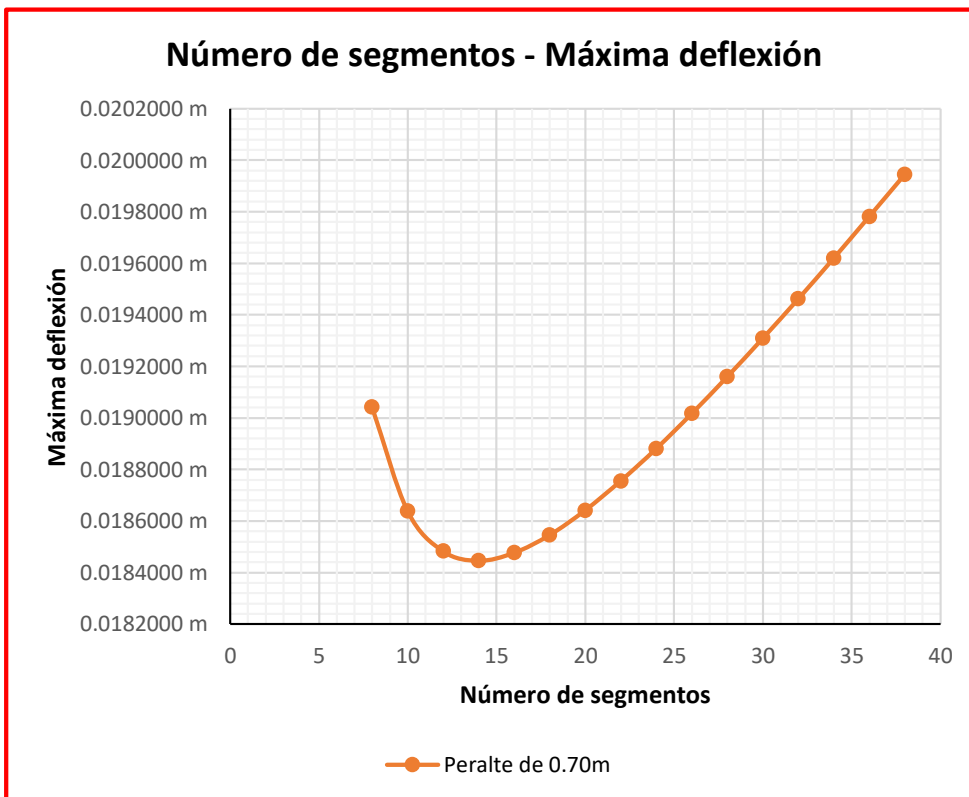


Figura 3.26. Número de segmentos - Máxima deflexión - Peralte de 0.70m - ejemplo N°1.

4. Peralte de 0.75m.

Tabla 3.12. Datos de simulación estructural - Peralte de 0.75m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	30.96 °	249.92 kN	0.0167140 m	81.95 kN	87.42 kN	33.30	1.25 m
10	36.87 °	265.13 kN	0.0163830 m	84.16 kN	87.67 kN	33.40	1.00 m
12	41.99 °	281.09 kN	0.0162700 m	85.49 kN	87.94 kN	33.50	0.83 m
14	46.40 °	297.66 kN	0.0162620 m	86.41 kN	88.21 kN	33.60	0.71 m
16	50.19 °	314.71 kN	0.0163110 m	87.11 kN	88.50 kN	33.70	0.63 m
18	53.47 °	332.15 kN	0.0163970 m	87.69 kN	88.79 kN	33.80	0.56 m
20	56.31 °	349.89 kN	0.0165050 m	88.19 kN	89.08 kN	33.90	0.50 m
22	58.78 °	367.89 kN	0.0166290 m	88.64 kN	89.38 kN	34.00	0.45 m
24	60.95 °	386.09 kN	0.0167660 m	89.06 kN	89.69 kN	34.10	0.42 m
26	62.85 °	404.46 kN	0.0169100 m	89.46 kN	89.99 kN	34.20	0.38 m
28	64.54 °	422.97 kN	0.0170620 m	89.84 kN	90.30 kN	34.40	0.36 m
30	66.04 °	441.59 kN	0.0172190 m	90.21 kN	90.61 kN	34.50	0.33 m
32	67.38 °	460.32 kN	0.0173810 m	90.57 kN	90.92 kN	34.60	0.31 m
34	68.59 °	479.13 kN	0.0175460 m	90.92 kN	91.24 kN	34.70	0.29 m
36	69.68 °	498.00 kN	0.0177150 m	91.27 kN	91.55 kN	34.80	0.28 m

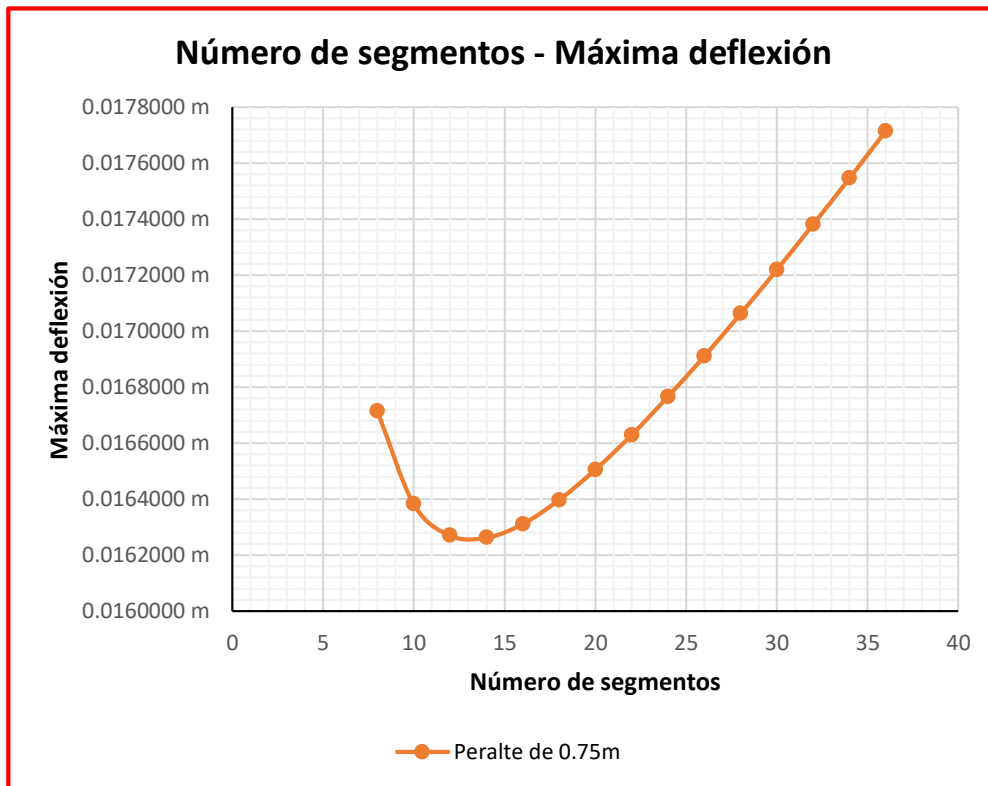


Figura 3.27. Número de segmentos - Máxima deflexión - Peralte de 0.75m - ejemplo N°1.

5. Peralte de 0.80m.

Tabla 3.13. Datos de simulación estructural – Peralte de 0.80m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	% de Utilización	Longitud de segmento de brida
8	32.62 °	254.22 kN	0.0148120 m	76.89 kN	82.02 kN	31.20	1.25 m
10	38.66 °	270.70 kN	0.0145410 m	78.98 kN	82.27 kN	31.30	1.00 m
12	43.83 °	287.98 kN	0.0144630 m	80.25 kN	82.54 kN	31.40	0.83 m
14	48.24 °	305.89 kN	0.0144800 m	81.13 kN	82.82 kN	31.50	0.71 m
16	52.00 °	324.29 kN	0.0145470 m	81.81 kN	83.11 kN	31.60	0.63 m
18	55.22 °	343.09 kN	0.0146470 m	82.37 kN	83.40 kN	31.70	0.56 m
20	57.99 °	362.19 kN	0.0147670 m	82.87 kN	83.70 kN	31.80	0.50 m
22	60.40 °	381.54 kN	0.0149020 m	83.31 kN	84.01 kN	32.00	0.45 m
24	62.49 °	401.10 kN	0.0150480 m	83.73 kN	84.31 kN	32.10	0.42 m
26	64.32 °	420.82 kN	0.0152020 m	84.12 kN	84.62 kN	32.20	0.38 m
28	65.94 °	440.67 kN	0.0153620 m	84.50 kN	84.93 kN	32.30	0.36 m
30	67.38 °	460.64 kN	0.0155280 m	84.86 kN	85.24 kN	32.40	0.33 m
32	68.66 °	480.71 kN	0.0156970 m	85.22 kN	85.55 kN	32.50	0.31 m
34	69.81 °	500.85 kN	0.0158710 m	85.57 kN	85.87 kN	32.70	0.29 m

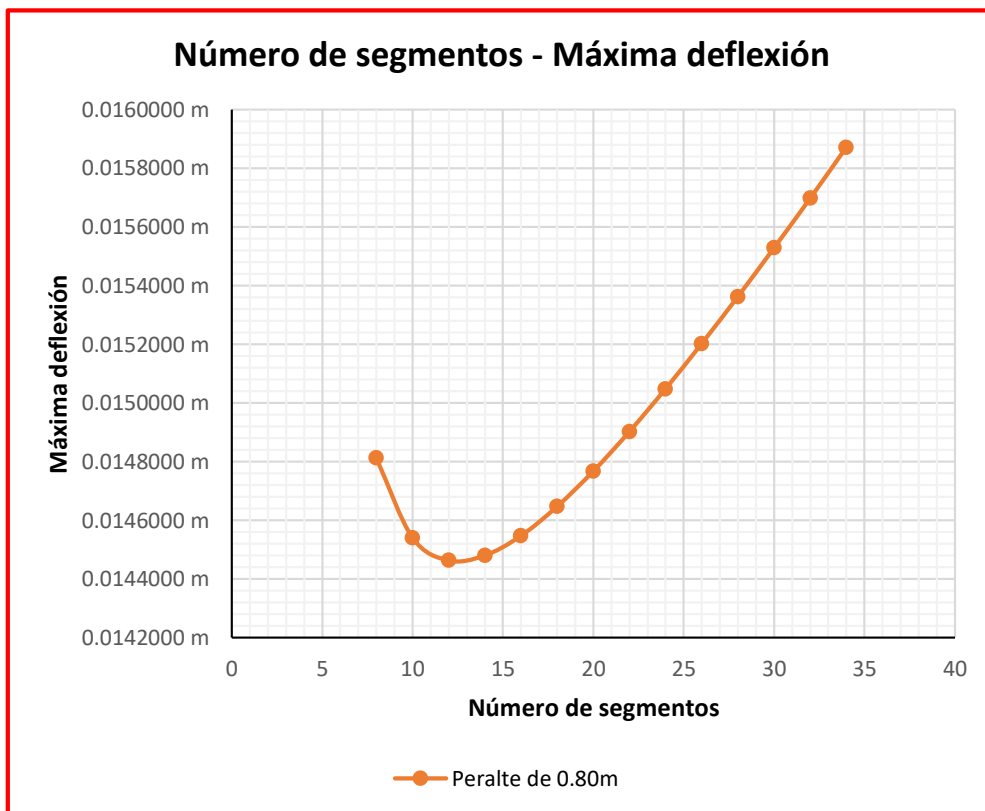


Figura 3.28. Número de segmentos - Máxima deflexión - Peralte de 0.80m - ejemplo N°1.

6. Peralte de 0.85m.

Tabla 3.14. Datos de simulación estructural – Peralte de 0.85m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	34.22 °	258.58 kN	0.0132380 m	72.42 kN	77.25 kN	29.40	1.25 m
10	40.36 °	276.35 kN	0.0130190 m	74.41 kN	77.51 kN	29.50	1.00 m
12	45.57 °	294.96 kN	0.0129720 m	75.62 kN	77.79 kN	29.60	0.83 m
14	49.96 °	314.21 kN	0.0130090 m	76.48 kN	78.07 kN	29.70	0.71 m
16	53.67 °	333.97 kN	0.0130930 m	77.14 kN	78.36 kN	29.80	0.63 m
18	56.83 °	354.12 kN	0.0132060 m	77.68 kN	78.66 kN	29.90	0.56 m
20	59.53 °	374.59 kN	0.0133380 m	78.17 kN	78.96 kN	30.00	0.50 m
22	61.86 °	395.29 kN	0.0134840 m	78.61 kN	79.26 kN	30.20	0.45 m
24	63.89 °	416.20 kN	0.0136390 m	79.02 kN	79.57 kN	30.30	0.42 m
26	65.65 °	437.27 kN	0.0138020 m	79.41 kN	79.88 kN	30.40	0.38 m
28	67.21 °	458.47 kN	0.0139710 m	79.78 kN	80.19 kN	30.50	0.36 m
30	68.59 °	479.78 kN	0.0141450 m	80.15 kN	80.50 kN	30.60	0.33 m
32	69.81 °	501.18 kN	0.0143220 m	80.50 kN	80.82 kN	30.70	0.31 m

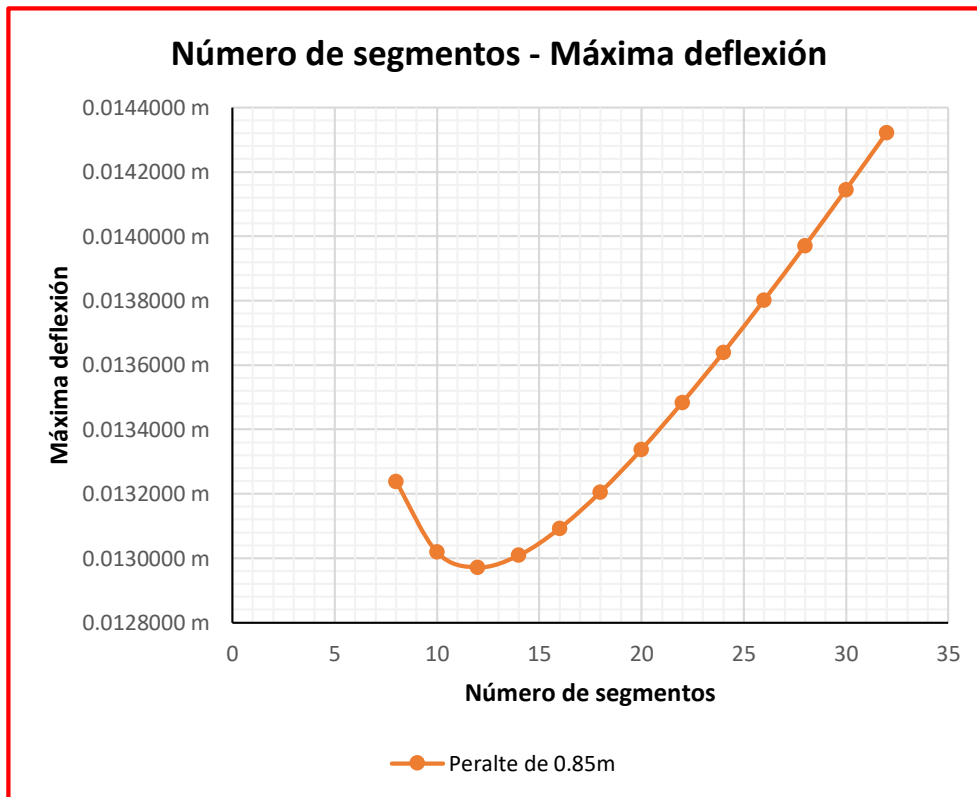


Figura 3.29. Número de segmentos - Máxima deflexión - Peralte de 0.85m - ejemplo N°1.

7. Peralte de 0.90m.

Tabla 3.15. Datos de simulación estructural – Peralte de 0.90m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
8	35.75 °	263.00 kN	0.0119210 m	68.45 kN	73.02 kN	27.80	1.25 m
10	41.99 °	282.07 kN	0.0117470 m	70.35 kN	73.28 kN	27.90	1.00 m
12	47.20 °	302.01 kN	0.0117270 m	71.51 kN	73.56 kN	28.00	0.83 m
14	51.56 °	322.62 kN	0.0117830 m	72.34 kN	73.84 kN	28.10	0.71 m
16	55.22 °	343.74 kN	0.0118820 m	72.98 kN	74.14 kN	28.20	0.63 m
18	58.31 °	365.25 kN	0.0120080 m	73.52 kN	74.44 kN	28.30	0.56 m
20	60.95 °	387.07 kN	0.0121500 m	73.99 kN	74.74 kN	28.40	0.50 m
22	63.20 °	409.13 kN	0.0123060 m	74.43 kN	75.05 kN	28.50	0.45 m
24	65.16 °	431.38 kN	0.0124710 m	74.83 kN	75.35 kN	28.70	0.42 m
26	66.86 °	453.79 kN	0.0126430 m	75.22 kN	75.67 kN	28.80	0.38 m
28	68.36 °	476.33 kN	0.0128210 m	75.59 kN	75.98 kN	28.90	0.36 m
30	69.68 °	498.98 kN	0.0130030 m	75.95 kN	76.29 kN	29.00	0.33 m

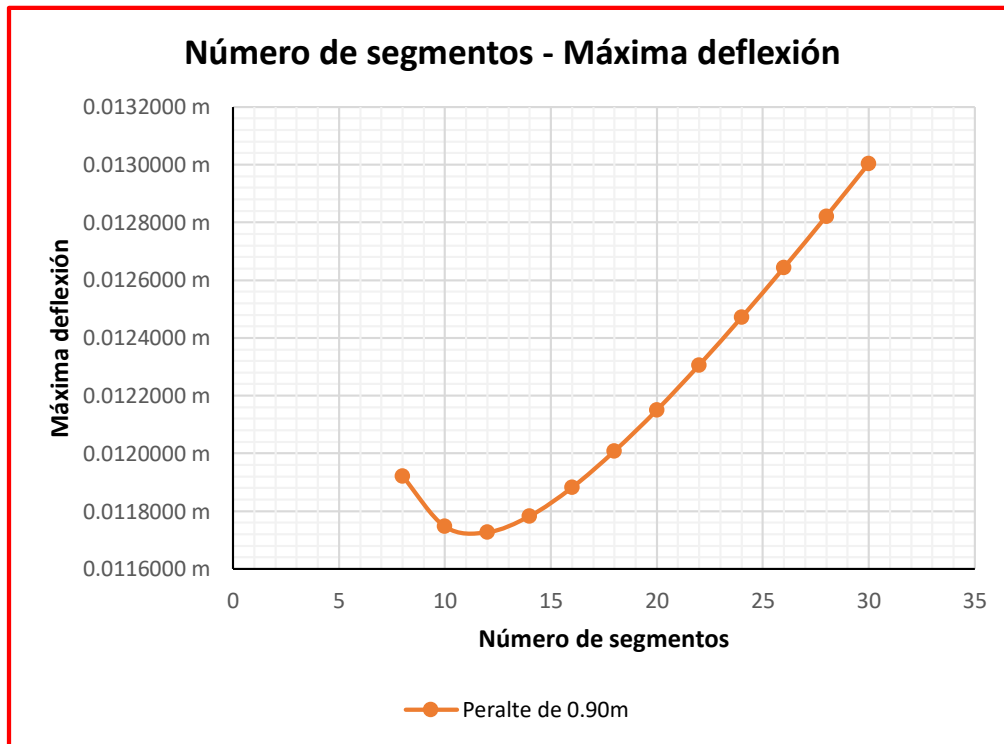


Figura 3.30. Número de segmentos - Máxima deflexión - Peralte de 0.90m - ejemplo N°1.

8. Peralte de 0.95m.

Tabla 3.16. Datos de simulación estructural – Peralte de 0.95m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
6	29.68 °	248.28 kN	0.0112890 m	61.31 kN	68.98 kN	26.20	1.67 m
8	37.23 °	267.47 kN	0.0108100 m	64.90 kN	69.23 kN	26.30	1.25 m
10	43.53 °	287.86 kN	0.0106740 m	66.72 kN	69.50 kN	26.40	1.00 m
12	48.74 °	309.14 kN	0.0106790 m	67.84 kN	69.78 kN	26.50	0.83 m
14	53.06 °	331.10 kN	0.0107520 m	68.63 kN	70.06 kN	26.70	0.71 m
16	56.66 °	353.58 kN	0.0108650 m	69.26 kN	70.36 kN	26.80	0.63 m
18	59.68 °	376.45 kN	0.0110020 m	69.79 kN	70.66 kN	26.90	0.56 m
20	62.24 °	399.62 kN	0.0111570 m	70.26 kN	70.97 kN	27.00	0.50 m
22	64.43 °	423.03 kN	0.0113220 m	70.69 kN	71.27 kN	27.10	0.45 m
24	66.32 °	446.63 kN	0.0114960 m	71.09 kN	71.58 kN	27.20	0.42 m
26	67.96 °	470.39 kN	0.0116770 m	71.47 kN	71.90 kN	27.40	0.38 m
28	69.40 °	494.27 kN	0.0118640 m	71.84 kN	72.21 kN	27.50	0.36 m

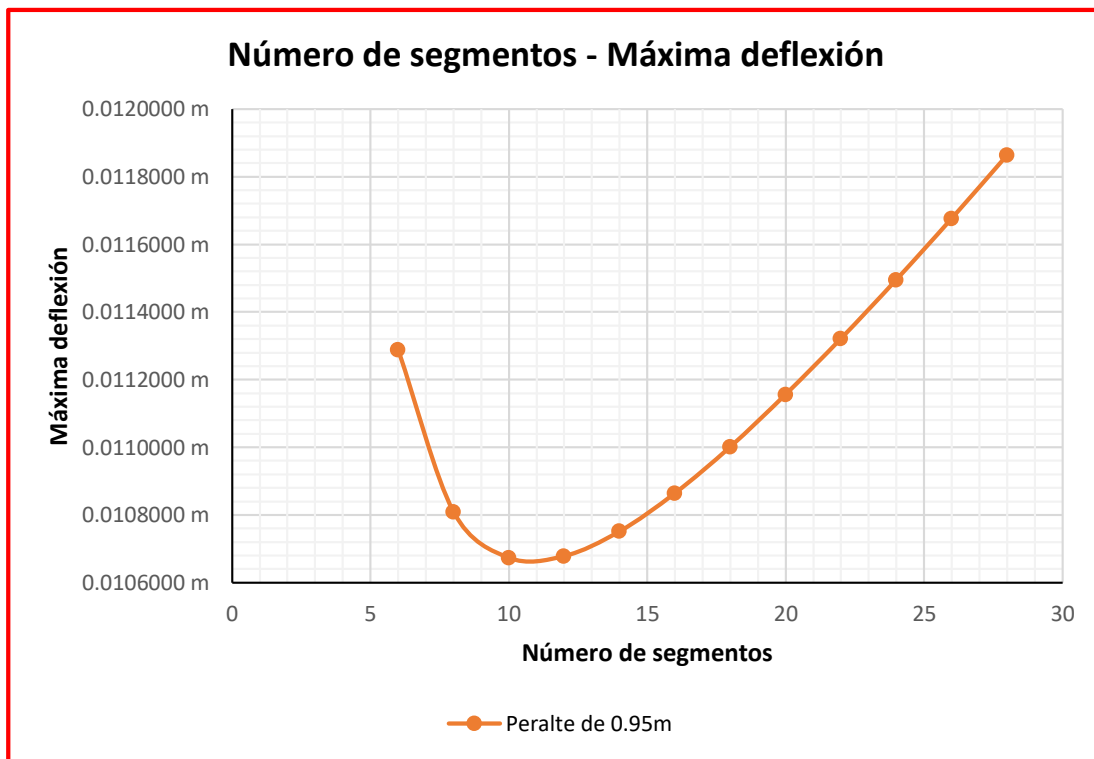


Figura 3.31. Número de segmentos - Máxima deflexión - Peralte de 0.95m - ejemplo N°1.

9. Peralte de 1.00m.

Tabla 3.17. Datos de simulación estructural – Peralte de 1.00m - ejemplo N°1.

Luz = L = 10.00 m Carga = W = 5.00 kN

N° Segmentos	Ángulo	Masa	Máxima deflexión	Máxima fuerza a compresión	Máxima fuerza a tensión	Máximo % de Utilización	Longitud de segmento de brida
6	30.96 °	251.54 kN	0.0102770 m	58.28 kN	65.56 kN	24.90	1.67 m
8	38.66 °	272.00 kN	0.0098630 m	61.71 kN	65.82 kN	25.00	1.25 m
10	45.00 °	293.71 kN	0.0097620 m	63.45 kN	66.09 kN	25.10	1.00 m
12	50.19 °	316.34 kN	0.0097880 m	64.53 kN	66.37 kN	25.20	0.83 m
14	54.46 °	339.66 kN	0.0098790 m	65.30 kN	66.66 kN	25.40	0.71 m
16	57.99 °	363.49 kN	0.0100050 m	65.92 kN	66.96 kN	25.50	0.63 m
18	60.95 °	387.72 kN	0.0101530 m	66.43 kN	67.27 kN	25.60	0.56 m
20	63.43 °	412.24 kN	0.0103180 m	66.90 kN	67.57 kN	25.70	0.50 m
22	65.56 °	437.00 kN	0.0104930 m	67.32 kN	67.88 kN	25.80	0.45 m
24	67.38 °	461.94 kN	0.0106760 m	67.72 kN	68.19 kN	25.90	0.42 m
26	68.96 °	487.04 kN	0.0108670 m	68.10 kN	68.51 kN	26.10	0.38 m

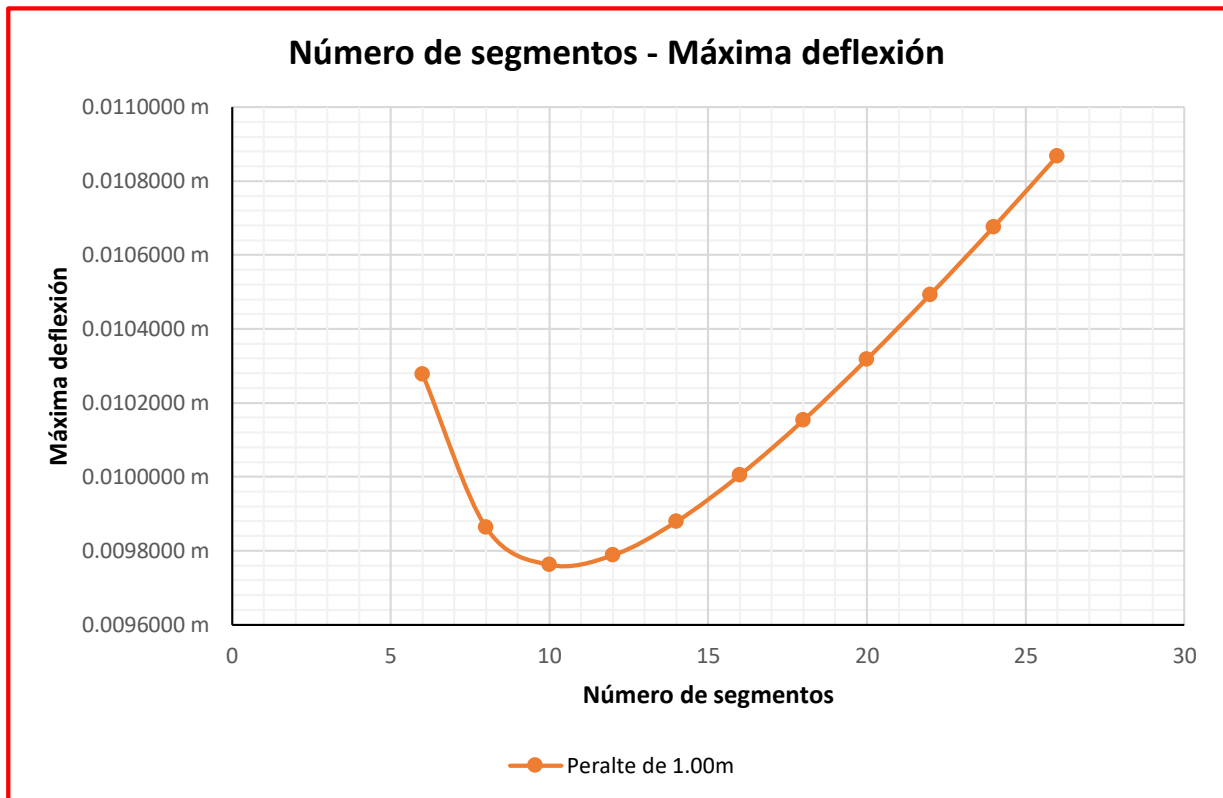


Figura 3.32. Número de segmentos - Máxima deflexión - Peralte de 1.00m - ejemplo N°1.

A continuación, se presenta la tabla 3.18, donde se evalúa el aumento de peralte con la disminución de la máxima deflexión.

Tabla 3.18. *Disminución de la máxima deflexión por aumento de peralte - ejemplo N°1.*

N° Segmentos	Disminución de la máxima deflexión por aumento de peralte							
	0.60-0.65	0.65-0.70	0.70-0.75	0.75-0.8	0.8-0.85	0.85-0.9	0.9-0.95	0.95-1.00
	0.05 m	0.05 m	0.05 m	0.05 m	0.05 m	0.05 m	0.05 m	0.05 m
	0.05 m	0.10 m	0.15 m	0.20 m	0.25 m	0.30 m	0.35 m	0.40 m
6	-	-	-	-	-	-	-	0.00 m
8	0.0036440 m	0.0028900 m	0.0023280 m	0.0019020 m	0.0015740 m	0.0013170 m	0.0011110 m	0.0009470 m
10	0.0035360 m	0.0028000 m	0.0022560 m	0.0018420 m	0.0015220 m	0.0012720 m	0.0010730 m	0.0009120 m
12	0.0034740 m	0.0027500 m	0.0022130 m	0.0018070 m	0.0014910 m	0.0012450 m	0.0010480 m	0.0008910 m
14	0.0034330 m	0.0027190 m	0.0021840 m	0.0017820 m	0.0014710 m	0.0012260 m	0.0010310 m	0.0008730 m
16	0.0034070 m	0.0026940 m	0.0021660 m	0.0017640 m	0.0014540 m	0.0012110 m	0.0010170 m	0.0008600 m
18	0.0033870 m	0.0026770 m	0.0021490 m	0.0017500 m	0.0014410 m	0.0011980 m	0.0010060 m	0.0008490 m
20	0.0033700 m	0.0026630 m	0.0021370 m	0.0017380 m	0.0014290 m	0.0011880 m	0.0009930 m	0.0008390 m
22	0.0033570 m	0.0026510 m	0.0021270 m	0.0017270 m	0.0014180 m	0.0011780 m	0.0009840 m	0.0008290 m
24	0.0033470 m	0.0026400 m	0.0021150 m	0.0017180 m	0.0014090 m	0.0011680 m	0.0009750 m	0.0008200 m
26	0.0033370 m	0.0026310 m	0.0021070 m	0.0017080 m	0.0014000 m	0.0011590 m	0.0009660 m	0.0008100 m
28	0.0033270 m	0.0026230 m	0.0020980 m	0.0017000 m	0.0013910 m	0.0011500 m	0.0009570 m	-
30	0.0033200 m	0.0026140 m	0.0020900 m	0.0016910 m	0.0013830 m	0.0011420 m	-	-
32	0.0033120 m	0.0026070 m	0.0020810 m	0.0016840 m	0.0013750 m	-	-	-
34	0.0033050 m	0.0026000 m	0.0020740 m	0.0016750 m	-	-	-	-
36	0.0032980 m	0.0025920 m	0.0020670 m	-	-	-	-	-
38	0.0032920 m	0.0025850 m	-	-	-	-	-	-
40	0.0032850 m	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-

Intervalo de aumento (m)

Aumento parcial

Aumento acumulado

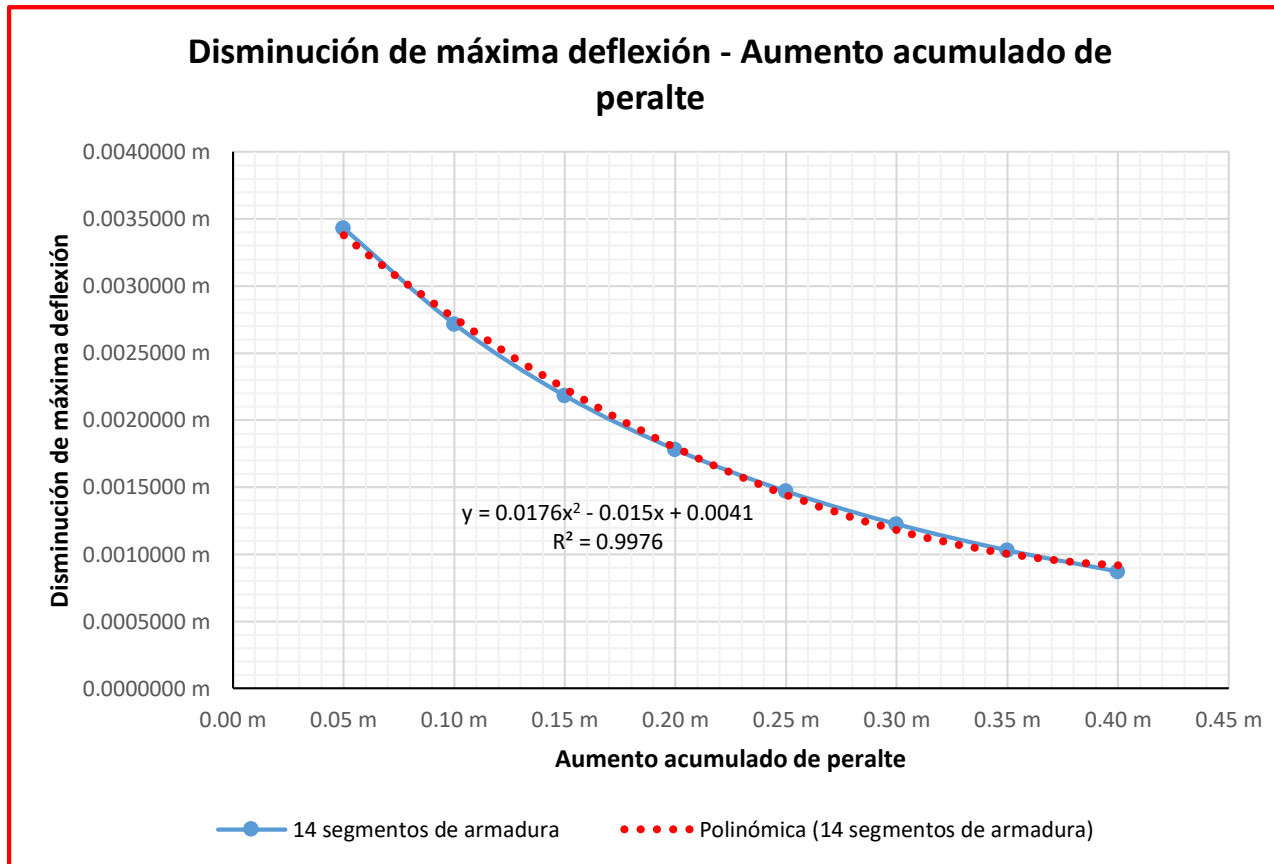


Figura 3.33. Disminución de máxima deflexión - Aumento acumulado de peralte - ejemplo N°1.

A continuación, se presenta la tabla 3.19, donde se evalúa el peralte con la máxima deflexión.

Tabla 3.19. Peralte - Máxima deflexión - ejemplo N°1.

N° Segmentos	Peralte - Máxima deflexión								
	0.60 m	0.65 m	0.70 m	0.75 m	0.80 m	0.85 m	0.90 m	0.95 m	1.00 m
6	-	-	-	-	-	-	-	0.0112890 m	0.010277
8	0.0255760 m	0.0219320 m	0.0190420 m	0.0167140 m	0.0148120 m	0.0132380 m	0.0119210 m	0.0108100 m	0.0098630 m
10	0.0249750 m	0.0214390 m	0.0186390 m	0.0163830 m	0.0145410 m	0.0130190 m	0.0117470 m	0.0106740 m	0.0097620 m
12	0.0247070 m	0.0212330 m	0.0184830 m	0.0162700 m	0.0144630 m	0.0129720 m	0.0117270 m	0.0106790 m	0.0097880 m
14	0.0245980 m	0.0211650 m	0.0184460 m	0.0162620 m	0.0144800 m	0.0130090 m	0.0117830 m	0.0107520 m	0.0098790 m
16	0.0245780 m	0.0211710 m	0.0184770 m	0.0163110 m	0.0145470 m	0.0130930 m	0.0118820 m	0.0108650 m	0.0100050 m
18	0.0246100 m	0.0212230 m	0.0185460 m	0.0163970 m	0.0146470 m	0.0132060 m	0.0120080 m	0.0110020 m	0.0101530 m
20	0.0246750 m	0.0213050 m	0.0186420 m	0.0165050 m	0.0147670 m	0.0133380 m	0.0121500 m	0.0111570 m	0.0103180 m
22	0.0247640 m	0.0214070 m	0.0187560 m	0.0166290 m	0.0149020 m	0.0134840 m	0.0123060 m	0.0113220 m	0.0104930 m
24	0.0248680 m	0.0215210 m	0.0188810 m	0.0167660 m	0.0150480 m	0.0136390 m	0.0124710 m	0.0114960 m	0.0106760 m
26	0.0249850 m	0.0216480 m	0.0190170 m	0.0169100 m	0.0152020 m	0.0138020 m	0.0126430 m	0.0116770 m	0.0108670 m
28	0.0251100 m	0.0217830 m	0.0191600 m	0.0170620 m	0.0153620 m	0.0139710 m	0.0128210 m	0.0118640 m	0.0000000 m
30	0.0252430 m	0.0219230 m	0.0193090 m	0.0172190 m	0.0155280 m	0.0141450 m	0.0130030 m	0.0000000 m	0.0000000 m
32	0.0253810 m	0.0220690 m	0.0194620 m	0.0173810 m	0.0156970 m	0.0143220 m	0.0000000 m	0.0000000 m	0.0000000 m
34	0.0255250 m	0.0222200 m	0.0196200 m	0.0175460 m	0.0158710 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m
36	0.0256720 m	0.0223740 m	0.0197820 m	0.0177150 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m
38	0.0258220 m	0.0225300 m	0.0199450 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m
40	0.0259750 m	0.0226900 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m
42	0.0261310 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m
44	0.0262890 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m	0.0000000 m

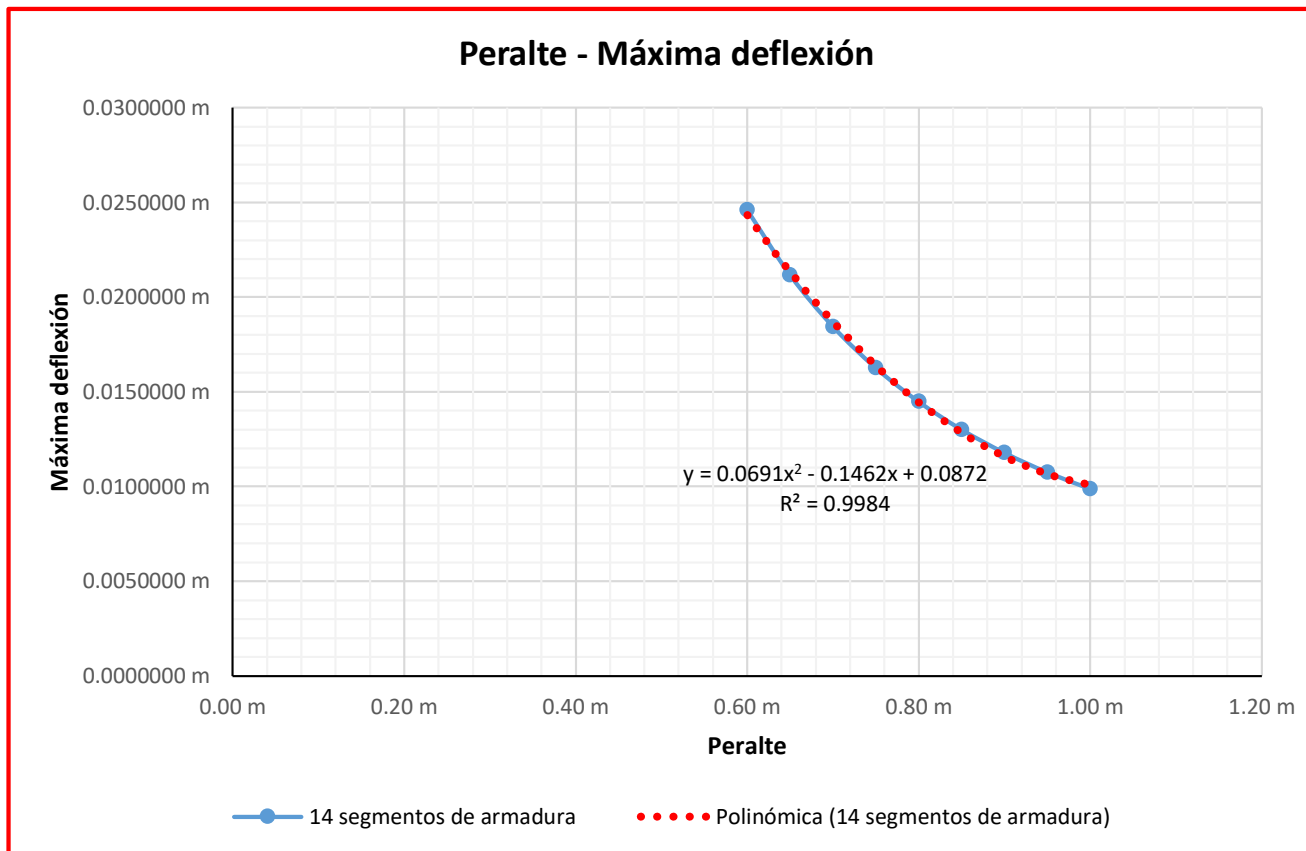


Figura 3.34. Peralte - Máxima deflexión - ejemplo N°1.

3.4.2.3.Optimización

Para el desarrollo de la optimización se ha realizado modificaciones al código gráfico definido para el análisis estructural, puesto que esta vez se tiene que diseñar los elementos de la armadura según la AISC 360-10. (con la nueva definición de código gráfico para la optimización también se puede realizar simulaciones estructurales, si no se tomó de manera directa la definición de código gráfico para la optimización en las simulaciones estructurales es porque es un código más complejo, entonces se trata de ir de códigos simples a los más complejos). Así se tiene el código gráfico del ejemplo en la siguiente figura:

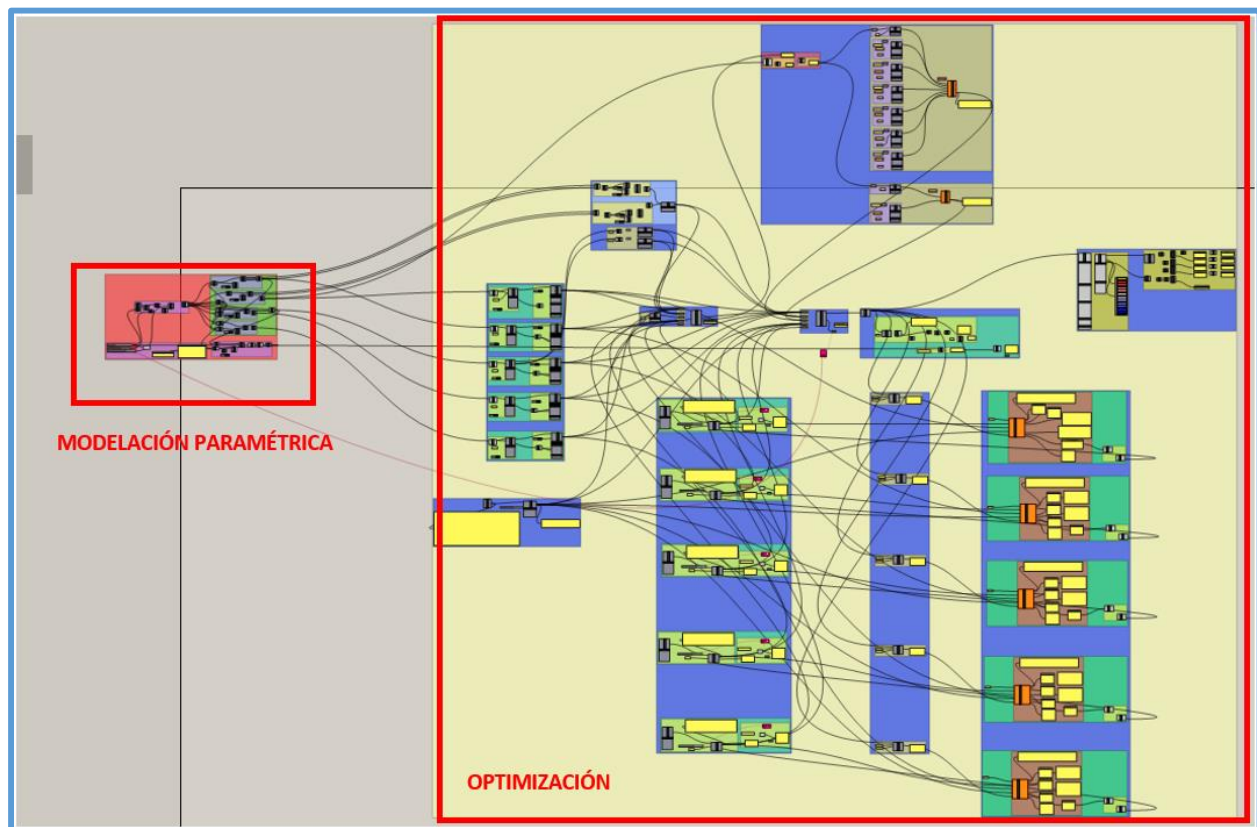


Figura 3.35. Código gráfico de modelación paramétrica y optimización - ejemplo N°1.

- Se han asignado cargas arbitrarias de 4 kN/m para la carga viva en azoteas (o carga viva de techo) y 4.5 kN/m para la carga de viento.
- La optimización geométrica se realizó con el complemento Octopus.
- La optimización de tamaño o de sección se realizó con el complemento Galápagos.
- Se ha colocado las secciones más pequeñas en los elementos o miembros de la armadura antes de realizar la optimización con Octopus, ya que este es el caso más desfavorable para las

secciones, así la geometría obtenida fue la más óptima posible puesto que trabaja con las secciones más desfavorables.

- En este ejemplo se trabajó con perfiles tubulares de secciones rectangulares, consecuentemente se usó el script “AISC - TENSION - COMPRESION R”.

➤ Optimización geométrica con Octopus

Una vez conectados los parámetros de la armadura (el peralte y el número de segmentos), y los objetivos a minimizar (el peso de la armadura y la deflexión máxima, se conecta la deflexión máxima brindada por el script “FILTRO1”, lo cual garantiza que Octopus adopte deflexiones máximas menores o iguales a la deflexión límite establecida en la N.T.E E 0.20), se da doble click izquierdo en el complemento Octopus, el cual mostrara la ventana que se muestra en la figura 3.36, luego se da click izquierdo en la opción “start”, así Octopus comenzara a realizar la optimización geométrica, para este caso se deja que Octopus optimice hasta la tercera generación y de las soluciones generadas por Octopus, se elige la que genere ratios de diseño próximos a “uno” en los elementos de la armadura, esto se realiza visualizando los ratios de diseño que brinda el componente “AISC – TENSION –COMPRESION R” a medida que se busca la solución más óptima de entre las soluciones generadas por Octopus. En la figura 3.37 se observa la solución

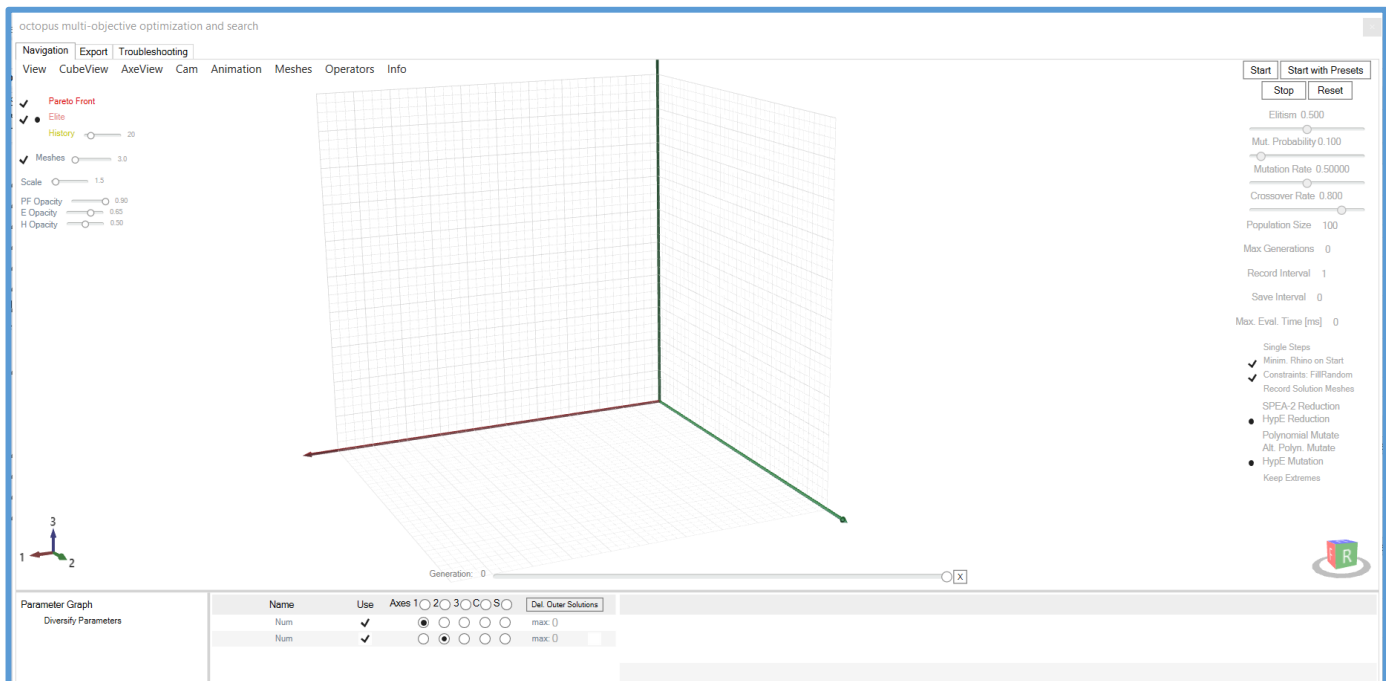


Figura 3.36. Ventana o entorno de “Octopus” - optimización geométrica - ejemplo N°1.

óptima (solución elegida dentro de las múltiples soluciones que nos brinda Octopus - geometría óptima para la armadura), la cual se encuentra dentro de una esfera amarilla en el entorno de Octopus y también se observa la solución óptima en el entorno de Rhino.

