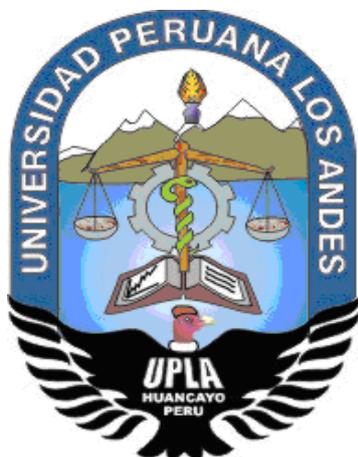


UNIVERSIDAD PERUANA LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA CIVIL



TESIS

**ANÁLISIS Y DISEÑO DE TECHOS CON TIJERALES METÁLICOS TIPO
HOWE MEDIANTE EL VISUAL C# Y EL API DEL SAP2000.**

PRESENTADO POR:

Bach. Ing.: Edwin Rubén Llacza Rojas

Línea de Investigación Institucional

Nuevas Tecnologías y Procesos

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO CIVIL

Huancayo – Perú

2021

ASESORA:

MG. Lourdes Graciela POMA BERNAOLA

El presente proyecto de tesis va dedicado a mi familia, mi Esposa y mi hijo Diago, que cada día me dan fortaleza de seguir superándome en lo personal, académico y sobre todo familiar.

Agradezco a mi asesora de Tesis, Mg. Lourdes Poma Bernaola, por dedicarme el tiempo necesario y el apoyo con su conocimiento para el logro de este proyecto de Tesis.

Agradezco a todas las personas involucradas para realizar este proyecto de Tesis, en especial a mi buen amigo Ing. Abel Quichca, que es un experto en estructuras, que con su ayuda hice realidad este proyecto.

HOJA DE CONFORMIDAD DE JURADOS

DR. RUBEN DARIO TAPIA SILGUERA
PRESIDENTE

ING. CARLOS GERARDO FLORES ESPINOZA
JURADO

ING. VLADIMIR ORDOÑEZ CAMPOSANO
JURADO

ING. NATALY LUCIA CORDOVA ZORRILLA
JURADO

MG. LEONEL UNTIVEROS PEÑALOZA
SECRETARIO DOCENTE

INDICE

1.	CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA	12
1.1.	Descripción de la realidad problemática	12
1.2.	Delimitación del problema	13
1.2.1.	Delimitación espacial	13
1.2.2.	Delimitación temporal	13
1.3.	Formulación del problema	14
1.3.1.	Problema General.....	14
1.3.2.	Problemas Específicos	14
1.4.	Justificación de la Investigación	14
1.4.1.	Justificación social	14
1.4.2.	Justificación teórica	14
1.4.3.	Justificación metodológica.....	15
1.5.	Objetivos	15
1.5.1.	Objetivo General.....	15
1.5.2.	Objetivos Específicos.....	15
2.	CAPÍTULO II MARCO TEÓRICO	16
2.1.	Antecedentes	16
2.1.1.	Antecedente Internacional	16
2.1.2.	Antecedente Nacional	18
2.1.3.	Antecedente Local	20
2.2.	Bases Teóricas o Científicas	20
2.2.1.	Introducción al Diseño Estructural en Acero.....	20
2.2.2.	Tipos de Estructuras de Acero	23
2.2.3.	Especificaciones y Códigos de Construcción	26
2.2.4.	Cargas y Combinaciones de Carga	28
2.2.5.	Métodos de diseño de Miembros de Acero Estructural	31
2.2.6.	Techos	32
2.2.7.	Lenguaje de programación Visual C#.....	41
2.2.8.	API del SAP2000 V16.0	43
2.3.	Marco Conceptual	44
3.	CAPÍTULO III HIPÓTESIS.....	50
3.1.1.	Hipótesis General.....	50

3.1.2.	Hipótesis Específicas	50
3.1.3.	Variables	50
4.	CAPÍTULO IV METODOLOGÍA	52
4.1.	Método de Investigación	52
4.2.	Tipo de Investigación	52
4.3.	Nivel de Investigación.....	52
4.4.	Diseño de Investigación	53
4.5.	Población y Muestra.....	53
4.5.1.	