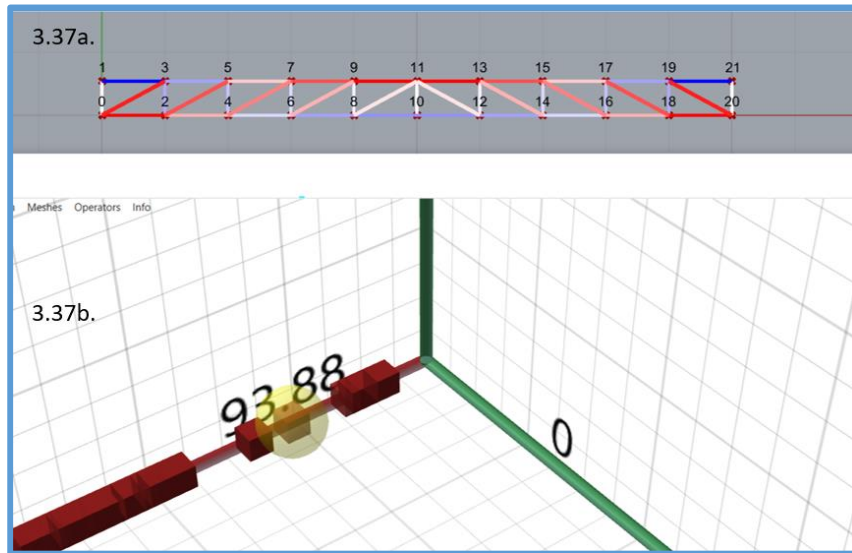


Figura 3.37. Solución óptima en los entornos de Rhino y “Octopus”, figuras 3.37a y 3.37b respectivamente - optimización geométrica - ejemplo N°1.

La solución óptima brinda un valor de deflexión menor que la deflexión límite establecida en la N.T.E E 0.20, como se ha trabajado con las secciones más pequeñas (caso más desfavorable) de la lista usada, entonces al modificar las secciones (asignar secciones de mayores dimensiones a los elementos), la deflexión siempre estará debajo de la deflexión límite, puesto que secciones de mayores dimensiones si bien proporcionan mayor peso, también proporcionan en mayor cantidad rigidez (el peso propio es una pequeña carga que produce deflexiones muy pequeñas). Concluida la optimización geométrica, los parámetros de la solución óptima se muestran la figura 3.38 y los valores de los objetivos se muestran en las figuras 3.39 y 3.40.

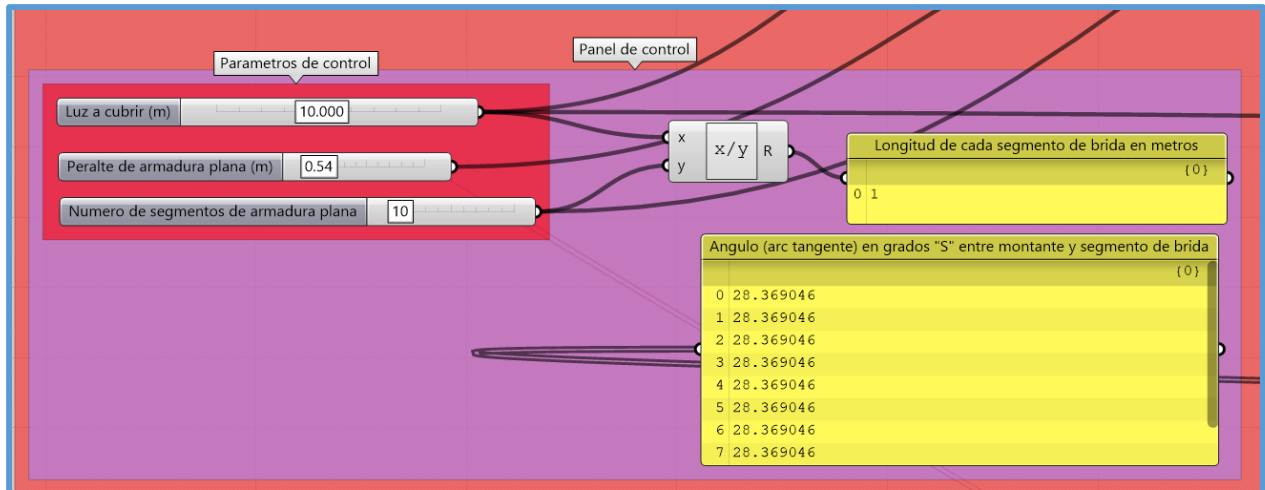


Figura 3.38. Parámetros de solución óptima - optimización geométrica - ejemplo N°1.

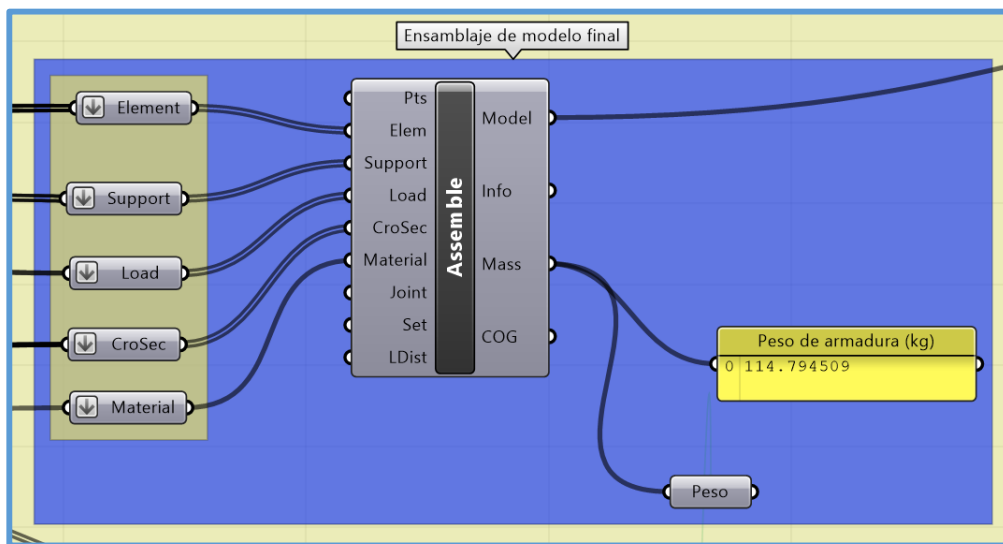


Figura 3.39. Peso de la armadura luego de la optimización geométrica - ejemplo N°1.

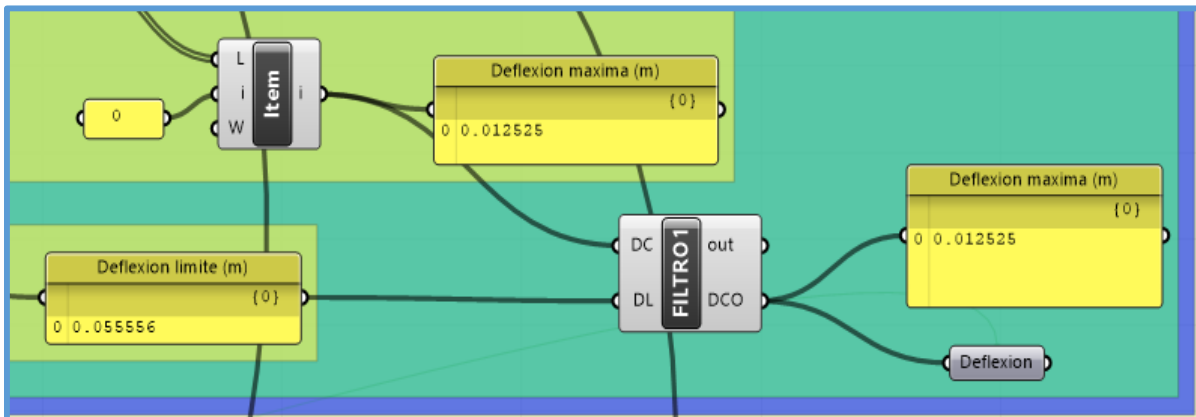


Figura 3.40. Deflexión máxima de la armadura luego de la optimización geométrica - ejemplo N°1.

➤ Optimización de tamaño con Galápagos

Una vez concluida la optimización geométrica con Octopus, se realiza la optimización de tamaño o de sección con Galápagos. Se realiza la optimización de tamaño para cada tipo de elementos (brida superior, brida inferior, montantes, diagonales izquierda y diagonales derecha) según los “Ids” (identificadores de cada tipo de elemento) establecidos en el modelo. Es así que Galápagos tiene como genome o parámetros las secciones transversales definidas en la lista de secciones y como objetivo que los ratios de diseño de los elementos sean “uno”, para lograr esto se usa la función objetivo ABS (x-1), donde x son los ratios de diseño de los elementos brindados por el script “FILTRO3” (se usa los ratios de diseño brindados por este script porque así se garantiza que Galápagos adopte ratios de diseño menores o iguales a uno), Galápagos minimiza el resultado de la función ABS(x-1), busca que sea cero, consecuentemente los ratios de diseño se aproximan a 1, el resultado de la función ABS(x-1) se conecta como el fitness u objetivo a minimizar de Galápagos; luego de realizadas las conexiones del genome y del fitness se da doble click en el complemento Galápagos, el cual mostrara la ventana que se encuentra en la figura 3.41.

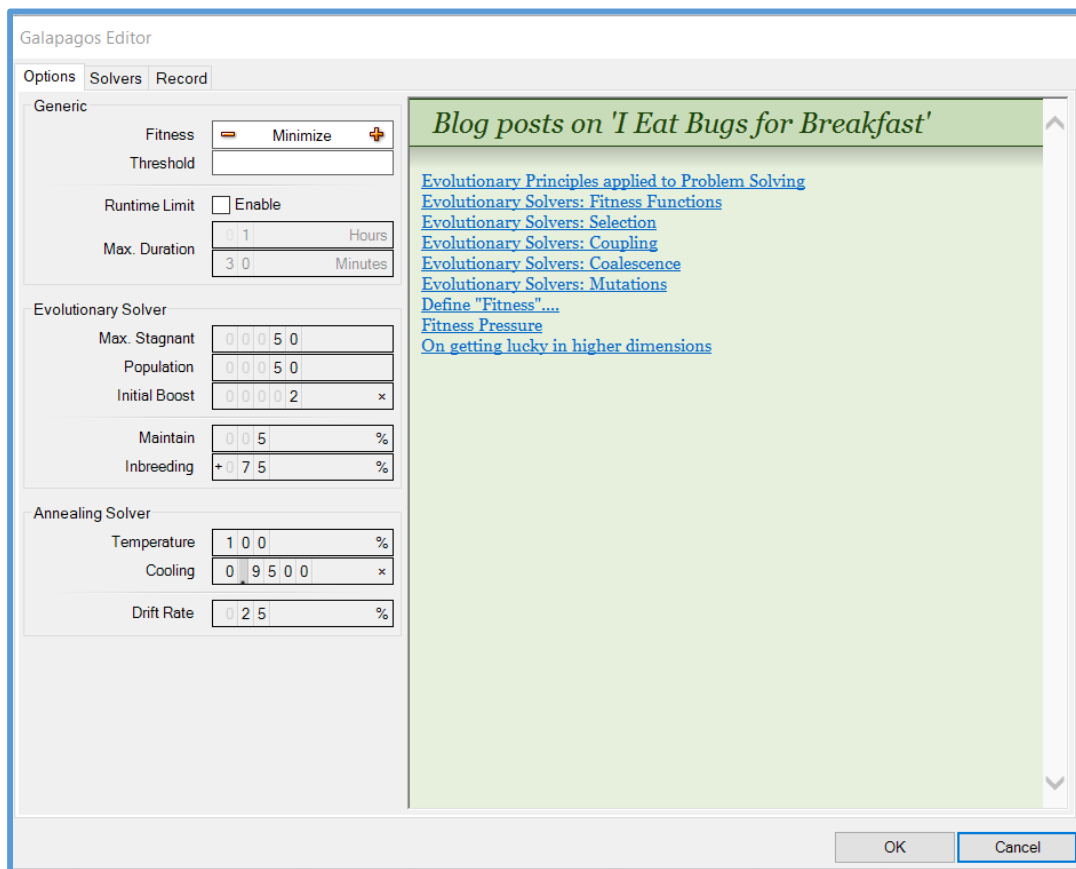


Figura 3.41. Ventana inicial de Galápagos - optimización de tamaño - ejemplo N°1.

Una vez en la ventana inicial de Galápagos (figura 3.41), en la opción “Fitness” se da click en el signo de menos, lo cual significa que se minimizará, para el resto de opciones de la ventana inicial se trabajara con las configuraciones por defecto, luego se cambia a la pestaña “Solvers” (figura 3.42), por defecto Galápagos usara el “Evolutionary solver” que es el solucionador con el cual se trabajara, entonces se da click en la opción “Start Solver” así Galápagos comenzara a realizar la optimización de tamaño. Cabe recalcar que la optimización de tamaño se realiza para cada tipo de elemento, según los “Ids” definidos en el modelo. En este caso se obtuvo la solución óptima por parte de Galápagos esperando a que se generen las 50 generaciones establecidas en la configuración por defecto (opción Max. Stagnant de la ventana inicial), pero si el valor objetivo se vuelve constante antes de llegar a las 50 generaciones, significa que se ha cumplido con el objetivo, entonces no es necesario esperar a que se generen las 50 generaciones, se puede parar ya el “Evolutionary solver” dando click en la opción “Stop Solver” y adoptar la solución que este obtuvo. En este ejemplo el “Evolutionary solver” obtuvo las soluciones sin necesidad de llegar a las 50 generaciones, se llegó a usar un máximo de 7 generaciones.

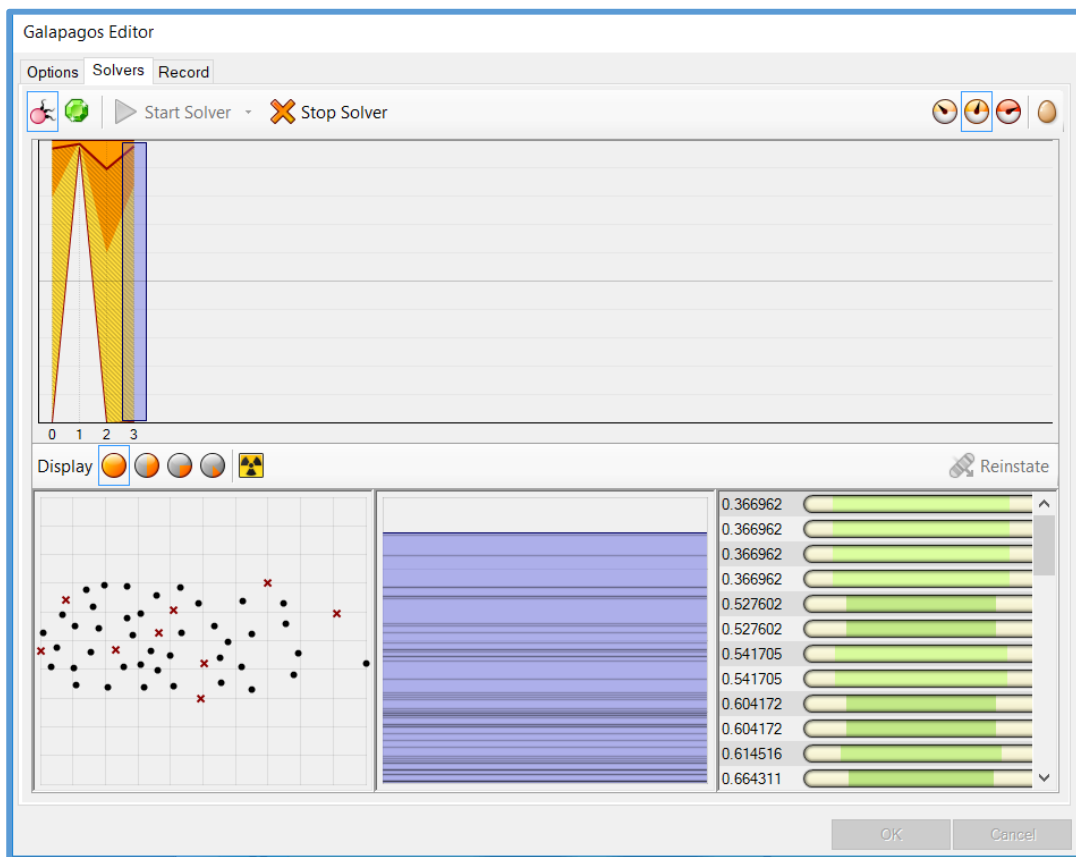


Figura 3.42. Optimización con el “Evolutionary Solver” de Galápagos - optimización de tamaño - ejemplo N°1.

Una vez realizada la optimización de tamaño para cada tipo de elemento se verifica que los ratios de diseño sean todos menores o iguales a 1, puesto que al realizar la optimización de tamaño de un tipo de elementos se modifican ligeramente los ratios de diseño de los elementos ya optimizados debido a la variación del peso propio, si es que hubiera algún ratio mayor a 1, entonces se hace una segunda iteración de la optimización de tamaño. En este ejemplo solo se realizó una iteración de la optimización de tamaño.

Producto de la optimización de tamaño se tienen los resultados que se muestran en las figuras 3.44, 3.45, 3.46, 3.47 y 3.48 (se tienen las secciones de cada tipo de elemento, la resistencia de diseño y el ratio de diseño) también se tienen nuevos valores de peso propio y deflexión máxima, los cuales se muestran en las figuras 3.49 y 3.50 respectivamente. Cabe indicar que en Karamba, al igual que en SAP2000, los elementos a compresión (con fuerzas actuantes negativas) son de color rojo y los elementos a tensión (con fuerzas actuantes positivas) son de color azul, en Karamba los elementos sin fuerzas actuantes son de color blanco. Los números identificadores de los elementos los asigna Karamba al momento de ensamblar el modelo y se visualizan con los componentes “ModelView” y “BeamView”, para cada tipo de elementos asignados con un “Id” el orden de los números identificadores es sucesivo, por ejemplo, las montantes tienen el id “mont”, entonces todos los elementos bajo ese id tendrán números identificadores con un orden sucesivo (del 0 al 10 para este ejemplo). Los scripts “AISC-TENSION-COMPRESIO C” y “AISC-TENSION-COMPRESIO R” trabajan con números identificadores parciales en los elementos, es decir cada elemento dentro de un grupo de elementos bajo un id tendrá un número identificador relativo al número de elementos de ese grupo, si en un grupo hay 5 elementos entonces los números que identifican a dichos elementos en los resultados de diseño van del 0 al 4; por lo tanto se cumple que el elemento con el número identificador (asignado por Karamba) más bajo en un grupo de elementos corresponde al número identificador 0 para los resultados obtenidos con los scripts “AISC-TENSION-COMPRESIO C” y “AISC-TENSION-COMPRESIO R” y estos van aumentando sucesivamente, por ejemplo en la figura 3.44 se tiene que los resultados de “Resistencia de diseño” y “Ratio de diseño” en el número identificador 0 (o índice 0) corresponden al elemento con el menor número identificador (asignado por Karamba) dentro del grupo de elementos bajo el id “brsup”, entonces corresponden al elemento con el número identificador 21, luego se continúan relacionando uno a uno en forma consecutiva, así al llegar a los resultados en el número identificador 9 (o índice 9)

sabemos que estos corresponden al elemento con el número identificador 30 (asignado por Karamba). Se muestran los resultados de diseño en las siguientes figuras:

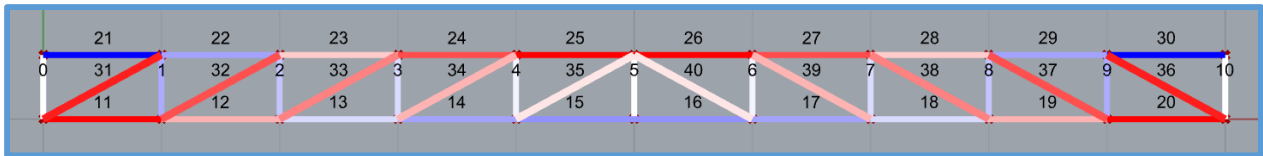


Figura 3.43. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización de tamaño - ejemplo N°1.

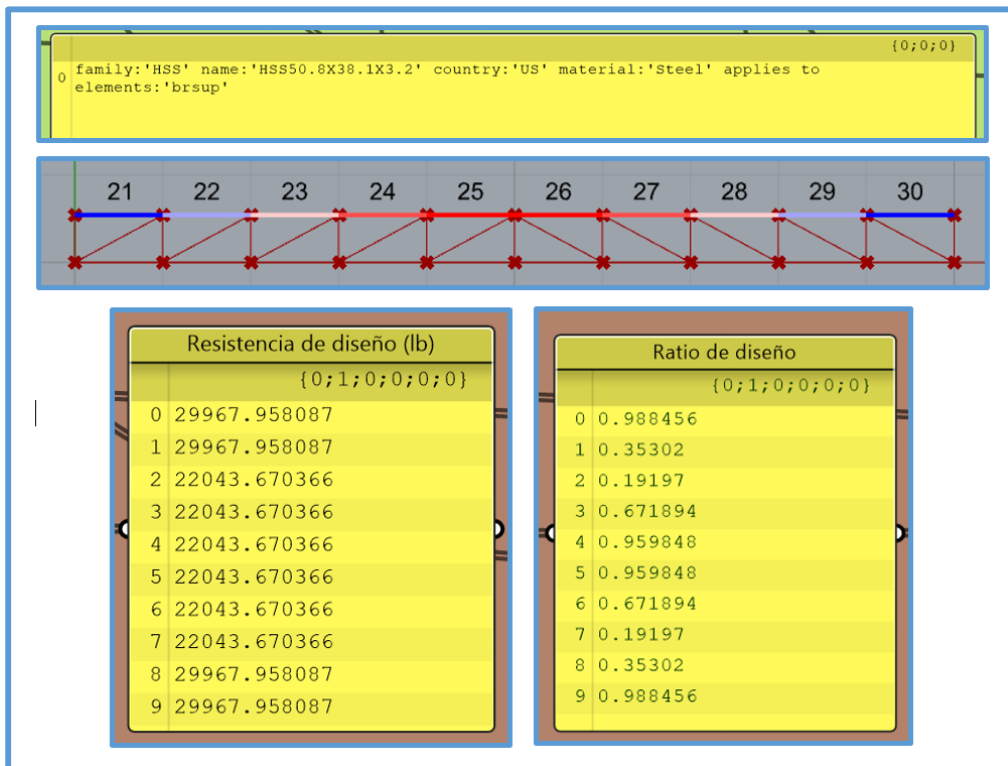


Figura 3.44. Resultados de diseño de la brida superior - optimización de tamaño - ejemplo N°1.

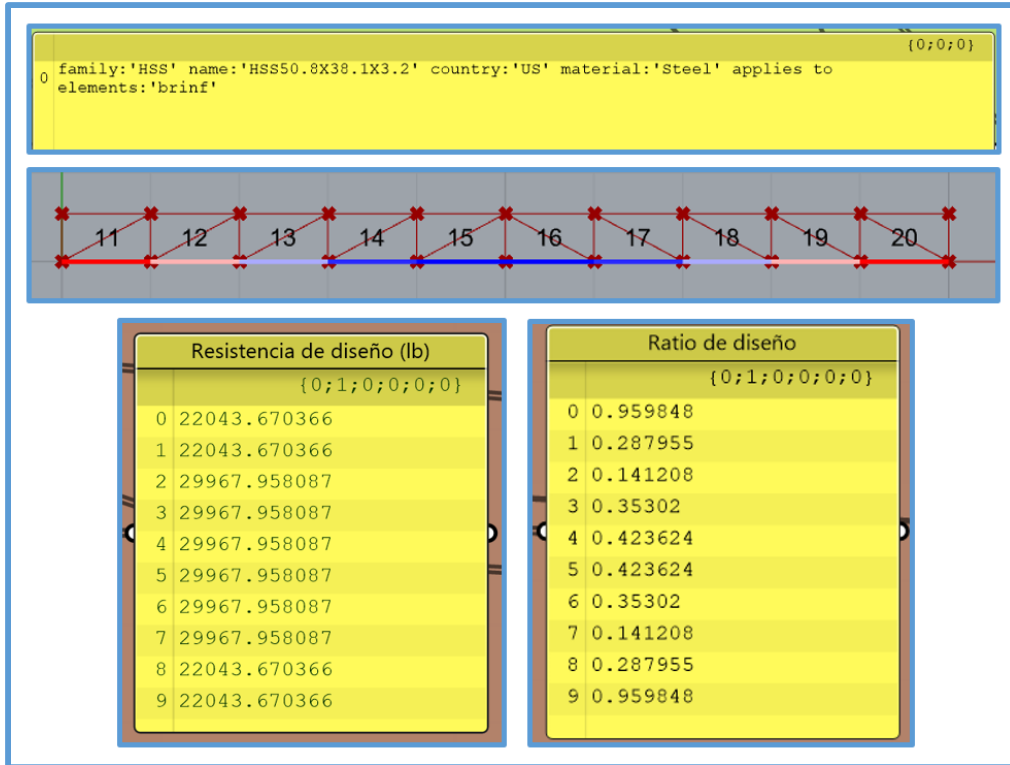


Figura 3.45. Resultados de diseño de la brida inferior - optimización de tamaño - ejemplo N°1.

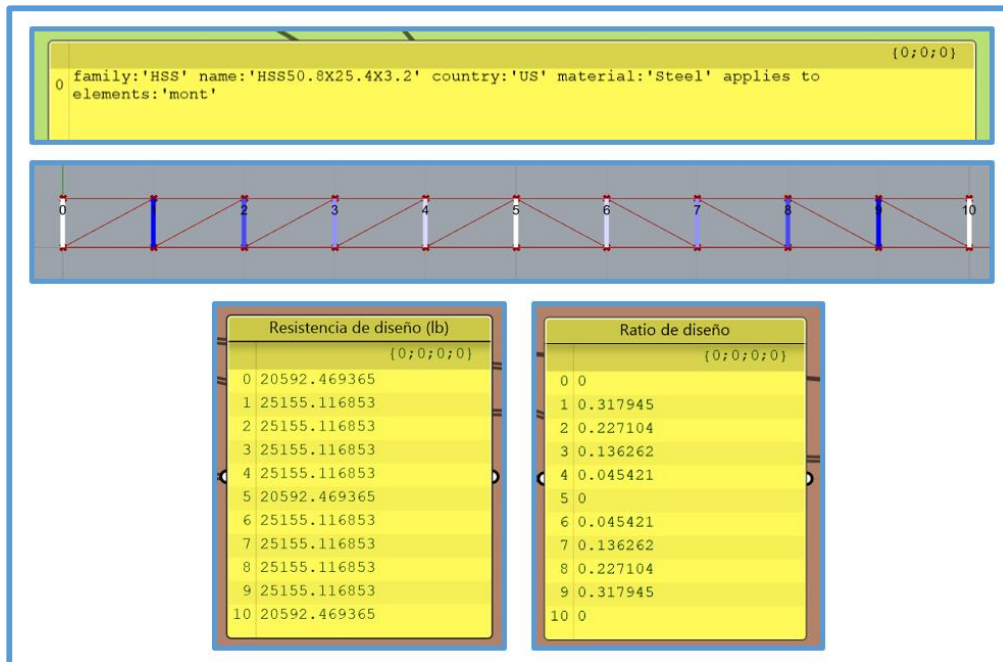


Figura 3.46. Resultados de diseño de las montantes - optimización de tamaño - ejemplo N°1.

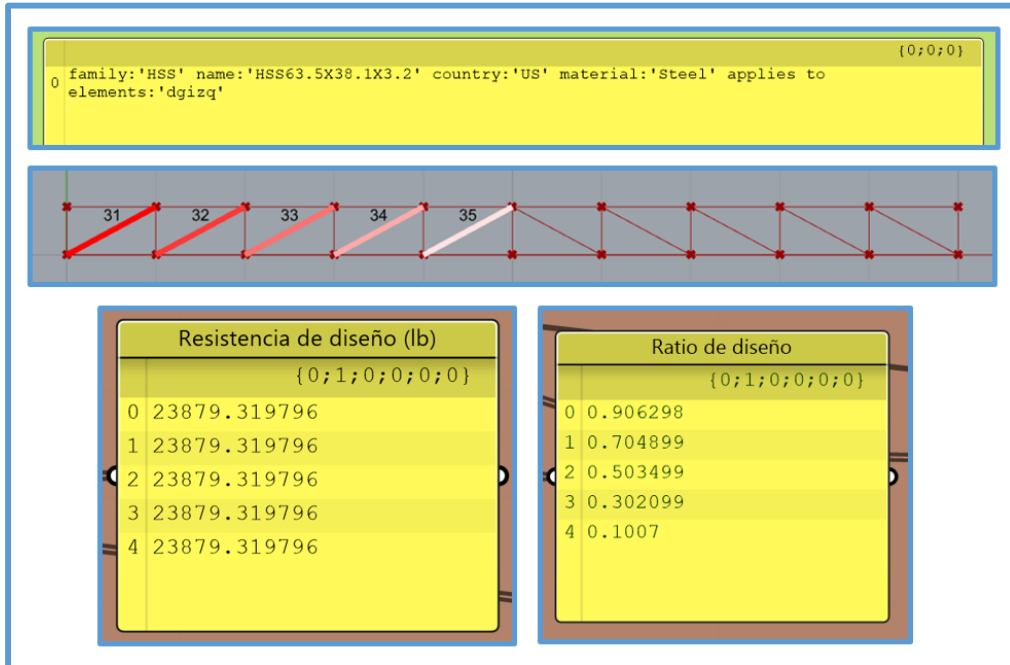


Figura 3.47. Resultados de diseño de las diagonales - izquierda - optimización de tamaño - ejemplo N°1.

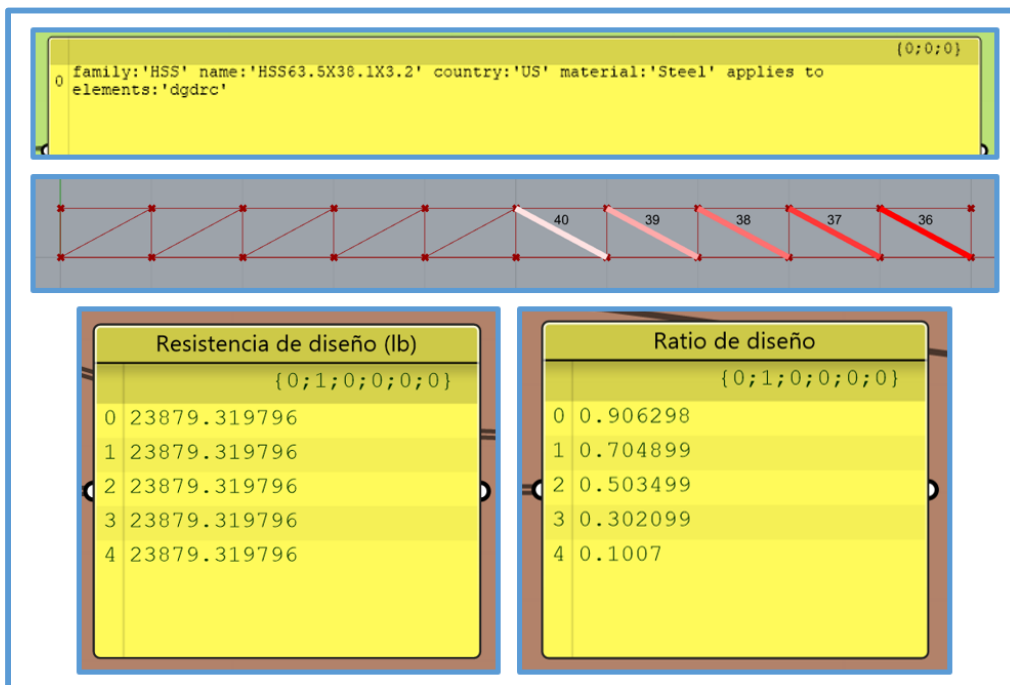


Figura 3.48. Resultados de diseño de las diagonales - derecha - optimización de tamaño - ejemplo N°1.

Nota: Las dimensiones de las secciones se encuentran en “mm”.

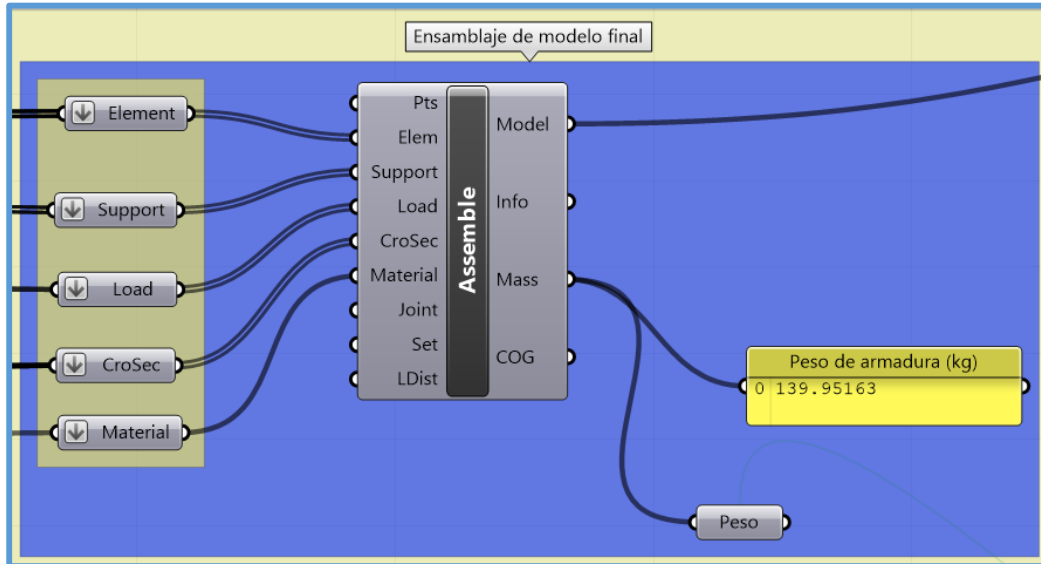


Figura 3.49. Peso de la armadura luego de la optimización de tamaño - ejemplo N°1.

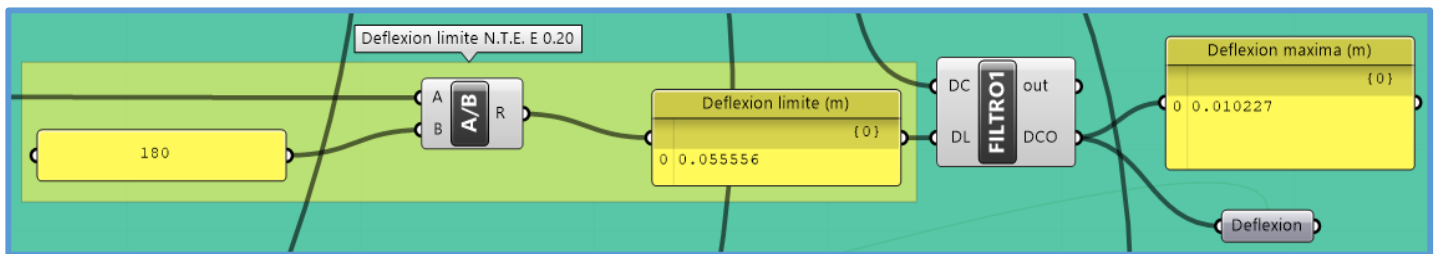


Figura 3.50. Deflexión máxima de la armadura luego de la optimización de tamaño - ejemplo N°1.

➤ Optimización topológica con el componente BESOBBeam

Concluida la optimización de tamaño se desarrolló la optimización “topológica” con el componente “BESOBBeam”, así tenemos la figura 3.51, que nos muestra las implementaciones hechas al código gráfico de la optimización (definido en la figura 3.35), necesarias para realizar la optimización “topológica”, para más detalles revisar los archivos en Grasshopper.

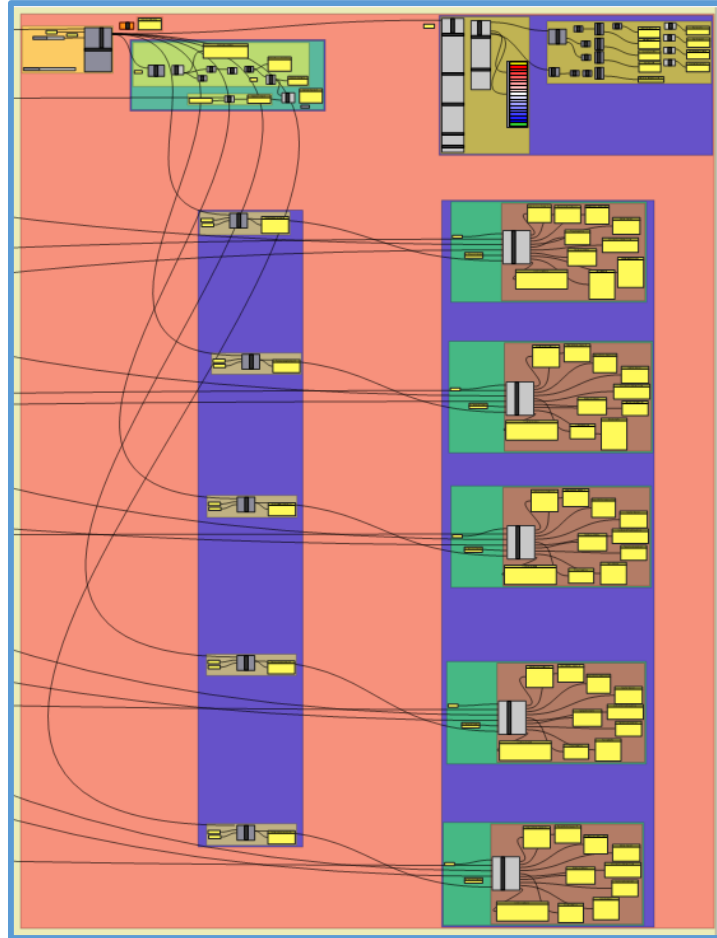


Figura 3.51. Implementaciones de código gráfico para realizar la optimización topológica - ejemplo N°1.

Con el componente “BESOBeam” para el caso de armaduras solo se pueden eliminar elementos que no afecten la longitud de un elemento, por ejemplo si eliminamos la montante central de este ejemplo entonces en la brida inferior se generaría un elemento de doble longitud (ya que todos los elementos de la brida inferior tienen la misma longitud) y para este caso “BESOBeam” no hace las actualizaciones respectivas, por eso solo se debe trabajar con elementos que no afecten la longitud de los elementos restantes, para este ejemplo se puede trabajar con las diagonales ya que al eliminarlas los elementos restantes mantienen su longitud (no se generan elementos de mayor longitud).

Para este ejemplo se puede apreciar (por simple inspección - sin necesidad de realizar la optimización topológica) que podemos eliminar las montantes de ambos extremos de la armadura puesto que no tienen ninguna fuerza actuante, se encuentran entre los apoyos, no afectan la

longitud del resto de elementos y la conexión a usar será plenamente soldada (la eliminación de estas montantes afecta muy levemente o nada los resultados de diseño obtenidos al final de la optimización de tamaño), en la figura 3.52 se muestran las implementaciones de código gráfico para eliminar las montantes de los extremos y en la figura 3.53 se muestra la armadura después de dichas implementaciones . Ahora quedara evaluar que diagonales se pueden eliminar mediante la optimización topológica, de tal manera que se siga cumpliendo con tener una deflexión máxima menor o igual a la deflexión limite y ratios de diseño de los elementos menores o iguales a 1.

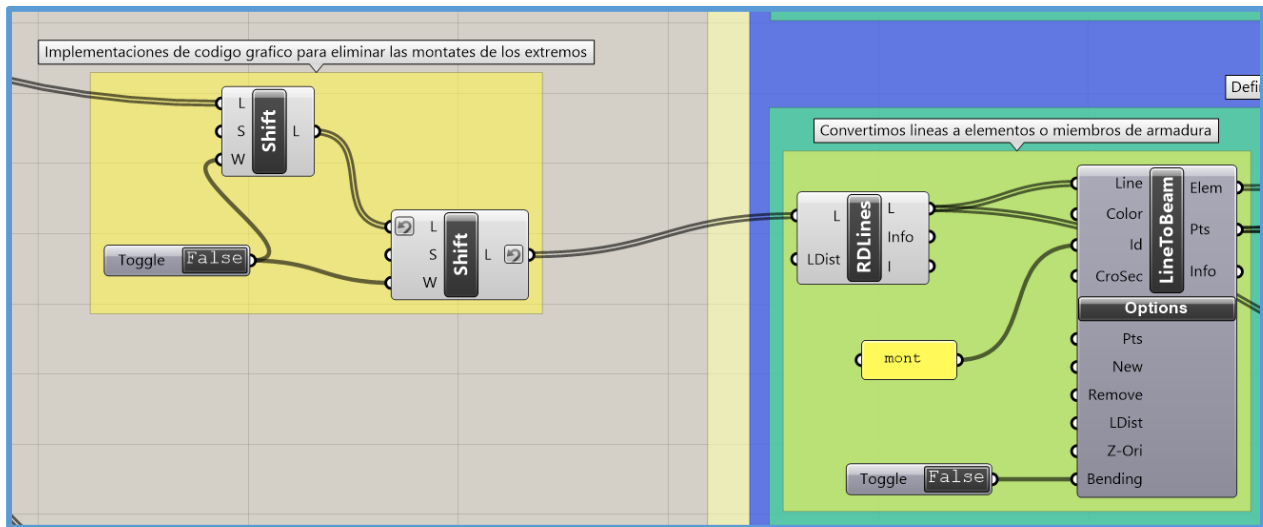


Figura 3.52. Implementaciones de código gráfico para eliminar las montantes de los extremos - ejemplo N°1.

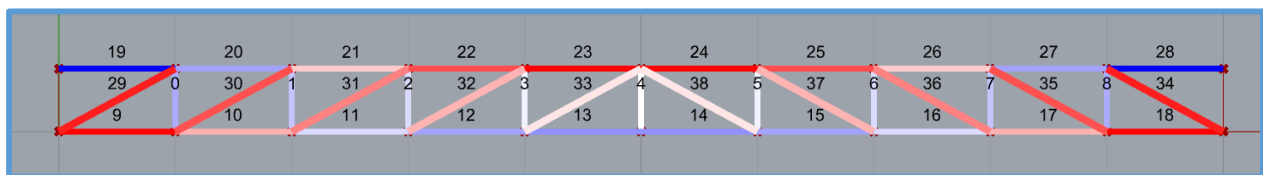


Figura 3.53. Armadura luego de eliminar las montantes de los extremos - ejemplo N°1.

En la figura 3.54 se observa el componente “BESOBeam”, al cual se le ha indicado que trabaje solo con las diagonales de la izquierda y para el caso de cargas 8, que es la combinación de cargas de mayor valor. Como estamos trabajando con estructuras discretas, entonces el “Target mass ratio” sirve de forma referencial, como se sabe el “Target mass ratio” sirve para asignar el peso de la estructura que se quiere conservar, pero al trabajar con armaduras esto se aplica hasta que hayamos completado cierto porcentaje de peso (el peso de uno o más elementos con respecto al total), se entiende esto con un ejemplo, se supone que se tiene 20 elementos de una armadura (con los mismos parámetros – longitud, sección y material) bajo un identificador “rel” entonces cada elemento representa un 5% de peso con respecto al peso total, si el “Target mass ratio” se encuentra en un intervalo de (<0.95 a 1.00], no se eliminara ningún elemento puesto que no se ha alcanzado un 5% que representa a un elemento, si se encuentra en un intervalo de (<0.90 a 0.95] solo se elimina un elemento puesto que no se ha alcanzado un porcentaje de 10% que representa 2 elementos y así sucesivamente.

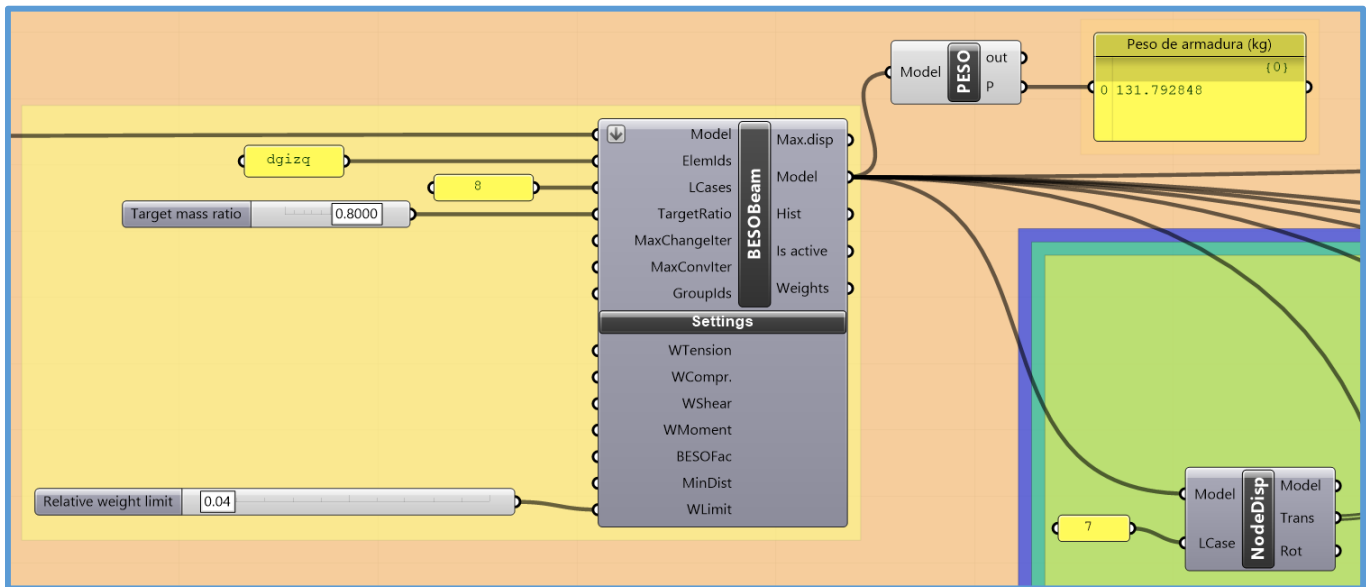


Figura 3.54. Componente “BESOBeam”, componente que realiza la optimización topológica - ejemplo N°1.

En la figura 3.54, también se aprecia la presencia del script “PESO”, el cual sirve para obtener el peso de la armadura (en kg) luego de la optimización topológica.

Si se trabaja con un valor de 0.80 en el “Target mass ratio” del componente “BESOBeam” (como se muestra en la figura 3.54), entonces se eliminará 1 diagonal, esto se observa en la figura 3.55.

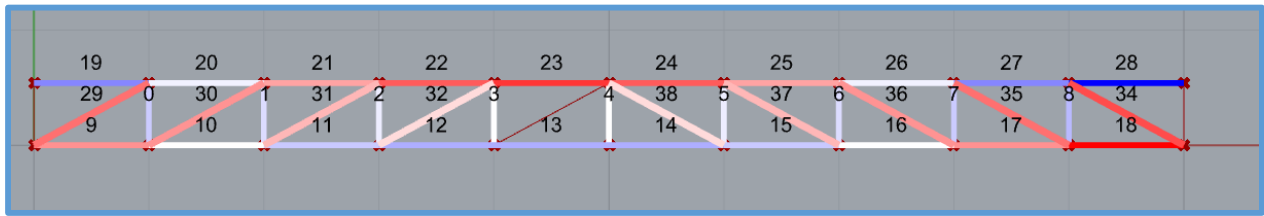


Figura 3.55. Armadura luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.

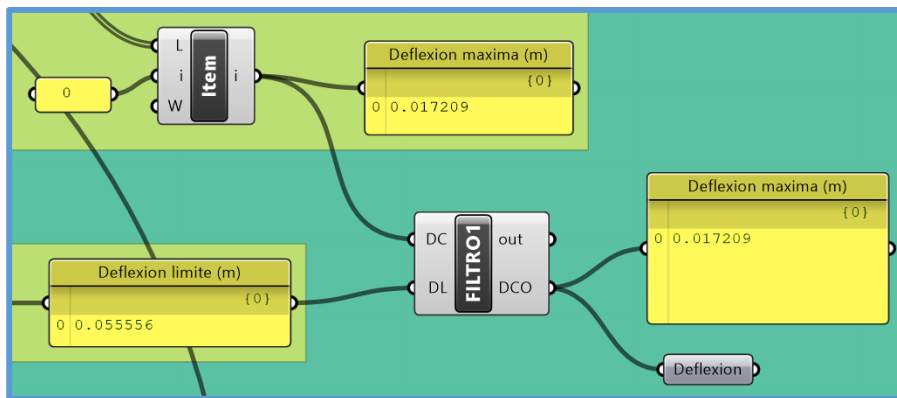


Figura 3.56. Deflexión máxima luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.

Al trabajar con 1 diagonal – izquierda eliminada, si bien la deflexión máxima sigue siendo aún menor a la deflexión limite (figura 3.56), no obstante se tiene ratios de diseño mayores a uno en algunos elementos, esto se verifica rápidamente al observar la leyenda del componente “BeamView”, la cual brinda los porcentajes de utilización de los elementos (los cuales se calculan al dividir la fuerza actuante entre el producto del área de la sección por el módulo de elasticidad), cuando los ratios de diseño son menores o iguales a uno se tienen porcentajes de utilización en un intervalo del (70%-85%), aproximadamente, pero en este caso dichos porcentajes son mayores al 100%, lo cual se muestra en la figura 3.57; también se puede realizar la verificación al observar directamente los ratios de diseño de los elementos brindados por el componente “AISC-TENSION-COMPRESION R”, es así que para los elementos de la brida superior se tienen algunos ratios de diseño mayores a uno (figura 3.58).

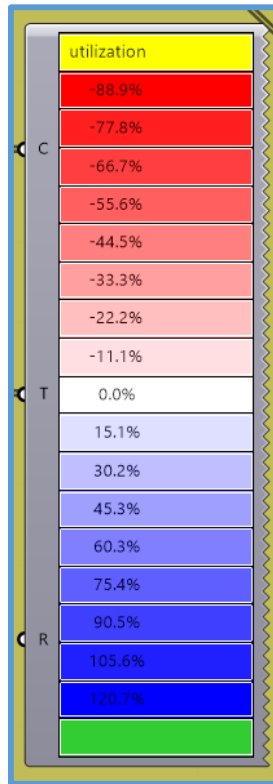


Figura 3.57. Porcentajes de utilización de los elementos luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.

Ratio de diseño	
{0;1;0;0;0;0}	
0	0.635192
1	0.070577
2	0.479739
3	0.863531
4	1.055427
5	0.863531
6	0.47974
7	0.070577
8	0.635191
9	1.340959

Figura 3.58. Ratios de diseño de los elementos de la brida superior luego de una optimización topológica con un “Target mass ratio” de 0.8 - ejemplo N°1.

Luego de eliminar la diagonal-izquierda anterior se nota que no se puede eliminar ninguna diagonal de la armadura (si eliminamos una diagonal-derecha se tendrá los mismos resultados que al eliminar una diagonal-izquierda, puesto que se trata de una armadura simétrica en todos sus parámetros de diseño), puesto que se produciría ratios de diseño de los elementos mayores a 1, entonces para este ejemplo no aplica la optimización topológica.

En conclusión, la armadura óptima final (solución óptima final) es la que se muestra en la figura 3.53 (luego de eliminar las montantes de los extremos), en la cual los resultados de diseño luego de aplicada la optimización de tamaño han tenido una muy leve variación y también ha surgido una variación en los números identificadores de los elementos luego de eliminar las montantes de los extremos, es así que se muestran los resultados de diseño en las figuras siguientes:

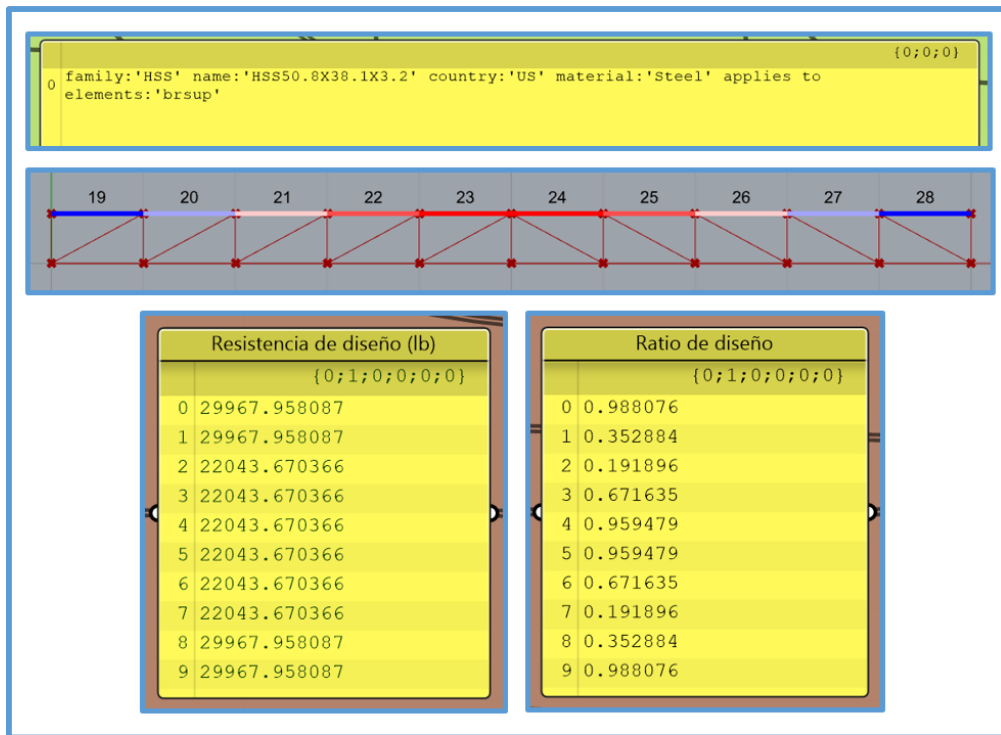


Figura 3.59. Resultados de diseño de la brida superior - armadura óptima final - ejemplo N°1.

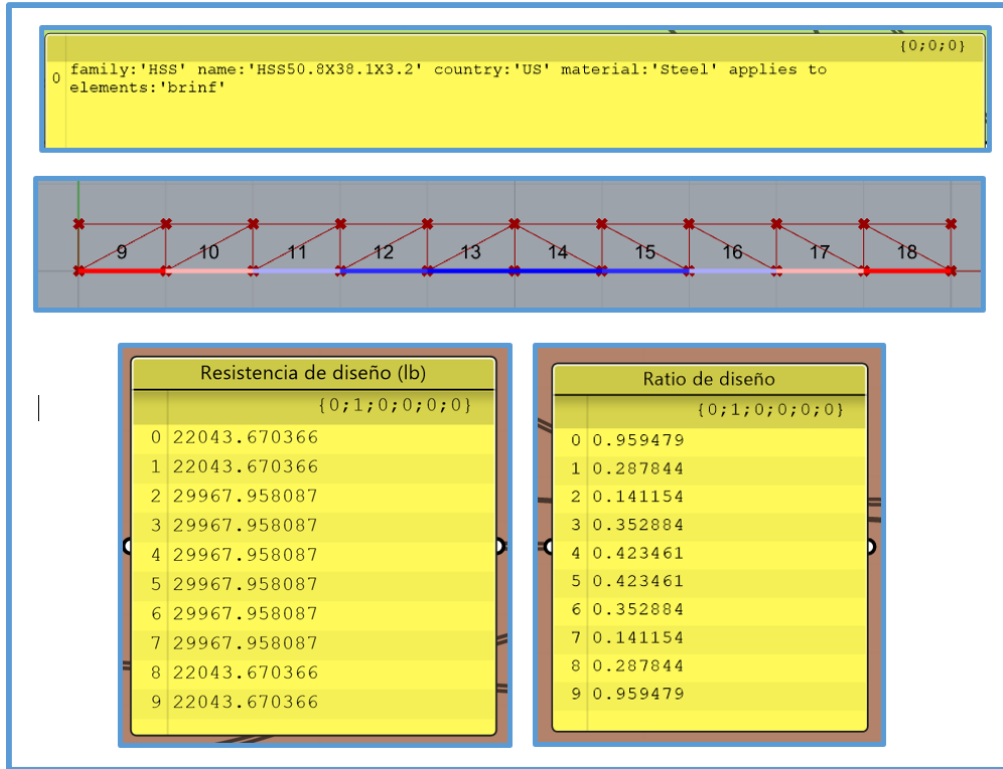


Figura 3.60. Resultados de diseño de la brida inferior - armadura óptima final - ejemplo N°1.

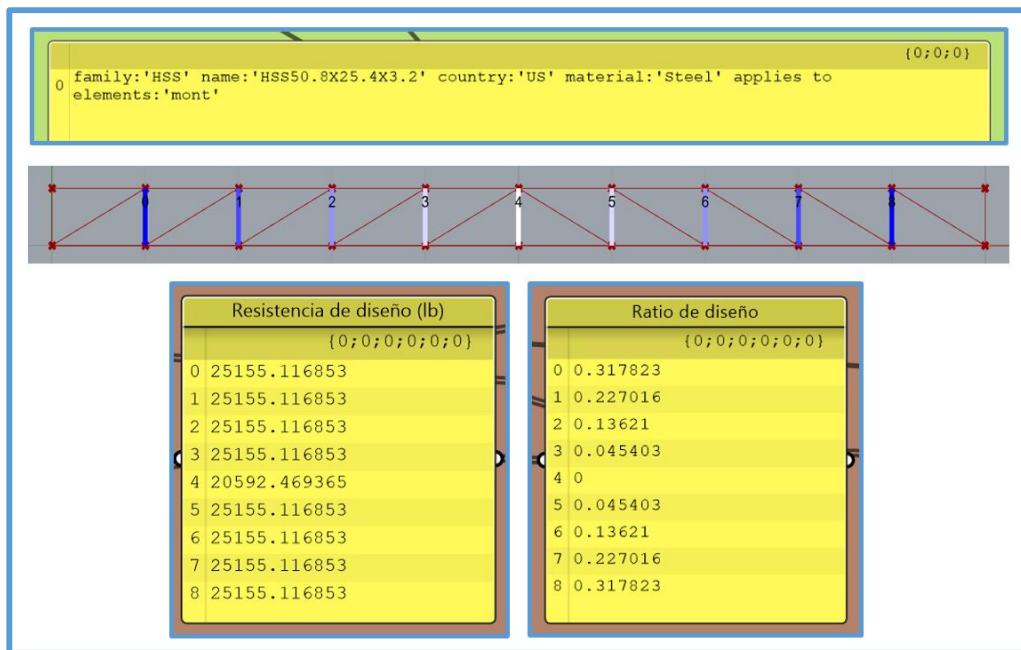


Figura 3.61. Resultados de diseño de las montantes - armadura óptima final - ejemplo N°1.

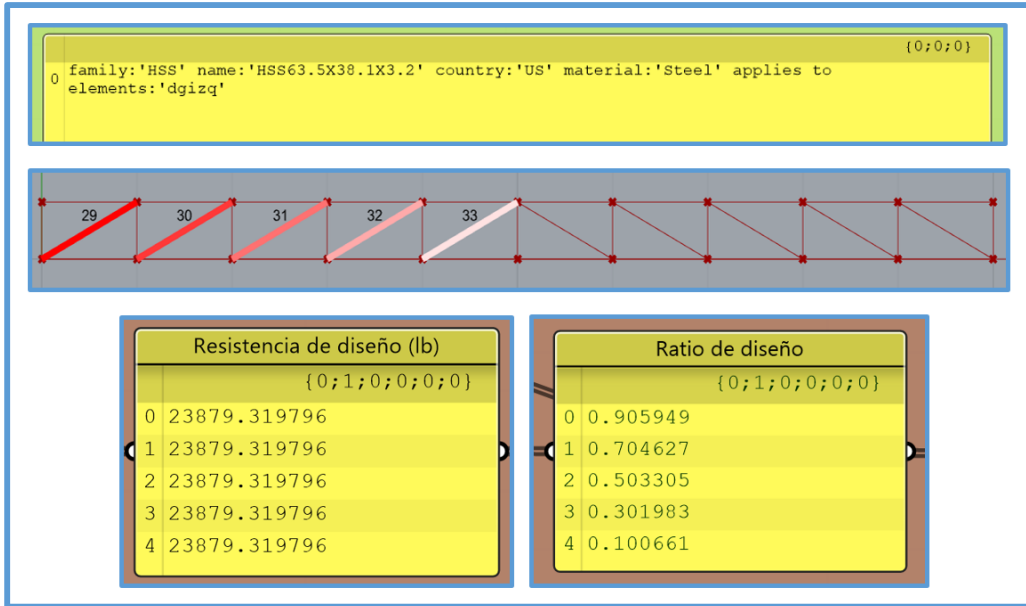


Figura 3.62. Resultados de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°1.

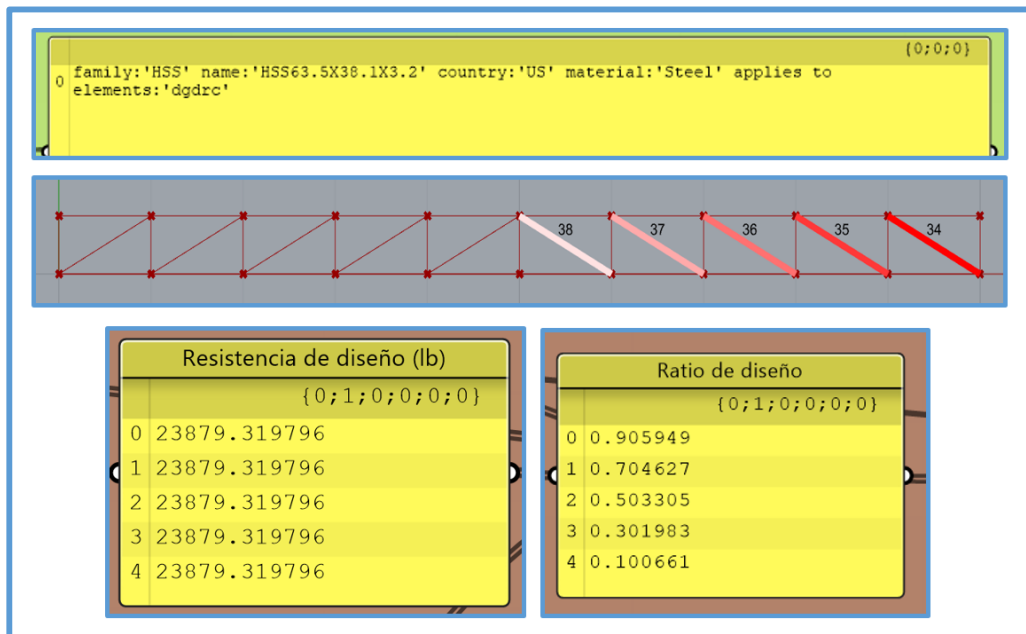


Figura 3.63. Resultados de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°1.

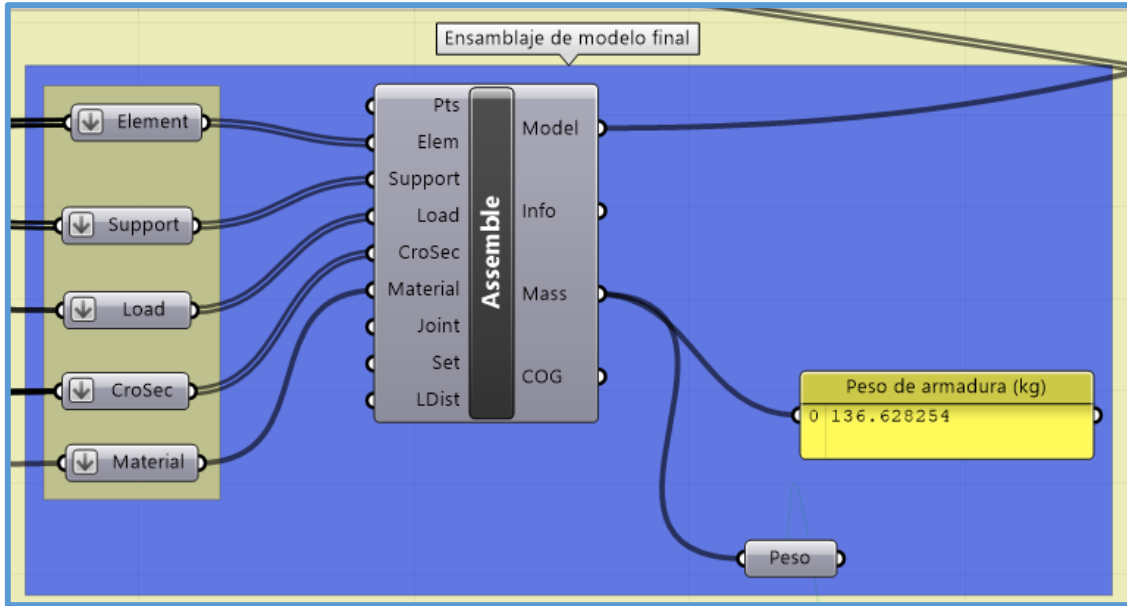


Figura 3.64. Peso de armadura óptima final - ejemplo N°1.

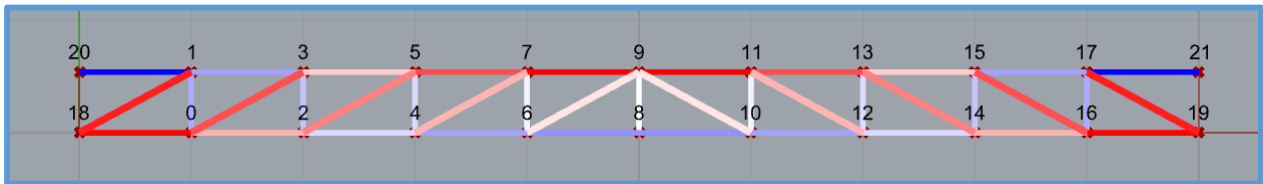


Figura 3.65. Nodos de armadura óptima final - ejemplo N°1.

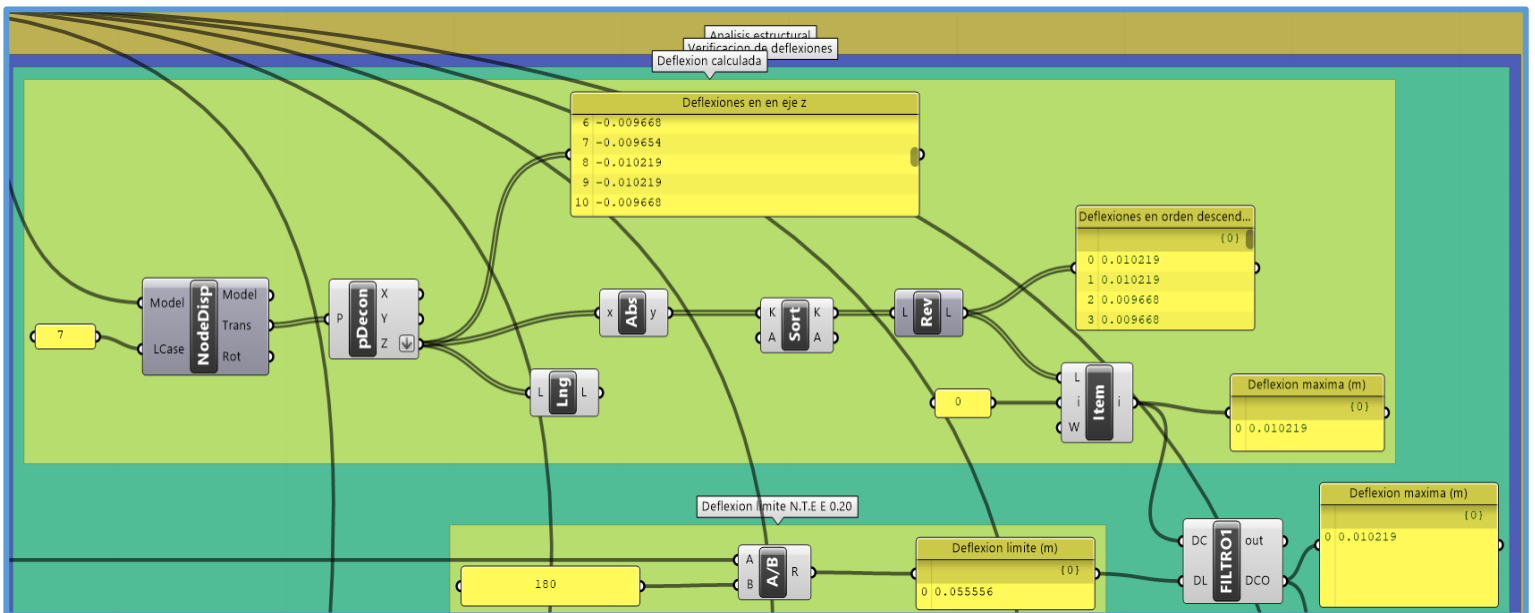


Figura 3.66. Deflexión máxima de armadura óptima final - ejemplo N°1.

Para el caso de la deflexión máxima de la armadura óptima se tiene dos deflexiones máximas (mismo valor), que corresponden a los dos extremos de la montante central, lo cual se puede verificar al observar los nodos 8 y 9 de la armadura óptima (figura 3.65) y los índices 8 y 9 del panel “Deflexiones en el eje Z” (figura 3.66), para realizar la optimización geométrica y la verificación de deflexiones, nuestro código gráfico está programado para seleccionar siempre la deflexión máxima, solo un valor de deflexión máxima, es así que para deflexiones máximas repetidas se seleccionara la que se encuentra en el nodo con menor número identificador, consecuentemente se tiene una deflexión máxima obtenida con el script “FILTRO1”(deflexión máxima con la cual se realiza la optimización geométrica y la verificación de deflexiones).

Para probar la veracidad de los resultados de diseño obtenidos con el proceso de diseño estructural propuesto en esta investigación se diseño la armadura óptima final (figura 3.53) en el software SAP2000.

➤ **Diseño de armadura óptima con “SAP2000”**

Se diseño la armadura óptima final en el software SAP2000 para una posterior comparación de resultados. Para el diseño de la armadura en SAP2000, se ha tenido en cuenta el mismo material, los mismos apoyos, mismas secciones, mismas cargas y misma geometría (la cual se exporto de Rhino a un archivo dwg, luego el archivo dwg se guardó como dxf y finalmente el archivo dxf se exporto a SAP2000). En las figuras 3.67, 3.68, 3.69 y 3.70 se muestran los resultados del diseño en SAP2000 de la armadura óptima final diseñada en el ejemplo N°1.

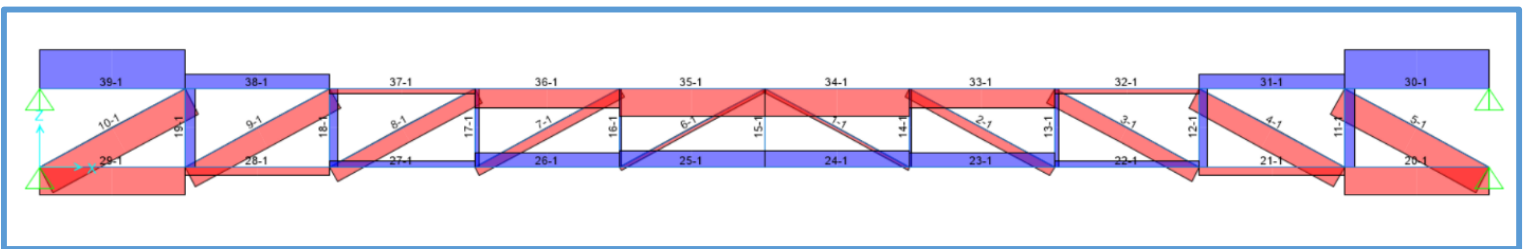


Figura 3.67. Elementos a tensión y/o compresión, con su respectivo número identificador en “SAP2000” - armadura óptima final - ejemplo N°1.

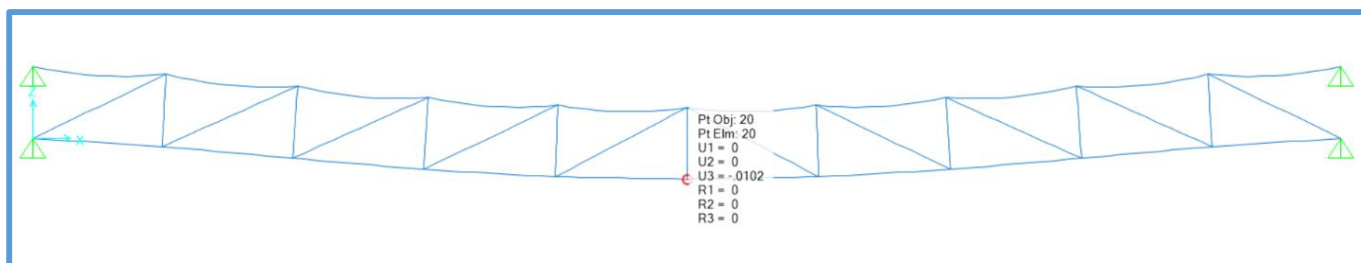


Figura 3.68. Deflexión máxima de armadura (en el extremo inferior de la montante central) obtenida con “SAP2000” - armadura óptima final - ejemplo N°1.

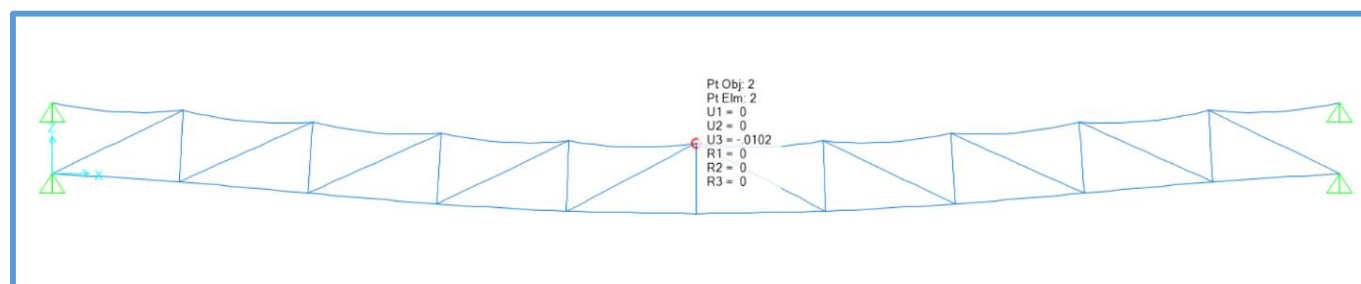


Figura 3.69. Deflexión máxima de armadura (en el extremo superior de la montante central) obtenida con “SAP2000” - armadura óptima final - ejemplo N°1.

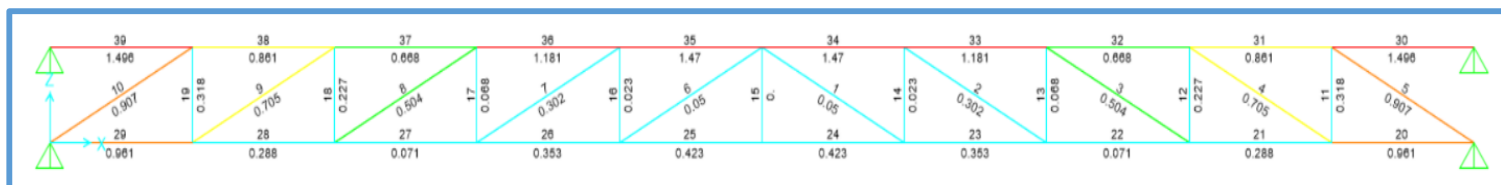


Figura 3.70. Ratios de diseño de los elementos de la armadura óptima final obtenidos con “SAP2000”- ejemplo N°1.

Como se trabajó con cargas distribuidas, SAP2000, para el caso de la brida superior también calculo flexión a pesar que con la opción “Assign Frame Releases and Partial Fixity” se ha indicado que los momentos en las direcciones 22 y 33 son cero, es por eso que los ratios mostrados en la brida superior son muy elevados, por lo tanto solo se considero los ratios generados por fuerzas axiales. A continuación, se muestra la tabla 3.20 en la cual se dan a conocer los resultados de diseño de la armadura óptima final, obtenidos con SAP2000.

Tabla 3.20. Resultados de diseño de armadura óptima final, obtenidos con “SAP2000”- ejemplo N°1.

Resultados de diseño de armadura óptima final con SAP2000					
Frame	DesignSect	Combo	PRatio	PcComp	PcTension
Text	Text	Text	Unitless	Lb	Lb
1	HSS2-1/2X1-1/2X1/8	COMB1	0.050389	23851.66	34776
2	HSS2-1/2X1-1/2X1/8	COMB1	0.302333	23851.66	34776
3	HSS2-1/2X1-1/2X1/8	COMB1	0.503889	23851.66	34776
4	HSS2-1/2X1-1/2X1/8	COMB1	0.705445	23851.66	34776
5	HSS2-1/2X1-1/2X1/8	COMB1	0.907	23851.66	34776
6	HSS2-1/2X1-1/2X1/8	COMB1	0.050389	23851.66	34776
7	HSS2-1/2X1-1/2X1/8	COMB1	0.302333	23851.66	34776
8	HSS2-1/2X1-1/2X1/8	COMB1	0.503889	23851.66	34776
9	HSS2-1/2X1-1/2X1/8	COMB1	0.705445	23851.66	34776
10	HSS2-1/2X1-1/2X1/8	COMB1	0.907	23851.66	34776
11	HSS2X1X1/8	COMB1	0.31762	20598.27	25171.2
12	HSS2X1X1/8	COMB1	0.226871	20598.27	25171.2
13	HSS2X1X1/8	COMB1	0.068061	20598.27	25171.2
14	HSS2X1X1/8	COMB1	0.022687	20598.27	25171.2
15	HSS2X1X1/8	COMB1	0	20598.27	25171.2
16	HSS2X1X1/8	COMB1	0.022687	20598.27	25171.2
17	HSS2X1X1/8	COMB1	0.068061	20598.27	25171.2
18	HSS2X1X1/8	COMB1	0.226871	20598.27	25171.2
19	HSS2X1X1/8	COMB1	0.31762	20598.27	25171.2
20	HSS2X1-1/2X1/8	COMB1	0.961483	21997.73	29973.6
21	HSS2X1-1/2X1/8	COMB1	0.288445	21997.73	29973.6
22	HSS2X1-1/2X1/8	COMB1	0.070564	21997.73	29973.6
23	HSS2X1-1/2X1/8	COMB1	0.352818	21997.73	29973.6
24	HSS2X1-1/2X1/8	COMB1	0.423381	21997.73	29973.6
25	HSS2X1-1/2X1/8	COMB1	0.423381	21997.73	29973.6
26	HSS2X1-1/2X1/8	COMB1	0.352818	21997.73	29973.6
27	HSS2X1-1/2X1/8	COMB1	0.070564	21997.73	29973.6
28	HSS2X1-1/2X1/8	COMB1	0.288445	21997.73	29973.6
29	HSS2X1-1/2X1/8	COMB1	0.961483	21997.73	29973.6
30	HSS2X1-1/2X1/8	COMB1	0.98789	21997.73	29973.6
31	HSS2X1-1/2X1/8	COMB1	0.352818	21997.73	29973.6
32	HSS2X1-1/2X1/8	COMB1	0.096148	21997.73	29973.6
33	HSS2X1-1/2X1/8	COMB1	0.673038	21997.73	29973.6
34	HSS2X1-1/2X1/8	COMB1	0.961483	21997.73	29973.6
35	HSS2X1-1/2X1/8	COMB1	0.961483	21997.73	29973.6
36	HSS2X1-1/2X1/8	COMB1	0.673038	21997.73	29973.6
37	HSS2X1-1/2X1/8	COMB1	0.096148	21997.73	29973.6
38	HSS2X1-1/2X1/8	COMB1	0.352818	21997.73	29973.6
39	HSS2X1-1/2X1/8	COMB1	0.98789	21997.73	29973.6

3.4.3. Ejemplo N°2

3.4.3.1. Modelación paramétrica

➤ Se han definido los siguientes parámetros para la armadura plana tipo Pratt.

- Luz a cubrir. Tiene un rango de 1m a 30m.
- Flecha superior de armadura plana. Tiene un rango de 0m a 3m.
- Peralte de armadura plana. Tiene un rango de 0.1m a 5m.
- Flecha inferior de armadura plana. Tiene un rango de -3m a 3m.
- Número de segmentos de armadura plana. Tiene un rango de 2 a 80.

Se ha trabajado con una luz a cubrir de 15m.

➤ Se puede apreciar el código gráfico de la modelación paramétrica del ejemplo N°2 en la siguiente figura:

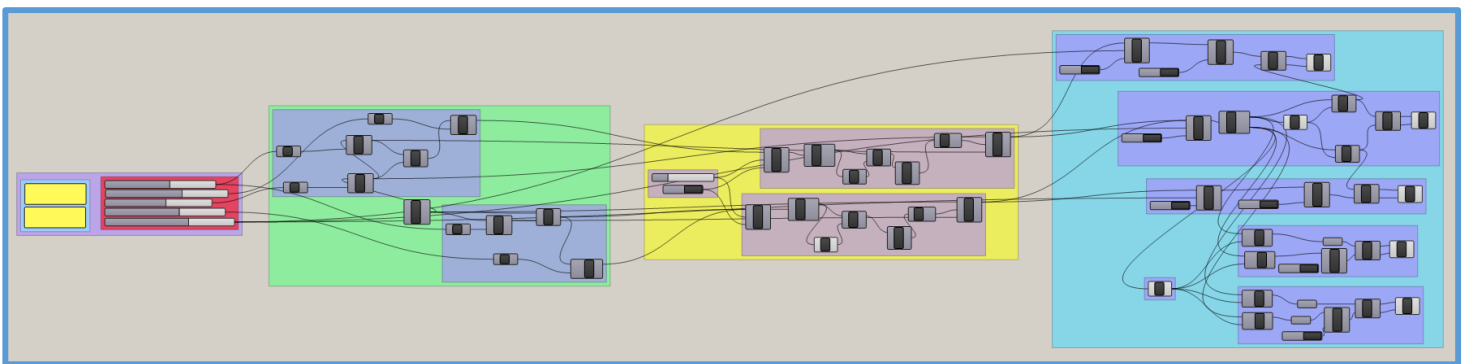


Figura 3.71. Código gráfico de la modelación paramétrica del ejemplo N°2.

3.4.3.2. Optimización

Para el desarrollo de la optimización se ha implementado el código gráfico de la modelación paramétrica, así tenemos el código gráfico del ejemplo en la siguiente figura:

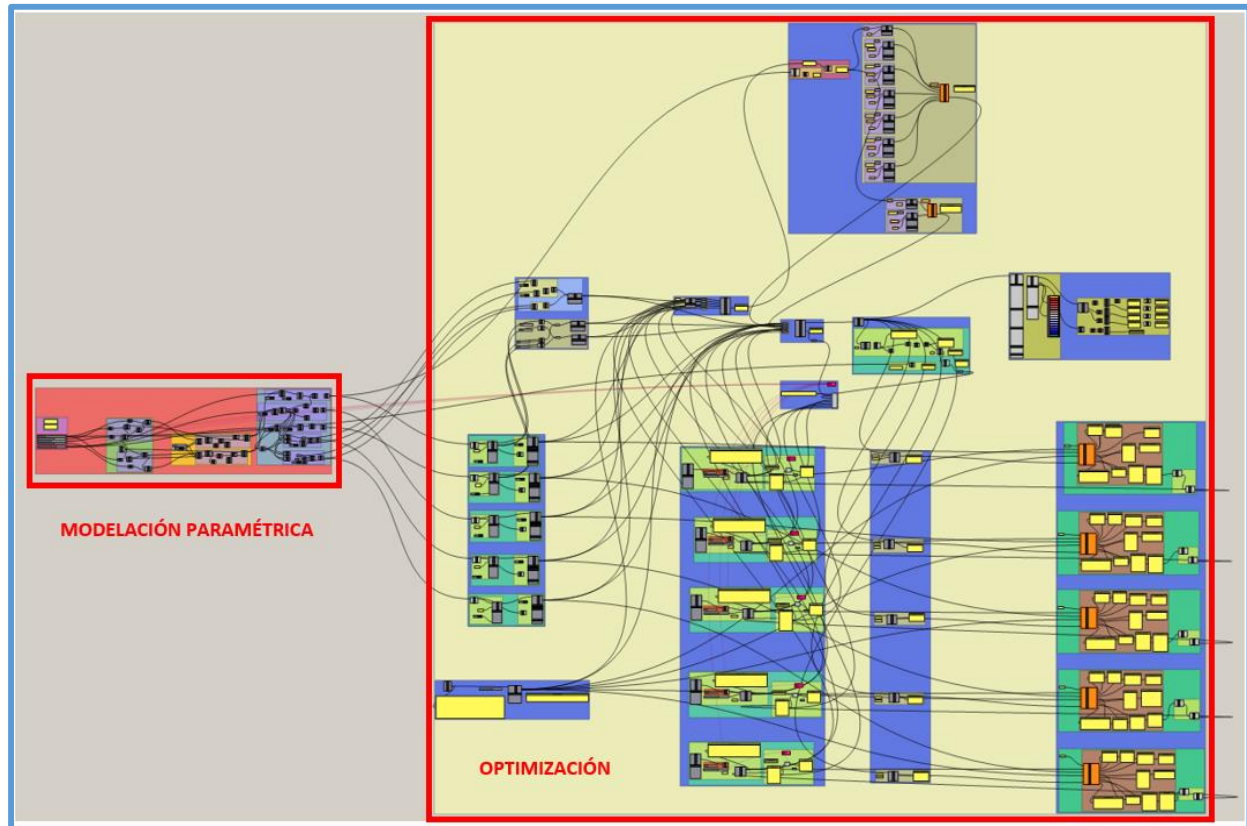


Figura 3.72. Código gráfico de modelación paramétrica y optimización - ejemplo N°2.

- Se han asignado cargas arbitrarias de 4 kN/m para la carga viva en azoteas (o carga viva de techo) y 4.5 kN/m para la carga de viento.
- Este ejemplo se realizó de dos formas, en la primera solo se realizó la optimización de tamaño (optimización similar a la usada en el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”); en la segunda se realizó la optimización geométrica y de tamaño en simultaneo, en ambas formas se utilizó el complemento Galápagos.
- Cada una de las formas en que se realizó este ejemplo cuenta con su respectivo archivo en Grasshopper, por lo cual se recomienda revisar dichos archivos para una mejor comprensión.
- En este ejemplo se trabajó con perfiles tubulares de secciones circulares, consecuentemente se usó el script “AISC - TENSION - COMPRESION C”.

➤ Optimización de tamaño con Galápagos

El objetivo de realizar este tipo de optimización fue encontrar una solución aproximada a la que se encontraría al usar el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, entonces antes de realizar la optimización de tamaño se debe definir la geometría de la armadura, para esto se usó los criterios de predimensionamiento de armaduras definidos en el anexo 1, es así que se adoptó los siguientes:

$$\text{Peralte} = L/12 = 15/12 = 1.25\text{m}$$

Angulo de triangulación = 45° (consecuentemente la longitud de segmento de brida es 1.25m).

Una vez definida la geometría de la armadura se procedió con la optimización de tamaño. Si bien la optimización de tamaño realizada en este ejemplo es similar a la optimización usada en el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, tiene la ventaja de optimizar toda la armadura a la vez y no elemento por elemento como se suele realizar en SAP2000. Galápagos suele trabajar con funciones objetivo o funciones de optimización para optimizar, en este caso se usó la función de optimización que se muestra en la figura 3.73.

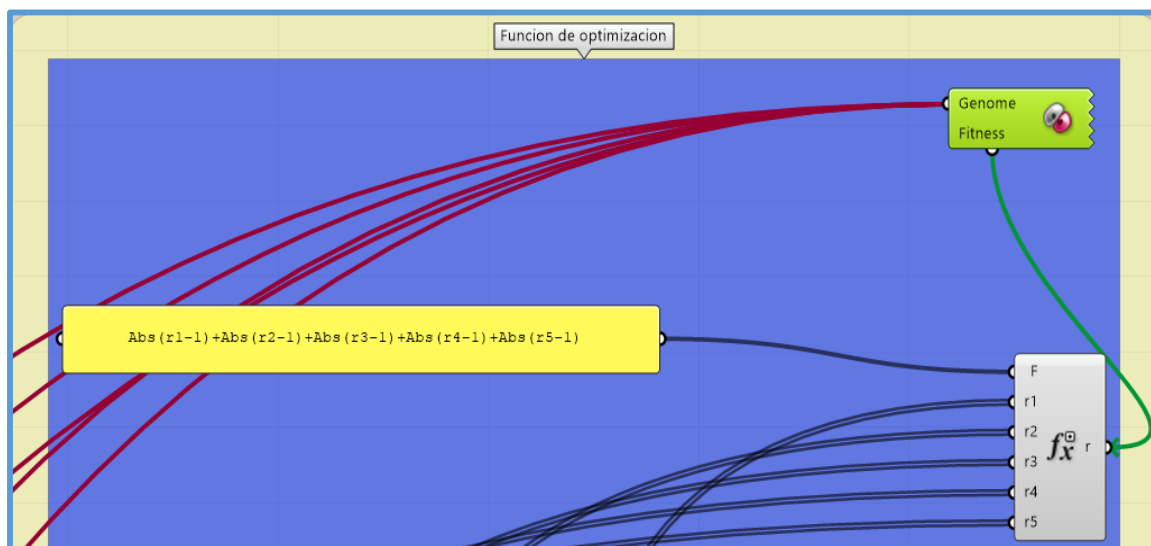


Figura 3.73. Función de optimización - optimización de tamaño - ejemplo N°2.

La función de optimización cuenta con las siguientes variables: r_1 , r_2 , r_3 , r_4 y r_5 que representan los ratios de diseño de los elementos de la brida superior, los ratios de diseño de las montantes, los ratios de diseño de los elementos de la brida inferior, los ratios de diseño de las diagonales – izquierda y los ratios de diseño de las diagonales – derecha respectivamente. Galápagos minimizó (óptimizo) el resultado de la función de optimización y dicho resultado mínimo indica que se ha

elegido los ratios de diseño de los elementos de la armadura más próximos a 1. También cabe indicar que los parámetros que tomo Galápagos para minimizar el resultado de la función de optimización fueron las secciones transversales dentro de listas, una lista por cada tipo de elemento de armadura (brida superior, montantes, brida inferior, diagonales – izquierda y diagonales – derecha).

Una vez definida la función de optimización se realizaron sus respectivas conexiones, también se realizó las conexiones de Galápagos, luego en el entorno de Galápagos se realizó la optimización en forma similar al ejemplo N°1. En este ejemplo el “Evolutionary solver” trabajó hasta la generación 50, la mejor solución encontrada hasta esa generación es la solución óptima, los resultados de la solución óptima se muestran en las siguientes figuras:

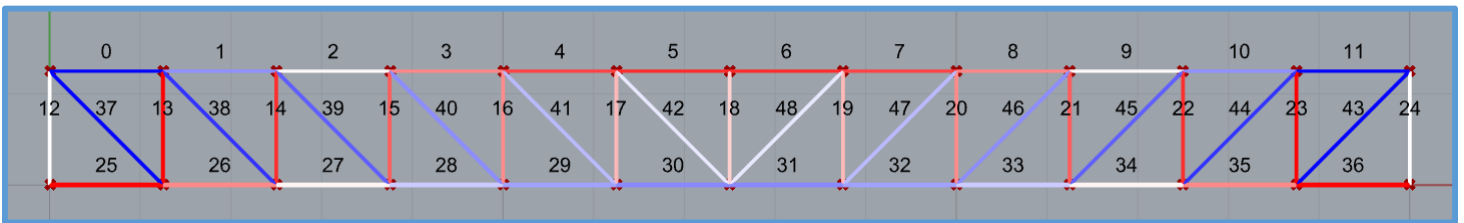


Figura 3.74. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización de tamaño - ejemplo N°2.

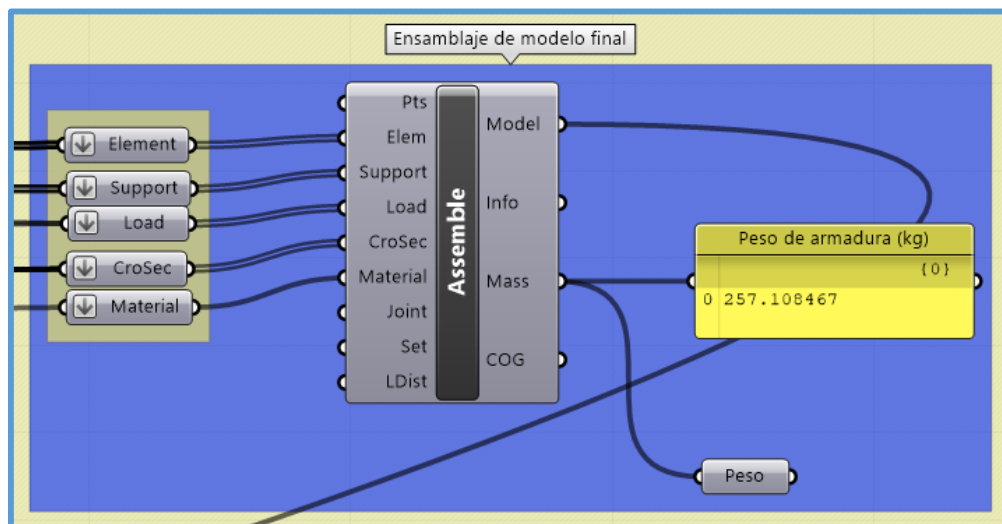


Figura 3.75. Peso de la armadura luego de la optimización de tamaño - ejemplo N°2.

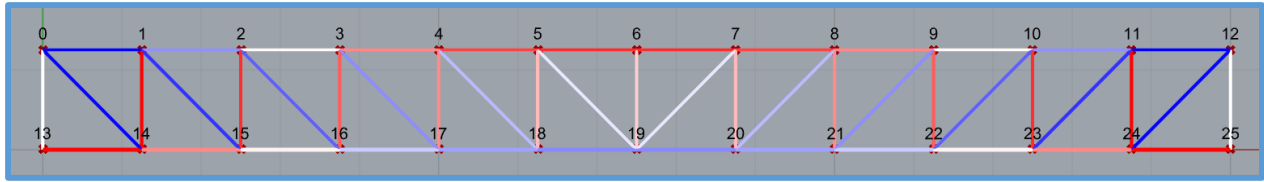


Figura 3.76. Nodos de armadura - optimización de tamaño - ejemplo N°2.

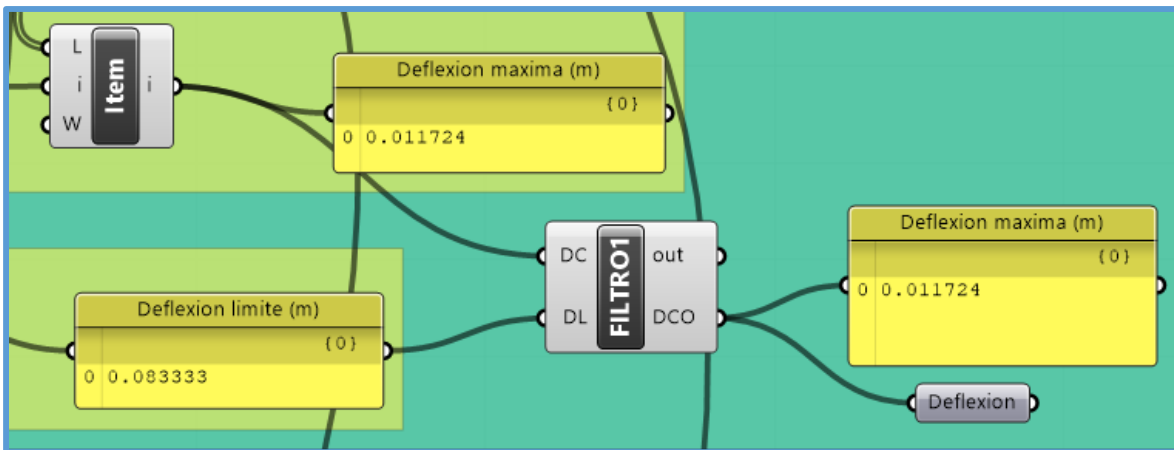


Figura 3.77. Deflexión máxima de la armadura luego de la optimización de tamaño - ejemplo N°2.

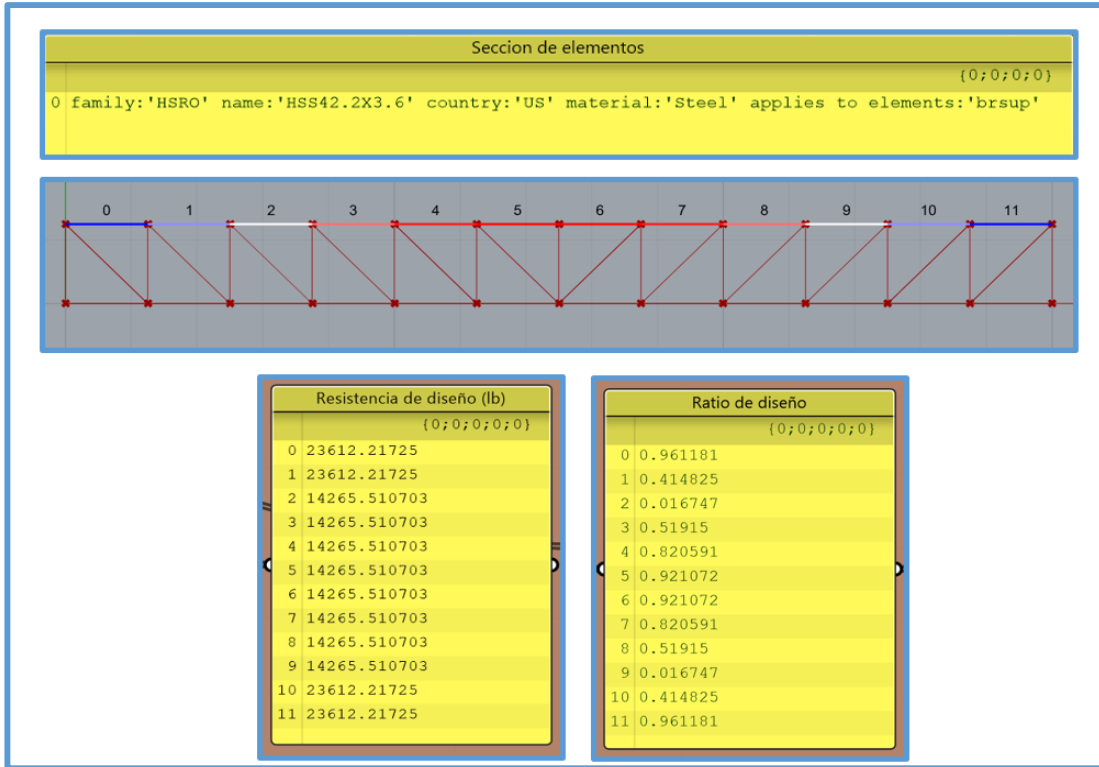


Figura 3.78. Resultados de diseño de la brida superior - optimización de tamaño - ejemplo N°2.

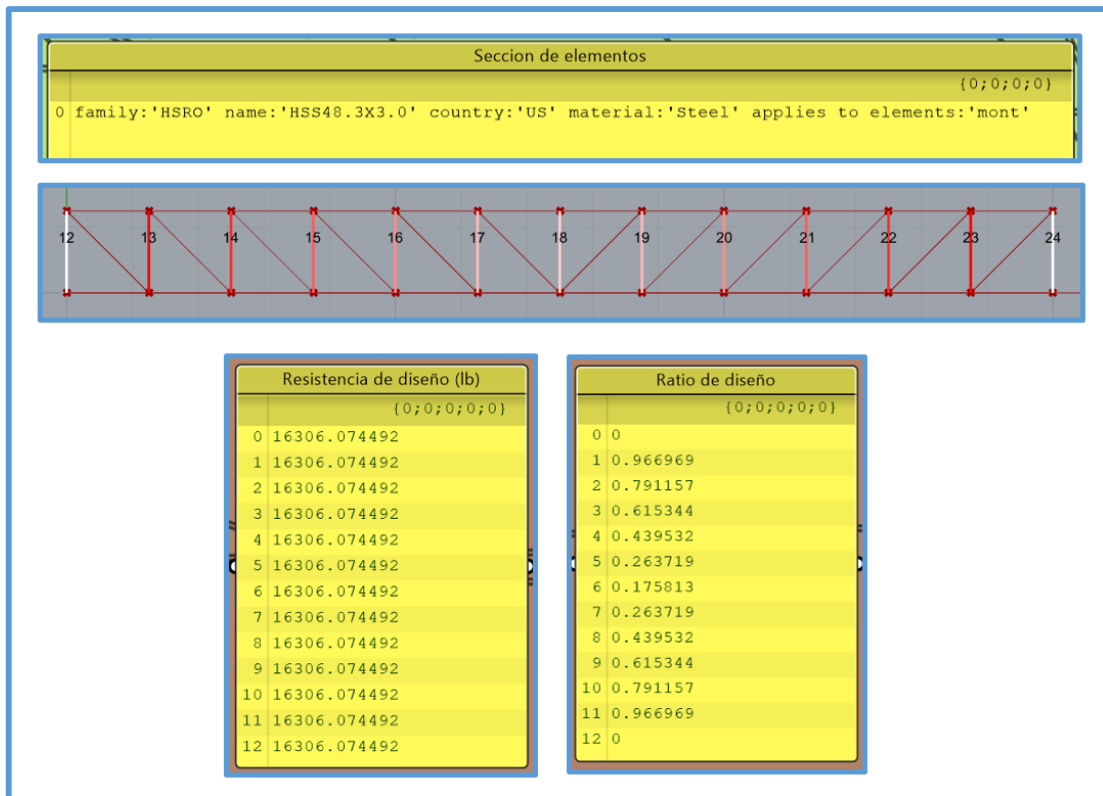


Figura 3.79. Resultados de diseño de las montantes - optimización de tamaño - ejemplo N°2.

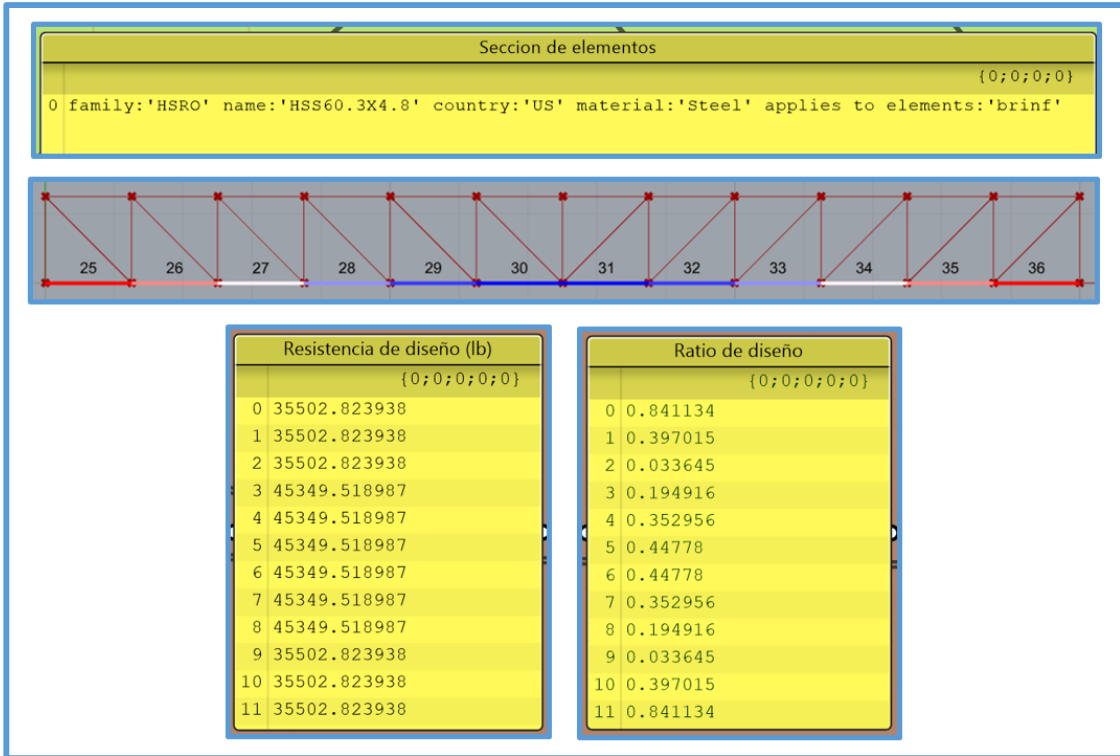


Figura 3.80. Resultados de diseño de la brida inferior - optimización de tamaño - ejemplo N°2.

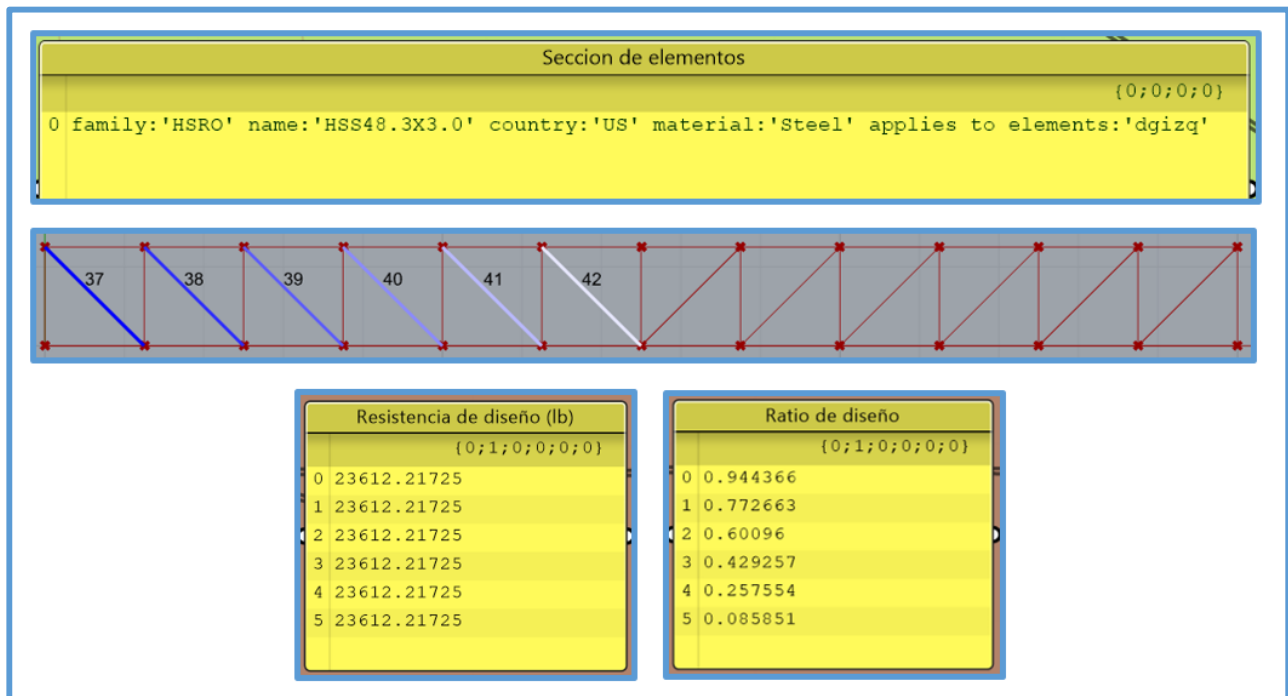


Figura 3.81. Resultados de diseño de las diagonales - izquierda - optimización de tamaño - ejemplo N°2.

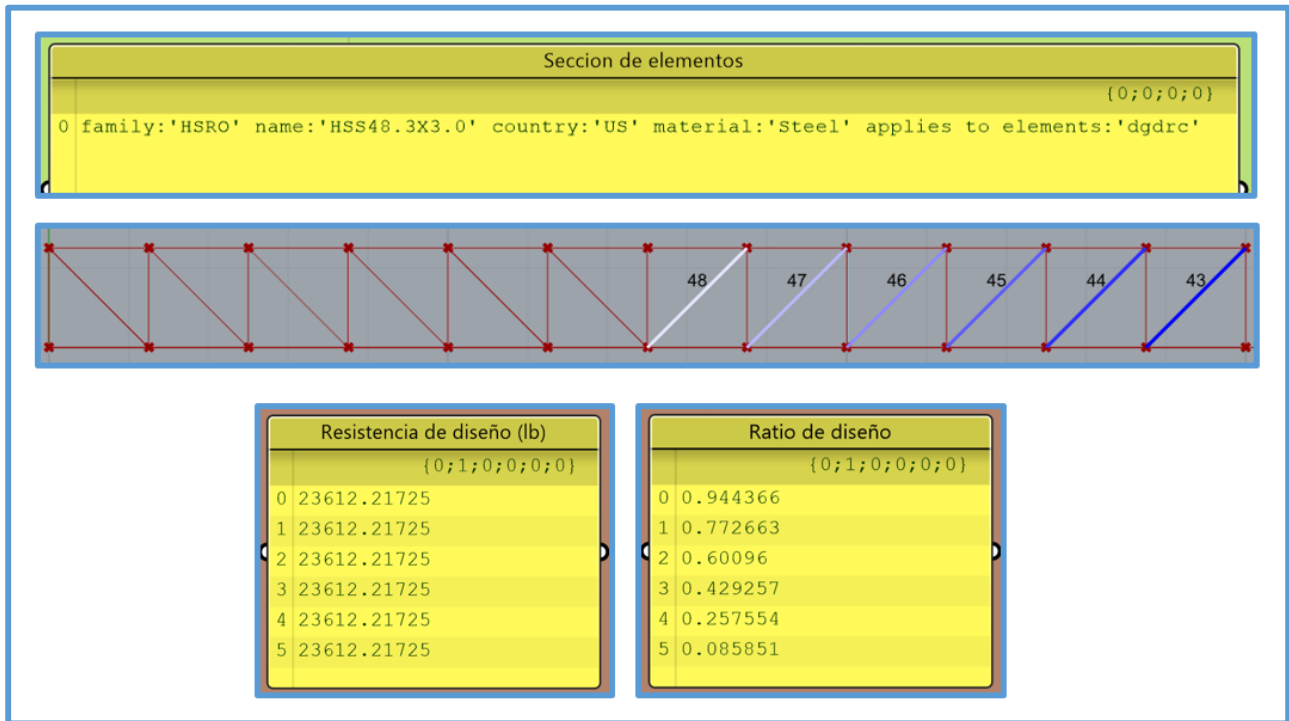


Figura 3.82. Resultados de diseño de las diagonales - derecha - optimización de tamaño - ejemplo N°2.

➤ **Diseño de armadura óptima con “SAP2000”- optimización de tamaño**

Para probar la veracidad de los resultados de diseño obtenidos con el proceso de diseño estructural propuesto en esta investigación se diseñó la armadura óptima obtenida en la optimización de tamaño (figura 3.74) en el software SAP2000, es así se muestran los resultados del diseño en SAP2000 en las siguientes figuras:

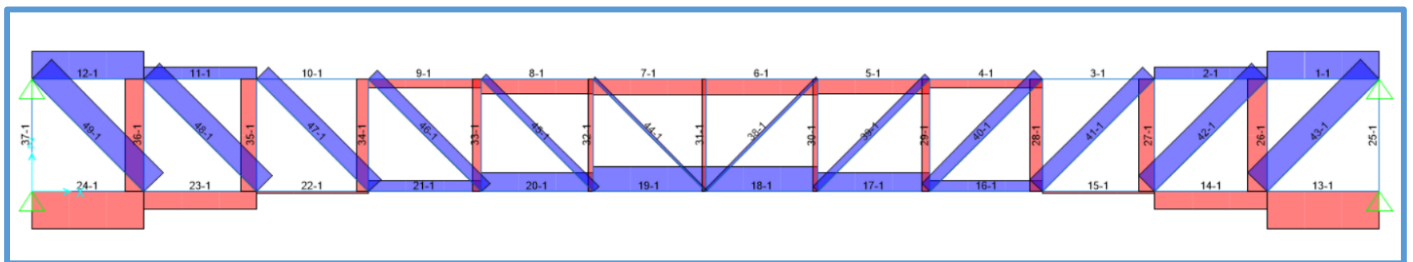


Figura 3.83. Elementos a tensión y/o compresión, con su respectivo número identificador en “SAP2000” - optimización de tamaño - ejemplo N°2.

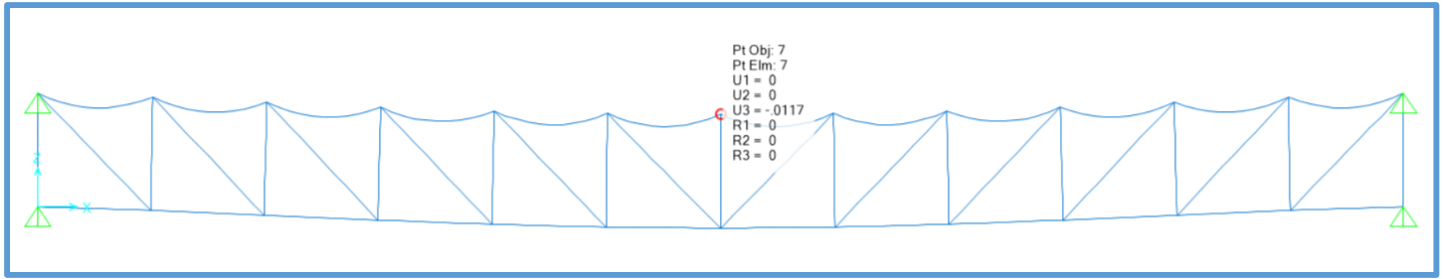


Figura 3.84. Deflexión máxima de armadura obtenida con “SAP2000”- optimización de tamaño - ejemplo N°2.

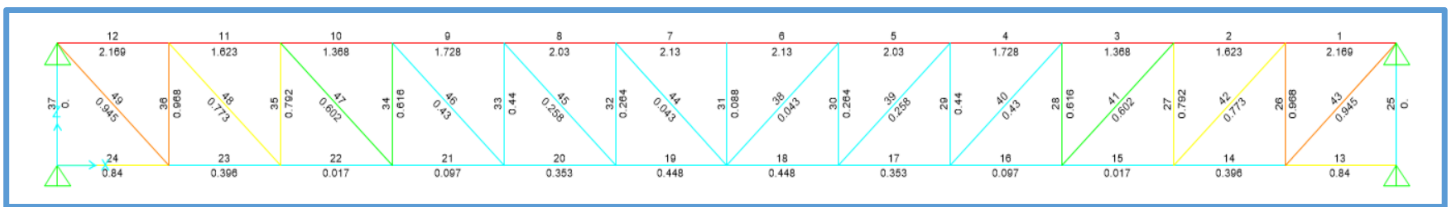


Figura 3.85. Ratios de diseño de los elementos de la armadura obtenidos con “SAP2000”- optimización de tamaño - ejemplo N°2.

Al igual que en el ejemplo N°1, como se trabajó con cargas distribuidas, SAP2000, para el caso de la brida superior también calculo flexión a pesar que con la opción “Assign Frame Releases and Partial Fixity” se ha indicado que los momentos en las direcciones 22 y 33 son cero, es por eso que los ratios mostrados en la brida superior son muy elevados, por lo tanto solo se considero los ratios generados por fuerzas axiales. A continuación, se muestra la tabla 3.21 en la cual se dan a conocer los resultados de diseño, de la armadura óptima obtenida en la optimización de tamaño, obtenidos con SAP2000.

Tabla 3.21. Resultados de diseño de armadura óptima, obtenidos con “SAP2000” - optimización de tamaño - ejemplo N°2.

Resultados de diseño de armadura óptima con SAP2000 - optimización de tamaño					
Frame	DesignSect	Combo	PRatio	PcComp	PcTension
Text	Text	Text	Unitless	Lb	Lb
1	HSS1.660X.140	COMB1	0.960661	14254.13	23625
2	HSS1.660X.140	COMB1	0.414601	14254.13	23625
3	HSS1.660X.140	COMB1	0.00838	14254.13	23625
4	HSS1.660X.140	COMB1	0.519564	14254.13	23625
5	HSS1.660X.140	COMB1	0.821247	14254.13	23625
6	HSS1.660X.140	COMB1	0.921808	14254.13	23625
7	HSS1.660X.140	COMB1	0.921808	14254.13	23625
8	HSS1.660X.140	COMB1	0.821247	14254.13	23625
9	HSS1.660X.140	COMB1	0.519564	14254.13	23625
10	HSS1.660X.140	COMB1	0.00838	14254.13	23625
11	HSS1.660X.140	COMB1	0.414601	14254.13	23625
12	HSS1.660X.140	COMB1	0.960661	14254.13	23625
13	HSS2.375X.188	COMB1	0.839872	35556.17	45360
14	HSS2.375X.188	COMB1	0.39642	35556.17	45360
15	HSS2.375X.188	COMB1	0.016797	35556.17	45360
16	HSS2.375X.188	COMB1	0.097435	35556.17	45360
17	HSS2.375X.188	COMB1	0.352874	35556.17	45360
18	HSS2.375X.188	COMB1	0.447676	35556.17	45360
19	HSS2.375X.188	COMB1	0.447676	35556.17	45360
20	HSS2.375X.188	COMB1	0.352874	35556.17	45360
21	HSS2.375X.188	COMB1	0.097435	35556.17	45360
22	HSS2.375X.188	COMB1	0.016797	35556.17	45360
23	HSS2.375X.188	COMB1	0.39642	35556.17	45360
24	HSS2.375X.188	COMB1	0.839872	35556.17	45360
25	HSS1.900X.120	COMB1	0	16295.81	23587.2
26	HSS1.900X.120	COMB1	0.967579	16295.81	23587.2
27	HSS1.900X.120	COMB1	0.791655	16295.81	23587.2
28	HSS1.900X.120	COMB1	0.615732	16295.81	23587.2
29	HSS1.900X.120	COMB1	0.439809	16295.81	23587.2
30	HSS1.900X.120	COMB1	0.263885	16295.81	23587.2

Tabla 3.21. Resultados de diseño de armadura óptima, obtenidos con “SAP2000” - optimización de tamaño - ejemplo N°2.

Resultados de diseño de armadura óptima con SAP2000 - optimización de tamaño					
Frame	DesignSect	Combo	PRatio	PcComp	PcTension
Text	Text	Text	Unitless	Lb	Lb
31	HSS1.900X.120	COMB1	0.087962	16295.81	23587.2
32	HSS1.900X.120	COMB1	0.263885	16295.81	23587.2
33	HSS1.900X.120	COMB1	0.439809	16295.81	23587.2
34	HSS1.900X.120	COMB1	0.615732	16295.81	23587.2
35	HSS1.900X.120	COMB1	0.791655	16295.81	23587.2
36	HSS1.900X.120	COMB1	0.967579	16295.81	23587.2
37	HSS1.900X.120	COMB1	0	16295.81	23587.2
38	HSS1.900X.120	COMB1	0.042971	11258.36	23587.2
39	HSS1.900X.120	COMB1	0.257828	11258.36	23587.2
40	HSS1.900X.120	COMB1	0.429713	11258.36	23587.2
41	HSS1.900X.120	COMB1	0.601598	11258.36	23587.2
42	HSS1.900X.120	COMB1	0.773483	11258.36	23587.2
43	HSS1.900X.120	COMB1	0.945368	11258.36	23587.2
44	HSS1.900X.120	COMB1	0.042971	11258.36	23587.2
45	HSS1.900X.120	COMB1	0.257828	11258.36	23587.2
46	HSS1.900X.120	COMB1	0.429713	11258.36	23587.2
47	HSS1.900X.120	COMB1	0.601598	11258.36	23587.2
48	HSS1.900X.120	COMB1	0.773483	11258.36	23587.2
49	HSS1.900X.120	COMB1	0.945368	11258.36	23587.2

➤ Optimización geométrica y de tamaño con Galápagos

Para realizar la optimización geométrica y de tamaño en simultaneo se usó Galápagos, en este caso la función de optimización fue la que se muestra en la figura 3.86, cuyas variables son p , d , r_1 , r_2 , r_3 , r_4 y r_5 que representan el peso, la deflexión máxima, los ratios de diseño de los elementos de la brida superior, los ratios de diseño de las montantes, los ratios de diseño de los elementos de la brida inferior, los ratios de diseño de las diagonales – izquierda y los ratios de diseño de las diagonales – derecha respectivamente. Galápagos minimizó (óptimizo) el resultado de la función de optimización y dicho resultado mínimo indica que se ha elegido el menor peso, la menor deflexión máxima, y los ratios de diseño de los elementos de la armadura más próximos a 1. También cabe indicar que los parámetros que toma Galápagos para minimizar el resultado de la función de optimización son la flecha superior, peralte, flecha inferior, el número de segmentos (estos cuatro definidos en la modelación paramétrica) y las secciones transversales dentro de listas, una lista por cada tipo de elemento de armadura (brida superior, montantes, brida inferior, diagonales – izquierda y diagonales – derecha).

Se define la función de optimización con la cual trabaja el complemento Galápagos.

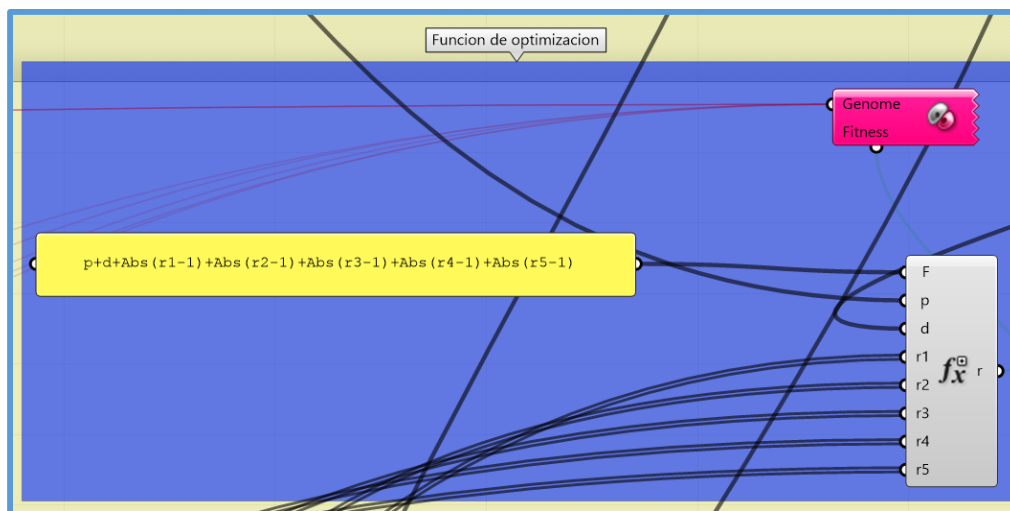


Figura 3.86. Función de optimización - optimización geométrica y de tamaño - ejemplo N°2.

Una vez definida la función de optimización se realizaron sus respectivas conexiones, también se realizaron las conexiones de Galápagos, luego en el entorno de Galápagos se realizó la optimización en forma similar al ejemplo N°1, cabe recalcar que en esta ocasión por medio de la función objetivo se realizó la optimización geométrica y de tamaño en simultaneo. Es recomendable en este caso de optimización dejar a Galápagos realizar la optimización unas 3 veces

y en un intervalo de tiempo determinado en cada vez, se recomienda un intervalo de 5 a 15 minutos, puesto que en periodos menores a 5 minutos puede que la solución encontrada no sea la más óptima y en periodos mayores a 15 minutos puede que las soluciones tengan parámetros inconsistentes como peraltes de 5cm o menores. En este ejemplo la solución óptima fue la obtenida en la primera vez, a los 8 minutos y los resultados de dicha solución son los que se muestran en las siguientes figuras:

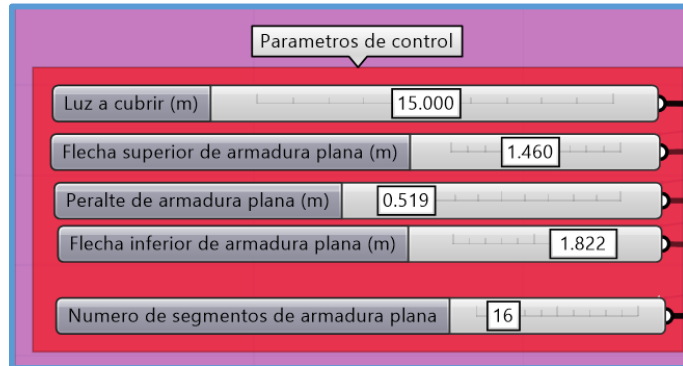


Figura 3.87. Parámetros de control luego de optimización geométrica y de tamaño - ejemplo N°2.

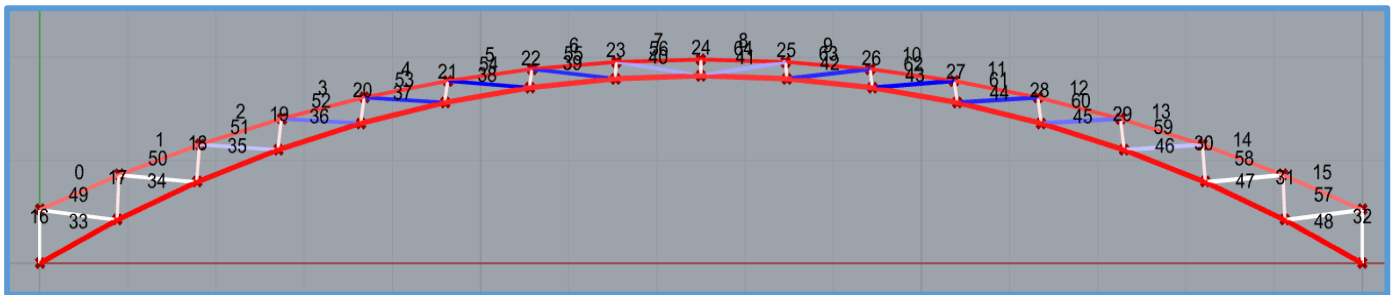


Figura 3.88. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por “Karamba” - optimización geométrica y de tamaño - ejemplo N°2.

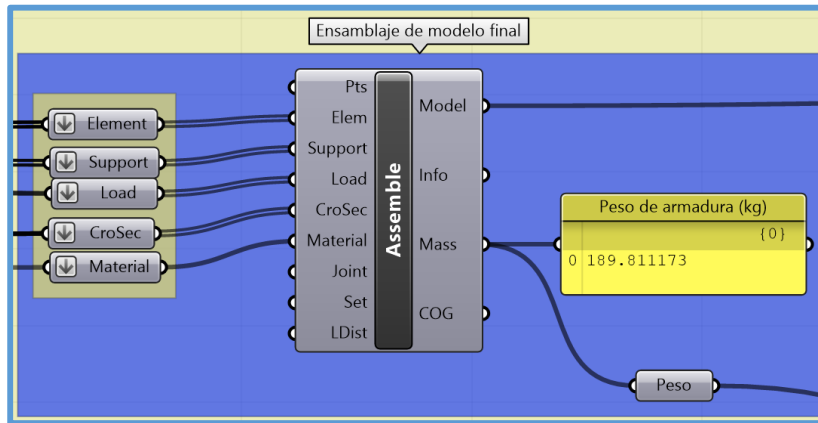


Figura 3.89. Peso de la armadura luego de la optimización geométrica y de tamaño - ejemplo N°2.

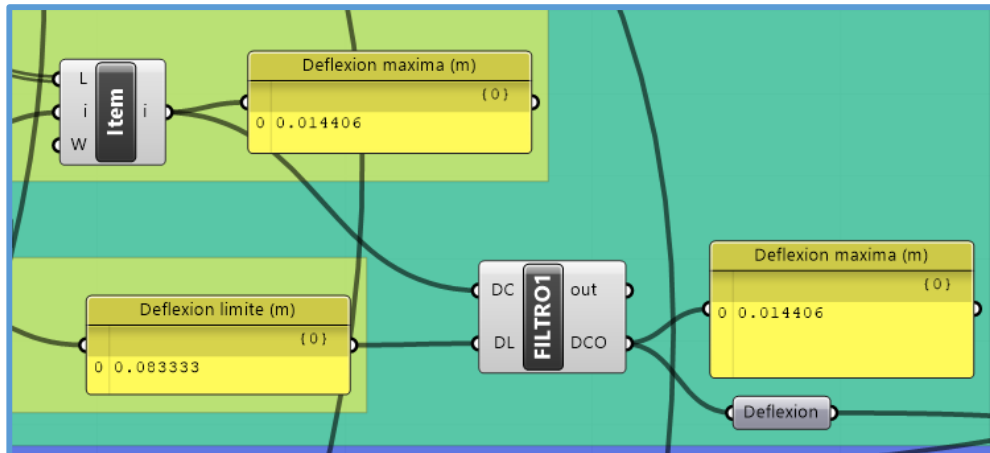


Figura 3.90. Deflexión máxima de la armadura luego de la optimización geométrica y de tamaño - ejemplo N°2.

➤ Optimización topológica con el componente BESOBBeam

Se aplicó la optimización topológica (usando el componente “BESOBBeam”) a la solución óptima obtenida en la optimización geométrica y de tamaño, para lo cual se realizó las implementaciones necesarias al código gráfico de la optimización (de manera similar al ejemplo N°1) definido en la figura 3.72, para más detalles revisar los archivos en Grasshopper.

En este ejemplo se mantuvo las montantes de los extremos (por motivos de conexiones) a pesar de que no presentan fuerzas actuantes. Se aplicó la optimización topológica a las diagonales - izquierda y a las diagonales - derecha en simultaneo, obteniéndose la solución óptima (armadura óptima) al trabajar con un valor de 0.88 en el “Target mass ratio” del componente “BESOBeam”, como se muestra en la figura 3.91.

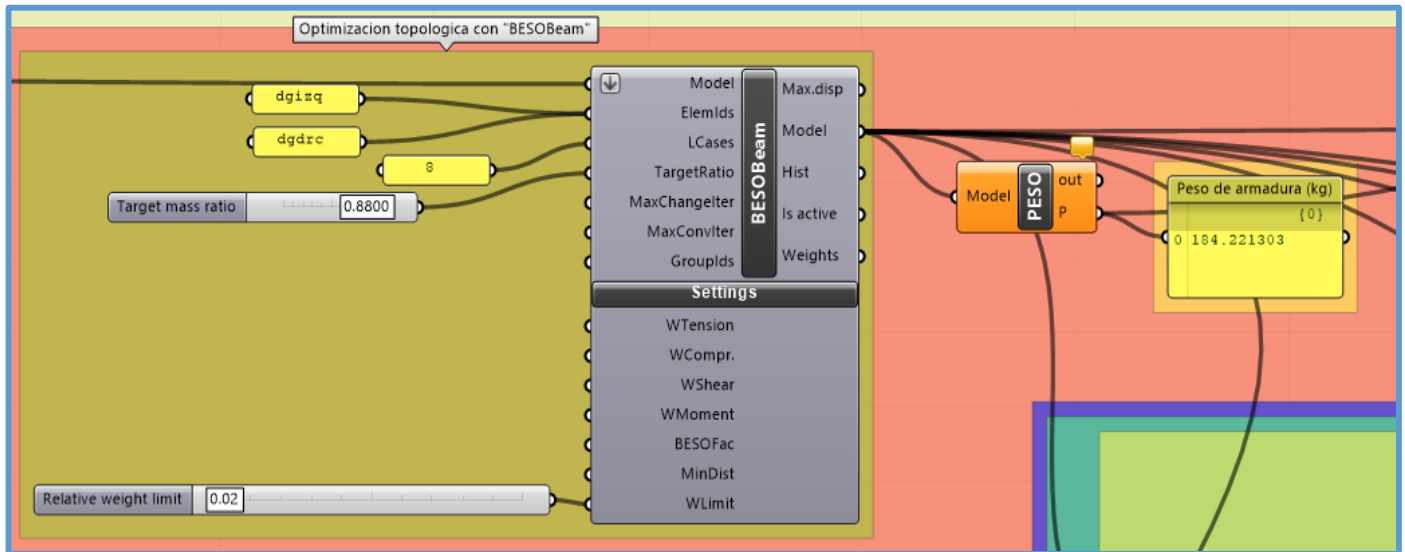


Figura 3.91. Optimización topológica con “BESOBBeam” - ejemplo N°2.

Al trabajar con un valor de 0.88 en el “Target mass ratio” el componente “BESOBBeam”, elimina dos diagonales de la armadura (una diagonal izquierda y una diagonal derecha) con los números identificadores 34 y 47, lo cual se puede apreciar en la figura 3.92.

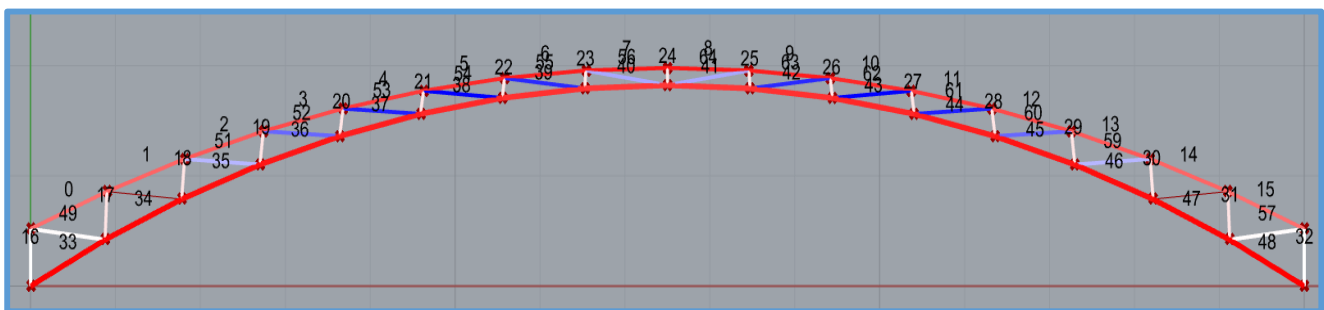


Figura 3.92. Armadura luego de una optimización topológica con un “Target mass ratio” de 0.88 - ejemplo N°2.

Los resultados de diseño de la armadura, luego de aplicar la optimización topológica se muestran en las siguientes figuras:

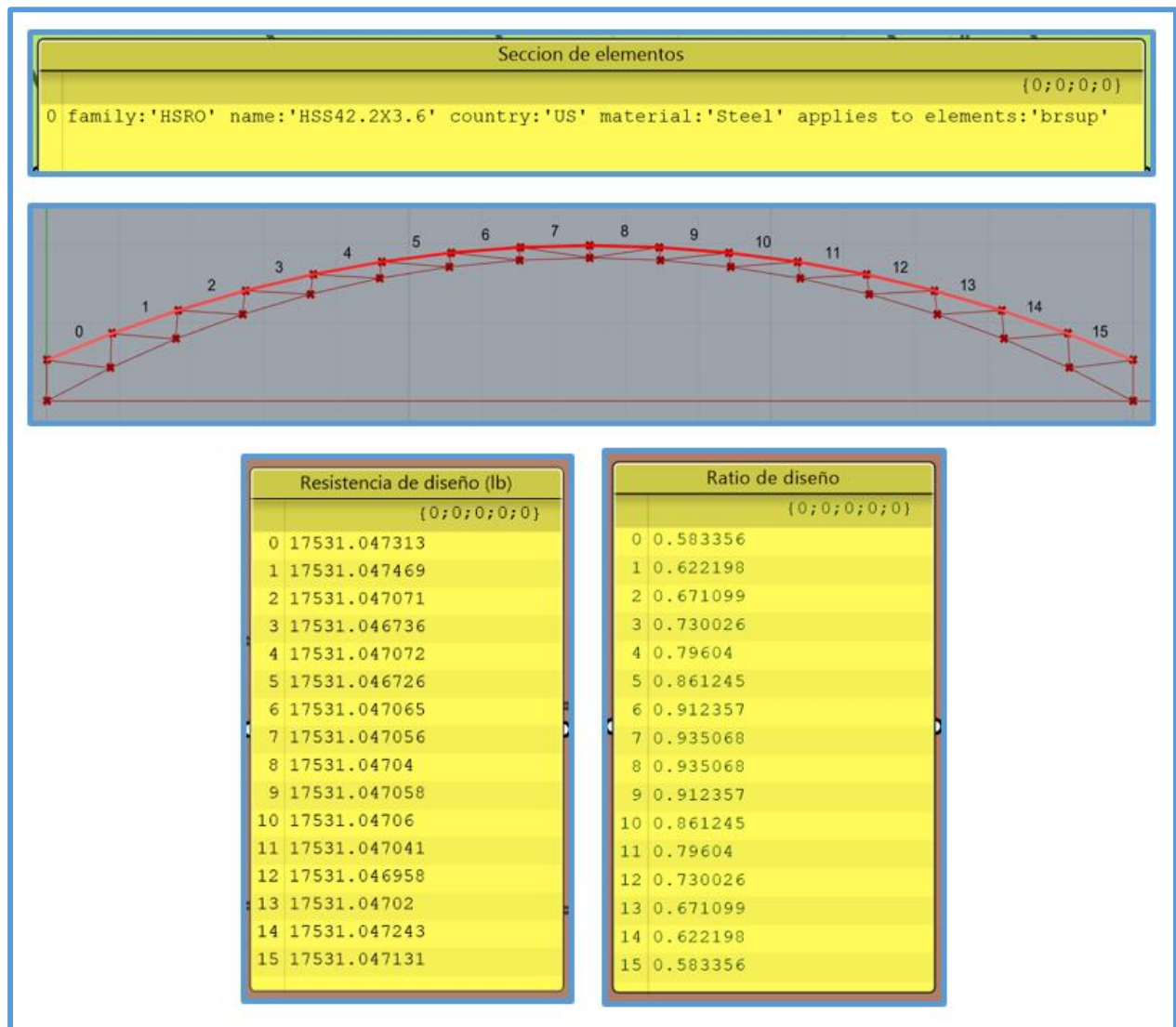


Figura 3.93. Resultados de diseño de la brida superior - optimización topológica - ejemplo N°2.

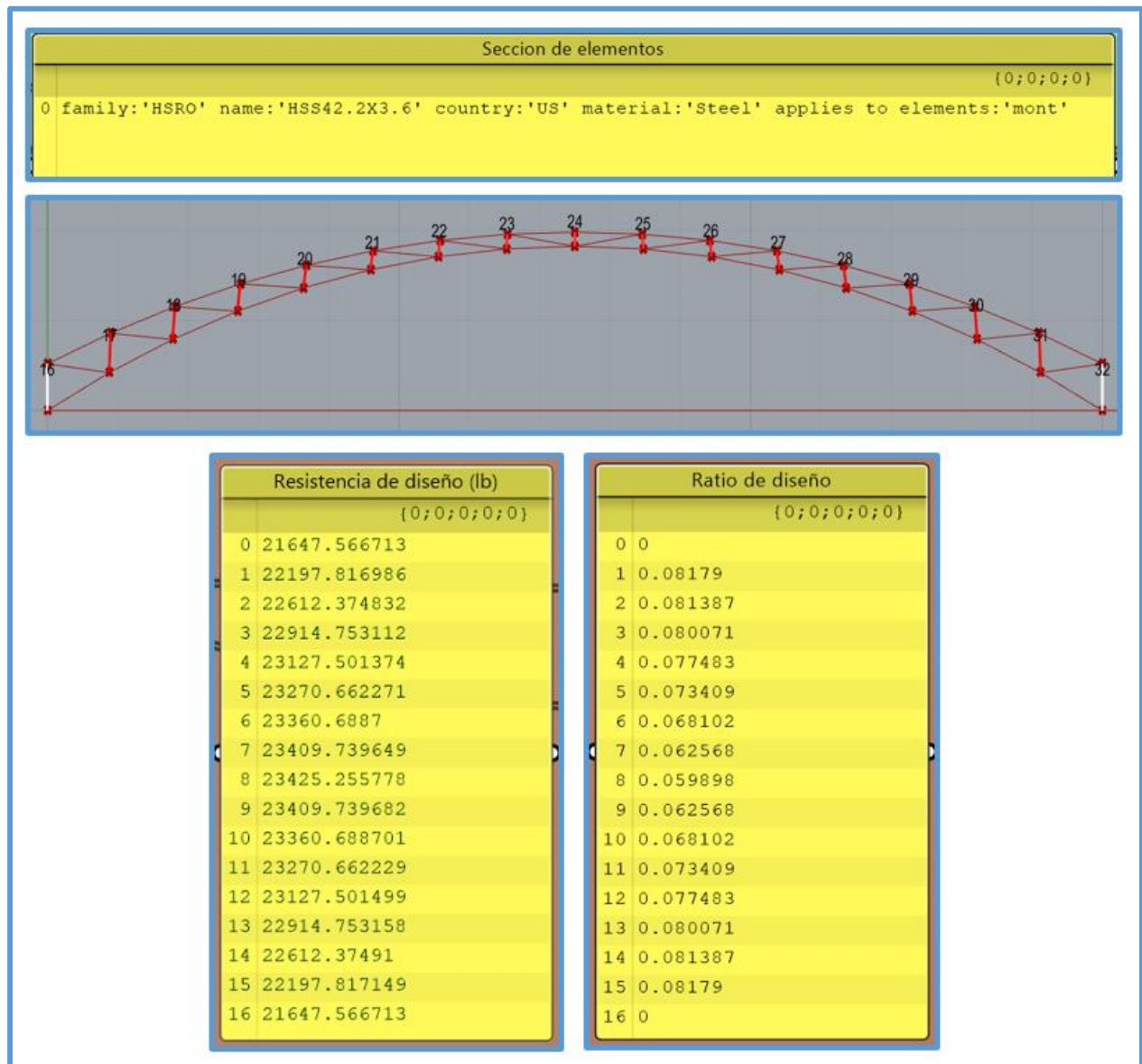


Figura 3.94. Resultados de diseño de las montantes - optimización topológica - ejemplo N°2.

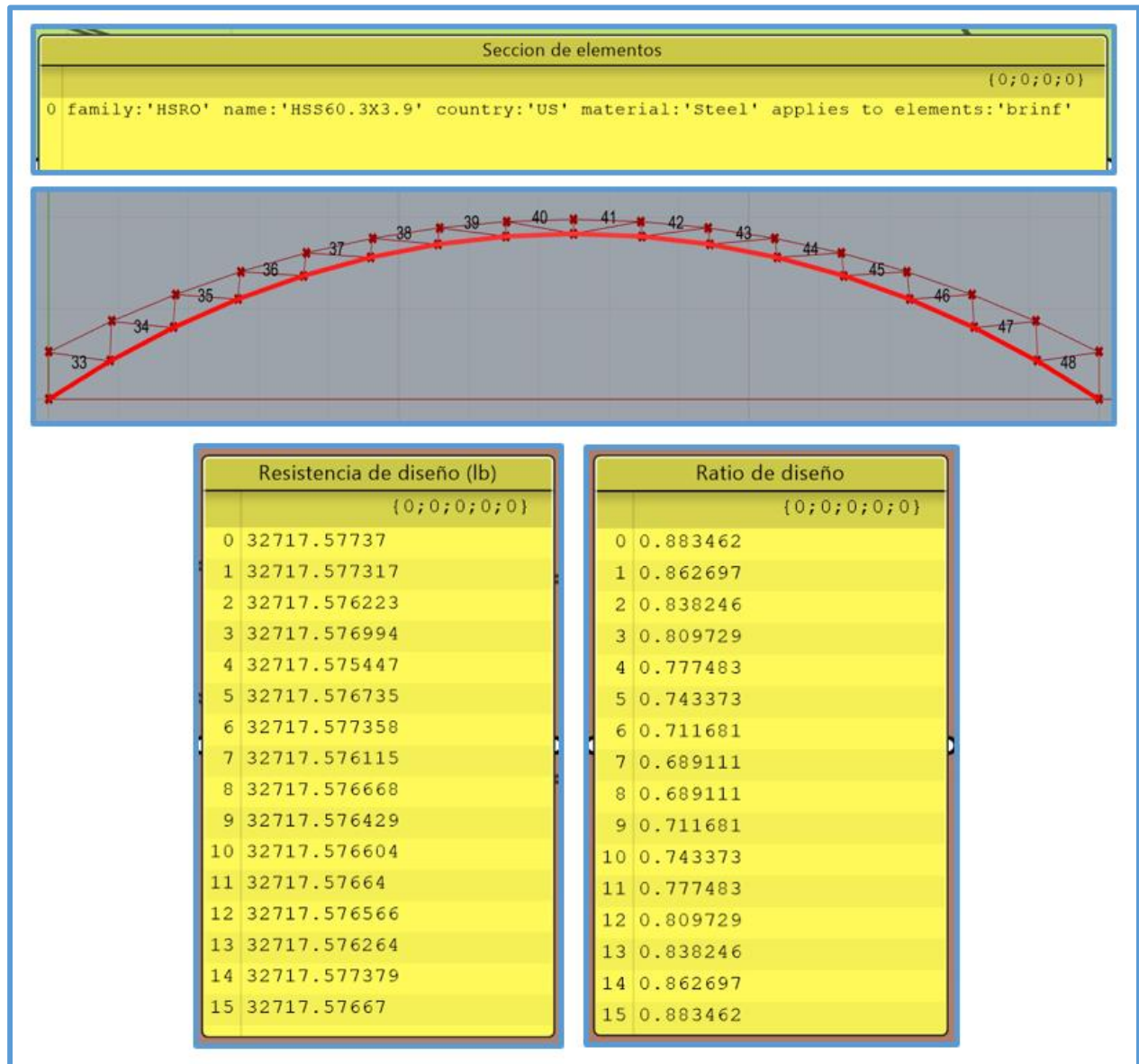


Figura 3.95. Resultados de diseño de la brida inferior - optimización topológica - ejemplo N°2.

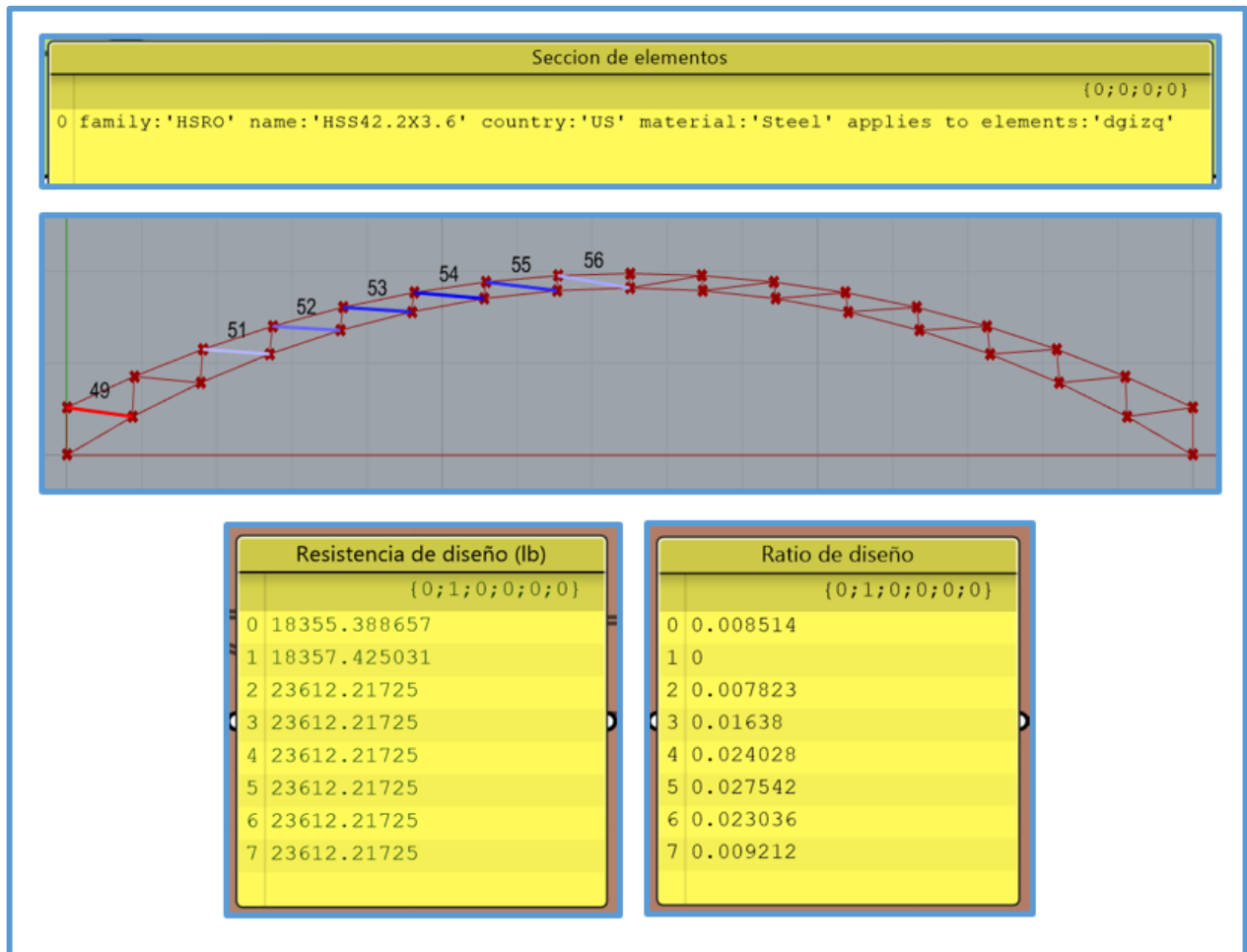


Figura 3.96. Resultados de diseño de las diagonales - izquierda - optimización topológica - ejemplo N°2.

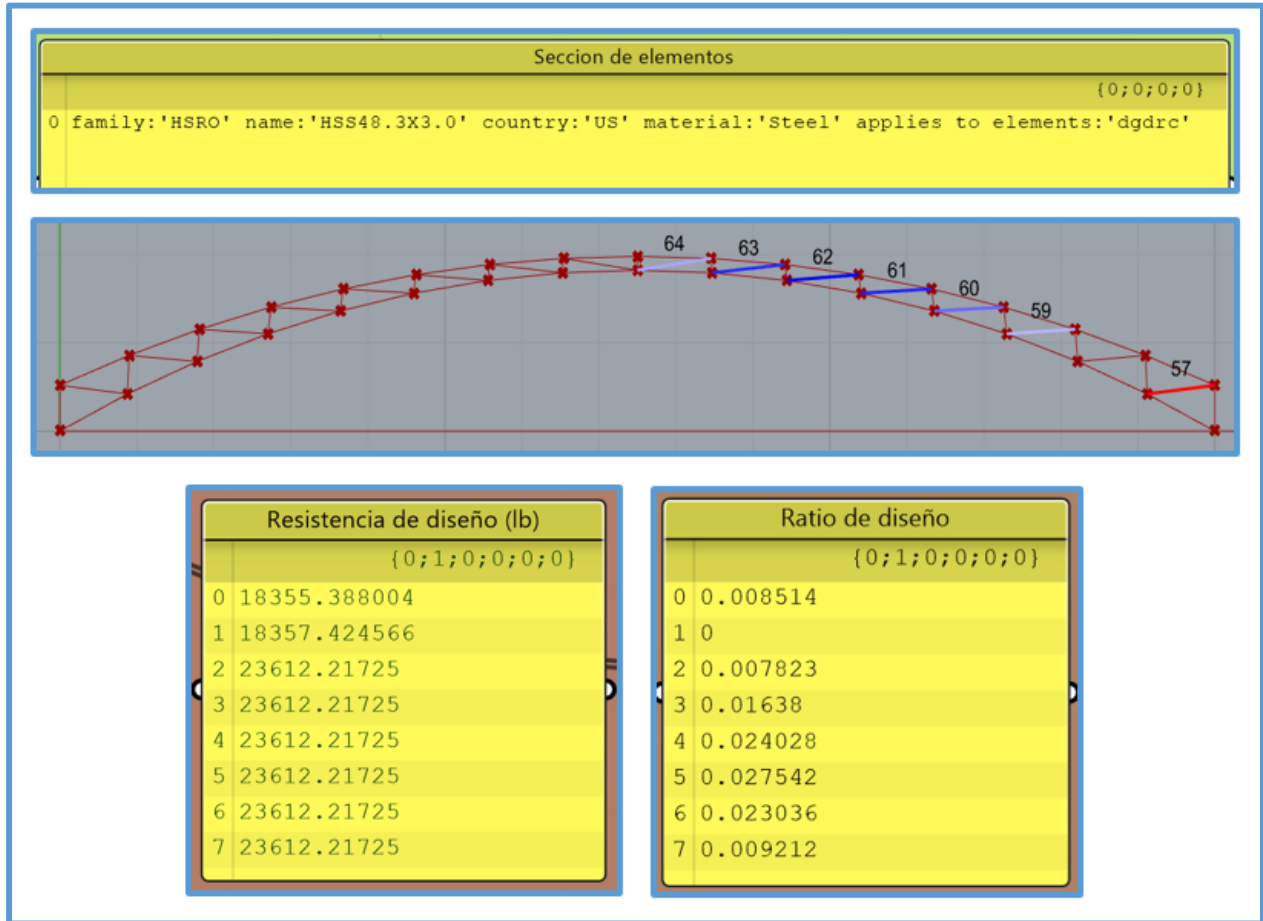


Figura 3.97. Resultados de diseño de las diagonales - derecha - optimización topológica - ejemplo N°2.

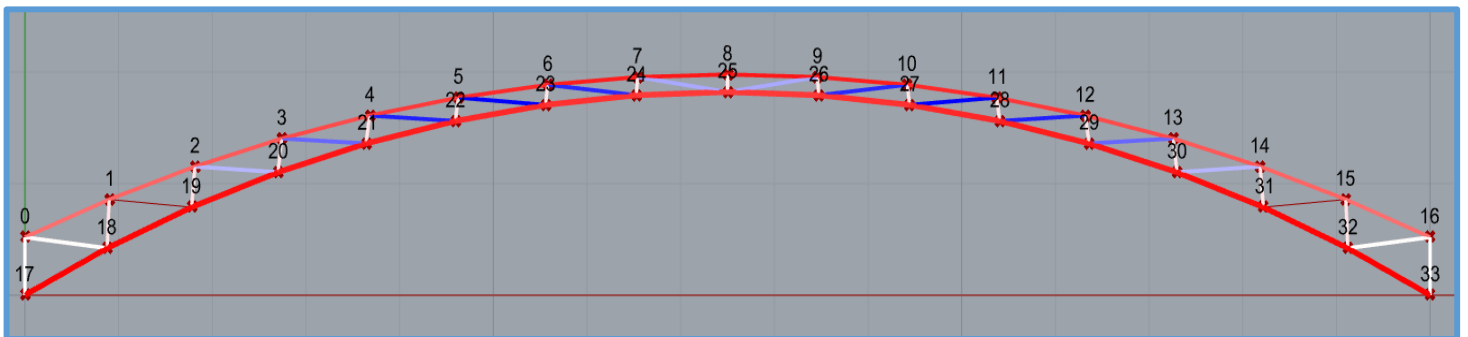


Figura 3.98. Nodos de armadura óptima - optimización topológica - ejemplo N°2.

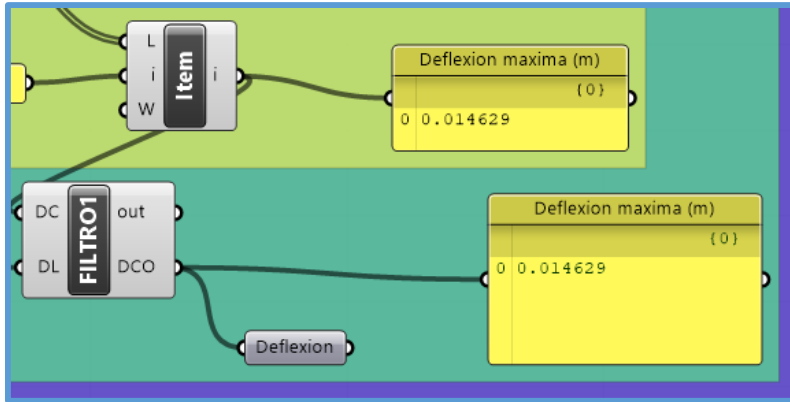


Figura 3.99. Deflexión máxima de armadura óptima - optimización topológica - ejemplo N°2.

➤ **Diseño de armadura óptima con “SAP2000” - optimización topológica**

Para probar la veracidad de los resultados de diseño obtenidos con el proceso de diseño estructural propuesto en esta investigación se diseñó la armadura óptima obtenida en la optimización topológica (figura 3.92) en el software SAP2000, es así que se muestran los resultados de diseño en SAP2000 en las siguientes figuras:

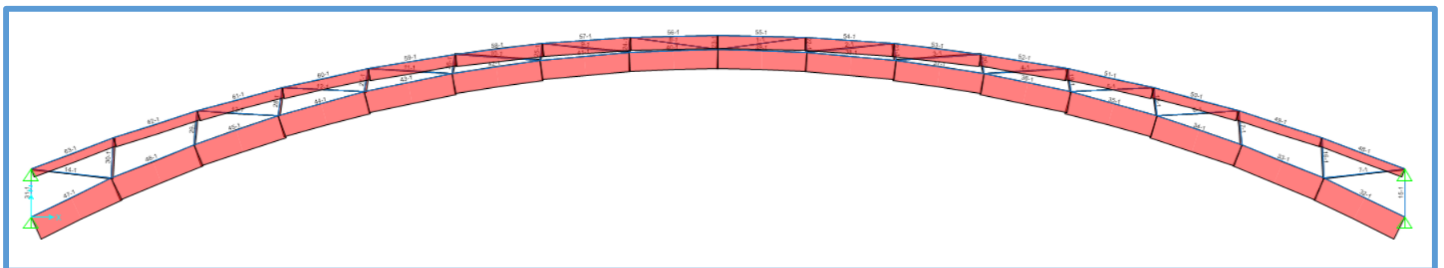


Figura 3.100. Elementos a tensión y/o compresión en “SAP2000” - optimización topológica - ejemplo N°2.

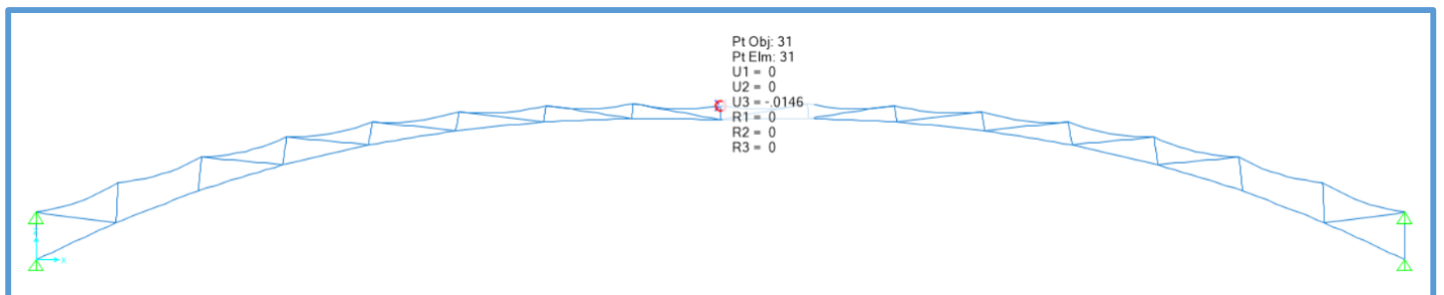


Figura 3.101. Deflexión máxima de armadura obtenida con “SAP2000”- optimización topológica - ejemplo N°2.

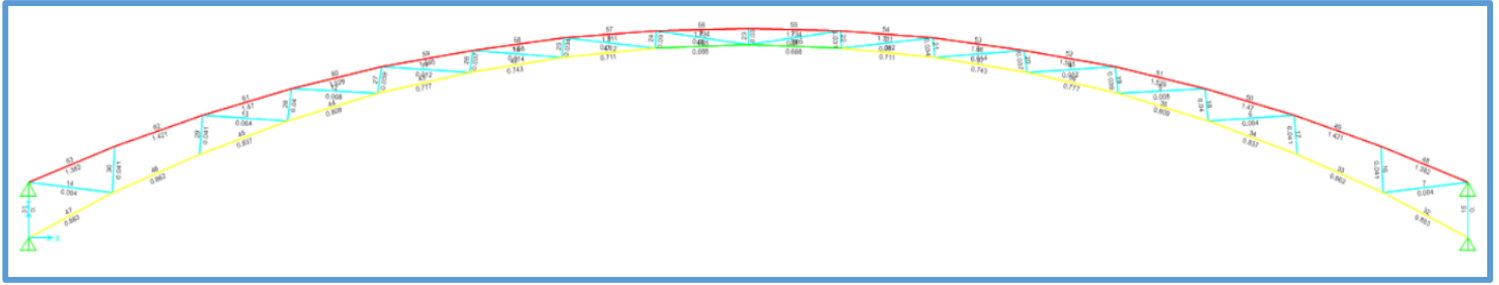


Tabla 3.22. Resultados de diseño de armadura óptima, obtenidos con “SAP2000”- optimización topológica - ejemplo N°2.

Resultados de diseño de armadura óptima con SAP2000 - optimización topológica					
Frame	DesignSect	Combo	PRatio	PcComp	PcTension
Text	Text	Text	Unitless	Kgf	Kgf
33	HSS2.375X.154	COMB1	0.861901	14846.55	17145.79
34	HSS2.375X.154	COMB1	0.837473	14846.55	17145.79
35	HSS2.375X.154	COMB1	0.808982	14846.55	17145.79
36	HSS2.375X.154	COMB1	0.776765	14846.55	17145.79
37	HSS2.375X.154	COMB1	0.742687	14846.55	17145.79
38	HSS2.375X.154	COMB1	0.711025	14846.55	17145.79
39	HSS2.375X.154	COMB1	0.688475	14846.55	17145.79
40	HSS2.375X.154	COMB1	0.688475	14846.55	17145.79
41	HSS2.375X.154	COMB1	0.711025	14846.55	17145.79
42	HSS2.375X.154	COMB1	0.742687	14846.55	17145.79
43	HSS2.375X.154	COMB1	0.776765	14846.55	17145.79
44	HSS2.375X.154	COMB1	0.808982	14846.55	17145.79
45	HSS2.375X.154	COMB1	0.837473	14846.55	17145.79
46	HSS2.375X.154	COMB1	0.8619	14846.55	17145.79
47	HSS2.375X.154	COMB1	0.882646	14846.55	17145.79
48	HSS1.660X.140	COMB1	0.583437	7949.96	10716.12
49	HSS1.660X.140	COMB1	0.622269	7949.96	10716.12
50	HSS1.660X.140	COMB1	0.671157	7949.96	10716.12
51	HSS1.660X.140	COMB1	0.730069	7949.96	10716.12
52	HSS1.660X.140	COMB1	0.796066	7949.96	10716.12
53	HSS1.660X.140	COMB1	0.861254	7949.96	10716.12
54	HSS1.660X.140	COMB1	0.912352	7949.96	10716.12
55	HSS1.660X.140	COMB1	0.935057	7949.96	10716.12
56	HSS1.660X.140	COMB1	0.935057	7949.96	10716.12
57	HSS1.660X.140	COMB1	0.912352	7949.96	10716.12
58	HSS1.660X.140	COMB1	0.861254	7949.96	10716.12
59	HSS1.660X.140	COMB1	0.796066	7949.96	10716.12
60	HSS1.660X.140	COMB1	0.730069	7949.96	10716.12
61	HSS1.660X.140	COMB1	0.671157	7949.96	10716.12
62	HSS1.660X.140	COMB1	0.622269	7949.96	10716.12
63	HSS1.660X.140	COMB1	0.583437	7949.96	10716.12

3.4.4. Ejemplo N°3

En este ejemplo se rediseño plenamente con el proceso de diseño estructural propuesto en esta investigación un arco de viga de celosía ya diseñado con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, con el objetivo de aplicar el proceso de diseño estructural propuesto en esta investigación en un caso real de diseño. Se ha considerado el proyecto: “MEJORAMIENTO DE LAS CONDICIONES DEL SERVICIO DE EDUCACIÓN SECUNDARIA EN LA I.E. FELIPE HUAMAN POMA DE AYALA, DEL CENTRO POBLADO EL TAMBO, DISTRITO DE BAMBAMARCA, PROVINCIA DE HUALGAYOC, REGIÓN CAJAMARCA”, que cuenta con 2 coberturas metálicas, en este ejemplo se rediseño un arco de viga de celosía central de la cobertura del bloque B1 (figura 3.103) del proyecto.

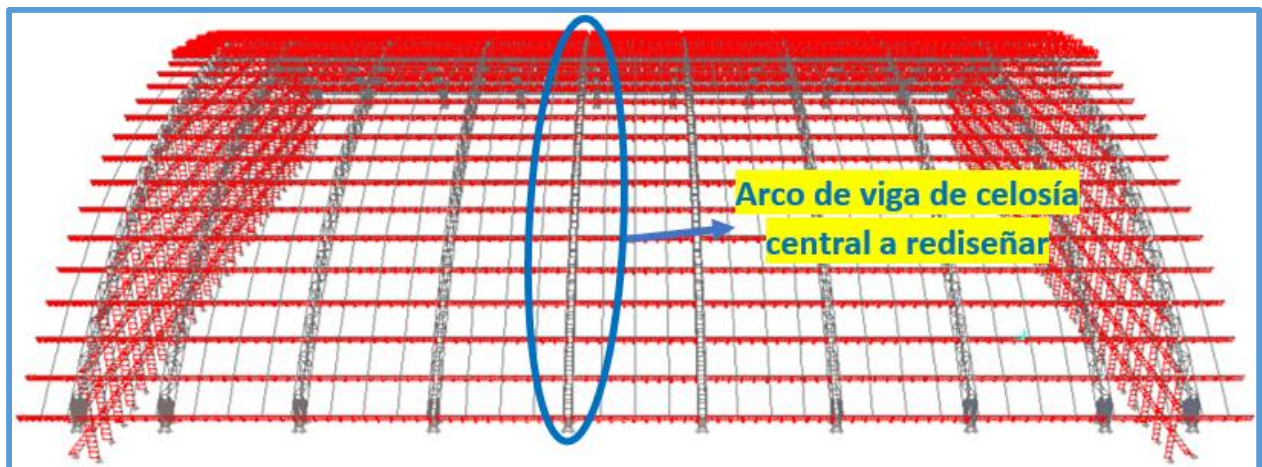


Figura 3.103. Cobertura metálica de bloque B1. Adaptado de (Quispe, 2016).

3.4.4.1. Modelación paramétrica

- Datos del proyecto:
 - Luz a cubrir de arco de viga de celosía: 41.00m.
 - Peralte de arco de viga de celosía: 0.95m.
 - Ancho de arco de viga de celosía: 0.30m.
 - Flecha de arco de viga de celosía: 8.20m.
 - Número de segmentos de arco de viga de celosía: 46.
- Se han definido los siguientes parámetros para el arco de viga de celosía:
 - Luz a cubrir. Tiene un rango de 20m a 60m.
 - Flecha de arco de viga de celosía: Tiene un rango de 1m a 15m.

- Peralte de arco de viga de celosía. Tiene un rango de 0.1m a 5m.
 - Número de segmentos de arco de viga de celosía. Tiene un rango de 2 a 200.
 - Ancho de arco de viga de celosía. Tiene un rango de 0.1m a 5m.
- Se puede apreciar el código gráfico de la modelación paramétrica del ejemplo N°3 en la siguiente figura:

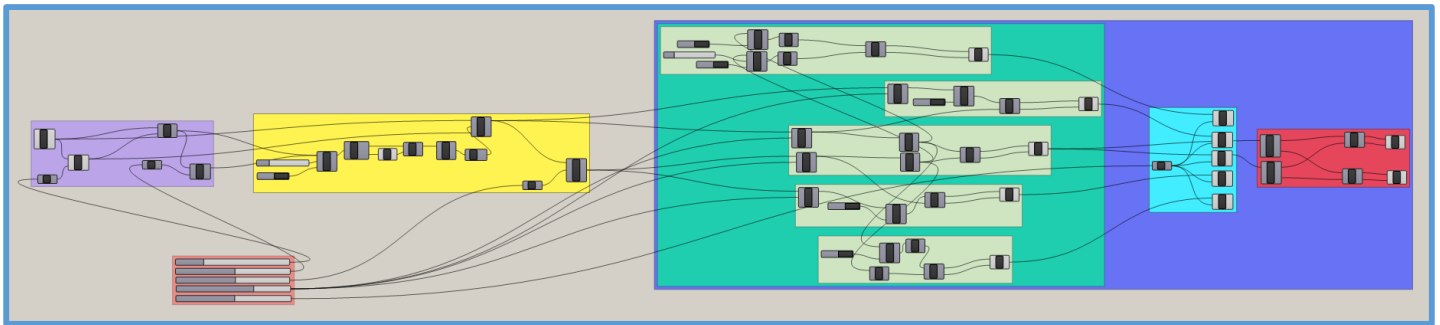


Figura 3.104. Código gráfico de la modelación paramétrica del ejemplo N°3.

En la siguiente figura se muestra el arco de viga de celosía definido en la modelación paramétrica, en el entorno de Rhinoceros 3D.

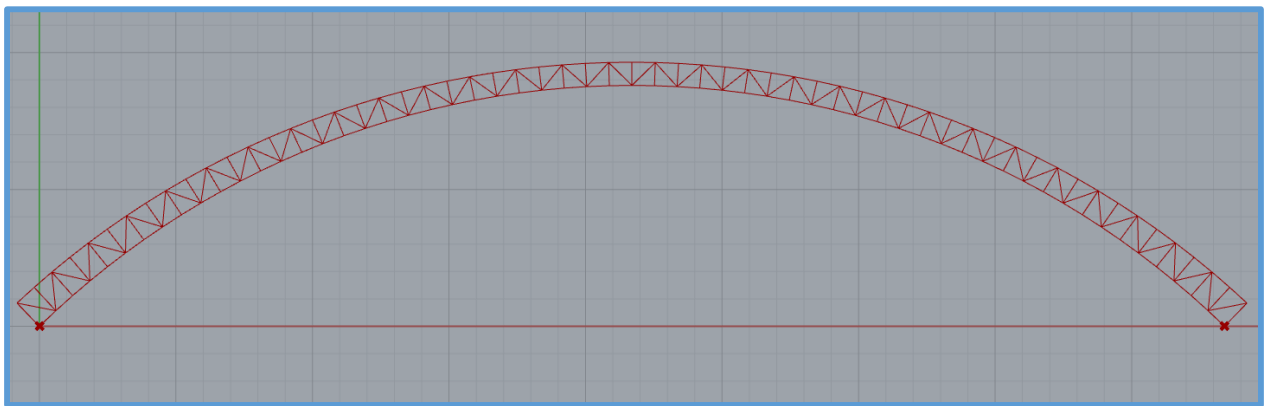


Figura 3.105. Arco de viga de celosía definido en modelación paramétrica - ejemplo N°3.

3.4.4.2. Optimización

Para el desarrollo de la optimización se implementó el código gráfico de la modelación paramétrica, así se tiene el código gráfico del ejemplo en la siguiente figura:

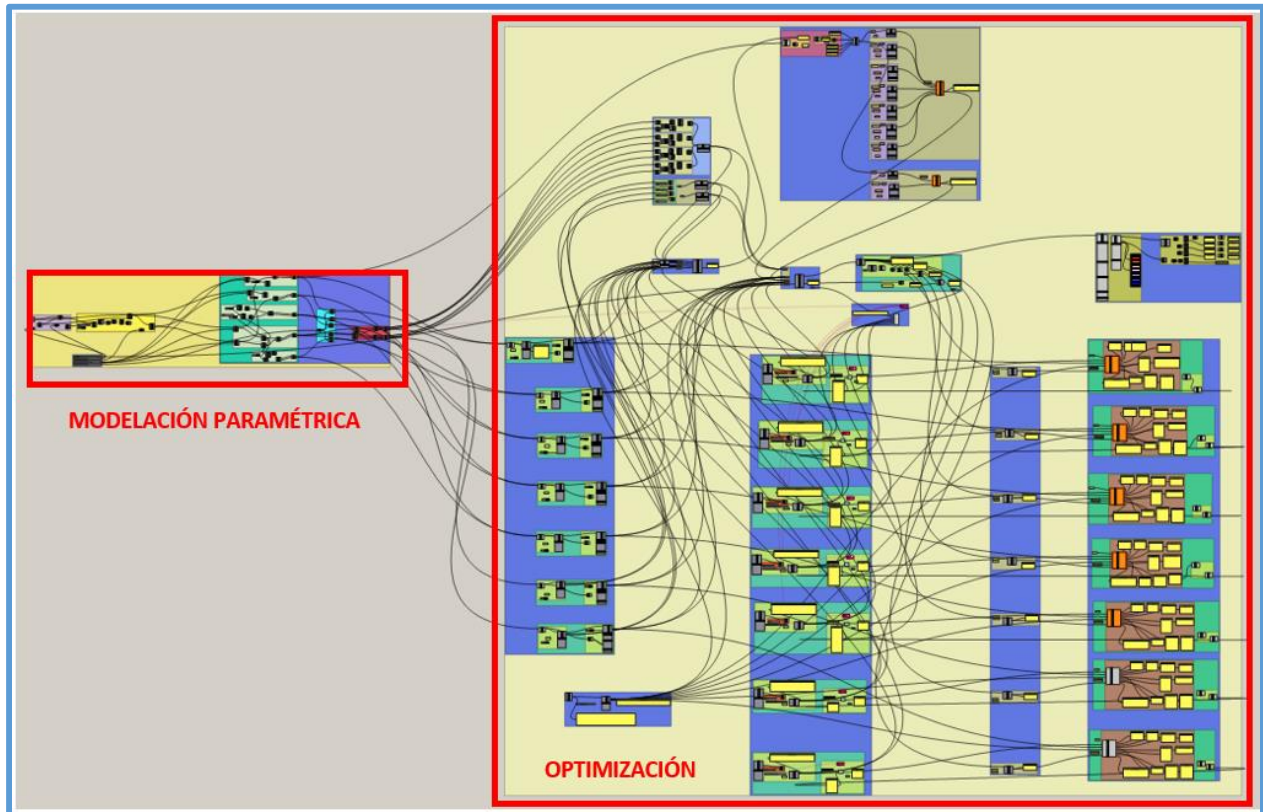


Figura 3.106. Código gráfico de modelación paramétrica y optimización - ejemplo N°3.

Antes de realizar el proceso de optimización se tuvo en cuenta lo siguiente:

- La carga de viento indicada en la memoria de cálculo del proyecto es de:

$$P_h = 26.89 \text{ kg/m}^2 \text{ para barlovento.}$$

$$P_h = 20.17 \text{ kg/m}^2 \text{ para Sotavento.}$$

Para la combinación de cargas se trabajó con el caso más desfavorable, el cual se produce cuando toda la superficie de la cobertura se encuentra a presión (barlovento), entonces se trabajó con una presión de viento de $+26.89 \text{ kg/m}^2$.

El ancho de influencia de un arco central es de 5.30m (dato del proyecto), por lo tanto, la carga de viento en el arco es de $26.89 \text{ kg/m}^2 * 5.30\text{m} = 142.517\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga de viento es de 1.398092kN/m, como el arco cuenta

con dos bridas superiores entonces se asigna la mitad de la carga de viento en cada brida, es decir 0.699046kN/m en cada brida superior y 0.699046kN/m es la carga que se coloca en el componente “COMBINACION DE CARGAS” para este caso de carga.

- La carga viva de techo indicada en la memoria de cálculo del proyecto es de 50 kg/m^2 , como el ancho de influencia es de 5.30m , la carga viva de techo en el arco es de $50\text{ kg/m}^2 * 5.30\text{m} = 265\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga viva de techo es de 2.59965kN/m , como el arco cuenta con dos bridas superiores entonces se asigna la mitad de la carga viva de techo en cada brida, es decir 1.299825kN/m en cada brida superior y 1.299825kN/m es la carga que se coloca en el componente “COMBINACION DE CARGAS” para este caso de carga.
- Para el caso de la carga muerta debida al peso de las viguetas en una primera instancia se la estimará con una carga de 0.2 kN/m sobre cada brida superior del arco, puesto que aún no sabemos el número de segmentos del arco, ya que en función a estos se colocan las viguetas y consecuentemente se puede hallar la carga muerta debido a estas, entonces se realizará 2 optimizaciones para el arco, en la primera se realizará una optimización geométrica y de tamaño con Galápagos, en forma similar al ejemplo N°2, cuyo objetivo es definir la geometría del arco, obteniendo así el número de segmentos de arco, en consecuencia se podrá ubicar las viguetas en el arco con su respectivo ancho de influencia; luego se diseñara las viguetas (ya que en esa instancia se contara con todos los datos necesarios para ello), al final del diseño de las viguetas se conocerá su peso, entonces se calculara con exactitud la carga muerta debida al peso de las viguetas, luego se retornara al diseño del arco y colocaremos la carga muerta debida al peso de las viguetas en el componente “COMBINACION DE CARGAS” y consecutivamente realizamos la segunda optimización para el arco que será la optimización geométrica y de tamaño en simultaneo pero sin tomar en cuenta los parámetros número de segmentos de arco, ancho de arco y flecha de arco.
- En la memoria de cálculo y en los planos del proyecto no se dan datos sobre el peso de los paneles de la cobertura, por lo tanto, asumiremos que se usara el “TR-4” (PRECOR) de espesor de 6mm , cuyo peso es de 5.26 kg/m^2 . Como el ancho de influencia es de 5.30m , la carga muerta debida al peso de los paneles en el arco es de $5.26\text{ kg/m}^2 * 5.30\text{m} = 27.878\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga muerta debida al peso de los paneles es de 0.273483kN/m , como el arco cuenta con dos bridas superiores entonces se

asigna la mitad de la carga viva de techo en cada brida, es decir 0.136742kN/m en cada brida superior y 0.136742kN/m es la carga que se coloca en el componente “COMBINACION DE CARGAS” para este caso de carga.

- Se considero una carga de 8 kg/m^2 que corresponde a otros tipos de carga muerta que puedan presentarse como luminarias, accesorios, etc., la cual tiene el nombre de carga auxiliar. Como el ancho de influencia es de 5.30m, la carga muerta debida a la carga auxiliar en el arco es de $8.00 \text{ kg/m}^2 * 5.30\text{m} = 42.4\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga muerta debida a la carga auxiliar es de 0.415944kN/m, como el arco cuenta con dos bridas superiores entonces se asigna la mitad de la carga auxiliar en cada brida, es decir 0.207972kN/m en cada brida superior y 0.207972kN/m es la carga que se coloca en el componente “COMBINACION DE CARGAS” para este caso de carga.
- En este ejemplo se trabajó con perfiles tubulares de secciones circulares, consecuentemente se usó el script “AISC - TENSION - COMPRESION C”.

➤ **Primera optimización geométrica y de tamaño con Galápagos**

Para realizar la optimización geométrica y de tamaño en simultaneo se usó Galápagos, y se trabajó de manera similar al ejemplo N°2, es así que la función de optimización se muestra en la figura 3.107, cuyas variables son p, d, r1, r2, r3, r4, r5, r6 y r7 que representan el peso, la deflexión máxima, los ratios de diseño de los elementos de la brida superior, los ratios de diseño de las montantes, los ratios de diseño de los elementos de la brida inferior, los ratios de diseño de las diagonales – izquierda, los ratios de diseño de las diagonales – derecha, los ratios de diseño de los enlaces superiores y los ratios de diseño de los enlaces inferiores respectivamente. Galápagos minimizó (óptimizo) el resultado de la función de optimización y dicho resultado mínimo nos indica que se ha elegido el menor peso, la menor deflexión máxima, y los ratios de diseño de los elementos de la armadura más próximos a 1. También cabe indicar que los parámetros que toma Galápagos para minimizar el resultado de la función de optimización son la flecha de arco de viga de celosía, el peralte de arco de viga de celosía, el número de segmentos de arco de viga de celosía, el ancho de arco de viga de celosía y las secciones transversales dentro de listas, una lista por cada

tipo de elemento (brida superior, montantes, brida inferior, diagonales – izquierda, diagonales – derecha, enlaces superiores y enlaces inferiores).

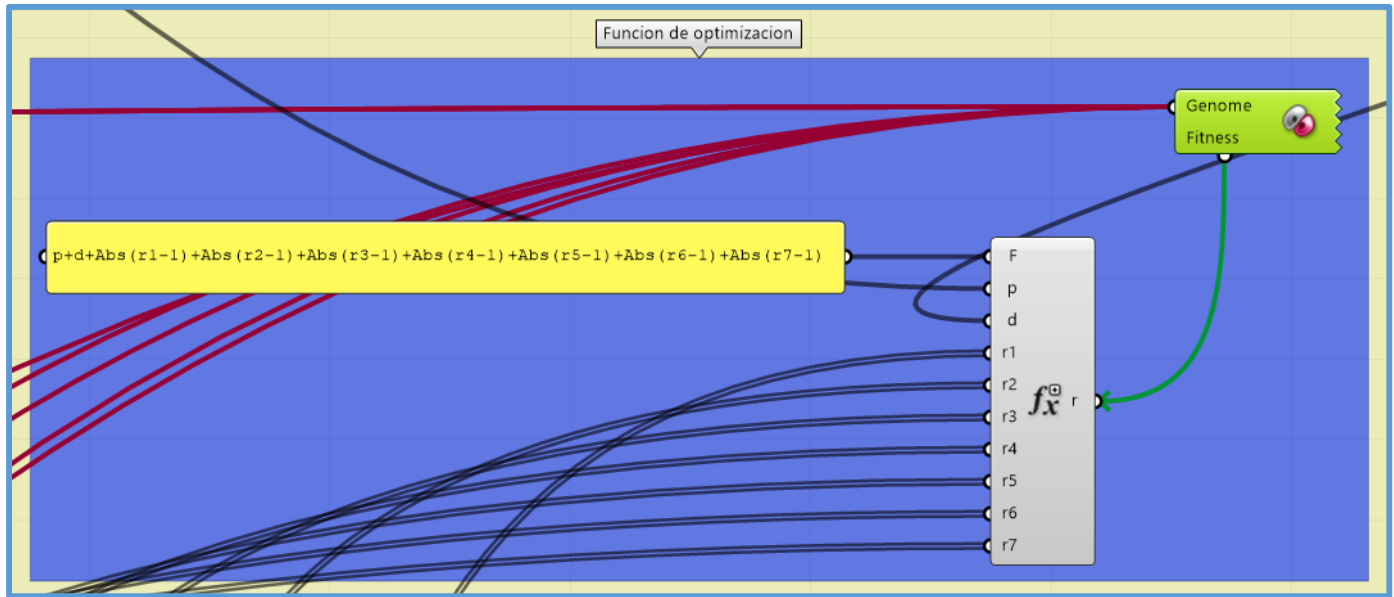


Figura 3.107. Función de optimización de arco de viga de celosía - primera optimización geométrica y de tamaño - ejemplo N°3.

Una vez definida la función de optimización se realizó sus respectivas conexiones, también se realizó las conexiones de Galápagos, luego en el entorno de Galápagos se realizó la optimización en forma similar al ejemplo N°1. En este caso se encontró la solución óptima a los 8 minutos y los resultados de dicha solución son los que se muestran en las siguientes figuras:

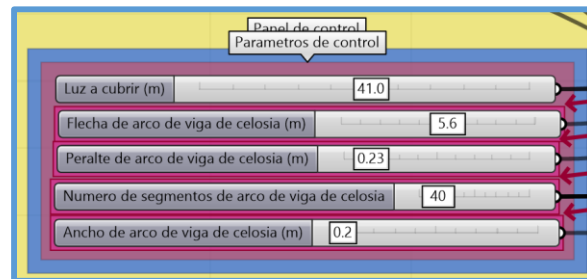


Figura 3.108. Parámetros de control de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.

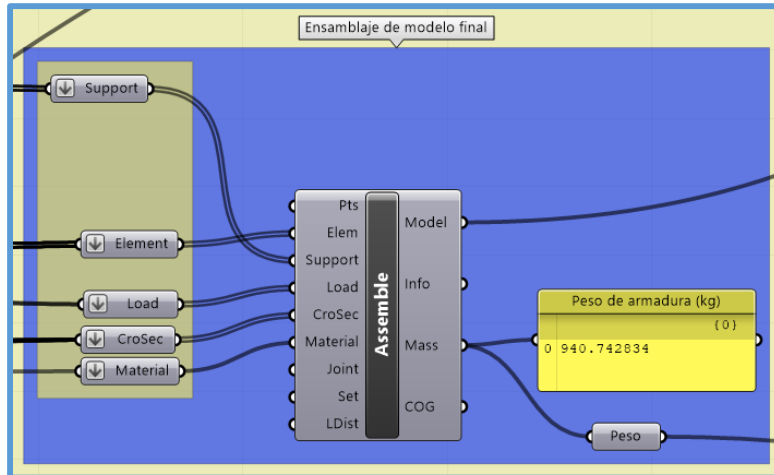


Figura 3.109. Peso de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.

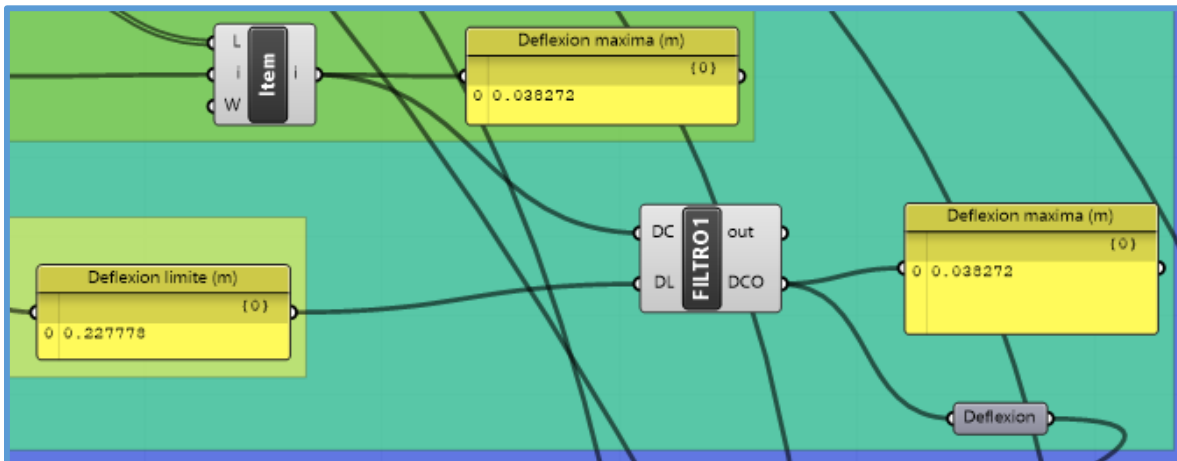


Figura 3.110. Deflexión máxima de arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.

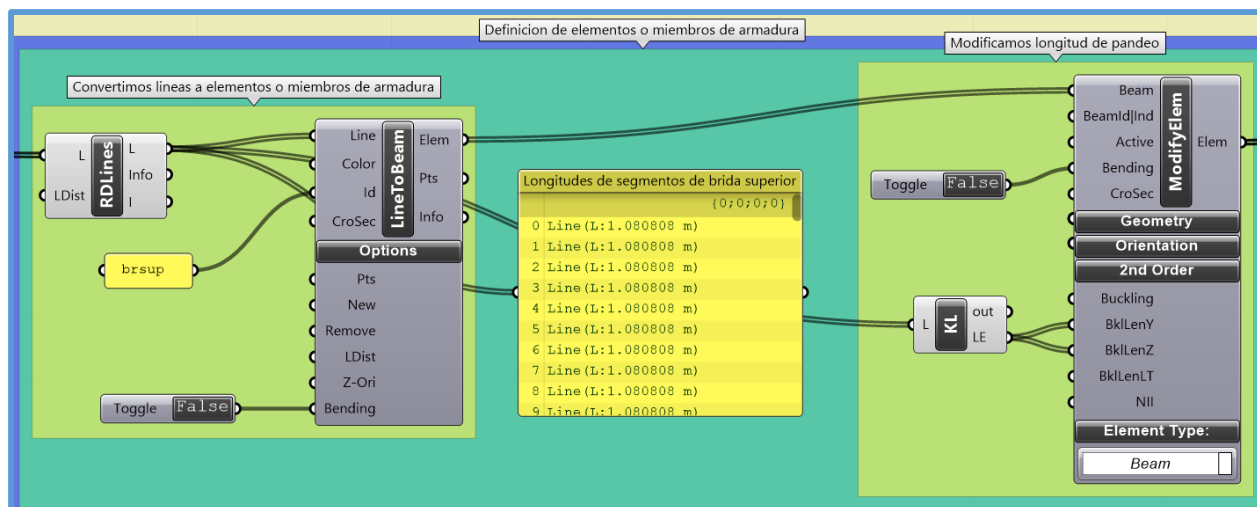


Figura 3.111. Longitudes de segmentos de la brida superior del arco de viga de celosía luego de la primera optimización geométrica y de tamaño - ejemplo N°3.

➤ Optimización de vigueta de cobertura

Para realizar la optimización de la vigueta se tuvo en cuenta lo siguiente:

- Se usó una configuración estructural Warren simple.
- Se diseñó el caso de una vigueta intermedia, para las viguetas en los extremos se adoptaron los resultados de diseño de las viguetas intermedias.
- El ancho de influencia de la vigueta (intermedia) es 1.08m (ver figura 3.111), haciendo una aproximación se adoptó 1.10m.
- La carga de viento sobre la vigueta es de $26.89 \text{ kg/m}^2 \cdot 1.10\text{m} = 29.579\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga de viento es de 0.29017kN/m.
- La carga viva de techo sobre la vigueta es de $50 \text{ kg/m}^2 \cdot 1.10\text{m} = 55\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga de viento es de 0.53955kN/m.
- La carga muerta debida al peso de los paneles, sobre la vigueta es $5.26 \text{ kg/m}^2 \cdot 1.10\text{m} = 5.786\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga muerta debida al peso de los paneles es de 0.056761kN/m.
- La carga muerta debida a la carga auxiliar, sobre la vigueta es de $8.00 \text{ kg/m}^2 \cdot 1.10\text{m} = 8.8\text{kg/m}$ (en coordenadas locales), convirtiendo a kN/m tenemos que la carga muerta debida a la carga auxiliar es de 0.086328kN/m.
- Como el ancho del arco de viga de celosía es de 0.2m (figura 3.108), y su ancho de influencia es de 5.30m, entonces la luz de la vigueta es de $5.30\text{m} - 0.20\text{m} = 5.10\text{m}$.

Se ha elegido un perfil tubular circular de 1.75in de diámetro y 1.5mm de espesor para todos los elementos de la vigueta, entonces no se ha realizado optimización de tamaño para la vigueta (no es recomendable en este caso realizar la optimización de tamaño, puesto que las cargas en la vigueta son pequeñas y los perfiles tubulares tienen buena resistencia, entonces lo más recomendable en este caso es trabajar directamente con las secciones más pequeñas y realizar la optimización geométrica).

Para la vigueta luego de realizada la optimización geométrica se realizó la optimización topológica, no se obtuvo resultados favorables en esta última optimización (no aplica la optimización topológica), para más detalles revisar los archivos en Grasshopper.

En la figura 3.112 se muestra la función de optimización para la optimización geométrica de la vigueta.

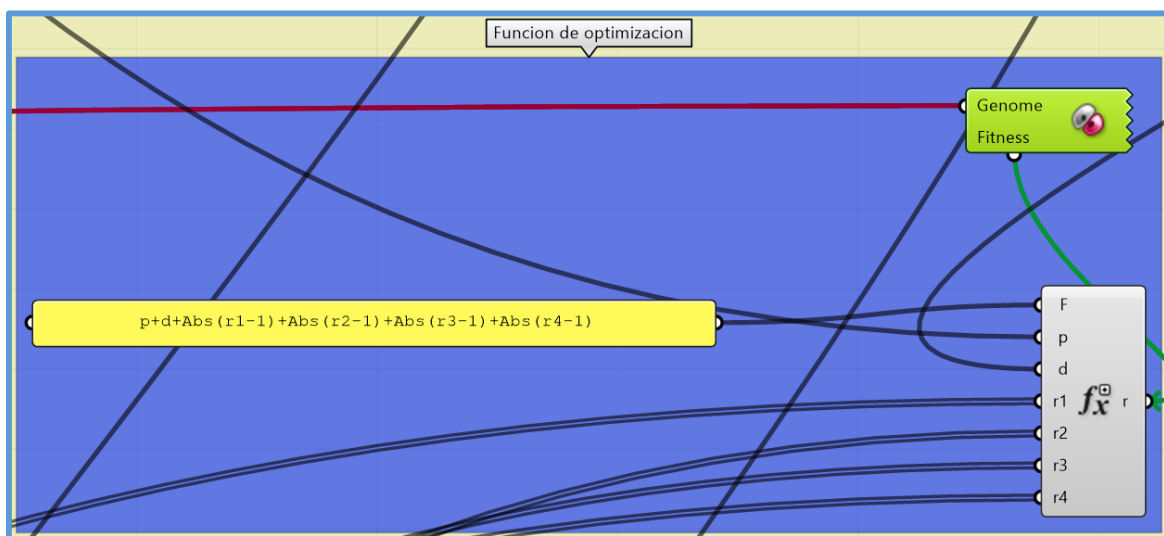


Figura 3.112. Función de optimización - optimización geométrica de vigueta - ejemplo N°3.

Al igual que en el ejemplo N°2 las variables p y d de la función de optimización representan el peso y la deflexión máxima, si bien no se realizara la optimización de tamaño es necesario tener las variables r1, r2, r3 y r4 en la función de optimización puesto que esto hará que en las soluciones que vaya encontrando Galápagos no se generen elementos con esbeltez efectiva superior a la máxima establecida en la AISC 360-10. Los parámetros que tomo Galápagos para minimizar el resultado de la función de optimización fueron el peralte y el número de segmentos. En este caso

Galápagos encontró la solución óptima a los 5 minutos, es así que se muestran los parámetros de control de dicha solución en la siguiente figura:

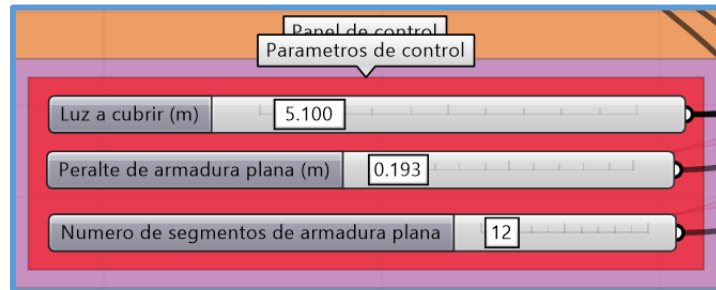


Figura 3.113. Parámetros de control luego de la optimización geométrica de vigueta - ejemplo N°3.

Como el peralte encontrado por Galápagos fue de 0.193m, por motivos constructivos se redondeo a 0.20m, al hacer esto se recalcula todo el modelo, el modelo recalculado se define como la armadura óptima final, solución óptima final o vigueta óptima final, cuyos resultados de diseño se muestran en las siguientes figuras:

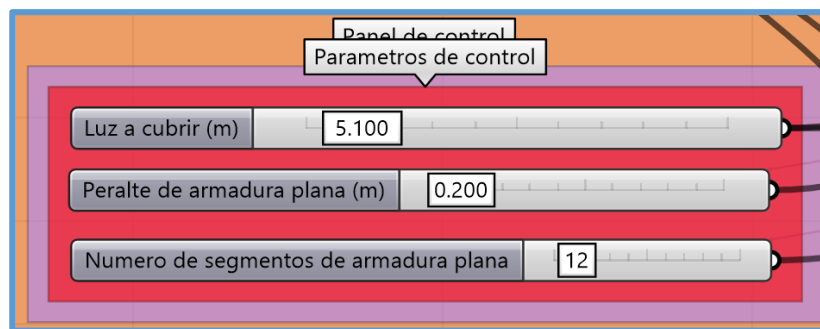


Figura 3.114. Parámetros de control - vigueta óptima final - ejemplo N°3.

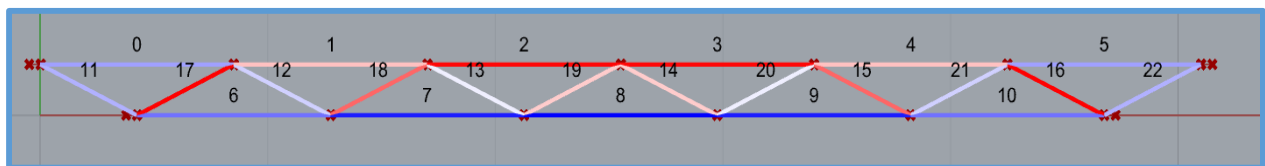


Figura 3.115. Elementos a tensión y/o compresión, con su respectivo número identificador asignado por "Karamba" - vigueta óptima final - ejemplo N°3.

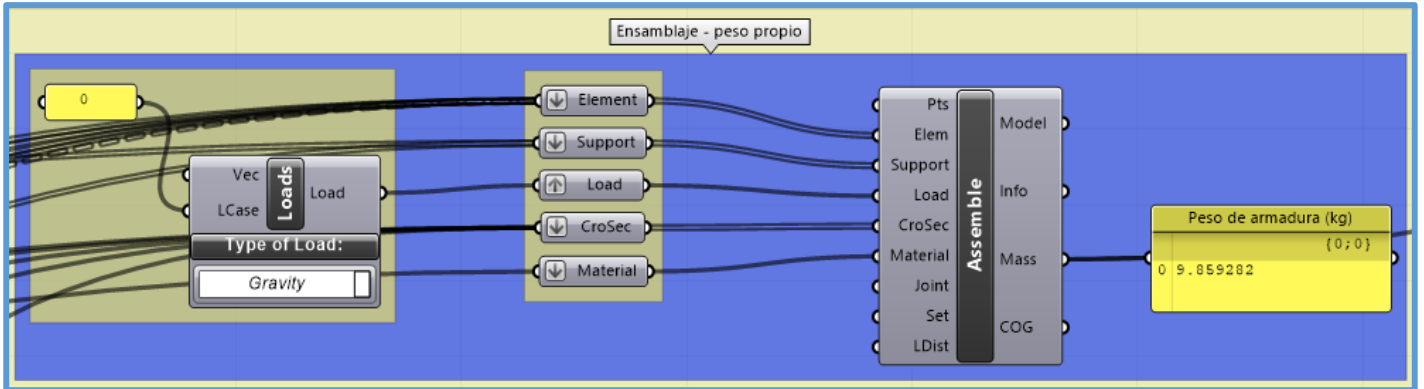


Figura 3.116. Peso de vigueta óptima final - ejemplo N°3.

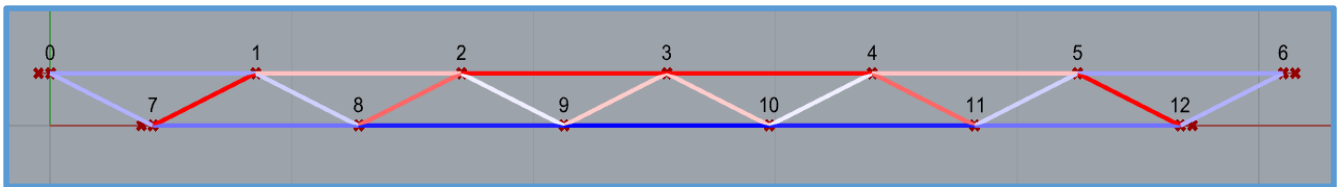


Figura 3.117. Nodos de vigueta óptima final - ejemplo N°3.

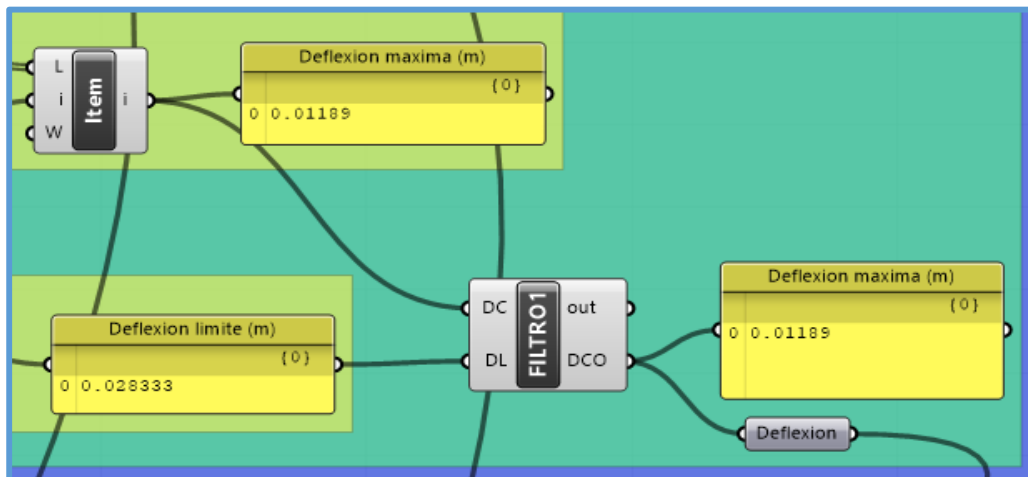


Figura 3.118. Deflexión máxima de vigueta óptima final - ejemplo N°3.

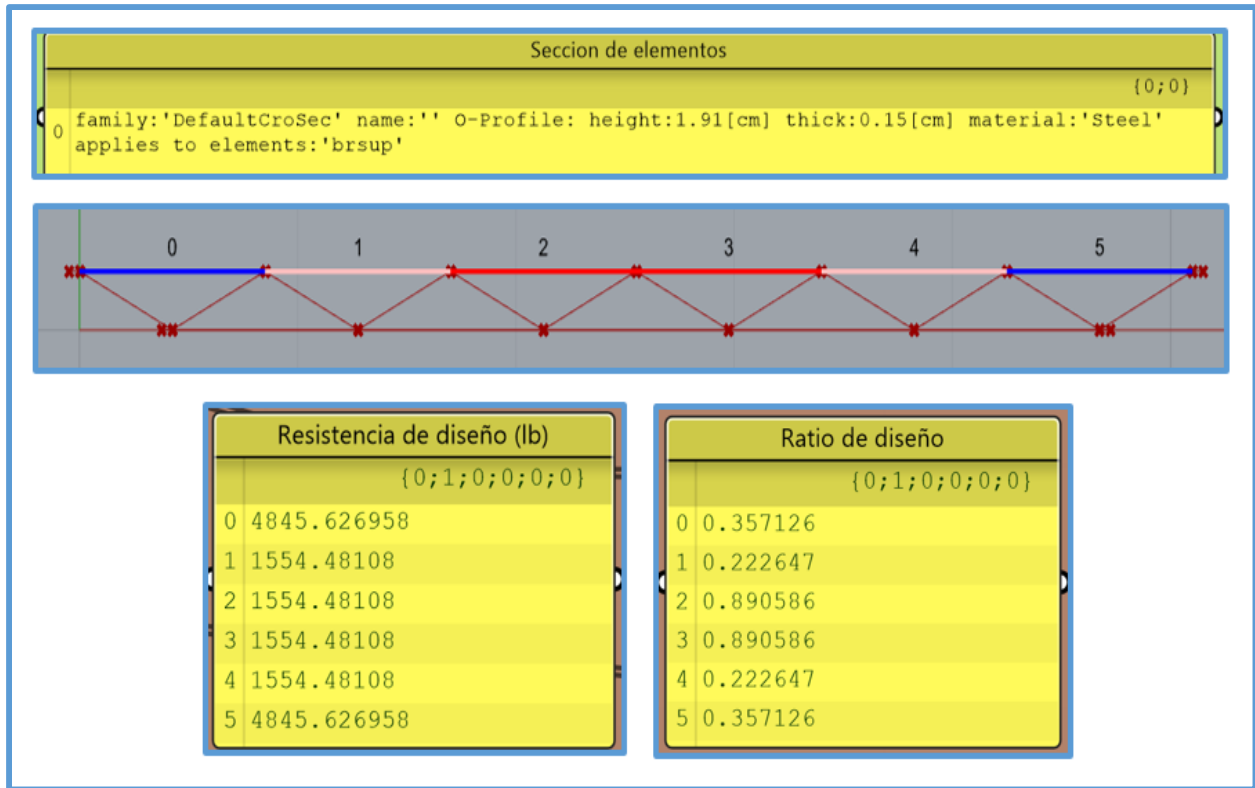


Figura 3.119. Resultados de diseño de la brida superior - vigueta óptima final - ejemplo N°3.

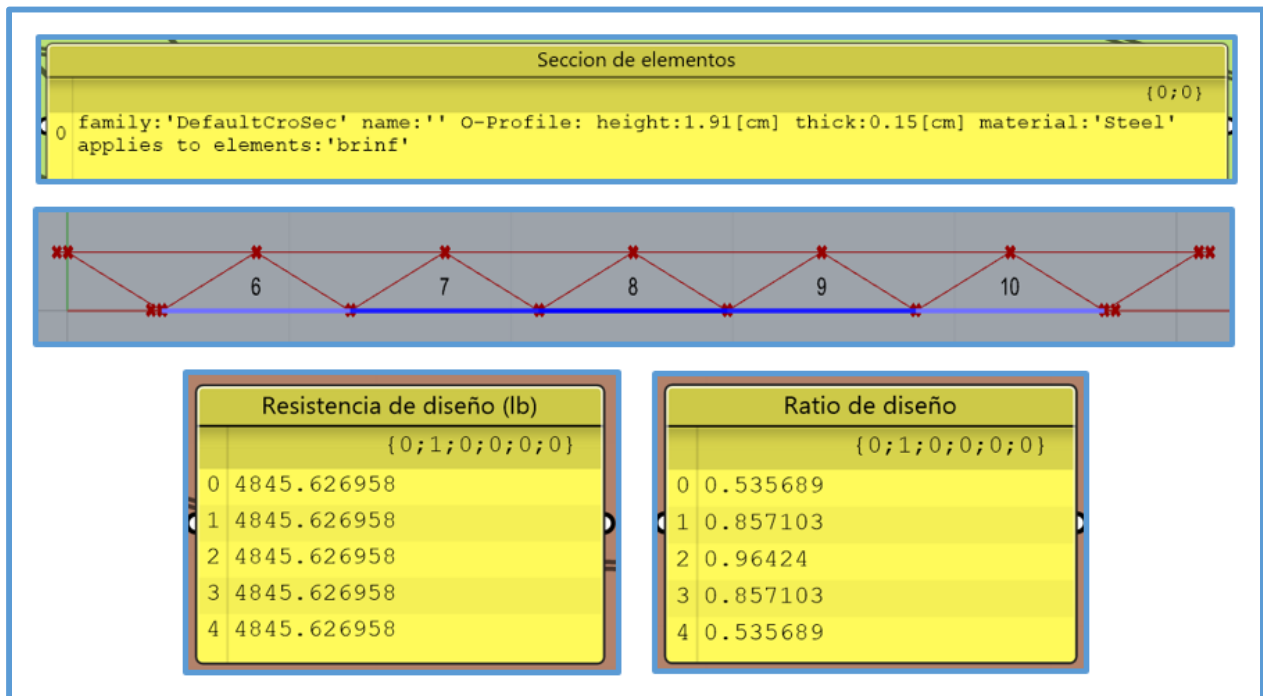


Figura 3.120. Resultados de diseño de la brida inferior - vigueta óptima final - ejemplo N°3.

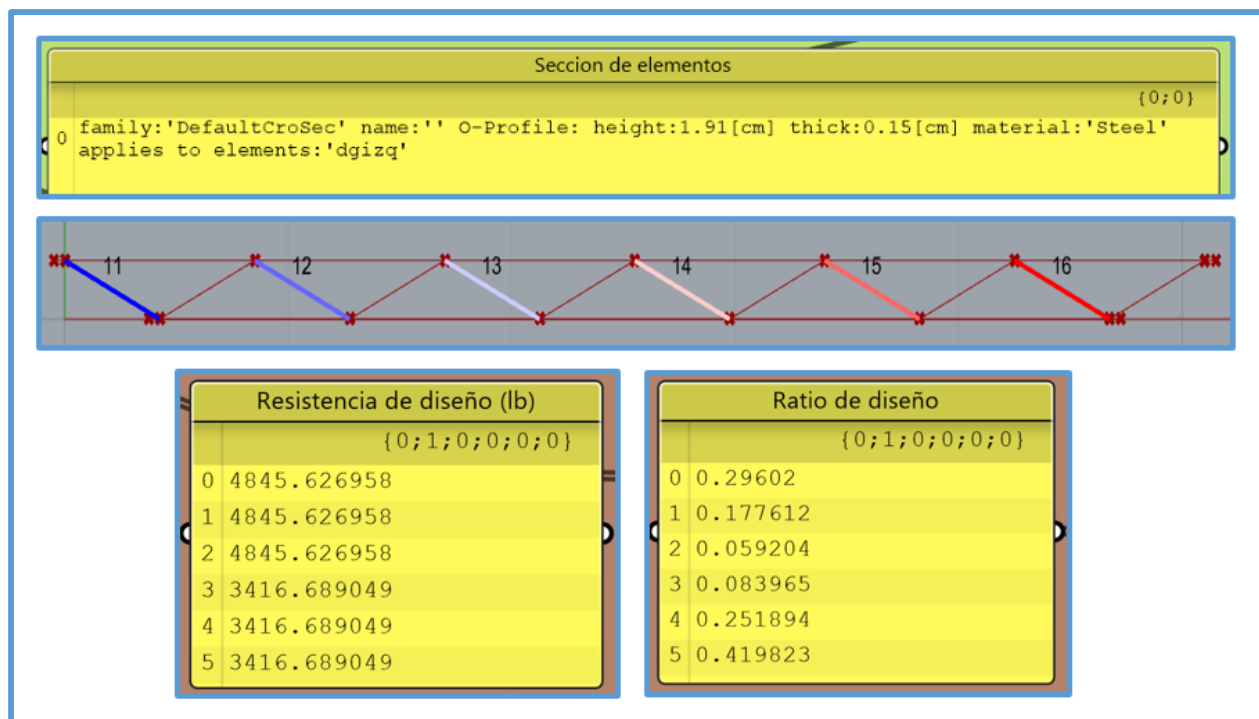


Figura 3.121. Resultados de diseño de las diagonales - izquierda - vigueta óptima final - ejemplo N°3.

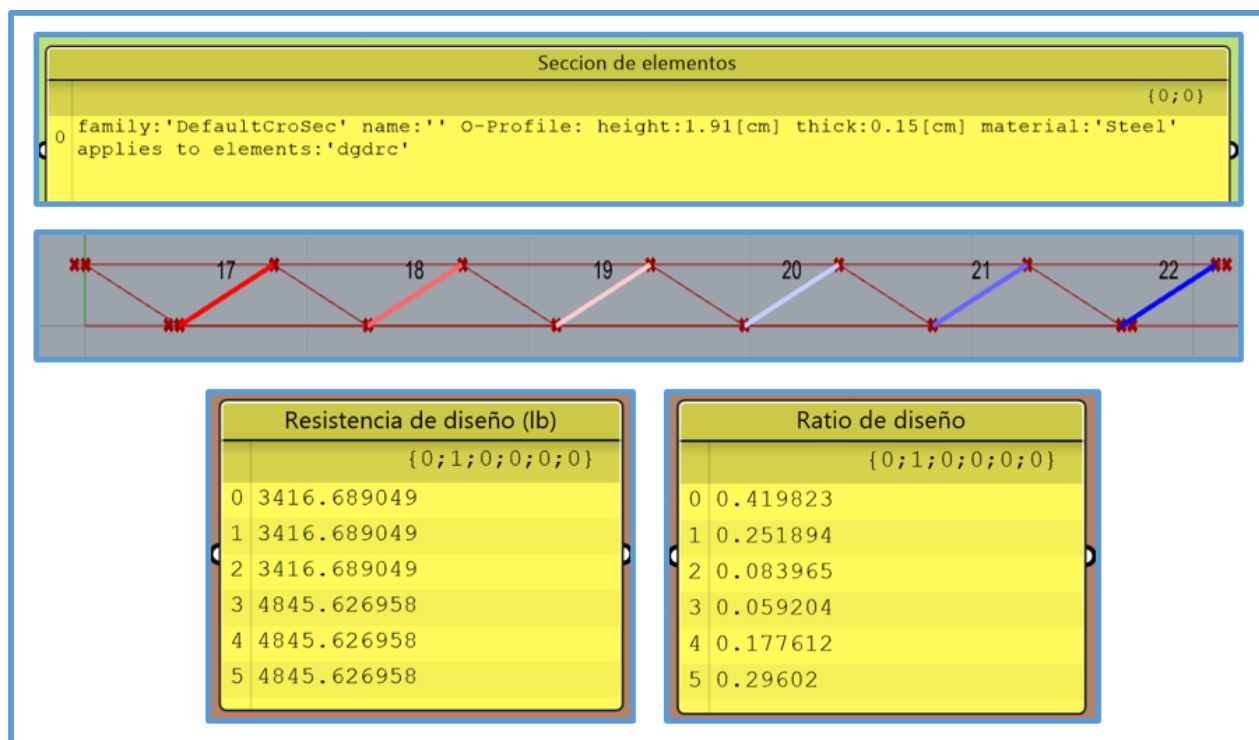


Figura 3.122. Resultados de diseño de las diagonales - derecha - vigueta óptima final - ejemplo N°3.

➤ Segunda optimización geométrica y de tamaño con Galápagos

Luego de diseñada la vigueta se procedió a realizar la segunda optimización geométrica y de tamaño en el arco, de la figura 3.116 sabemos que el peso de una vigueta es de 9.859282kg, como el número de segmentos de la brida superior del arco es de 40, entonces habrá 41 viguetas, la carga muerta debida al peso de las viguetas en el arco es de $9.859282\text{kg} \cdot 41 = 404.230562\text{kg}$, convirtiendo a carga distribuida se tiene que la carga muerta debida al peso de las viguetas es de $404.230562\text{kg}/43.203164\text{m} = 9.356504\text{kg/m}$, luego convirtiendo a kN/m tenemos 0.091787 Kn/m , a cada brida superior del arco le corresponde la mitad de esa carga, es decir 0.045894 Kn/m (carga que se coloca en el componente “COMBINACION DE CARGAS”).

Una vez colocadas las cargas actuantes ya exactas sobre el arco se realizó algunas modificaciones en la función de optimización, la cual se aprecia en la figura 3.123, en esta ocasión no se cuenta con las variables r_6 y r_7 que corresponden a los ratios de diseño de los enlaces superiores y los ratios de diseño de los enlaces inferiores respectivamente, puesto que dichos elementos no cuentan con fuerzas actuantes, entonces basta con asignarles la sección de menor dimensión de sus listas de secciones. También se han desactivado los parámetros número de segmentos de arco, ancho de arco y flecha de arco, puesto que alguna variación en estos alterara el número de viguetas (consecuentemente el peso de las viguetas), la luz de las viguetas y el ancho de influencia de las viguetas respectivamente, lo cual ya se ha definido en pasos anteriores y no se debe alterar.

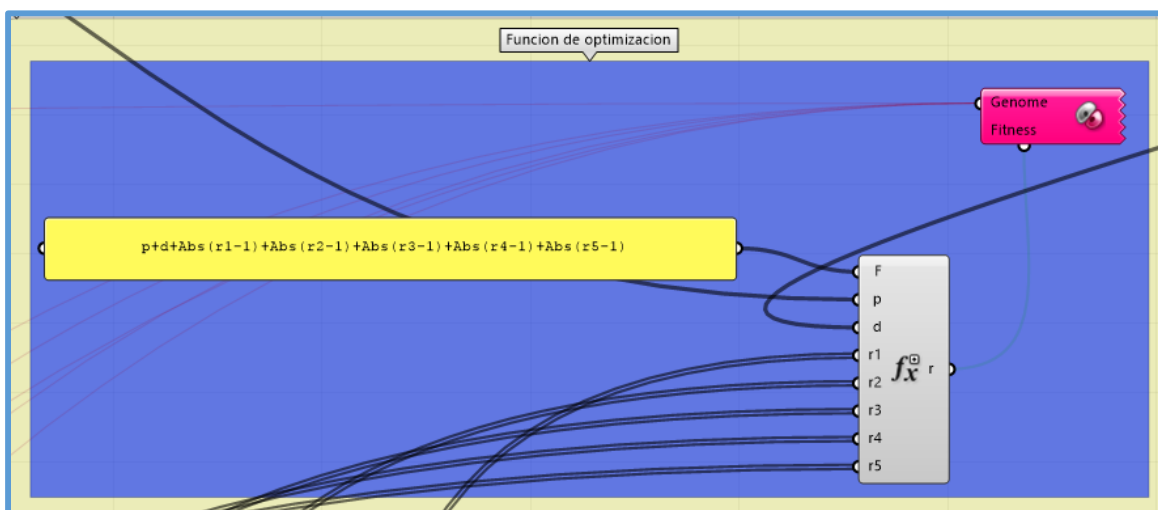


Figura 3.123. Función de optimización de arco de viga de celosía - segunda optimización geométrica y de tamaño - ejemplo N°3.

Una vez definida la función de optimización se realizó sus respectivas conexiones, también se realizó las conexiones de Galápagos, luego en el entorno de Galápagos se realizó la optimización en forma similar al ejemplo N°1. En este caso Galápagos encontró la solución óptima a los 5 minutos, es así que se muestran los parámetros de control de dicha solución en la siguiente figura:

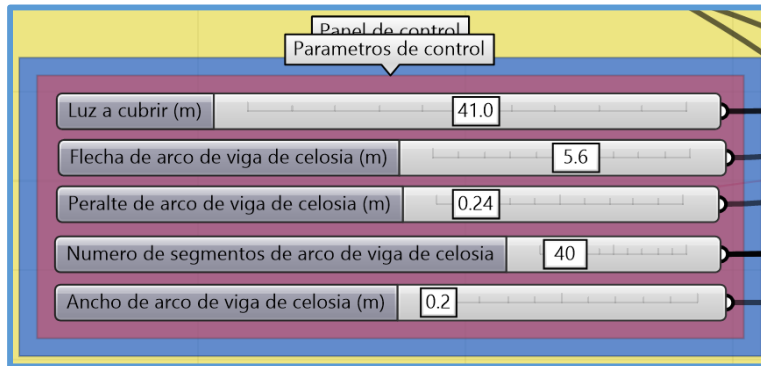


Figura 3.124. Parámetros de control de arco de viga de celosía luego de la segunda optimización geométrica y de tamaño - ejemplo N°3.

Como el peralte encontrado por Galápagos es fue de 0.24m, por motivos constructivos se lo redondeo a 0.25m, al hacer esto se recalcula todo el modelo, luego se aplicó la optimización topológica a las diagonales - izquierda y a las diagonales - derecha en simultaneo, se obtuvo la solución óptima al trabajar con un valor de 0.97 en el “Target mass ratio” del componente “BESOBeam”, como se muestra en la figura 3.125, en la solución óptima se han eliminado 2 diagonales derecha (figura 3.126), para más detalles se recomienda revisar los archivos en Grasshopper.

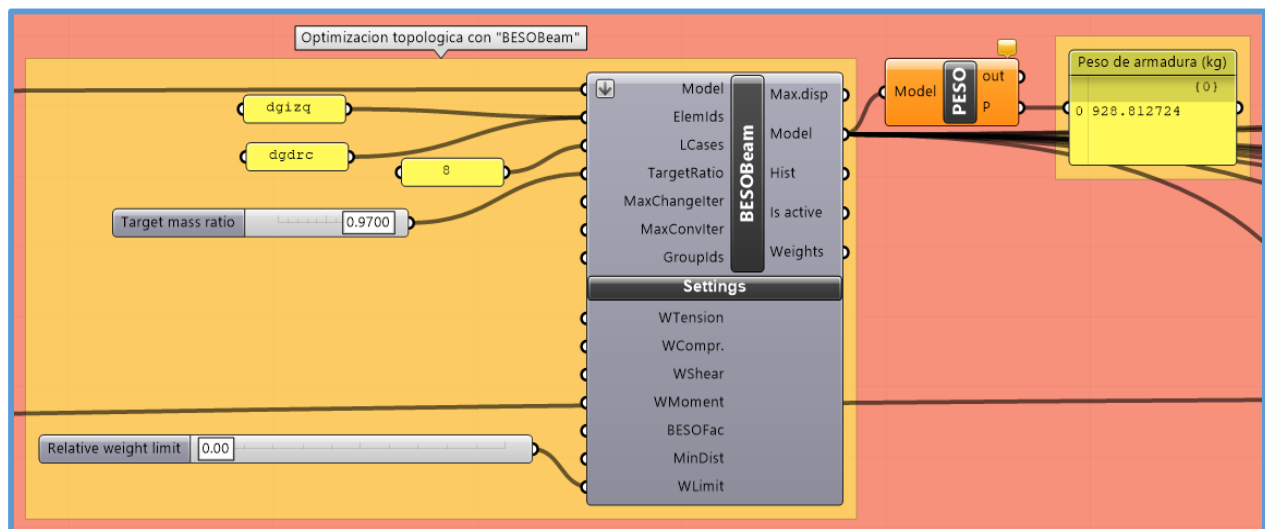


Figura 3.125. Optimización topológica con “BESOBeam” de arco de viga de celosía - ejemplo N°3.

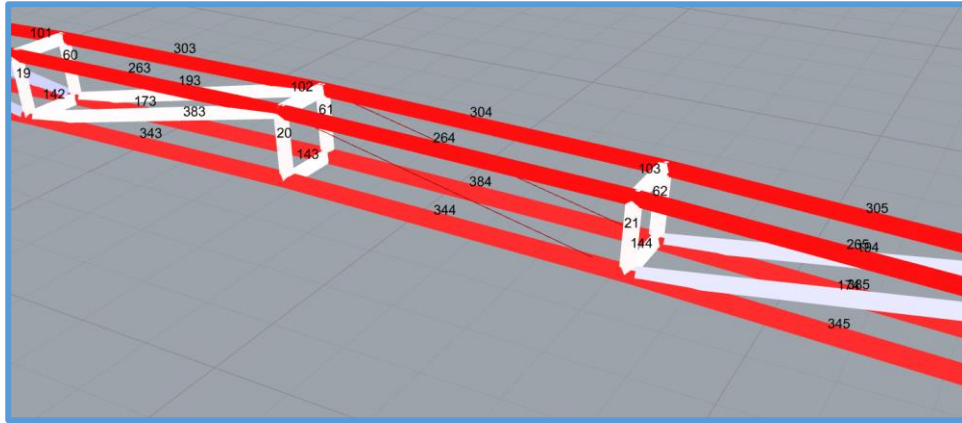


Figura 3.126. Diagonales - derecha de arco de viga de celosía eliminadas con la optimización topológica - ejemplo N°3.

Concluida la optimización topológica, se procedió a realizar unos reajustes finales en el diseño, puesto que en las montantes, diagonales izquierda y derecha, y enlaces superiores e inferiores, los ratios de diseño son muy bajos a pesar de que las secciones en estos elementos son las más pequeñas de sus respectivas listas de secciones, entonces se les asigno una sección de 1.5in de diámetro y 2mm de espesor (de menor dimensión que la sección de menor dimensión en sus respectivas listas de secciones), quedando así definida la armadura óptima final o solución óptima final para el arco de viga de celosía y sus resultados de diseño se muestran en las siguientes figuras:

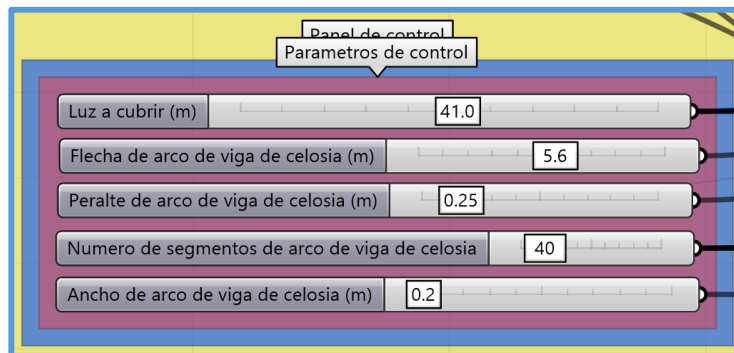


Figura 3.127. Parámetros de control - armadura óptima final - ejemplo N°3.

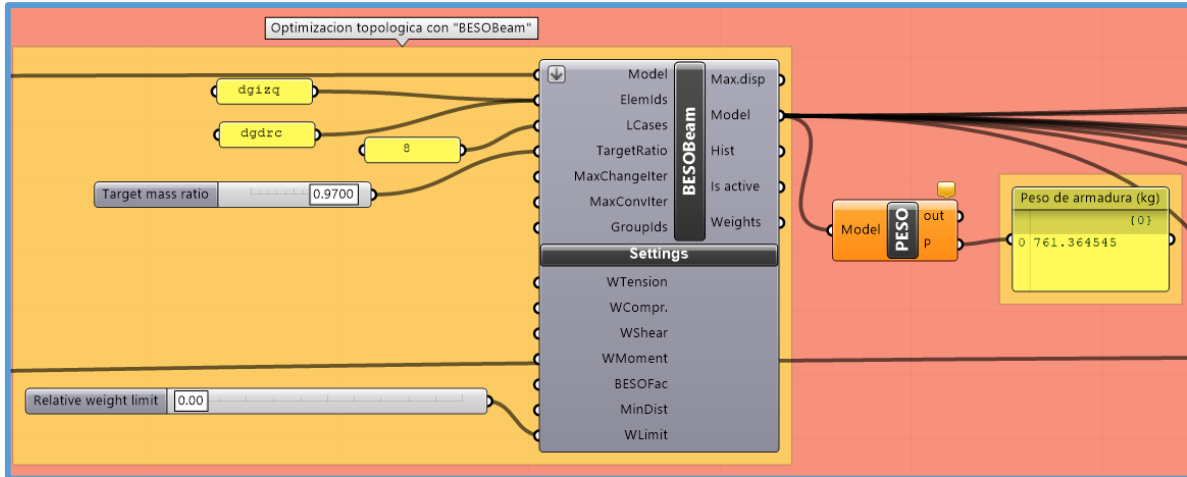


Figura 3.128. Optimización topológica con “BESOBeam” - peso - armadura óptima final - ejemplo N°3.

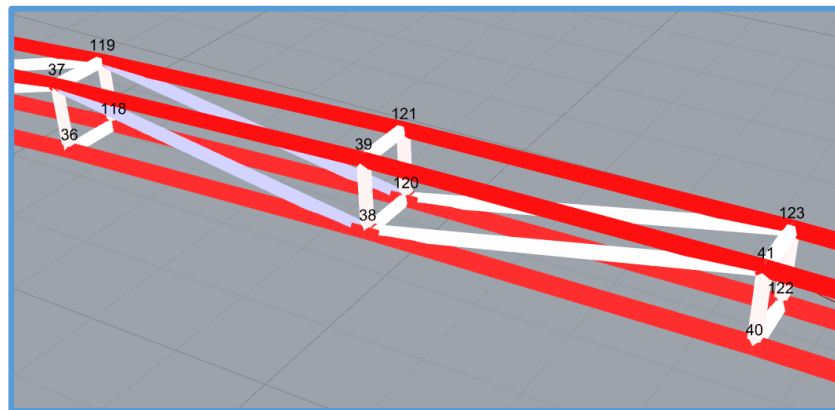


Figura 3.129. Nodos de armadura óptima final - ejemplo N°3.

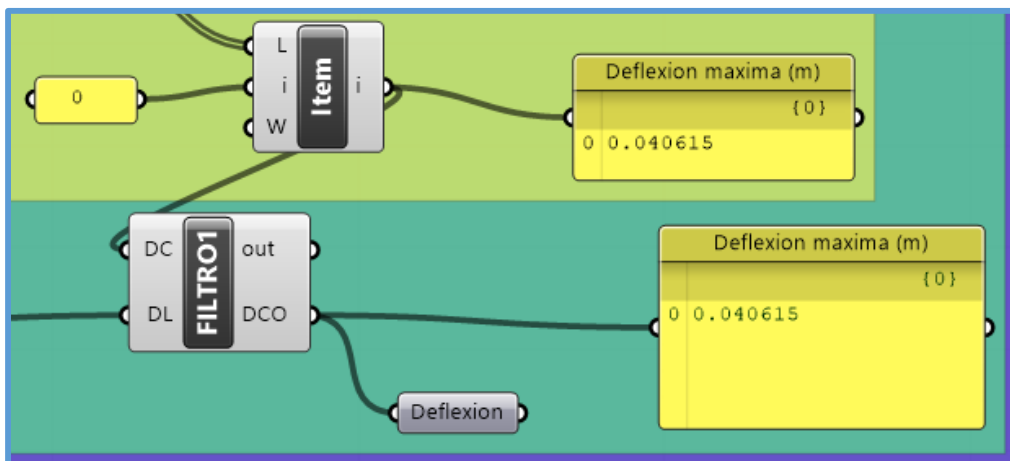


Figura 3.130. Deflexión máxima - armadura óptima final - ejemplo N°3.

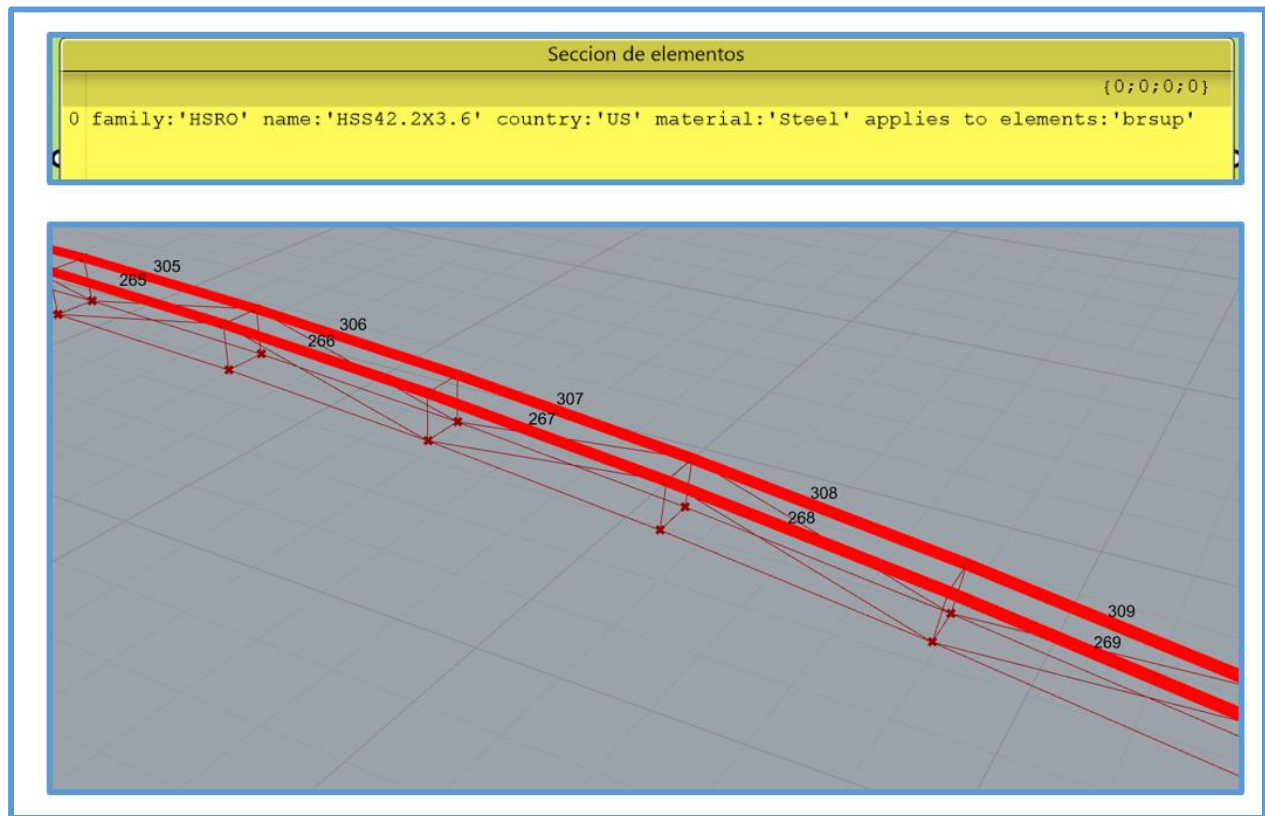


Figura 3.131. Sección de la brida superior - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)		Resistencia de diseño (lb)	
	{0;0;0;0;0}	23	16194.774741	47	16194.77609
0	16194.77902	24	16194.774812	48	16194.777803
1	16194.777614	25	16194.774903	49	16194.776103
2	16194.776308	26	16194.775015	50	16194.775504
3	16194.777078	27	16194.77515	51	16194.770651
4	16194.778182	28	16194.777464	52	16194.776186
5	16194.776627	29	16194.775352	53	16194.775343
6	16194.776693	30	16194.775532	54	16194.774984
7	16194.77609	31	16194.775741	55	16194.77483
8	16194.777803	32	16194.775981	56	16194.774962
9	16194.776103	33	16194.776257	57	16194.774924
10	16194.775504	34	16194.776938	58	16194.774523
11	16194.770651	35	16194.77634	59	16194.774733
12	16194.776186	36	16194.776656	60	16194.774194
13	16194.775343	37	16194.777017	61	16194.774019
14	16194.774984	38	16194.777431	62	16194.771668
15	16194.77483	39	16194.779632	63	16194.774741
16	16194.774962	40	16194.77902	64	16194.774812
17	16194.774924	41	16194.777614	65	16194.774903
18	16194.774523	42	16194.776308	66	16194.775015
19	16194.774733	43	16194.777078	67	16194.77515
20	16194.774194	44	16194.778182	68	16194.777464
21	16194.774019	45	16194.776627	69	16194.775352
22	16194.771668	46	16194.776693	70	16194.775532
		Resistencia de diseño (lb)			
		71	16194.775741		
		72	16194.775981		
		73	16194.776257		
		74	16194.776938		
		75	16194.77634		
		76	16194.776656		
		77	16194.777017		
		78	16194.777431		
		79	16194.779632		

Figura 3.132. Resistencia de diseño de los elementos de la brida superior - armadura óptima final - ejemplo N°3.

Ratio de diseño	
	{0;0;0;0;0}
0	0.792161
1	0.792027
2	0.819034
3	0.818809
4	0.84522
5	0.844937
6	0.870166
7	0.869855
8	0.893278
9	0.892968
10	0.913949
11	0.913663
12	0.931585
13	0.931342
14	0.945646
15	0.945462
16	0.955681
17	0.955566
18	0.961359
19	0.96132
20	0.962502
21	0.962541
22	0.959096

Ratio de diseño	
23	0.95921
24	0.951292
25	0.951475
26	0.939394
27	0.939634
28	0.923833
29	0.924116
30	0.905135
31	0.905441
32	0.883884
33	0.88419
34	0.860684
35	0.860963
36	0.836131
37	0.836352
38	0.810782
39	0.810913
40	0.792161
41	0.792027
42	0.819034
43	0.818809
44	0.84522
45	0.844937
46	0.870166

Ratio de diseño	
47	0.869855
48	0.893278
49	0.892968
50	0.913949
51	0.913663
52	0.931585
53	0.931342
54	0.945646
55	0.945462
56	0.955681
57	0.955566
58	0.961359
59	0.96132
60	0.962502
61	0.962541
62	0.959096
63	0.95921
64	0.951292
65	0.951475
66	0.939394
67	0.939634
68	0.923833
69	0.924116
70	0.905135

Ratio de diseño	
71	0.905441
72	0.883884
73	0.88419
74	0.860684
75	0.860963
76	0.836131
77	0.836352
78	0.810782
79	0.810913

Figura 3.133. Ratios de diseño de los elementos de la brida superior - armadura óptima final - ejemplo N°3.

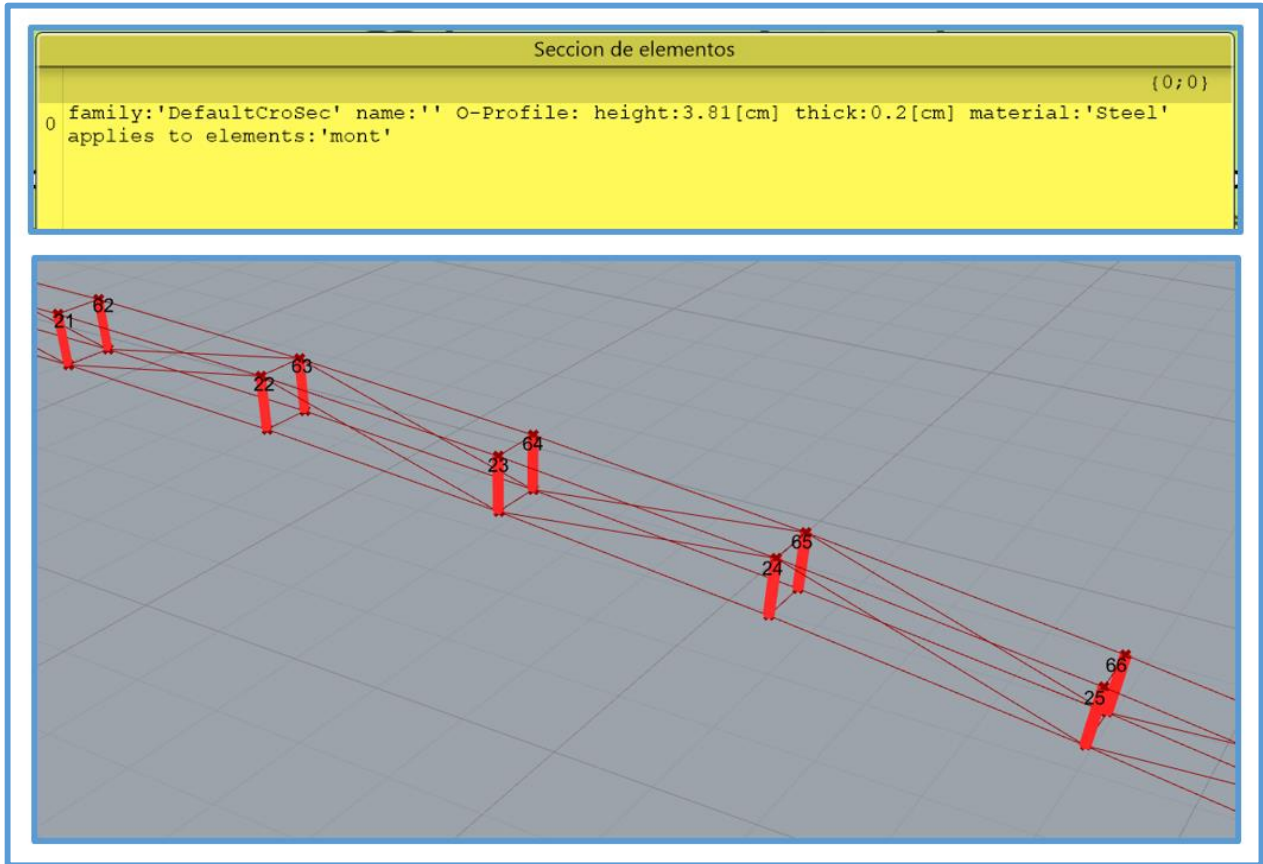


Figura 3.134. Sección de montantes - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)		Resistencia de diseño (lb)	
	{0;0;0}	23	13058.379915	47	13019.751844
0	12981.242293	24	13056.8858	48	13025.150361
1	12988.274256	25	13054.969757	49	13030.20626
2	12995.09501	26	13052.635645	50	13034.90476
3	13001.677151	27	13049.888332	51	13039.232844
4	13007.995563	28	13046.733778	52	13043.179135
5	13014.027366	29	13043.179136	53	13046.733781
6	13019.751844	30	13039.232845	54	13049.888334
7	13025.150361	31	13034.904761	55	13052.635647
8	13030.20626	32	13030.206259	56	13054.969758
9	13034.90476	33	13025.15036	57	13056.885801
10	13039.232844	34	13019.751844	58	13058.379916
11	13043.179135	35	13014.027366	59	13059.449179
12	13046.733781	36	13007.995563	60	13060.091543
13	13049.888334	37	13001.67715	61	13060.305798
14	13052.635647	38	12995.095009	62	13060.091544
15	13054.969758	39	12988.274255	63	13059.449179
16	13056.885801	40	12981.242293	64	13058.379915
17	13058.379916	41	12981.242293	65	13056.8858
18	13059.449179	42	12988.274256	66	13054.969757
19	13060.091543	43	12995.09501	67	13052.635645
20	13060.305798	44	13001.677151	68	13049.888332
21	13060.091544	45	13007.995563	69	13046.733778
22	13059.449179	46	13014.027366	70	13043.179136
		Resistencia de diseño (lb)			
		71	13039.232845		
		72	13034.904761		
		73	13030.206259		
		74	13025.15036		
		75	13019.751844		
		76	13014.027366		
		77	13007.995563		
		78	13001.67715		
		79	12995.095009		
		80	12988.274255		
		81	12981.242293		

Figura 3.135. Resistencia de diseño de las montantes - armadura óptima final - ejemplo N°3.

Ratio de diseño	
	{0;0;0}
0	0
1	0.033833
2	0.033277
3	0.032932
4	0.032353
5	0.032057
6	0.031466
7	0.031226
8	0.030634
9	0.030458
10	0.029878
11	0.029772
12	0.029218
13	0.029188
14	0.028672
15	0.028722
16	0.028257
17	0.02839
18	0.027986
19	0.028201
20	0.027869
21	0.028162
22	0.027908

Ratio de diseño	
23	0.028274
24	0.028101
25	0.02853
26	0.02844
27	0.028923
28	0.028914
29	0.029437
30	0.029506
31	0.030057
32	0.030197
33	0.030763
34	0.030969
35	0.031537
36	0.031801
37	0.032359
38	0.032673
39	0.033211
40	0
41	0
42	0.033833
43	0.033277
44	0.032932
45	0.032353
46	0.032057

Ratio de diseño	
47	0.031466
48	0.031226
49	0.030634
50	0.030458
51	0.029878
52	0.029772
53	0.029218
54	0.029188
55	0.028672
56	0.028722
57	0.028257
58	0.02839
59	0.027986
60	0.028201
61	0.027869
62	0.028162
63	0.027908
64	0.028274
65	0.028101
66	0.02853
67	0.02844
68	0.028923
69	0.028914
70	0.029437

Ratio de diseño	
71	0.029506
72	0.030057
73	0.030197
74	0.030763
75	0.030969
76	0.031537
77	0.031801
78	0.032359
79	0.032673
80	0.033211
81	0

Figura 3.136. Ratios de diseño de las montantes - armadura óptima final - ejemplo N°3.

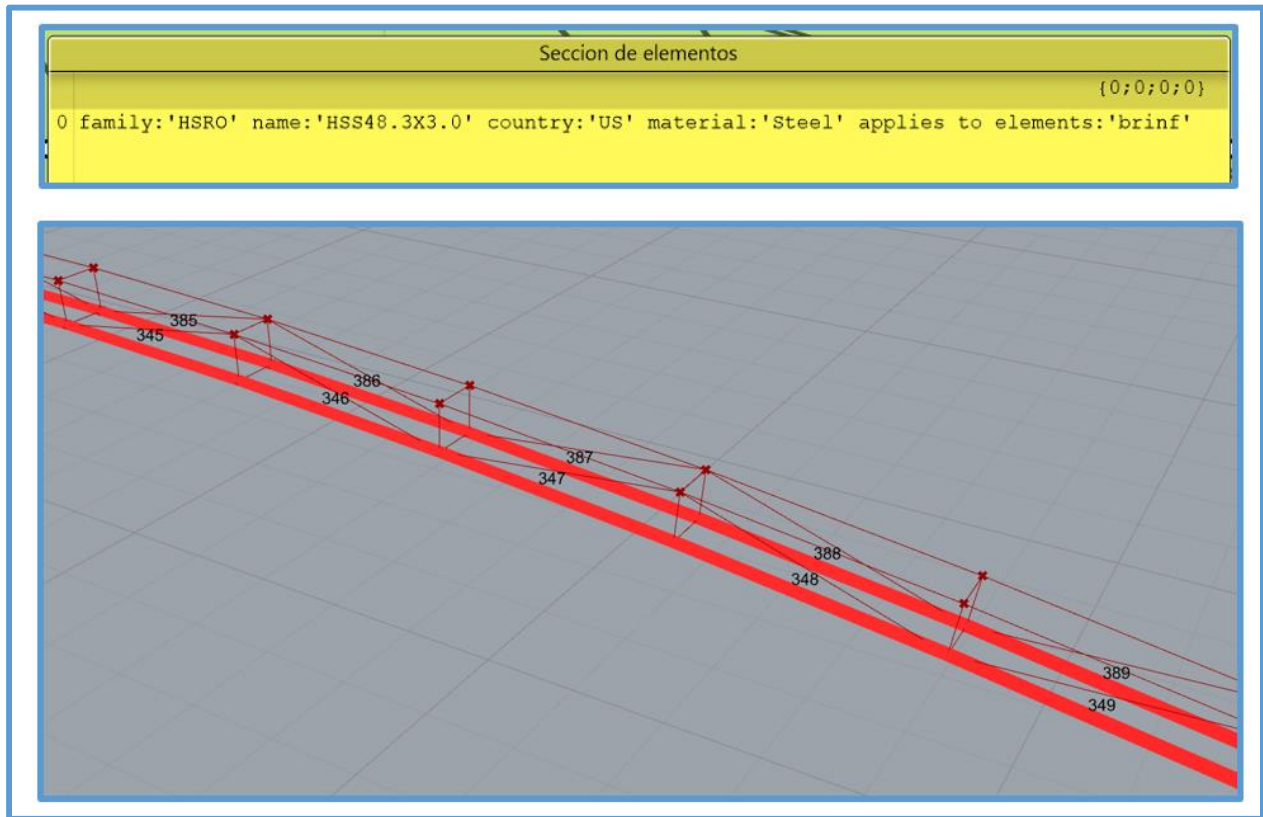


Figura 3.137. Sección de la brida inferior - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)		Resistencia de diseño (lb)	
	{0;0;0;0;0}	23	17954.331782	47	17954.331819
0	17954.332637	24	17954.331779	48	17954.33323
1	17954.331798	25	17954.331778	49	17954.332122
2	17954.331019	26	17954.331779	50	17954.331817
3	17954.33182	27	17954.331783	51	17954.328043
4	17954.332792	28	17954.332735	52	17954.332573
5	17954.331863	29	17954.331699	53	17954.33203
6	17954.332556	30	17954.331699	54	17954.33184
7	17954.331819	31	17954.331708	55	17954.331793
8	17954.33323	32	17954.331728	56	17954.331981
9	17954.332122	33	17954.331764	57	17954.331716
10	17954.331817	34	17954.33226	58	17954.331701
11	17954.328043	35	17954.331467	59	17954.331869
12	17954.332573	36	17954.331504	60	17954.331419
13	17954.33203	37	17954.331566	61	17954.331223
14	17954.33184	38	17954.331659	62	17954.330456
15	17954.331793	39	17954.332932	63	17954.331782
16	17954.331981	40	17954.332637	64	17954.331779
17	17954.331716	41	17954.331798	65	17954.331778
18	17954.331701	42	17954.331019	66	17954.331779
19	17954.331869	43	17954.33182	67	17954.331783
20	17954.331419	44	17954.332792	68	17954.332735
21	17954.331223	45	17954.331863	69	17954.331699
22	17954.330456	46	17954.332556	70	17954.331699
Resistencia de diseño (lb)		71	17954.331708		
		72	17954.331728		
		73	17954.331764		
		74	17954.33226		
		75	17954.331467		
		76	17954.331504		
		77	17954.331566		
		78	17954.331659		
		79	17954.332932		

Figura 3.138. Resistencia de diseño de los elementos de la brida inferior - armadura óptima final - ejemplo N°3.

Ratio de diseño	
	{0;0;0;0;0}
0	0.927824
1	0.903176
2	0.903279
3	0.878927
4	0.879103
5	0.855558
6	0.855777
7	0.833591
8	0.833828
9	0.813575
10	0.813808
11	0.796061
12	0.79627
13	0.781566
14	0.781736
15	0.770548
16	0.770667
17	0.763363
18	0.763425
19	0.760245
20	0.760245
21	0.761284
22	0.761222

Ratio de diseño	
23	0.766415
24	0.766296
25	0.77543
26	0.775261
27	0.787992
28	0.787785
29	0.803661
30	0.803431
31	0.82193
32	0.821695
33	0.842253
34	0.842037
35	0.864084
36	0.863911
37	0.886894
38	0.886793
39	0.910199
40	0.927824
41	0.903176
42	0.903279
43	0.878927
44	0.879103
45	0.855558
46	0.855777

Ratio de diseño	
47	0.833591
48	0.833828
49	0.813575
50	0.813808
51	0.796061
52	0.79627
53	0.781566
54	0.781736
55	0.770548
56	0.770667
57	0.763363
58	0.763425
59	0.760245
60	0.760245
61	0.761284
62	0.761222
63	0.766415
64	0.766296
65	0.77543
66	0.775261
67	0.787992
68	0.787785
69	0.803661
70	0.803431

Ratio de diseño	
71	0.82193
72	0.821695
73	0.842253
74	0.842037
75	0.864084
76	0.863911
77	0.886894
78	0.886793
79	0.910199

Figura 3.139. Ratios de diseño de los elementos de la brida inferior - armadura óptima final - ejemplo N°3.

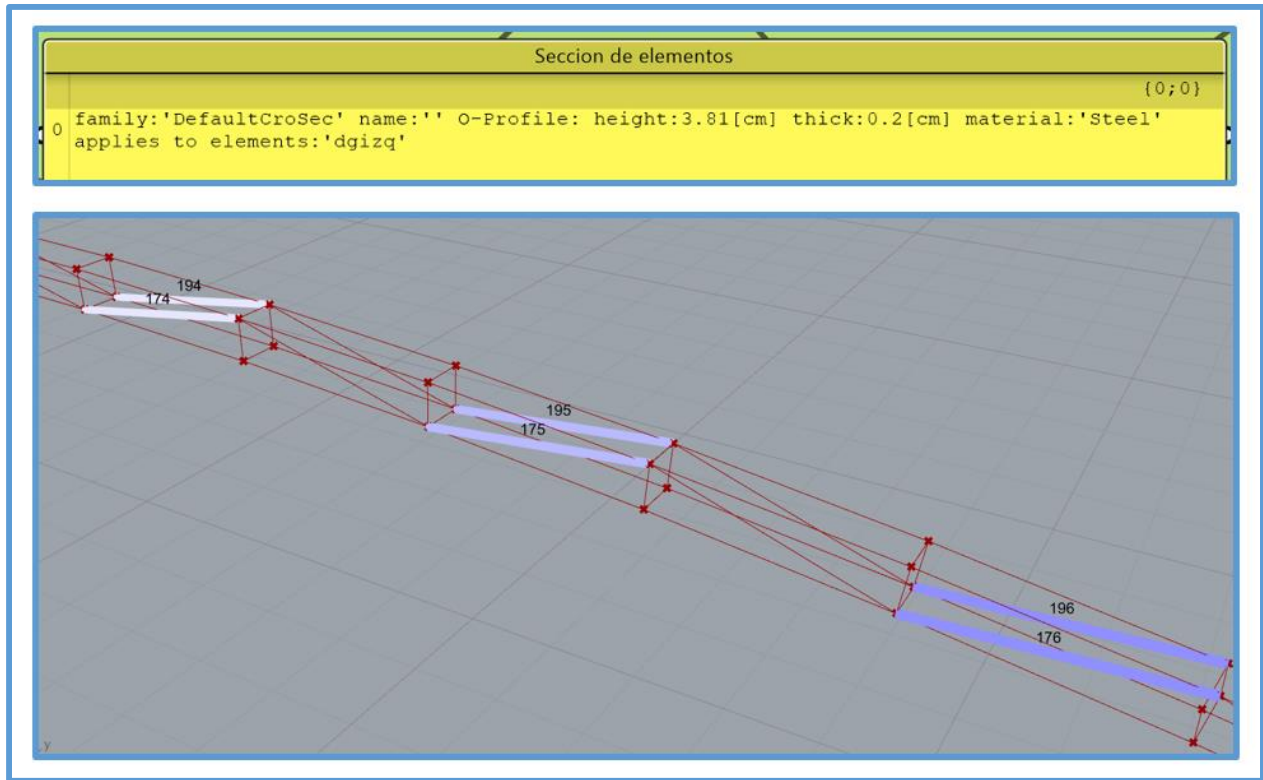


Figura 3.140. Sección de las diagonales - izquierda - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)	
{0;0;0;0}			
0	8400.011723	23	8433.664618
1	8413.905448	24	8439.658743
2	8425.111549	25	8443.234029
3	8433.664618	26	8444.588493
4	8439.658743	27	8443.938248
5	8443.234029	28	8441.541525
6	8444.588493	29	8437.674795
7	8443.938248	30	13289.810689
8	8441.541525	31	13289.810689
9	8437.674795	32	13289.810689
10	13289.810689	33	13289.810689
11	13289.810689	34	13289.810689
12	13289.810689	35	13289.810689
13	13289.810689	36	13289.810689
14	13289.810689	37	13289.810689
15	13289.810689	38	13289.810689
16	13289.810689	39	13289.810689
17	13289.810689		
18	13289.810689		
19	13289.810689		
20	8400.011723		
21	8413.905448		
22	8425.111549		

Figura 3.141. Resistencia de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°3.

Ratio de diseño	
{0;0;0;0}	
0	0.026988
1	0.026527
2	0.025489
3	0.023836
4	0.021554
5	0.018655
6	0.015185
7	0.011224
8	0.006888
9	0.00232
10	0.00147
11	0.004345
12	0.007049
13	0.009492
14	0.011603
15	0.013334
16	0.014659
17	0.015574
18	0.016092
19	0.016239
20	0.026988
21	0.026527
22	0.025489

Ratio de diseño	
23	0.023836
24	0.021554
25	0.018655
26	0.015185
27	0.011224
28	0.006888
29	0.00232
30	0.00147
31	0.004345
32	0.007049
33	0.009492
34	0.011603
35	0.013334
36	0.014659
37	0.015574
38	0.016092
39	0.016239

Figura 3.142. Ratios de diseño de las diagonales - izquierda - armadura óptima final - ejemplo N°3.

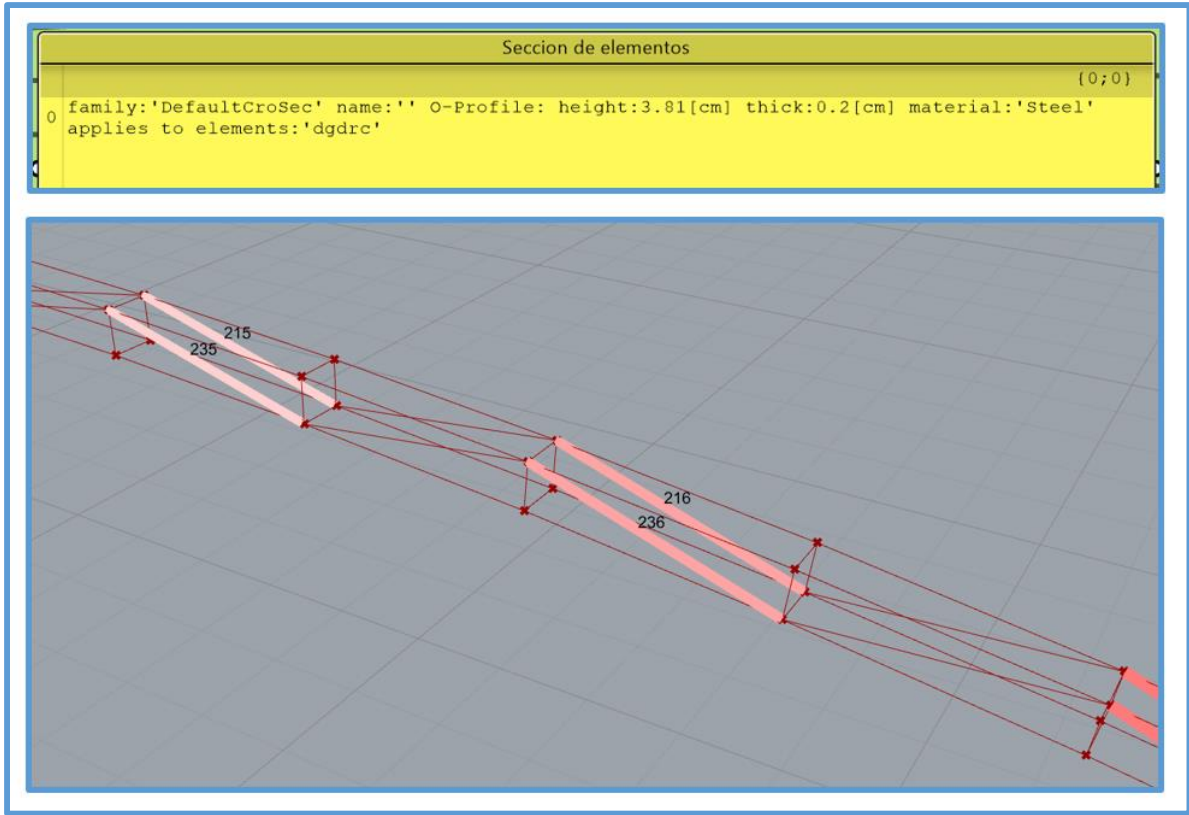


Figura 3.143. Sección de las diagonales - derecha - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)	
	{0;0;0;0}
0	13289.810689
1	13289.810689
2	13289.810689
3	13289.810689
4	13289.810689
5	13289.810689
6	13289.810689
7	13289.810689
8	13289.810689
9	13289.810689
10	8437.67418
11	8441.540238
12	8443.938899
13	8444.588969
14	8443.238247
15	8439.658542
16	8433.664797
17	8425.11176
18	8413.905353
19	8400.01176
20	13289.810689
21	13289.810689
22	13289.810689

Resistencia de diseño (lb)	
23	13289.810689
24	13289.810689
25	13289.810689
26	13289.810689
27	13289.810689
28	13289.810689
29	13289.810689
30	8437.67418
31	8441.540238
32	8443.938899
33	8444.588969
34	8443.238247
35	8439.658542
36	8433.664797
37	8425.11176
38	8413.905353
39	8400.01176

Figura 3.144. Resistencia de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°3.

Ratio de diseño	
	{0;0;0;0}
0	0.017078
1	0.017017
2	0.016589
3	0.015764
4	0.014526
5	0.012876
6	0.010836
7	0.008452
8	0.00579
9	0.002939
10	0
11	0.004594
12	0.008985
13	0.013028
14	0.016602
15	0.019621
16	0.022034
17	0.023823
18	0.025
19	0.025596
20	0.017078
21	0.017017
22	0.016589

Ratio de diseño	
23	0.015764
24	0.014526
25	0.012876
26	0.010836
27	0.008452
28	0.00579
29	0.002939
30	0
31	0.004594
32	0.008985
33	0.013028
34	0.016602
35	0.019621
36	0.022034
37	0.023823
38	0.025
39	0.025596

Figura 3.145. Ratios de diseño de las diagonales - derecha - armadura óptima final - ejemplo N°3.

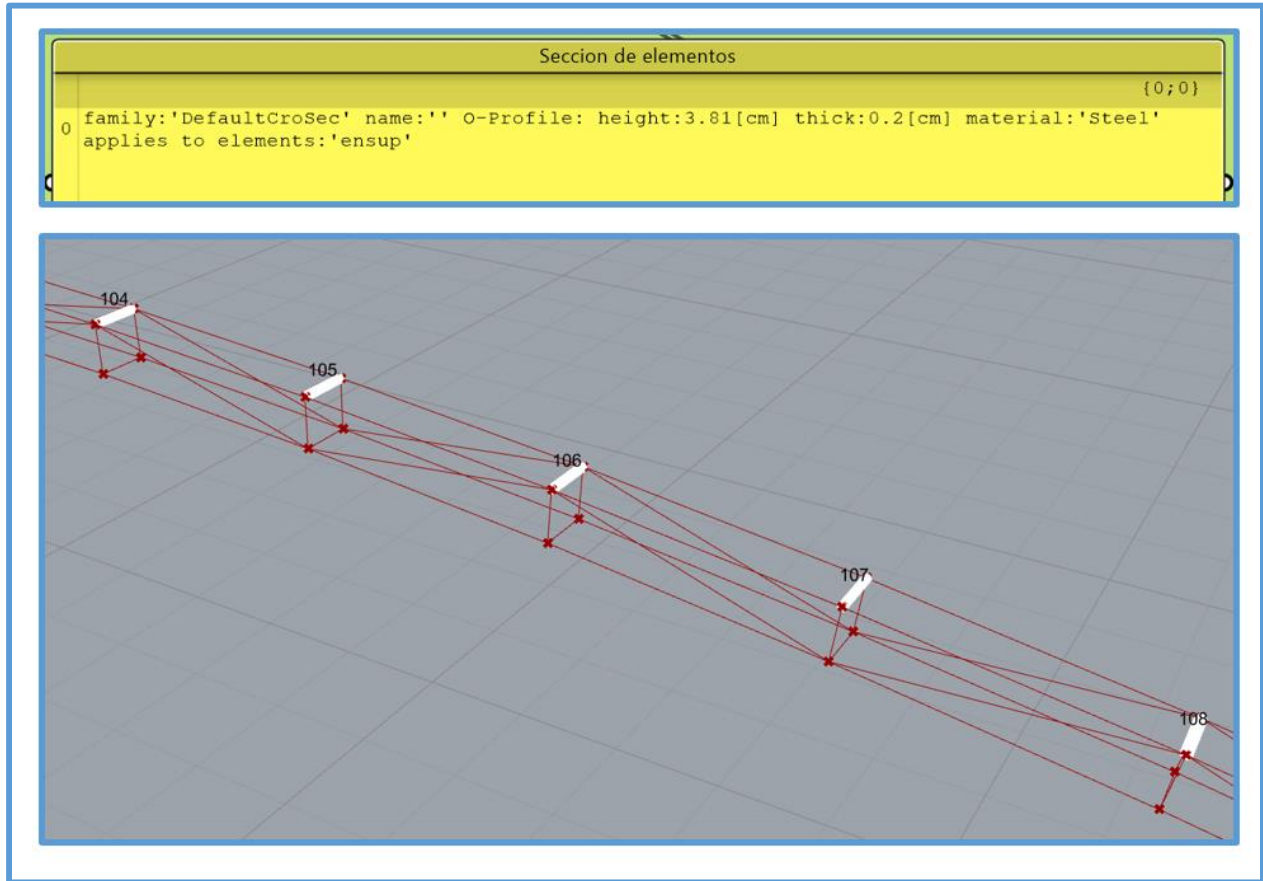


Figura 3.146. Sección de los enlaces superiores - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)	
	{0;0;0}	21	13091.49276
0	13091.49276	22	13091.49276
1	13091.49276	23	13091.49276
2	13091.49276	24	13091.49276
3	13091.49276	25	13091.49276
4	13091.49276	26	13091.49276
5	13091.49276	27	13091.49276
6	13091.49276	28	13091.49276
7	13091.49276	29	13091.49276
8	13091.49276	30	13091.49276
9	13091.49276	31	13091.49276
10	13091.49276	32	13091.49276
11	13091.49276	33	13091.49276
12	13091.49276	34	13091.49276
13	13091.49276	35	13091.49276
14	13091.49276	36	13091.49276
15	13091.49276	37	13091.49276
16	13091.49276	38	13091.49276
17	13091.49276	39	13091.49276
18	13091.49276	40	13091.49276
19	13091.49276		
20	13091.49276		

Figura 3.147. Resistencia de diseño de los enlaces superiores - armadura óptima final - ejemplo N°3.

Ratio de diseño		Ratio de diseño	
	{0;0;0}	21	0
0	0	22	0
1	0	23	0
2	0	24	0
3	0	25	0
4	0	26	0
5	0	27	0
6	0	28	0
7	0	29	0
8	0	30	0
9	0	31	0
10	0	32	0
11	0	33	0
12	0	34	0
13	0	35	0
14	0	36	0
15	0	37	0
16	0	38	0
17	0	39	0
18	0	40	0
19	0		
20	0		

Figura 3.148. Ratios de diseño de los enlaces superiores - armadura óptima final - ejemplo N°3.

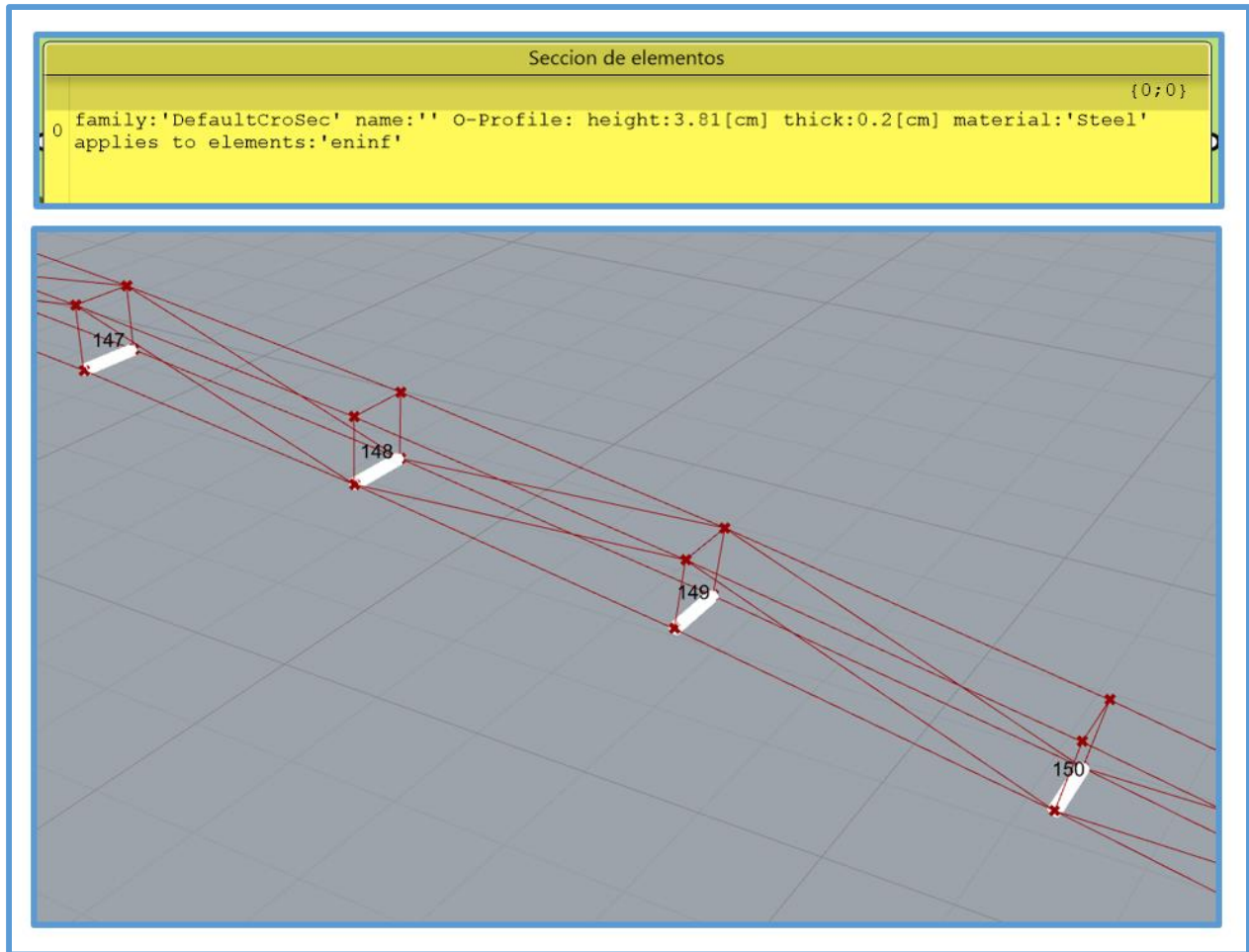


Figura 3.149. Sección de los enlaces inferiores - armadura óptima final - ejemplo N°3.

Resistencia de diseño (lb)		Resistencia de diseño (lb)	
	{0;0;0}		
0	13091.49276	21	13091.49276
1	13091.49276	22	13091.49276
2	13091.49276	23	13091.49276
3	13091.49276	24	13091.49276
4	13091.49276	25	13091.49276
5	13091.49276	26	13091.49276
6	13091.49276	27	13091.49276
7	13091.49276	28	13091.49276
8	13091.49276	29	13091.49276
9	13091.49276	30	13091.49276
10	13091.49276	31	13091.49276
11	13091.49276	32	13091.49276
12	13091.49276	33	13091.49276
13	13091.49276	34	13091.49276
14	13091.49276	35	13091.49276
15	13091.49276	36	13091.49276
16	13091.49276	37	13091.49276
17	13091.49276	38	13091.49276
18	13091.49276	39	13091.49276
19	13091.49276	40	13091.49276
20	13091.49276		

Figura 3.150. Resistencia de diseño de los enlaces inferiores - armadura óptima final - ejemplo N°3.

Ratio de diseño		Ratio de diseño	
	{0;0;0}		
0	0	21	0
1	0	22	0
2	0	23	0
3	0	24	0
4	0	25	0
5	0	26	0
6	0	27	0
7	0	28	0
8	0	29	0
9	0	30	0
10	0	31	0
11	0	32	0
12	0	33	0
13	0	34	0
14	0	35	0
15	0	36	0
16	0	37	0
17	0	38	0
18	0	39	0
19	0	40	0
20	0		

Figura 3.151. Ratios de diseño de los enlaces inferiores - armadura óptima final - ejemplo N°3.

Notas:

- Cada lista de secciones de perfiles tubulares usada en la optimización de tamaño debe contener solo secciones de perfiles tubulares elaborados bajo un mismo proceso de soldadura, es decir solo “ERW” o solo “SAW”.
- En los elementos eliminados mediante la optimización topológica, los scripts “AISC – TENSION –COMPRESION C” y “AISC – TENSION –COMPRESION R” continúan calculando la resistencia de diseño de dichos elementos, no obstante, sus ratios de diseño son cero.

CAPITULO IV: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

4.1. EJEMPLO N°1

El ejemplo N°1, constituye una guía básica e introductoria para la aplicación del “Diseño integral de armaduras metálicas usando Rhinoceros 3D”, en este se optimizó una armadura plana tipo Howe, la cual cubre una luz de 10m. Como primer paso se realizó la modelación paramétrica para la armadura, es así que se definieron 3 parámetros (Luz a cubrir, peralte de armadura y número de segmentos de armadura), de los cuales el parámetro “Luz a cubrir” permaneció constante (este parámetro permanece constante para cada proyecto), los otros parámetros restantes se usaron para realizar la optimización geométrica.

Luego de realizada la modelación paramétrica se realizó simulaciones de comportamientos estructurales mediante variación de parámetros, dichos resultados se presentaron en tablas, a partir de los cuales se puede notar que son muchos los comportamientos estructurales que se pueden suscitar en una armadura al variar sus parámetros. Mediante la simulación estructural se pudo determinar los parámetros más óptimos para una armadura, la simulación estructural es la base de la optimización pues los componentes Galápagos y Octopus probaron soluciones (de diseño de armadura), hasta elegir la solución más óptima, es decir lo que hacen es simular comportamientos estructurales hasta encontrarse con los parámetros más óptimos. Como ya se indicó con anterioridad, al ser muchos los comportamientos estructurales generados mediante variación de parámetros, solo se realizó simulaciones estructurales para este ejemplo, para el resto de ejemplos y/o casos se los puede realizar de manera similar.

Luego de realizada la simulación estructural se realizó la optimización, en una primera instancia se realizó una optimización geométrica con el complemento Octopus, el cual es un optimizador multi-objeto, entonces los objetos a optimizar fueron el peso y la deflexión máxima, se dejó trabajar a Octopus hasta la tercera generación de soluciones y luego en el entorno de este se eligió la mejor solución de las muchas presentadas por este, como el objetivo de Octopus es encontrar la armadura con menor peso y menor deflexión (para la carga de servicio), entonces puede generar soluciones inconsistentes con valores de esbeltez efectiva de los elementos superiores a los máximos establecidos en la AISC 360-10, por lo cual a medida que se iba buscado la mejor solución (solución óptima) se visualizaba que los ratios de diseño de los elemento brindados por

el script “AISC - TENSION - COMPRESION R” fuesen lo más próximos a uno; finalmente la solución óptima (de la optimización geométrica) fue la que contaba con menor peso, menor deflexión máxima y ratios de diseño de sus elementos lo más próximos a uno.

Luego de realizada la optimización geométrica, se realizó la optimización de tamaño con ayuda del complemento Galápagos, el cual es un optimizador de un solo objetivo, entonces se realizó la optimización de tamaño en forma parcial para cada tipo de elemento (brida superior, brida inferior, montantes, diagonales izquierda, diagonales derecha y montantes), lo que hizo Galápagos fue buscar la mejor sección para cada tipo de elementos, con la cual estos tuvieran ratios de diseño según la AISC 360-10 lo más próximos posibles a uno. En este ejemplo se usó el script “AISC - TENSION - COMPRESION R”, el cual diseña elementos o miembros (de una armadura) con secciones de perfiles tubulares rectangulares a tensión y/o compresión según la AISC 360-10.

También cabe indicar que una vez que se realizó la optimización de tamaño para cada tipo de elemento, se verifico que los ratios de diseño sean todos menores o iguales a uno, puesto que al realizar la optimización de tamaño de un tipo de elementos se modifican ligeramente los ratios de diseño de los elementos ya optimizados debido a la variación del peso propio, si es que hubiera algún ratio mayor a uno entonces se realiza una segunda iteración de la optimización de tamaño. En el este ejemplo solo se realizó una iteración de la optimización de tamaño. Antes de realizar la optimización topológica se eliminó las montantes de los extremos por que no cuentan con fuerzas actuantes, se encuentran entre los apoyos, no afectan la longitud del resto de elementos y la conexión a usada es plenamente soldada. Se aplicó la optimización topológica a las diagonales izquierda, pero no se pudo eliminar ninguna de ellas, pues al eliminar una sola diagonal izquierda si bien la nueva deflexión máxima aún estaba por debajo de la deflexión límite establecida en la N.T.E. E 0.20, los ratios de diseño de algunos elementos en las bridas superior e inferior sobrepasaban de uno. Como es una armadura simétrica al eliminar una diagonal derecha se produce el mismo efecto, por lo tanto, no se pudo aplicar la optimización topológica en este ejemplo. La solución óptima para este ejemplo, obtenida mediante la aplicación del “Diseño integral de armaduras metálicas usando Rhinoceros 3D” es la que se obtuvo luego de eliminar las montantes de los extremos a la solución óptima obtenida en la optimización de tamaño.

Para comprobar la veracidad de los resultados de diseño de la solución óptima obtenida aplicando el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” (haciendo uso de “Karamba” y del script “AISC – TENSION – COMPRESION R”), se diseñó dicha solución en el software “SAP2000”, presentándose así los respectivos resultados de diseño en las tablas siguientes:

Tabla 4.1. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION R” y “SAP2000” - solución óptima - ejemplo N°1.

Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Brida superior	30	28	29973.600000lb	29967.958087lb	0.98789	0.988076
	31	27	29973.600000lb	29967.958087lb	0.352818	0.352884
	32	26	21997.730000lb	22043.670366lb	0.096148	0.191896
	33	25	21997.730000lb	22043.670366lb	0.673038	0.671635
	34	24	21997.730000lb	22043.670366lb	0.961483	0.959479
	35	23	21997.730000lb	22043.670366lb	0.961483	0.959479
	36	22	21997.730000lb	22043.670366lb	0.673038	0.671635
	37	21	21997.730000lb	22043.670366lb	0.096148	0.191896
	38	20	29973.600000lb	29967.958087lb	0.352818	0.352884
	39	19	29973.600000lb	29967.958087lb	0.98789	0.988076
Brida inferior	20	18	21997.730000lb	22043.670366lb	0.961483	0.959479
	21	17	21997.730000lb	22043.670366lb	0.288445	0.287844
	22	16	29973.600000lb	29967.958087lb	0.070564	0.141154
	23	15	29973.600000lb	29967.958087lb	0.352818	0.352884
	24	14	29973.600000lb	29967.958087lb	0.423381	0.423461
	25	13	29973.600000lb	29967.958087lb	0.423381	0.423461
	26	12	29973.600000lb	29967.958087lb	0.352818	0.352884
	27	11	29973.600000lb	29967.958087lb	0.070564	0.141154
	28	10	21997.730000lb	22043.670366lb	0.288445	0.287844
	29	9	21997.730000lb	22043.670366lb	0.961483	0.959479

Tabla 4.1. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION R” y “SAP2000” - solución óptima - ejemplo N°1.

Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Montantes	11	8	25171.200000lb	25155.116853lb	0.31762	0.317823
	12	7	25171.200000lb	25155.116853lb	0.226871	0.227016
	13	6	25171.200000lb	25155.116853lb	0.068061	0.13621
	14	5	25171.200000lb	25155.116853lb	0.022687	0.045403
	15	4	20598.270000lb	20592.469365lb	0	0
	16	3	25171.200000lb	25155.116853lb	0.022687	0.045403
	17	2	25171.200000lb	25155.116853lb	0.068061	0.13621
	18	1	25171.200000lb	25155.116853lb	0.226871	0.227016
	19	0	25171.200000lb	25155.116853lb	0.31762	0.317823
Diagonales - izquierda	6	33	23851.660000lb	23879.319796lb	0.050389	0.100661
	7	32	23851.660000lb	23879.319796lb	0.302333	0.301983
	8	31	23851.660000lb	23879.319796lb	0.503889	0.503305
	9	30	23851.660000lb	23879.319796lb	0.705445	0.704627
	10	29	23851.660000lb	23879.319796lb	0.907	0.905949
Diagonales - derecha	1	38	23851.660000lb	23879.319796lb	0.050389	0.100661
	2	37	23851.660000lb	23879.319796lb	0.302333	0.301983
	3	36	23851.660000lb	23879.319796lb	0.503889	0.503305
	4	35	23851.660000lb	23879.319796lb	0.705445	0.704627
	5	34	23851.660000lb	23879.319796lb	0.907	0.905949

Tabla 4.2. Deflexiones máximas obtenidas con “Karamba”- “AISC - TENSION - COMPRESION R” y “SAP2000” - solución óptima - ejemplo N°1.

Karamba		SAP2000		Deflexión limite N.T.E E 0.20
N° de nodo	Deflexión máxima	N° de nodo	Deflexión máxima	
8	0.010219m	20	0.010218m	0.055556m
9	0.010219m	2	0.010218m	

Como se aprecia en las tablas 4.1 y 4.2, los resultados de diseño obtenidos con “Karamba” y el script “AISC – TENSION – COMPRESION R”, son similares a los resultados de diseño obtenidos con “SAP2000”, las pequeñas variaciones de milésimas en los ratios de diseño se deben a los factores de conversión de unidades, por lo tanto, se ha confirmado la correcta funcionalidad del script “AISC - TENSION - COMPRESION R” para el diseño de elementos o miembros (de una armadura) con secciones de perfiles tubulares rectangulares a tensión y/o compresión según la AISC 360-10.

Nota: Cabe indicar que a pesar de tratarse de solo elementos a tensión y/o compresión “SAP2000”, aplica el criterio de considerar solo la mitad del ratio de diseño para ratios de diseño menores a 0.2, lo cual se establece en la formula H1-1b de la AISC 360-10 y se aplica a Miembros con Simetría Doble y Simple Solicitados a Flexión y Compresión, el script “AISC - TENSION - COMPRESION R” no aplica tal criterio puesto que los elementos de una armadura solo están sometidos a tensión y/o compresión, por lo tanto los ratios de diseño menores a 0.2 obtenidos con el script “AISC - TENSION - COMPRESION R” solo tendrán la mitad de su valor al ser obtenidos con “SAP2000” (lo cual se aprecia en las tablas de resultados de diseño).

4.2. EJEMPLO N°2

En el ejemplo N°2 se optimizó una armadura tipo Pratt aplicando el “Diseño integral de armaduras metálicas usando Rhinoceros 3D”. Como primer paso se realizó la modelación paramétrica para la armadura, es así que se definieron 5 parámetros (Luz a cubrir, peralte de armadura, flecha superior de armadura, flecha inferior de armadura y número de segmentos de armadura), de los cuales el parámetro “Luz a cubrir” tuvo un valor de 15m.

En este ejemplo se obtuvieron 2 soluciones óptimas, en la primera se hizo una aproximación a una solución óptima obtenida aplicando al “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” (ya que SAP2000 realiza la optimización de tamaño para el caso de armaduras), por lo cual se adoptaron criterios típicos de predimensionamiento para la armadura. Para realizar la optimización de tamaño se usó el complemento Galápagos y esta optimización de tamaño usando Galápagos comparada con la optimización de tamaño realizada con SAP2000 tiene la ventaja de optimizar las secciones transversales de todos los elementos a la vez y no una por una como lo hace SAP2000, es por eso que una solución óptima obtenida adoptando criterios típicos de predimensionamiento y aplicando una optimización de tamaño con Galápagos, se aproxima a una solución óptima obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, entonces en este caso para efectos de análisis comparativos se considero la solución óptima obtenida adoptando criterios típicos de predimensionamiento y aplicando una optimización de tamaño con Galápagos como una solución óptima obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”. Los criterios típicos de predimensionamiento usados fueron $L/12$ para el peralte y un ángulo de triangulación de 45° (ángulo de triangulación óptimo).

En la segunda solución óptima se aplicó plenamente el “Diseño integral de armaduras metálicas usando Rhinoceros 3D”, se realizó una optimización geométrica y de tamaño en simultaneo usando Galápagos, luego se aplicó una optimización topológica, es así que la solución óptima para este caso fue la que se obtuvo luego de aplicar la optimización topológica. Por otro lado, las dos soluciones óptimas obtenidas en este ejemplo fueron diseñadas en SAP2000 con el objetivo de probar la veracidad de los resultados de diseño brindados por el script “AISC - TENSION - COMPRESION C” desarrollado en esta investigación. Es así que se tiene los resultados de diseño de este ejemplo en las tablas siguientes:

Tabla 4.3. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima – “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” - ejemplo N°2.

“Proceso de diseño estructural tradicional de prueba y error usando SAP2000”						
Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Brida superior	1	11	23625.000000lb	23612.217250lb	0.960661	0.961181
	2	10	23625.000000lb	23612.217250lb	0.414601	0.414825
	3	9	14254.130000lb	14265.510703lb	0.00838	0.016747
	4	8	14254.130000lb	14265.510703lb	0.519564	0.51915
	5	7	14254.130000lb	14265.510703lb	0.821247	0.820591
	6	6	14254.130000lb	14265.510703lb	0.921808	0.921072
	7	5	14254.130000lb	14265.510703lb	0.921808	0.921072
	8	4	14254.130000lb	14265.510703lb	0.821247	0.820591
	9	3	14254.130000lb	14265.510703lb	0.519564	0.51915
	10	2	14254.130000lb	14265.510703lb	0.00838	0.016747
	11	1	23625.000000lb	23612.217250lb	0.414601	0.414825
	12	0	23625.000000lb	23612.217250lb	0.960661	0.961181
Brida inferior	13	36	35556.170000lb	35502.823938lb	0.839872	0.841134
	14	35	35556.170000lb	35502.823938lb	0.39642	0.397015
	15	34	35556.170000lb	35502.823938lb	0.016797	0.033645
	16	33	45360.000000lb	45349.518987lb	0.097435	0.194916
	17	32	45360.000000lb	45349.518987lb	0.352874	0.352956
	18	31	45360.000000lb	45349.518987lb	0.447676	0.44778
	19	30	45360.000000lb	45349.518987lb	0.447676	0.44778
	20	29	45360.000000lb	45349.518987lb	0.352874	0.352956
	21	28	45360.000000lb	45349.518987lb	0.097435	0.194916
	22	27	35556.170000lb	35502.823938lb	0.016797	0.033645
	23	26	35556.170000lb	35502.823938lb	0.39642	0.397015
	24	25	35556.170000lb	35502.823938lb	0.839872	0.841134
Montantes	25	24	16295.810000lb	16306.074492lb	0	0
	26	23	16295.810000lb	16306.074492lb	0.967579	0.966969
	27	22	16295.810000lb	16306.074492lb	0.791655	0.791157
	28	21	16295.810000lb	16306.074492lb	0.615732	0.615344
	29	20	16295.810000lb	16306.074492lb	0.439809	0.439532
	30	19	16295.810000lb	16306.074492lb	0.263885	0.263719
	31	18	16295.810000lb	16306.074492lb	0.087962	0.175813
	32	17	16295.810000lb	16306.074492lb	0.263885	0.263719
	33	16	16295.810000lb	16306.074492lb	0.439809	0.439532
	34	15	16295.810000lb	16306.074492lb	0.615732	0.615344
	35	14	16295.810000lb	16306.074492lb	0.791655	0.791157

Tabla 4.3. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima – “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” - ejemplo N°2.

" Proceso de diseño estructural tradicional de prueba y error usando SAP2000"						
Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Montantes	36	13	16295.810000lb	16306.074492lb	0.967579	0.966969
	37	12	16295.810000lb	16306.074492lb	0	0
Diagonales - izquierda	44	42	23587.200000lb	23612.217250lb	0.042971	0.085851
	45	41	23587.200000lb	23612.217250lb	0.257828	0.257554
	46	40	23587.200000lb	23612.217250lb	0.429713	0.429257
	47	39	23587.200000lb	23612.217250lb	0.601598	0.60096
	48	38	23587.200000lb	23612.217250lb	0.773483	0.772663
	49	37	23587.200000lb	23612.217250lb	0.945368	0.944366
Diagonales - derecha	38	48	23587.200000lb	23612.217250lb	0.042971	0.085851
	39	47	23587.200000lb	23612.217250lb	0.257828	0.257554
	40	46	23587.200000lb	23612.217250lb	0.429713	0.429257
	41	45	23587.200000lb	23612.217250lb	0.601598	0.60096
	42	44	23587.200000lb	23612.217250lb	0.773483	0.772663
	43	43	23587.200000lb	23612.217250lb	0.945368	0.944366

Tabla 4.4. Deflexiones máximas obtenidas con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima – “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”- ejemplo N°2.

"Proceso de diseño estructural tradicional de prueba y error usando SAP2000"				
Karamba		SAP2000		Deflexión límite N.T.E E 0.20
N° de nodo	Deflexión máxima	N° de nodo	Deflexión máxima	
6	0.011724m	7	0.011727m	0.083333m

Tabla 4.5. Resultados de diseño obtenidos con “Karamba” - “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°2.

“Diseño integral de armaduras metálicas usando Rhinoceros 3D”						
Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Brida superior	48	15	17526.660000lb	17531.047131lb	0.583437	0.583356
	49	14	17526.660000lb	17531.047243lb	0.622269	0.622198
	50	13	17526.660000lb	17531.047020lb	0.671157	0.671099
	51	12	17526.660000lb	17531.046958lb	0.730069	0.730026
	52	11	17526.660000lb	17531.047041lb	0.796066	0.79604
	53	10	17526.660000lb	17531.047060lb	0.861254	0.861245
	54	9	17526.660000lb	17531.047058lb	0.912352	0.912357
	55	8	17526.660000lb	17531.047040lb	0.935057	0.935068
	56	7	17526.660000lb	17531.047056lb	0.935057	0.935068
	57	6	17526.660000lb	17531.047065lb	0.912352	0.912357
	58	5	17526.660000lb	17531.046726lb	0.861254	0.861245
	59	4	17526.660000lb	17531.047072lb	0.796066	0.79604
	60	3	17526.660000lb	17531.046736lb	0.730069	0.730026
	61	2	17526.660000lb	17531.047071lb	0.671157	0.671099
	62	1	17526.660000lb	17531.047469lb	0.622269	0.622198
63	0	17526.660000lb	17531.047313lb	0.583437	0.583356	
Brida inferior	32	48	32731.030000lb	32717.576670lb	0.882647	0.883462
	33	47	32731.030000lb	32717.577379lb	0.861901	0.862697
	34	46	32731.030000lb	32717.576264lb	0.837473	0.838246
	35	45	32731.030000lb	32717.576566lb	0.808982	0.809729
	36	44	32731.030000lb	32717.576640lb	0.776765	0.777483
	37	43	32731.030000lb	32717.576604lb	0.742687	0.743373
	38	42	32731.030000lb	32717.576429lb	0.711025	0.711681
	39	41	32731.030000lb	32717.576668lb	0.688475	0.689111
	40	40	32731.030000lb	32717.576115lb	0.688475	0.689111
	41	39	32731.030000lb	32717.577358lb	0.711025	0.711681
	42	38	32731.030000lb	32717.576735lb	0.742687	0.743373
	43	37	32731.030000lb	32717.575447lb	0.776765	0.777483
	44	36	32731.030000lb	32717.576994lb	0.808982	0.809729
	45	35	32731.030000lb	32717.576223lb	0.837473	0.838246
	46	34	32731.030000lb	32717.577317lb	0.8619	0.862697
47	33	32731.030000lb	32717.577370lb	0.882646	0.883462	

Tabla 4.5. Resultados de diseño obtenidos con “Karamba”- “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°2.

“Diseño integral de armaduras metálicas usando Rhinoceros 3D”						
Tipo de elemento	N° de elemento		Resistencia de diseño		Ratio de diseño	
	SAP2000	Karamba	SAP2000	Script AISC - TENSION - COMPRESION R	SAP2000	Script AISC - TENSION - COMPRESION R
Montantes	15	32	21654.280000lb	21647.566713lb	0	0
	16	31	22206.190000lb	22197.817149lb	0.040859	0.08179
	17	30	22622.010000lb	22612.374910lb	0.040655	0.081387
	18	29	22925.330000lb	22914.753158lb	0.039997	0.080071
	19	28	23138.750000lb	23127.501499lb	0.038703	0.077483
	20	27	23282.360000lb	23270.662229lb	0.036667	0.073409
	21	26	23372.670000lb	23360.688701lb	0.034016	0.068102
	22	25	23421.880000lb	23409.739682lb	0.031252	0.062568
	23	24	23437.440000lb	23425.255778lb	0.029918	0.059898
	24	23	23421.880000lb	23409.739649lb	0.031252	0.062568
	25	22	23372.670000lb	23360.688700lb	0.034016	0.068102
	26	21	23282.360000lb	23270.662271lb	0.036667	0.073409
	27	20	23138.750000lb	23127.501374lb	0.038703	0.077483
	28	19	22925.330000lb	22914.753112lb	0.039997	0.080071
	29	18	22622.010000lb	22612.374832lb	0.040655	0.081387
	30	17	22206.190000lb	22197.816986lb	0.040859	0.08179
31	16	21654.280000lb	21647.566713lb	0	0	
Diagonales - izquierda	8	56	23625.000000lb	23612.217250lb	0.004601	0.009212
	9	55	23625.000000lb	23612.217250lb	0.011506	0.023036
	10	54	23625.000000lb	23612.217250lb	0.013757	0.027542
	11	53	23625.000000lb	23612.217250lb	0.012001	0.024028
	12	52	23625.000000lb	23612.217250lb	0.008181	0.01638
	13	51	23625.000000lb	23612.217250lb	0.003907	0.007823
	14	49	18353.040000lb	18355.388657lb	0.004255	0.008514
Diagonales - derecha	1	64	23625.000000lb	23612.217250lb	0.004601	0.009212
	2	63	23625.000000lb	23612.217250lb	0.011506	0.023036
	3	62	23625.000000lb	23612.217250lb	0.013757	0.027542
	4	61	23625.000000lb	23612.217250lb	0.012001	0.024028
	5	60	23625.000000lb	23612.217250lb	0.008182	0.01638
	6	59	23625.000000lb	23612.217250lb	0.003907	0.007823
	7	57	18353.030000lb	18355.388004lb	0.004255	0.008514

Tabla 4.6. Deflexiones máximas obtenidas con “Karamba” - “AISC - TENSION - COMPRESION C” y “SAP2000” - solución óptima - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°2.

“Diseño integral de armaduras metálicas usando Rhinoceros 3D”				
Karamba		SAP2000		Deflexión límite N.T.E E 0.20
N° de nodo	Deflexión máxima	N° de nodo	Deflexión máxima	
8	0.014629m	31	0.014611m	0.083333m

Tabla 4.7. Pesos de armaduras en cada proceso de diseño empleado - ejemplo N°2.

	“Proceso de diseño estructural tradicional de prueba y error usando SAP2000”	“Diseño integral de armaduras metálicas usando Rhinoceros 3D”
Peso	257.108467kg	184.221303kg

Tabla 4.8. Secciones y longitudes totales de los elementos de las armaduras en cada proceso de diseño empleado - ejemplo N°2.

Elementos	Secciones transversales (mm)		Longitud total de elementos (m)	
	“Proceso de diseño estructural tradicional de prueba y error usando SAP2000”	“Diseño integral de armaduras metálicas usando Rhinoceros 3D”	“Proceso de diseño estructural tradicional de prueba y error usando SAP2000”	“Diseño integral de armaduras metálicas usando Rhinoceros 3D”
Brida superior	HSS42.2X3.6	HSS42.2X3.6	15.00m	15.37m
Brida inferior	HSS60.3X4.8	HSS60.3X3.9	15.00m	15.58m
Montantes	HSS48.3X3.0	HSS42.2X3.6	16.25m	5.04m
Diagonal izquierda	HSS48.3X3.0	HSS42.2X3.6	10.61m	7.33m
Diagonal derecha	HSS48.3X3.0	HSS42.2X3.6	10.61m	7.33m

Al observar la tabla 4.7 se aprecia que el peso de la armadura obtenida con el “Diseño integral de armaduras metálicas usando Rhinoceros 3D”, es mucho menor que el peso de la armadura obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, son 72.887164kg menos, los cuales traducidos a porcentaje con respecto al mayor peso representan un 28.35% de peso ahorrado, lo cual se corrobora con la tabla 4.8 donde se muestra las secciones y longitudes totales de los elementos de cada armadura, la armadura obtenida con el “Diseño integral

de armaduras metálicas usando Rhinoceros 3D” cuenta con secciones mucho más pequeñas que la armadura obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”, además de tener longitudes totales de elementos menores (menores metrados). Luego el contar con menor peso debido a secciones de menor dimensión y menores metrados constituye un ahorro de costo de material.

También cabe indicar que la deflexión máxima de la armadura obtenida con el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” y la deflexión máxima de la armadura obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” se encuentran muy por debajo de la deflexión límite establecida en la N.T.E E 0.20, siendo la deflexión máxima de la armadura obtenida con el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” superior en aproximadamente 3mm a la deflexión máxima de la armadura obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000” (ver tablas 4.4 y 4.6).

Como se aprecia en las tablas 4.3, 4.4, 4.5 y 4.6 los resultados de diseño obtenidos con “Karamba” y el script “AISC - TENSION - COMPRESION C”, son similares a los resultados de diseño obtenidos con “SAP2000”, las pequeñas variaciones de milésimas en los ratios de diseño se deben a los factores de conversión de unidades, por lo tanto, se ha confirmado la correcta funcionalidad del script “AISC - TENSION - COMPRESION C” para el diseño de elementos o miembros (de una armadura) con secciones de perfiles tubulares circulares a tensión y/o compresión según la AISC 360-10.

4.3. EJEMPLO N°3

En el ejemplo N°3 se aplicó el plenamente el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” a un proyecto real (una cobertura metálica), se optimizó un arco de viga de celosía central (con configuración estructural Warren con montantes consecutivos) y la vigueta (con configuración estructura Warren simple) que va en cada uno de sus nodos. En la modelación paramétrica para el arco de viga de celosía se definieron 5 parámetros (luz a cubrir, flecha de arco de viga de celosía, peralte de viga de celosía, número de segmentos de arco de viga de celosía y ancho de viga de celosía), de los cuales el parámetro “luz a cubrir” tuvo un valor de 41m. En la modelación paramétrica para la vigueta se definieron 3 parámetros (luz a cubrir, peralte de armadura y número de segmentos de armadura), de los cuales el parámetro “luz a cubrir” tuvo un valor de 5.1m.

Para el arco de viga de celosía se aplicó la optimización geométrica, de tamaño y topológica, además se realizaron unos reajustes finales al colocar una sección más pequeña que la sección más pequeña de las listas de secciones definidas para la optimización de tamaño en los elementos con ratios de diseño muy bajos o cero como son el caso de las montantes, diagonales y enlaces.

Por otro lado, en la vigueta solo se aplicó la optimización geométrica, no se pudo aplicar la optimización de tamaño, pues al usar perfiles tubulares estos tienen una mayor resistencia comparados con las varillas lisas, entonces se trabajó directamente con secciones pequeñas para no tener una vigueta sobredimensionada con ratios de diseño demasiado bajos, luego al trabajar con secciones pequeñas la optimización geométrica se encargó de obtener ratios de diseño lo más próximos posibles a uno. También cabe indicar que no se pudo aplicar la optimización topológica en la vigueta, pues al usar una configuración estructural Warren simple, la eliminación de diagonales se traducía en grandes deflexiones (muy superiores a la deflexión límite establecida en la N.T.E E 0.20), entonces la solución óptima para la vigueta se obtuvo solo por optimización geométrica.

Se obtuvieron resultados muy satisfactorios al optimizar el arco de viga de celosía, pues para una luz a cubrir de 41m, el peralte del arco es de 0.25m, el ancho del arco es de 0.20m, la flecha del arco es de 5.60m, el número de segmentos de arco es de 40 y el peso del arco es de 765.279545 kg (se usaron secciones pequeñas, como se muestra en la tabla 4.9). También para la vigueta se obtuvieron resultados satisfactorios, pues se obtuvo un peralte de 0.20m, con 12 segmentos de

vigueta y un peso de 9.859282kg (también se usaron secciones pequeñas, como se muestra en la tabla 4.10).

Tabla 4.9. Secciones y longitudes totales de los elementos del arco de viga de celosía - ejemplo N°3.

“Diseño integral de armaduras metálicas usando Rhinoceros 3D” - Arco de viga de celosía		
Elementos	Secciones transversales (mm)	Longitud total de elementos (m)
Brida superior	HSS42.2X3.6	86.50m
Brida inferior	HSS48.3X3.0	86.02m
Montantes	HSS38.1X2.0	18.71m
Diagonal izquierda	HSS38.1X2.0	44.08m
Diagonal derecha	HSS38.1X2.0	41.88m
Enlaces superiores	HSS38.1X2.0	8.20m
Enlaces inferiores	HSS38.1X2.0	8.20m

Tabla 4.10. Secciones y longitudes totales de los elementos de la vigueta - ejemplo N°3.

“Diseño integral de armaduras metálicas usando Rhinoceros 3D” - Vigueta		
Elementos	Secciones transversales (mm)	Longitud total de elementos (m)
Brida superior	HSS19.05X1.5	5.30m
Brida inferior	HSS19.05X1.5	4.25m
Diagonal izquierda	HSS19.05X1.5	2.80m
Diagonal derecha	HSS19.05X1.5	2.80m

Tabla 4.11. Deflexión máxima - solución óptima final - “Diseño integral de armaduras metálicas usando Rhinoceros 3D” - ejemplo N°3.

Armadura	“Diseño integral de armaduras metálicas usando Rhinoceros 3D”		
	Karamba		Deflexión limite N.T.E E 0.20
	N° de nodo	Deflexión máxima	
Arco de viga de celosía	38 y 120	0.040615m	0.227778m
Vigueta	3	0.01189m	0.028333m

4.4. ANÁLISIS DEL “DISEÑO INTEGRAL DE ARMADURAS METÁLICAS USANDO RHINOCEROS 3D”

Una vez desarrollados los ejemplos del “Diseño integral de armaduras metálicas usando Rhinoceros 3D”, se puede describir las ventajas que este brinda, así tenemos que:

- No es necesario realizar el predimensionamiento de una armadura.
- Permite realizar simulaciones estructurales de manera rápida y dinámica, bastara con solo cambiar el valor del slider (control de magnitud deslizante) que representa un parámetro de diseño para poder generar nuevos comportamientos estructurales.
- Aplicando el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” se puede ahorrar tiempo de diseño por las siguientes razones:
 - Un modelo de armadura en un archivo de Grasshopper se adapta a varios proyectos mediante el parámetro luz a cubrir, así ya no se tiene que realizar trabajos tediosos de modelamiento de la armadura, bastara con hacer una copia del archivo en Grasshopper del modelo de armadura y cambiar la luz a cubrir para empezar a trabajar otro proyecto.
 - La optimización de tamaño tiene la ventaja de aplicarse en forma simultánea a todos los elementos de la armadura y durar entre 5 a 10 minutos aproximadamente, comparada con la optimización de tamaño en SAP2000, donde se prueba secciones de una a una en cada elemento, pudiendo este proceso durar horas para el caso de armaduras complejas. También se puede aplicar la optimización de tamaño de forma parcial a un solo tipo de elemento de armadura y suele durar entre 1 a 5 minutos.
 - La optimización geométrica y de tamaño no solo se pueden aplicar de forma individual sino también en forma conjunta y suelen durar entre 5 a 10 minutos. Para geometrías complejas con muchos parámetros y un amplio rango para cada uno de esos parámetros puede que se demore más tiempo que el intervalo de tiempo antes mencionado, para encontrar la solución óptima.
 - La optimización topológica se aplica de forma rápida, con solo cambiar el valor del slider del “Target mass ratio” del componente “BESOBeam” se va evaluando si se pueden eliminar elementos de armadura.
- Permite optimizar armaduras; se aplica la optimización geométrica, de tamaño y topológica con el objetivo de buscar el menor peso de una armadura, y el menor peso se obtiene debido

a que se buscan menores longitudes de los elementos (menores metrados), secciones transversales de menor dimensión y se eliminan elementos que no son esenciales para el correcto funcionamiento de la armadura; en consecuencia, el tener el menor peso para una armadura se traduce en un ahorro de costo de material.

También cabe señalar las siguientes ventajas que nos brindan Grasshopper, Rhinoceros 3D y Karamba y que consecuentemente formaran parte del “Diseño integral de armaduras metálicas usando Rhinoceros 3D”:

- Facilitan la elaboración de los planos, pues a través de la opción “Bake” (obtenida al dar click derecho en cada componente de Grasshopper) en los componentes que contienen definiciones de geometría, se puede dibujar rápidamente dichas definiciones en Rhinoceros3D usando capas (de manera similar a AutoCAD) y luego la geometría dibujada en Rhinoceros3D se puede exportar rápidamente a AutoCAD al guardar el archivo de Rhinoceros3D como “dwg”.
- Se puede exportar la solución óptima obtenida al aplicar el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” a Revit (integración en un ambiente BIM), proceso que se explica detalladamente en los tutoriales que se encuentran en el sitio web de Karamba.

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.1.CONCLUSIONES

- Los tres tipos de optimización desarrollados en el “Diseño integral de armaduras metálicas usando Rhinoceros 3D” están enfocados a optimizar el peso de una armadura, en un caso específico se obtuvo un ahorro de 28.35% en peso, con respecto a una armadura óptima obtenida con el “Proceso de diseño estructural tradicional de prueba y error usando SAP2000”.
- El “Diseño integral de armaduras metálicas usando Rhinoceros 3D” reduce el tiempo de diseño mediante la modelación paramétrica y los complementos Octopus y Galápagos.
- El “Diseño integral de armaduras metálicas usando Rhinoceros 3D” permite optimizar el peso de las armaduras metálicas teniendo en cuenta los requisitos de diseño de la norma AISC 360-10.

5.2.RECOMENDACIONES

- Se recomienda en futuras investigaciones realizar plantillas de proyectos en base a armaduras metálicas (coberturas, puentes, edificios, etc.) de tal manera que se optimice todo un proyecto con el complemento Galápagos.
- Se recomienda desarrollar scripts para incluir dentro del “Diseño integral de armaduras metálicas usando Rhinoceros 3D” perfiles angulares, varillas lisas, perfiles “T”, perfiles “C”, etc.

REFERENCIAS BIBLIOGRÁFICAS

- Akos, G., & Parsons, R. (2015). *FOUNDATIONS : THE GRASSHOPPER PRIMER THIRD EDITION*. New York: Studio Mode. Obtenido de The Grasshopper Primer: <http://grasshopperprimer.com/es/0-about/1-grasshopper-an-overview.html>
- Alegria Mira, L. (2014). *PARAMETRIC STRUCTURAL ASSESSMENT OF DEPLOYABLE SCISSOR SYSTEMS Optimising the Universal Scissor Component*. Zelzate: University Press.
- Alfamege García , A. (2015). *ESTUDIO PARAMÉTRICO DE CÚPULAS DE BARRAS: GENERACIÓN PARAMÉTRICA, CÁLCULO AUTOMATIZADO Y COMPARATIVA DEL COMPORTAMIENTO DE CÚPULAS DE HACES ANTE VARIACIONES GEOMÉTRICAS*. Coruña: Universidad de Coruña.
- Almaraz, A. (2015). *EVOLUTIONARY OPTIMIZATION OF PARAMETRIC STRUCTURES*. Coruña: University of A Coruña.
- AMERICAN INSTITUTE OF STEEL CONSTRUCTION. (2010). *Especificaciones para Construcciones de Acero Estructural*. Chicago: American Institute of Steel Construction.
- Aroca Hernandez, R. (2001). *VIGAS TRIANGULADAS Y CERCHAS*. Madrid: Instituto Juan de Herrera Escuela Técnica Superior de Arquitectura de Madrid.
- Bonelli, M. (8 de Noviembre de 2016). *ResearchGate*. Obtenido de ResearchGate: https://www.researchgate.net/publication/310731033_OPTIMIZACION_DE_ARMADURAS_PLANAS_MEDIANTE_DISENO_PARAMETRICO_Y_ALGORITMOS_GENETICOS_EFECTOS_DE_LA_NO_CORRESPONDENCIA_OBJETO_REAL_-_MODELO_IDEALIZADO
- Bonelli, M., & Gudiño, S. (2016). *Diseño óptimo de una cercha de nave industrial existente mediante algoritmos genéticos*. Rafaela: Universidad Tecnológica Nacional.
- Christensen, P. W., & Klarbring, A. (2009). *An Introduction to Structural Optimization*. Waterloo: Springer.
- Cuevas, M. (2016). *DIBUJO Y PINTURA DIGITAL HERRAMIENTAS DE SOFTWARE LIBRE PARA LA CREACION ARTISTICA*. Obtenido de GRASSHOPER: <https://softwarelibrebellasartes.wordpress.com/grasshoper/>
- Davidson, S. (19 de Noviembre de 2017). *Grasshopper: Modelado algoritmico para Rhino*. Obtenido de Grasshopper: Modelado algoritmico para Rhino: <http://www.grasshopper3d.com/>
- Filomeno Coelho, R., Tysmans, T., & Verwimp, E. (2013). *Form finding & structural optimization*. Bruselas: Springer.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Alabama: ADDISON-WESLEY.

- Instituto para la Construcción Tubular. (2007). *GUIA DE DISEÑO PARA ESTRUCTURAS EN CELOSÍA RESUELTAS CON PERFILES TUBULARES DE ACERO*. Álava: Instituto para la Construcción Tubular.
- Jarquín Laguna, R. (2014). *APLICACIÓN DE ALGORITMOS GENÉTICOS EN INGENIERÍA CIVIL*. Distrito Federal: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
- Loja González, J. A. (2014). *Algoritmos genéticos como herramientas en el proceso de diseño generativo*. Loja: UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA.
- Mackenzie, J. S. (2012). *VISUALIZING LANDSCAPE SYSTEMS WITH PARAMETRIC MODELING*. ATHENS: University of Georgia.
- Maltagliatti, I. (2016). *DISEÑO PARAMÉTRICO DE ESTEREO ESTRUCTURAS CON OPTIMIZACIÓN DE SECCIONES*. Rafaela: Universidad Tecnológica Nacional.
- Marczyk, A. (23 de Abril de 2004). *The TalkOrigins Archive*. Obtenido de Algoritmos genéticos y computación evolutiva: <http://www.talkorigins.org/faqs/genalg/genalg.html>
- Mateos Andaluz, A. (2004). *ALGORITMOS EVOLUTIVOS Y ALGORITMOS GENÉTICOS*. Madrid: Universidad Carlos III de Madrid.
- McNeel, R. (2001). *Rhinoceros Modelador NURBS para Windows*. Seattle: Publications, Robert & Associates.
- McNeel, R. (28 de Noviembre de 2017). *Rhinoceros diseñar, modelar, presentar, analizar, realizar*. Obtenido de Rhinoceros diseñar, modelar, presentar, analizar, realizar: <https://www.rhino3d.com/>
- Nagy, D. (19 de Febrero de 2017). *Medium*. Obtenido de Medium: <https://medium.com/generative-design/structural-analysis-with-karamba-a73b959587c0>
- Namoncura, C., & Vásquez, M. (Marzo de 2011). *ResearchGate*. Obtenido de ResearchGate: https://www.researchgate.net/profile/Pedro_L/publication/269809277_Optimizacion_de_formas_arquitectonicas_con_algoritmos_geneticos_OTAG/links/5496df6e0cf20f487d31630c/Optimizacion-de-formas-arquitectonicas-con-algoritmos-geneticos-OTAG.pdf
- Payne, A., & Issa, R. (2009). *GRASSHOPPER PRIMER*. Cambridge: DUM DUM.
- Preisinger, C. (2013). *Linking Structure and Parametric Geometry*. London: Helen Castle.
- Preisinger, C. (2016). *PARAMETRIC STRUCTURAL MODELING KARAMBA*. Viena: KARAMBA.
- Preisinger, C. (29 de Noviembre de 2017). *Karamba 3D*. Obtenido de Karamba 3D: <http://www.karamba3d.com/>
- Quispe Salazar, S. N. (2016). *“MEJORAMIENTO DE LAS CONDICIONES DEL SERVICIO DE EDUCACIÓN SECUNDARIA EN LA I.E. FELIPE HUAMAN POMA DE AYALA, DEL*

CENTRO POBLADO EL TAMBO, DISTRITO DE BAMBAMARCA, PROVINCIA DE HUALGAYOC, REGIÓN CAJAMARCA". Cajamarca.

- Radman, A. (2013). *Bi-directional Evolutionary Structural Optimization (BESO) for Topology Optimization of Material's Microstructure*. Melbourne: RMIT University.
- Reboredo, A. (1999). *EL PROCESO DE DISEÑO ESTRUCTURAL Separata del "Manual de Construcción Sismorresistente"*. Mendoza: Centro de Ingenieros, Arquitectos y Agrimensores de Mendoza y de la Sociedad de Arquitectos de Mendoza.
- Reglamento Nacional de Edificaciones. (2006). *Norma Técnica de Edificaciones E 0.20 "CARGAS"*. Lima: El Peruano.
- Reglamento Nacional de Edificaciones. (2006). *Norma Técnica de Edificaciones E0.90 "ESTRUCTURAS METÁLICAS"*. Lima: El Peruano.
- Richardson, J. (2010). *Multicriteria Topology Optimization of Discrete*. Bruselas: Vrije Universiteit Brussel.
- Rutten, D. (24 de Setiembre de 2010). *Grasshopper MODELADO ALGORITMICO PARA RHINO*. Obtenido de Grasshopper MODELADO ALGORITMICO PARA RHINO: <http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>
- Rutten, D. (14 de Agosto de 2011). *Grasshopper MODELADO ALGORÍTMICO PARA RHINO*. Obtenido de Grasshopper MODELADO ALGORÍTMICO PARA RHINO: <https://www.grasshopper3d.com/forum/topics/galapagos-multiple-fitness>
- Rutten, D. (2013). *GALAPAGOS: ON THE LOGIC AND LIMITATIONS OF GENERIC SOLVERS*. London: Helen Castle.
- Rutten, D. (Noviembre de 30 de 2017). *Grasshopper: Modelado algoritmico para Rhino*. Obtenido de Grasshopper: Modelado algoritmico para Rhino: <https://www.grasshopper3d.com/group/galapagos>
- Sanchez Caballero, S. (2012). *Optimización estructural y topológica de estructuras morfológicamente no definidas mediante algoritmos genéticos*. Valencia: UNIVERSITAT POLITECNICA DE VALENCIA.
- Tedeschi, A. (2014). *AAD_Algorithms-Aided Design-Parametric Strategies using Grasshopper®*. Brienza: Le Penseur.
- Velázquez Villegas, F., & Santillán Gutiérrez, S. (2013). *Optimización de forma de un cuerpo suspendido basada en reglas evolutivas y modelado paramétrico: la forma de un fruto*. Ciudad de Mexico: Ingeniería Investigación y Tecnología.
- Vidal Calvet, M., & Ramírez Camarero, G. (26 de Febrero de 2011). *ISSUU*. Obtenido de Manual de ejercicios de Grasshopper Nivel I: https://issuu.com/frikeyarq/docs/cursogh_nivel_1_manual

Vidal Calvet, M., & Ramírez Camarero, G. (26 de Julio de 2011). *ISSUU*. Obtenido de Manual Basico de Scripting con C# en Grasshopper:
https://issuu.com/frikearq/docs/cursoghscripting_manual_1_ed

Vier, R. (06 de Diciembre de 2012). *Grasshopper ALGORITHMIC MODELING FOR RHINO*. Obtenido de Grasshopper ALGORITHMIC MODELING FOR RHINO:
<https://www.grasshopper3d.com/group/octopus?groupUrl=octopus&id=2985220%3AGroup%3A742529&page=6#comments>

Zapata Baglietto, L. F. (1997). *Diseno Estructural en Acero*. Lima: Zapata Baglietto, Luis F.

ANEXOS

Anexo 1. Predimensionamiento de armaduras planas

➤ Recomendación para peralte de armadura rectangular

El Instituto para la Construcción Tubular (ICT, 2007) nos brinda la siguiente recomendación:

Peralte (p) : $\frac{L}{15} \leq p \leq \frac{L}{10}$; generalmente se adopta $p = \frac{L}{15}$

➤ Recomendaciones para la triangulación de armaduras

Aroca (2001) menciona que el ángulo óptimo de triangulación es de 45° y que un intervalo razonable para los ángulos de triangulación es de 45° a 60°.

Anexo 2. Código en lenguaje de programación C Sharp (C#) del script N°1

```
//Instanciamos el objeto "lineas" el cual nos proporcionara las longitudes de cada uno de los
elementos o miembros con los cuales estamos trabajando.
Rhino.Geometry.Line lineas = (Rhino.Geometry.Line) L;
double l = 0.00;
l = lineas.Length;
//Definimos el factor de longitud efectiva, para miembros en armaduras,por estar simplemente
apoyadas se considera igual a 1.
double k = 1.00;
//Definimos la longitud efectiva o longitud de pandeo como producto del factor de longitud
efectiva y la longitud del miembro.
LE = l * k;
```

Anexo 3. Código en lenguaje de programación C Sharp (C#) del script N°2

```
//Instanciamos el objeto "lineas" el cual nos proporcionara las longitudes de cada uno de los
elementos o miembros que conforman
//la brida superior de la armadura.
Rhino.Geometry.Line lineas = (Rhino.Geometry.Line) Li;
//Definimos una lista donde agregaremos las longitudes de los elementos o miembros que
conforma la brida superior de la armadura.
Rhino.Collections.RhinoList<double> lngt = new Rhino.Collections.RhinoList<double>();
lngt.Add(lineas.Length);
//Definimos la variable "suma" la cual nos ayudara a convertir cada una de las longitudes de los
elementos o miembros de la brida
//superior de la armadura, al tipo object.
double suma = 0.00;
```

```

int i = 0;
for (i = 0; i < lngt.Count; i++)
{
    suma += lngt[i];
}
//Asignamos las longitudes de los elementos o miembros de la brida superior de la armadura
como tipo object a la salida "Lt"
Lt = lngt;

```

Anexo 4. Código en lenguaje de programación C Sharp (C#) del script N°3

```

//Definimos La variable "cn", con la cual vamos a convertir el peso (en kg) asignado en la
entrada "W" a newtons.
double cn = W * 0.00981;
//Definimos la variable "cknm" con la cual vamos a obtener la carga distribuida en kN/m y le
asignamos a la salida
//"Pp", con su respectivo signo que indica la direccion de la carga
double cknm = cn / L;
Pp = -cknm;

```

Anexo 5. Código en lenguaje de programación C Sharp (C#) del script N°4

```

//Instanciamos los objetos "Carga1" y "Carga2", los cuales nos proporcionaran los valores de las
cargas asignadas en las entradas "D" y "Lr".
Karamba.Loads.GravityLoad Carga1 = (Karamba.Loads.GravityLoad) D;
Karamba.Loads.UniformlyDistLoad Carga2 = (Karamba.Loads.UniformlyDistLoad) Lr;
//Clonamos para evitar efectos secundarios.
Carga1 = (Karamba.Loads.GravityLoad) Carga1.Clone();
Carga2 = (Karamba.Loads.UniformlyDistLoad) Carga2.Clone();
//Definimos las variables de los tipos carga.
double d = 0.00;
double lr = 0.00;
//Asignamos los valores de los tipos de cargas a sus respectivas variables.
d = Carga1.maxAbsLoad();
lr = Carga2.maxAbsLoad();
//Definimos la variable de la carga de servicio.
double cservicio = 0.00;
//Asignamos el valor de la carga de servicio a su respectiva variable.
cservicio = d + lr;
//Definimos un nuevo vector, el cual tendra el valor de la carga de servicio, colocamos signo
negativo para indicar la direccion de la carga.
Rhino.Geometry.Vector3d P = new Rhino.Geometry.Vector3d(0, 0, -cservicio);
//Instanciamos el objeto "cs" de tipo Load, asignando sus parametros respectivos, segun nos
indica su constructor.
Karamba.Loads.UniformlyDistLoad cs = new Karamba.Loads.UniformlyDistLoad(nameid, P,
Carga2.q_orient, 7);

```

```
//Envolvemos el objeto instanciado en el paso anterior para que pueda adaptarse al entorno de "Grasshopper".
```

```
Cs = new Karamba.Loads.GH_Load(cs);
```

Anexo 6. Código en lenguaje de programación C Sharp (C#) del script N°5

```
//Instanciamos los objetos de carga recibidos.
```

```
Karamba.Loads.GravityLoad Carga1 = (Karamba.Loads.GravityLoad) D;  
Karamba.Loads.UniformlyDistLoad Carga2 = (Karamba.Loads.UniformlyDistLoad) L;  
Karamba.Loads.UniformlyDistLoad Carga3 = (Karamba.Loads.UniformlyDistLoad) Lr;  
Karamba.Loads.UniformlyDistLoad Carga4 = (Karamba.Loads.UniformlyDistLoad) W;  
Karamba.Loads.UniformlyDistLoad Carga5 = (Karamba.Loads.UniformlyDistLoad) S;  
Karamba.Loads.UniformlyDistLoad Carga6 = (Karamba.Loads.UniformlyDistLoad) E;  
Karamba.Loads.UniformlyDistLoad Carga7 = (Karamba.Loads.UniformlyDistLoad) R;
```

```
//Clonamos para evitar efectos secundarios.
```

```
Carga1 = (Karamba.Loads.GravityLoad) Carga1.Clone();  
Carga2 = (Karamba.Loads.UniformlyDistLoad) Carga2.Clone();  
Carga3 = (Karamba.Loads.UniformlyDistLoad) Carga3.Clone();  
Carga4 = (Karamba.Loads.UniformlyDistLoad) Carga4.Clone();  
Carga5 = (Karamba.Loads.UniformlyDistLoad) Carga5.Clone();  
Carga6 = (Karamba.Loads.UniformlyDistLoad) Carga6.Clone();  
Carga7 = (Karamba.Loads.UniformlyDistLoad) Carga7.Clone();
```

```
//Definimos las variables de los tipos carga.
```

```
double d = 0.00;  
double l = 0.00;  
double lr = 0.00;  
double w = 0.00;  
double s = 0.00;  
double e = 0.00;  
double r = 0.00;
```

```
//Definimos las variables de las combinaciones de carga.
```

```
double combinacion1 = 0.00;  
double combinacion2 = 0.00;  
double combinacion3 = 0.00;  
double combinacion4 = 0.00;  
double combinacion5 = 0.00;  
double combinacion6 = 0.00;
```

```
//Asignamos los valores de carga recibidos segun el tipo de carga.
```

```
d = Carga1.maxAbsLoad();  
l = Carga2.maxAbsLoad();  
lr = Carga3.maxAbsLoad();  
w = Carga4.maxAbsLoad();  
s = Carga5.maxAbsLoad();  
e = Carga6.maxAbsLoad();  
r = Carga7.maxAbsLoad();
```

```
//Definimos listas auxiliares, que ayudaran a realizar las combinaciones de carga.
```

```

Rhino.Collections.RhinoList<double> lcomb2 = new Rhino.Collections.RhinoList<double>();
lcomb2.Add(lr);
lcomb2.Add(s);
lcomb2.Add(r);
Rhino.Collections.RhinoList<double> lcomb3 = new Rhino.Collections.RhinoList<double>();
lcomb3.Add(lr);
lcomb3.Add(s);
lcomb3.Add(r);
Rhino.Collections.RhinoList<double> lcomb4 = new Rhino.Collections.RhinoList<double>();
lcomb4.Add(lr);
lcomb4.Add(s);
lcomb4.Add(r);
//Realizamos las combinaciones de carga.
combinacion1 = 1.4 * d;
combinacion2 = 1.2 * d + 1.6 * l + 0.5 * (lcomb2.Max());
combinacion3 = 1.2 * d + 1.6 * (lcomb3.Max()) + Math.Max(0.5 * l, 0.8 * w);
combinacion4 = 1.2 * d + 1.3 * w + 0.5 * l + 0.5 * (lcomb4.Max());
combinacion5 = 1.2 * d + 1 * e + 0.5 * l + 0.2 * s;
combinacion6 = 0.9 * d + Math.Max(1.3 * w, 1 * e);
//Definimos una nueva lista donde agregamos las combinaciones de carga.
Rhino.Collections.RhinoList<double> combinaciones = new
Rhino.Collections.RhinoList<double>();
combinaciones.Add(combinacion1);
combinaciones.Add(combinacion2);
combinaciones.Add(combinacion3);
combinaciones.Add(combinacion4);
combinaciones.Add(combinacion5);
combinaciones.Add(combinacion6);
//Definimos la variable que contendra a la mayor carga de entre las combinaciones de carga y le
asignamos su respectivo valor.
double combinacionf = combinaciones.Max();
//Definimos un nuevo vector, el cual tendra el valor de la mayor carga de entre las
combinaciones de carga, colocamos signo
//negativo para indicar la direccion de la carga.
Rhino.Geometry.Vector3d P = new Rhino.Geometry.Vector3d(0, 0, -combinacionf);
//Instanciamos el objeto "cf" de tipo Load, asignando sus parametros respectivos, segun nos
indica su constructor.
Karamba.Loads.UniformlyDistLoad cf = new Karamba.Loads.UniformlyDistLoad(nameid, P,
Carga2.q_orient, 8);
//Envolvemos el objeto instanciado en el paso anterior para que pueda adaptarse al entorno de
"Grasshopper".
Cu = new Karamba.Loads.GH_Load(cf);

```

Anexo 7. Código en lenguaje de programación C Sharp (C#) del script N°6

```
//Definimos las variables que contendran los valores de la deflexion calculada("dc") y de la
deflexion limite("dl")
//y les asignamos sus respectivos valores.
double dc = 0.00;
double dl = 0.00;
dc = DC;
dl = DL;
//Definimos la variable "dco", la cual servira a manera de filtro ya que nos ayudara a obtener
valores de deflexion
//menores o iguales a la deflexion limite.
double dco = 0.00;
if (dc <= dl)
dco = dc;
else
dco = 1000;
//Asignamos su respectivo valor a la salida "DCO", el cual sera un valor de deflexion menor o
igual a la deflexion limite.
DCO = dco;
```

Anexo 8. Código en lenguaje de programación C Sharp (C#) del script N°7

```
//*****
*****
//DEFINICION DE VARIABLES PARA EL DISEÑO DE ELEMENTOS DE ARMADURA
SEGUN LA AISC 360-10.
//*****
*****
//Instanciamos el objeto "secciones" el cual nos proporcionara los datos necesarios para el diseño
con secciones tubulares circulares.
Karamba.CrossSections.CroSec_Circle secciones = (Karamba.CrossSections.CroSec_Circle)
Secciones;
//Obtenemos el area de la seccion y convertimos de metros2 a pulgadas2.
double a = 0.00;
a = secciones.A * 1550.00310001;
//Obtenemos el radio de giro local de la seccion en la direccion y.
double ry = 0.00;
ry = secciones.iy;
//Obtenemos el radio de giro local de la seccion en la direccion z.
double rz = 0.00;
rz = secciones.iz;
//Obtenemos el espesor de la seccion y convertimos de metros a pulgadas.
double es = 0.00;
es = secciones.thick * 39.370079;
//Obtenemos el diametro de la seccion y convertimos de metros a pulgadas.
```

```

double dm = 0.00;
dm = secciones.getHeight() * 39.370079;
//Definimos una variable para el espesor de la seccion tubular.
double t = 0.00;

//Elegimos menor radio de giro y convertimos de metros a pulgadas.
double rgf = Math.Min(ry, rz) * 39.370079;
//Instanciamos el objeto "elementos", el cual nos proporcionara las longitudes de pandeo (KL) de
los elementos o miembros a diseñar.
Karamba.Elements.GrassBeam elementos = (Karamba.Elements.GrassBeam) Elementos;
//Agregamos las longitudes de pandeo de los elementos o miembros a diseñar a una nueva lista
definida y las convertimos de metros a pulgadas.
Rhino.Collections.RhinoList<double> lppp = new Rhino.Collections.RhinoList<double>();
lppp.Add(elementos.buckling_length_y * 39.370079);
//Instanciamos el objeto "material" el cual nos proporcionara los datos necesarios(que intervienen
en el diseño) del material
//con el cual estamos trabajando.
Karamba.Materials.FemMaterial_Isotrop material = (Karamba.Materials.FemMaterial_Isotrop)
Material;
//Definimos las siguientes variables:
//ee: Esbeltez efectiva, fe: Esfuerzo critico de Euler, Me: Modulo de elasticidad, fy: Resistencia a
la fluencia del acero;
//rn: Resistencia nominal, rd: Resistencia de diseno, fu : Tension ultima o resistencia ultima del
acero.
//Convertimo Me y fy a psi(lb/in^2).
double ee = 0.00;
double fe = 0.00;
double Me = (material.E(1) / Math.Pow(10, 4) * 1450.3774389728);
double fy = (material.fy(1) / Math.Pow(10, 4) * 1450.3774389728);
double fu = Convert.ToDouble(Fu);
double rn = 0.00;
double rd = 0.00;
string Diseñoelemento = "";
//Definimos la variable para la relacion ancho - espesor de la seccion.
double rae = 0.00;
//Definimos la variable soldadura que contendra el proceso de soldadura usado en la elaboracion
de los perfiles tubulares circulares(HSS)
//con los cuales estamos trabajando y le asignamos su respectivo valor de entrada.
string soldadura = "";
soldadura = Convert.ToString(Soldadura);
//Definimos las varibles soldadura1 y soldadura2 las cuales contienen los 2 procesos de
soldadura usados en la elaboracion de perfiles
//tubulares (HSS), indicados en la AISC 360 - 10.
string soldadura1 = "ERW";
string soldadura2 = "SAW";

```

```

//Definimos la variable que contendra la relacion ancho/espesor limite entre secciones de
elementos no esbeltos y secciones de elementos esbeltos
//y se asignamos su respectivo valor segun la tabla B4.1a de la AISC 360 - 10.
double rael = 0.00;
rael = 0.11 * (Me / fy);

//Definimos la variable que contendra el limite superior de relacion ancho/espesor para secciones
de elementos esbeltos.
double reale = 0.00;
reale = 0.45 * (Me / fy);
//Definimos variable para el factor de reduccion neto para elementos esbeltos en compresion.
double Q = 0.00;
//Para el caso de secciones tubulares (Hss), circulares o rectangulares, Q es igual al factor de
reduccion Qa., asi que definimos la variable
//para Qa.
double Qa = 0.00;
//Asignamos sus respectivos valores a la variable que contendra las fuerzas actuantes en los
elementos o miembros, convertimos de kN a lb.
double fuerzas = Fuerzas * 224.80894309971;
//Definimos una lista a la cual agregamos las fuerzas actuantes.
Rhino.Collections.RhinoList<double> lfs = new Rhino.Collections.RhinoList<double>();
lfs.Add(fuerzas);
/*****
*****
*****
*****
*****

//PROCESO DE DISEÑO DE LOS ELEMENTOS DE UNA ARMADURA (A
COMPRESION O TENSION) SEGUN AISC 360-10.
/*****
*****

////////////////////////////////////
////
//Soldadura ERW.
  if (soldadura == soldadura1)
  {
    //Calculamos relacion ancho -
espesor.~~~~~
~~~~~

    t = 0.93 * es;
    rae = dm / t;
    foreach (var lp in lppp)
    {
      foreach (var fi in lfs)
      {
        //-----
-----

```

```

//Diseño a tension.
if (fi > 0)
{
    ee = lp / rgf;
    Ranchoespesor = "Elemento a tension";
    RanchoespesorL = "Elemento a tension";
    RanchoespesorLEE = "Elemento a tension";
    //L/r menor igual a
300#####
#####
    if (ee <= 300)
    {
        EsbeltezSeccion = "Elemento a tension";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 300 - Esbeltez efectiva adecuada
- elemento a tension";
        PandeoElemento = "Elemento a tension";
        if (fu == 0)
        {
            //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
            rd = fy * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
            //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
            rd = Math.Min(fy * 0.9 * secciones.A * 1550.00310001, fu * 0.75 * secciones.A *
1550.00310001);
        }
    }
    //L/r mayor a
300#####
#####
    else
    {
        EsbeltezSeccion = "Elemento a tension";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores
de los elementos - elemento a tension";
        PandeoElemento = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores de
los elementos - elemento a tension";
        rn = 0.00;
        //Multiplicamos por el factor de resistencia para el caso de tension y convertimos el area
de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
}

```



```

//-----
//Diseño a compresion.
if (fi <= 0)
{

    ee = lp / rgf;
    Ranchoespesor = rae;
    RanchoespesorL = rael;
    RanchoespesorLEE = reale;
    //kL/r menor igual a
    200#####
    #####
    if (ee <= 200)
    {
//Caso1#####
*****
        if (rae <= rael)
        {
            EsbeltezSeccion = "Seccion de elementos no esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / fy))
            {
                rn = Math.Pow(0.658, (fy / fe)) * fy;
                PandeoElemento = "Pandeo inelastico";
            }
            else
            {
                rn = 0.877 * fe;
                PandeoElemento = "Pandeo elastico";
            }
            //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
            rd = rn * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
//Caso2#####
*****
            if (rae > rael & rae < reale)
            {
                EsbeltezSeccion = "Seccion de elementos esbeltos";
                REsbeltezEfectiva = ee;

```

```

CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = (0.038 * Me) / (fy * (rae)) + 2 / 3;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
PandeoElemento = "Pandeo inelastico";
}
else
{
rn = 0.877 * fe;
PandeoElemento = "Pandeo elastico";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
else
{
EsbeltezSeccion = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = 1;
Q = 1;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
rn = 0.00;
PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
}
else
{
rn = 0.00;
}
}

```

```

        PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
}
//kL/r mayor a
200#####
#####
else
{
//Caso1#####
*****
    if (rae <= rael)
    {
        EsbeltezSeccion = "Seccion de elementos no esbeltos";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
        if (ee <= 4.71 * Math.Sqrt(Me / fy))
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        else
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    else
    {
//Caso2#####
*****
        if (rae > rael & rae < reale)
        {
            EsbeltezSeccion = "Seccion de elementos esbeltos";

```

```

REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = (0.038 * Me) / (fy * (rae)) + 2 / 3;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
else
{
    EsbeltezSeccion = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = 1;
Q = 1;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
    rn = 0.00;
    PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
}
}

```



```

CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 300 - Esbeltez efectiva adecuada
- elemento a tension";
PandeoElemento = "Elemento a tension";
if (fu == 0)
{
//Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
rd = fy * 0.9 * secciones.A * 1550.00310001;
}
else
{
//Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
rd = Math.Min(fy * 0.9 * secciones.A * 1550.00310001, fu * 0.75 * secciones.A *
1550.00310001);
}
//L/r mayor a
300#####
#####
else
{
EsbeltezSeccion = "Elemento a tension";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores
de los elementos - elemento a tension";
PandeoElemento = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores de
los elementos - elemento a tension";
rn = 0.00;
//Multiplicamos por el factor de resistencia para el caso de tension y convertimos el area
de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
}
//-----
-----
//Diseño a compresion.
if (fi <= 0)
{
ee = lp / rgf;
Ranchoespesor = rae;
RanchoespesorL = rael;
RanchoespesorLEE = reale;
//kL/r menor igual a
200#####
#####
if (ee <= 200)

```

```

{
//Caso1*****
*****
    if (rae <= rael)
    {
        EsbeltezSeccion = "Seccion de elementos no esbeltos";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
        fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
        if (ee <= 4.71 * Math.Sqrt(Me / fy))
        {
            rn = Math.Pow(0.658, (fy / fe)) * fy;
            PandeoElemento = "Pandeo inelastico";
        }
        else
        {
            rn = 0.877 * fe;
            PandeoElemento = "Pandeo elastico";
        }
        //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    else
    {
//Caso2*****
*****
        if (rae > rael & rae < reale)
        {
            EsbeltezSeccion = "Seccion de elementos esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
            //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
            //el diseño de todos los elementos.
            Qa = (0.038 * Me) / (fy * (rae)) + 2 / 3;
            Q = Qa;
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
            {
                rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
                PandeoElemento = "Pandeo inelastico";
            }
            else

```

```

    {
        rn = 0.877 * fe;
        PandeoElemento = "Pandeo elastico";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
    el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
else
{
    EsbeltezSeccion = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
    //Como cada script trabaja con una sola seccion para todos los elementos de la lista
    entonces, Q sera un valor constante en
    //el diseño de todos los elementos.
    Qa = 1;
    Q = 1;
    fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
    if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
    {
        rn = 0.00;
        PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    }
    else
    {
        rn = 0.00;
        PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
    el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
}
}
}
//kL/r mayor a
200#####
#####
else

```



```

{
//Caso1*****
*****
    if (rae <= rael)
    {
        EsbeltezSeccion = "Seccion de elementos no esbeltos";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
        if (ee <= 4.71 * Math.Sqrt(Me / fy))
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        else
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    else
    {
//Caso2*****
*****
        if (rae > rael & rae < reale)
        {
            EsbeltezSeccion = "Seccion de elementos esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
            //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
            //el diseño de todos los elementos.
            Qa = (0.038 * Me) / (fy * (rae)) + 2 / 3;
            Q = Qa;
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
            {
                rn = 0.00;
            }
        }
    }
}

```

```

    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
else
{
    EsbeltezSeccion = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
    //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
    Qa = 1;
    Q = 1;
    fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
    if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
    {
        rn = 0.00;
        PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    }
    else
    {
        rn = 0.00;
        PandeoElemento = "Seccion con elementos de relacion ancho/espesor fuera de los
rangos establecidos en la AISC 360-10, cambiar de seccion";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
}
}
}
}

```

```

    }
  }
}
}
//Calculamos los ratios de diseño en cada uno de los elementos.
double ratio = Math.Abs(fuerzas) / rd;
//Definimos una lista y agregamos los ratios de diseño de los elementos.
Rhino.Collections.RhinoList<double> ratios = new Rhino.Collections.RhinoList<double>();
ratios.Add(ratio);
//Realizamos el proceso de verificación de los ratios de diseño de los elementos o miembros,
donde un ratio menor o igual a 1, indicara un
//diseño correcto y un ratio mayor a 1 indicara un diseño incorrecto.
foreach (var rto in ratios)
{
  if (rto <= 1)
  {
    Diseñoelemento = "Diseño correcto";
  }
  else
  {
    Diseñoelemento = "No se cumple con el diseño, cambiar parametros de diseño";
  }
}
//*****
//*****
//*****
//DEFINICION DE SALIDAS RESTANTES DEL COMPONENTE.
//*****
//*****
//Estas salidas nos brindan datos obtenidos del proceso de diseño de los elementos o miembros
de una armadura.
RDiseño = rd;
Ratio = ratio;
DiseñoElemento = Diseñoelemento;
//*****
//*****
//DEFINICION DE SALIDAS CUANDO SE PROPORCIONA PROCEDIMIENTOS DE
SOLDADURA (PARA LA ELABORACION DE PERFILES TUBULARES) NO
ESPECIFICADOS EN LA AISC 360-10.
//*****
//*****
if (soldadura != soldadura2 & soldadura != soldadura1)
{

```

```

    EsbeltezSeccion = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    RDiseño = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    Ratio = 0.00;
    DiseñoElemento = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    Ranchoespesor = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    RanchoespesorL = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    RanchoespesorLEE = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    REsbeltezEfectiva = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    CREsbeltezEfectiva = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
    PandeoElemento = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
    tubulares(HSS), especificado en la AISC 360-10";
}
//*****
*****

```

Anexo 9. Código en lenguaje de programación C Sharp (C#) del script N°8

```

//*****
*****
//DEFINICION DE VARIABLES PARA EL DISEÑO DE ELEMENTOS DE ARMADURA
SEGUN LA AISC 360-10.
//*****
*****
//Instanciamos el objeto "secciones" el cual nos proporcionara los datos necesarios para el diseño
con secciones tubulares rectangulares.
Karamba.CrossSections.CroSec_Box secciones = (Karamba.CrossSections.CroSec_Box)
Secciones;
//Obtenemos el area de la seccion y convertimos de metros2 a pulgadas2.
double a = 0.00;
a = secciones.A * 1550.00310001;
//Obtenemos el radio de giro local de la seccion en la direccion y.
double ry = 0.00;
ry = secciones.iy;
//Obtenemos el radio de giro local de la seccion en la direccion z.
double rz = 0.00;
rz = secciones.iz;
//Obtenemos el espesor de la seccion (estamos trabajando con secciones de espesor uniforme) y
convertimos de metros a pulgadas.

```

```

double es = 0.00;
es = secciones.w_thick * 39.370079;
//Definimos una variable para el ancho efectivo de la seccion segun AISC 360-10.
double dm1 = 0.00;
//Definimos una variable para el peralte efectivo de la seccion segun AISC 360-10.
double dm2 = 0.00;

//Definimos una variable para el ancho de la seccion, le asignamos su respectivo valor y
convertimos de metros a pulgadas.
double at = 0.00;
at = secciones.maxWidth() * 39.370079;
//Definimos una variable para el peralte de la seccion, le asignamos su respectivo valor y
convertimos de metros a pulgadas.
double pt = 0.00;
pt = secciones.getHeight() * 39.370079;
//Definimos una variable para el espesor de la seccion tubular, segun proceso de soldadura
especificado en AISC 360-10.
double t = 0.00;
//Definimos una variable para el radio de la esquina interior del perfil, le asignamos su respectivo
valor y convertimos de metros a pulgadas.
double rfil = 0.00;
rfil = secciones.fillet_r * 39.370079;
//Elegimos menor radio de giro y convertimos de metros a pulgadas.
double rgf = Math.Min(ry, rz) * 39.370079;
//Instanciamos el objeto "elementos", el cual nos proporcionara las longitudes de pandeo (KL) de
los elementos o miembros a diseñar.
Karamba.Elements.GrassBeam elementos = (Karamba.Elements.GrassBeam) Elements;
//Agregamos las longitudes de pandeo de los elementos o miembros a diseñar a una nueva lista
definida y las convertimos de metros a pulgadas.
Rhino.Collections.RhinoList<double> lppp = new Rhino.Collections.RhinoList<double>();
lppp.Add(elementos.buckling_length_y * 39.370079);
//Instanciamos el objeto "material" el cual nos proporcionara los datos necesarios(que intervienen
en el diseño) del material
//con el cual estamos trabajando.
Karamba.Materials.FemMaterial_Isotrop material = (Karamba.Materials.FemMaterial_Isotrop)
Material;
//Definimos las siguientes variables:
//ee: Esbeltez efectiva, fe: Esfuerzo critico de Euler, Me: Modulo de elasticidad, fy: Resistencia a
la fluencia del acero;
//rn: Resistencia nominal, rd: Resistencia de diseno, fu : Tension ultima o resistencia ultima del
acero.
//Convertimo Me y fy a psi(lb/in^2).
double ee = 0.00;
double fe = 0.00;
double Me = (material.E(1) / Math.Pow(10, 4) * 1450.3774389728);
double fy = (material.fy(1) / Math.Pow(10, 4) * 1450.3774389728);

```

```

double fu = Convert.ToDouble(Fu);
double rn = 0.00;
double rd = 0.00;
string Diseñoelemento = "";
//Definimos una variable para la relacion ancho - espesor de la seccion considerando el ancho.
double rae1 = 0.00;
//Definimos una variable para la relacion ancho - espesor de la seccion considerando el peralte.
double rae2 = 0.00;
//Definimos una variable soldadura que contendra el proceso de soldadura usado en la
elaboracion de los perfiles tubulares circulares(HSS)
//con los cuales estamos trabajando y le asignamos su respectivo valor de entrada.
string soldadura = "";
soldadura = Convert.ToString(Soldadura);
//Definimos las variables soldadura1 y soldadura2 las cuales contienen los 2 procesos de
soldadura usados en la elaboracion de perfiles
//tubulares (HSS), indicados en la AISC 360 - 10.
string soldadura1 = "ERW";
string soldadura2 = "SAW";
//Definimos una variable para la relacion ancho espesor limite y se asignamos su respectivo valor
segun la tabla B4.1a de la AISC 360-10.
double rael = 0.00;
rael = 1.4 * Math.Sqrt(Me / fy);
//Variables para calculo de la reduccion de la resistencia por pandeo local.
//*****
//*****
//Definimos una variable para el factor de reduccion neto para elementos esbeltos en compresion.
double Q = 0.00;
//Para el caso de secciones tubulares (Hss), circulares o rectangulares, Q es igual al factor de
reduccion Qa., asi que definimos una variable
//para Qa.
double Qa = 0.00;
//Definimos una variable ff, que es el factor f necesario para hallar el ancho efectivo reducido
segun la ecuacion E7-18 de la AISC 360-10 y le asignamos
//su respectivo valor.
double ff = 0.00;
ff = fy;// fy esta en lb/in2.
//Definimos una variable para el ancho efectivo reducido1.
double be1 = 0.00;
//Definimos una variable para el ancho efectivo reducido2.
double be2 = 0.00;
//Definimos una variable para el area efectiva.
double Ae = 0.00;
//Definimos una variable para longitud inefectiva1.
double lie1 = 0.00;
//Definimos una variable para longitud inefectiva2.
double lie2 = 0.00;

```

```

//*****
*****
///Asignamos sus respectivos valores a la variable que contendra las fuerzas actuantes en los
elementos o miembros, convertimos de kN a lb.
double fuerzas = Fuerzas * 224.80894309971;
//Definimos una lista a la cual agregamos las fuerzas actuantes.
Rhino.Collections.RhinoList<double> lfs = new Rhino.Collections.RhinoList<double>();
lfs.Add(fuerzas);
//*****
*****
//*****
*****
//PROCESO DE DISEÑO DE LOS ELEMENTOS DE UNA ARMADURA (A COMPRESION
O TENSION) SEGUN AISC 360-10.
//*****
*****
////////////////////////////////////
////////
//Soldadura ERW.
    if (soldadura == soldadura1)
    {
        //Calculamos relacion ancho -
        espesor.~~~~~
        ~~~~~
        double rfl1 = 0.00;
        t = 0.93 * es;
        if (rfl1 == 0)
        {
            rfl1 = t;
        }
        else
        {
            rfl1 = 2*rfl1;
        }
        dm1 = at - 2 * t - rfl1;
        dm2 = pt - 2 * t - rfl1;
        rae1 = dm1 / t;
        rae2 = dm2 / t;
        foreach (var lp in lppp)
        {
            foreach (var fi in lfs)
            {
                //-----
                -----
                //Diseño a tension.
                if (fi > 0)

```

```

{
  ee = lp / rgf;
  RanchoespesorA = "Elemento a tension";
  RanchoespesorP = "Elemento a tension";
  RanchoespesorL = "Elemento a tension";

  //L/r menor igual a
  300#####
  #####
  if (ee <= 300)
  {
    EsbeltezSeccion = "Elemento a tension";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 300 - Esbeltez efectiva adecuada
- elemento a tension";
    PandeoElemento = "Elemento a tension";
    if (fu == 0)
    {
      //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
      rd = fy * 0.9 * secciones.A * 1550.00310001;
    }
    else
    {
      //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
      rd = Math.Min(fy * 0.9 * secciones.A * 1550.00310001, fu * 0.75 * secciones.A *
1550.00310001);
    }
  }
  //L/r mayor a
  300#####
  #####
  else
  {
    EsbeltezSeccion = "Elemento a tension";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores
de los elementos - elemento a tension";
    PandeoElemento = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores de
los elementos - elemento a tension";
    rn = 0.00;
    //Multiplicamos por el factor de resistencia para el caso de tension y convertimos el area
de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
  }
}

```



```

//-----
//Diseño a compresion.
if (fi <= 0)
{
    ee = lp / rgf;
    RanchoespesorA = rae1;
    RanchoespesorP = rae2;
    RanchoespesorL = rae1;
    //kL/r menor igual a
    200#####
    #####
    if (ee <= 200)
    {
//Caso1#####
*****
        if (rae1 <= rael & rae2 <= rael )
        {
            EsbeltezSeccion = "Seccion de elementos no esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / fy))
            {
                rn = Math.Pow(0.658, (fy / fe)) * fy;
                PandeoElemento = "Pandeo inelastico";
            }
            else
            {
                rn = 0.877 * fe;
                PandeoElemento = "Pandeo elastico";
            }
            //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
            rd = rn * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
//Caso2#####
*****
            if (rae1 > rael & rae2 <= rael )
            {
                EsbeltezSeccion = "Seccion de ancho esbelto";
                REsbeltezEfectiva = ee;

```

```

CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie1;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
rn = Q * (Math.Pow(0.658, ((Q * fy) / fe))) * fy;
PandeoElemento = "Pandeo inelastico";
}
else
{
rn = 0.877 * fe;
PandeoElemento = "Pandeo elastico";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
if (rae1 <= rael & rae2 > rael )
{
EsbeltezSeccion = "Seccion de peralte esbelto";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie2;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));

```

```

if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
  rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
  PandeoElemento = "Pandeo inelastico";
}
else

{
  rn = 0.877 * fe;
  PandeoElemento = "Pandeo elastico";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso4*****
*****
if (rael > rael & rae2 > rael )
{
  EsbeltezSeccion = "Seccion de ancho y peralte esbeltos";
  REsbeltezEfectiva = ee;
  CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
  be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
  lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
  be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
  lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
  Ae = a - 2 * t * lie1 - 2 * t * lie2;
  //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
  Qa = Ae / a;
  Q = Qa;
  fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
  rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
  PandeoElemento = "Pandeo inelastico";
}
else
{
  rn = 0.877 * fe;

```

```

        PandeoElemento = "Pandeo elastico";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    }
    }
    //kL/r mayor a
200#####
#####
    else
    {
//Caso1#####
#####
        if (rae1 <= rael & rae2 <= rael )
        {
            EsbeltezSeccion = "Seccion de elementos no esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / fy))
            {
                rn = 0.00;
                PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
            }
            else
            {
                rn = 0.00;
                PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
            }
            //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
            rd = rn * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
//Caso2#####
#####
            if (rae1 > rael & rae2 <= rael )
            {
                EsbeltezSeccion = "Seccion de ancho esbelto";
                REsbeltezEfectiva = ee;

```

```

CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie1;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
rn = 0.00;
PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
rn = 0.00;
PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
if (rae1 <= rael & rae2 > rael )
{
EsbeltezSeccion = "Seccion de peralte esbelto";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie2;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;

```

```

Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso4*****
*****
if (rae1 > rael & rae2 > rael )
{
    EsbeltezSeccion = "Seccion de ancho y peralte esbeltos";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
    be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
    lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff))));
    be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
    lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff))));
    Ae = a - 2 * t * lie1 - 2 * t * lie2;
    //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
    Qa = Ae / a;
    Q = Qa;
    fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
    if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
    {
        rn = 0.00;
        PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";

```



```

if (fi > 0)
{
    ee = lp / rgf;
    RanchoespesorA = "Elemento a tension";
    RanchoespesorP = "Elemento a tension";
    RanchoespesorL = "Elemento a tension";

    //L/r menor igual a
    300#####
    #####
    if (ee <= 300)
    {
        EsbeltezSeccion = "Elemento a tension";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 300 - Esbeltez efectiva adecuada
- elemento a tension";
        PandeoElemento = "Elemento a tension";
        if (fu == 0)
        {
            //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
            rd = fy * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
            //Hallamos la resistencia de diseño, convertimos el area de m2 a in2.
            rd = Math.Min(fy * 0.9 * secciones.A * 1550.00310001, fu * 0.75 * secciones.A *
1550.00310001);
        }
    }
    //L/r mayor a
    300#####
    #####
    else
    {
        EsbeltezSeccion = "Elemento a tension";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores
de los elementos - elemento a tension";
        PandeoElemento = "Esbeltez efectiva mayor a 300 - Reducir las longitudes mayores de
los elementos - elemento a tension";
        rn = 0.00;
        //Multiplicamos por el factor de resistencia para el caso de tension y convertimos el area
de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
}

```



```

//-----
//Diseño a compresion.
if (fi <= 0)
{
    ee = lp / rgf;
    RanchoespesorA = rae1;
    RanchoespesorP = rae2;
    RanchoespesorL = rae1;
    //kL/r menor igual a
    200#####
    #####
    if (ee <= 200)
    {
//Caso1#####
*****
        if (rae1 <= rael & rae2 <= rael )
        {
            EsbeltezSeccion = "Seccion de elementos no esbeltos";
            REsbeltezEfectiva = ee;
            CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
            fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
            if (ee <= 4.71 * Math.Sqrt(Me / fy))
            {
                rn = Math.Pow(0.658, (fy / fe)) * fy;
                PandeoElemento = "Pandeo inelastico";
            }
            else
            {
                rn = 0.877 * fe;
                PandeoElemento = "Pandeo elastico";
            }
            //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
            rd = rn * 0.9 * secciones.A * 1550.00310001;
        }
        else
        {
//Caso2#####
*****
            if (rae1 > rael & rae2 <= rael )
            {
                EsbeltezSeccion = "Seccion de ancho esbelto";
                REsbeltezEfectiva = ee;

```

```

CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie1;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) * fy;
PandeoElemento = "Pandeo inelastico";
}
else
{
rn = 0.877 * fe;
PandeoElemento = "Pandeo elastico";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
if (rae1 <= rael & rae2 > rael )
{
EsbeltezSeccion = "Seccion de peralte esbelto";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie2;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));

```

```

if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
{
    rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
    PandeoElemento = "Pandeo inelastico";
}
else
{
    rn = 0.877 * fe;
    PandeoElemento = "Pandeo elastico";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso4*****
*****
if (rae1 > rael & rae2 > rael )
{
    EsbeltezSeccion = "Seccion de ancho y peralte esbeltos";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva menor igual a 200 - Esbeltez efectiva
adecuada - elemento a compresion";
    be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
    lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
    be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
    lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
    Ae = a - 2 * t * lie1 - 2 * t * lie2;
    //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
    Qa = Ae / a;
    Q = Qa;
    fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
    if (ee <= 4.71 * Math.Sqrt(Me / ( Q * fy)))
    {
        rn = Q * (Math.Pow(0.658, ((Q * fy) / fe)) ) * fy;
        PandeoElemento = "Pandeo inelastico";
    }
    else
    {
        rn = 0.877 * fe;
        PandeoElemento = "Pandeo elastico";
    }
}

```

```

    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
    el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    }
}
//kL/r mayor a
200#####
#####
else
{
//Caso1*****
*****
    if (rae1 <= rael & rae2 <= rael )
    {
        EsbeltezSeccion = "Seccion de elementos no esbeltos";
        REsbeltezEfectiva = ee;
        CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
        if (ee <= 4.71 * Math.Sqrt(Me / fy))
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        else
        {
            rn = 0.00;
            PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
        }
        //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
        el area de m2 a in2.
        rd = rn * 0.9 * secciones.A * 1550.00310001;
    }
    else
    {
//Caso2*****
*****
        if (rae1 > rael & rae2 <= rael )
        {
            EsbeltezSeccion = "Seccion de ancho esbelto";
            REsbeltezEfectiva = ee;

```

```

CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie1;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;
Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
rn = 0.00;
PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
rn = 0.00;
PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso3*****
*****
if (rae1 <= rael & rae2 > rael )
{
EsbeltezSeccion = "Seccion de peralte esbelto";
REsbeltezEfectiva = ee;
CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
Ae = a - 2 * t * lie2;
//Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
Qa = Ae / a;

```

```

Q = Qa;
fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
else
{
    rn = 0.00;
    PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
}
//Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
rd = rn * 0.9 * secciones.A * 1550.00310001;
}
//Caso4*****
*****
if (rae1 > rael & rae2 > rael )
{
    EsbeltezSeccion = "Seccion de ancho y peralte esbeltos";
    REsbeltezEfectiva = ee;
    CREsbeltezEfectiva = "Esbeltez efectiva mayor a 200 - Reducir las longitudes
mayores de los elementos - elemento a compresion";
    be1 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) * Math.Sqrt(Me /
ff))), dm1);
    lie1 = Math.Abs(dm1 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae1) *
Math.Sqrt(Me / ff)));
    be2 = Math.Min((1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) * Math.Sqrt(Me /
ff))), dm2);
    lie2 = Math.Abs(dm2 - (1.92 * t * Math.Sqrt(Me / ff) * (1 - (0.38 / rae2) *
Math.Sqrt(Me / ff)));
    Ae = a - 2 * t * lie1 - 2 * t * lie2;
    //Como cada script trabaja con una sola seccion para todos los elementos de la lista
entonces, Q sera un valor constante en
//el diseño de todos los elementos.
    Qa = Ae / a;
    Q = Qa;
    fe = (Math.Pow(Math.PI, 2) * Me) / (Math.Pow(ee, 2));
    if (ee <= 4.71 * Math.Sqrt(Me / (Q * fy)))
    {
        rn = 0.00;
        PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";

```

```

    }
    else
    {
        rn = 0.00;
        PandeoElemento = "Esbeltez efectiva mayor a 200 - Reducir las longitudes mayores
de los elementos - elemento a compresion";
    }
    //Multiplicamos por el factor de resistencia para el caso de compresion y convertimos
el area de m2 a in2.
    rd = rn * 0.9 * secciones.A * 1550.00310001;
}
}
}
}
}
}
}
}
}
//Calculamos los ratios de diseño en cada uno de los elementos.
double ratio = Math.Abs(fuerzas) / rd;
//Definimos una lista y agregamos los ratios de diseño de los elementos.
Rhino.Collections.RhinoList<double> ratios = new Rhino.Collections.RhinoList<double>();
ratios.Add(ratio);
//Realizamos el proceso de verificacion de los ratios de diseño de los elementos o miembros,
donde un ratio menor o igual a 1, indicara un
//diseño correcto y un ratio mayor a 1 indicara un diseño incorrecto.
foreach (var rto in ratios)
{
    if (rto <= 1)
    {
        Diseñoelemento = "Diseño correcto";
    }
    else
    {
        Diseñoelemento = "No se cumple con el diseño, cambiar parametros de diseño";
    }
}
}
}
}
//*****
//*****
//*****
//*****
//DEFINICION DE SALIDAS RESTANTES DEL COMPONENTE.
//*****
//*****

//Estas salidas nos brindan datos obtenidos del proceso de diseño de los elementos o miembros
de una armadura.
RDiseño = rd;

```

```

Ratio = ratio;
DiseñoElemento = Diseñoelemento;
//*****
//*****
//*****
//DEFINICION DE SALIDAS CUANDO SE PROPORCIONA PROCEDIMIENTOS DE
SOLDADURA (PARA LA ELABORACION DE PERFILES TUBULARES) NO
ESPECIFICADOS EN LA AISC 360-10.
//*****
//*****
if (soldadura != soldadura2 & soldadura != soldadura1)
{
    EsbeltezSeccion = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    RDiseño = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    Ratio = 0.00;
    DiseñoElemento = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    RanchoespesorA = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    RanchoespesorP = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    RanchoespesorL = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    REsbeltezEfectiva = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    CREsbeltezEfectiva = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
    PandeoElemento = "Indicar un proceso de soldadura usado para la elaboracion de perfiles
tubulares(HSS), especificado en la AISC 360-10";
}
//*****
//*****

```

Anexo 10. Código en lenguaje de programación C Sharp (C#) del script N°9

```

//Definimos la variable "ratio" que contendra los ratios de diseño.
double ratio = 0.00;
ratio = Ratio;
//Definimos la lista "Irts" y le agregamos los ratios de diseño.
Rhino.Collections.RhinoList <double> Irts = new Rhino.Collections.RhinoList<double>();
Irts.Add(Ratio);
//Definimos la variable "nRatio" la cual servira a manera de filtro ya que nos ayudara a obtener
valores de ratio de diseño

```



```

//menores o iguales a 1 y le asignamos sus respectivos valores.
double nRatio = 0.00;
foreach( var ri in lrts)
{
    if (ri <= 1)
        nRatio = ratio;
    else
        nRatio = 1000.00;
}
//Asignamos sus respectivos valores a la salida "NRatio", los cuales seran valores de ratio de
diseño menores o iguales a 1.
NRatio = nRatio;

```

Anexo 11. Código en lenguaje de programación C Sharp (C#) del script N°10

```

//Definimos la lista "lr" y le agregamos los valores de ratio de diseño.
Rhino.Collections.RhinoList<double> lr = new Rhino.Collections.RhinoList<double>();
lr.Add(Ratio);
//Definimos la variable "nratio" la cual servira a manera de filtro ya que nos ayudara a obtener
valores de ratio de diseño
//menores o iguales a 1 y le asignamos sus respectivos valores.
double nratio = 0.00;
foreach (var pratio in lr)
{
    if (pratio == 1000.00)
    {
        nratio = 1000.00;
    }
    else
    {
        nratio = pratio;
    }
}
//Asignamos sus respectivos valores a la salida "NRatio", los cuales seran valores de ratio de
diseño menores o iguales a 1.
NRatio = nratio;

```

Anexo 12. Código en lenguaje de programación C Sharp (C#) del script N°11

```

//Instanciamos un objeto de tipo "modelo" para obtener el peso de la armadura y valiendonos de
sus propiedades le asignamos su respectivo valor.
Karamba.Models.Model modelo = (Karamba.Models.Model) Model;
P = modelo.mass() * 1000;

```

Anexo 13. Peso propio

Dentro del cd que contiene los archivos en Grasshopper de los ejemplos desarrollados, también se encuentra el archivo “Peso propio”, en el cual se realizó el análisis estructural de una armadura de dos formas, en la primera forma solo se consideró una carga distribuida por peso propio (obtenida al dividir el peso propio de la armadura entre la longitud de la brida superior) aplicada sobre la brida superior, haciendo uso de los scripts “LONGITUD” Y “PP(kN)” y en la segunda forma solo se considera el peso propio según Karamba, entonces se calcularon las fuerzas actuantes en los elementos de esas dos formas, obteniendo así resultados similares, la máxima diferencia es de 8.863914kg a tensión en una montante (para visualizar los resultados a detalle revisar el archivo en Grasshopper), dicha diferencia no es determinante en el diseño de los miembros o elementos de armadura, no es una variación excesiva que perjudique el diseño, entonces se puede usar una carga distribuida por peso propio aplicada sobre la brida superior en lugar del peso propio según Karamba. El uso de una carga distribuida por peso propio aplicada sobre la brida en lugar del peso propio según Karamba se justifica porque Karamba solo calcula el peso propio mas no realiza combinaciones de carga usando factores de carga para cada tipo de carga (caso de las combinaciones de carga del LRFD), entonces la única forma de usar las combinaciones de carga del LRFD es trabajando con una carga distribuida por peso propio aplicada sobre la brida superior (la cual formara parte de la carga muerta).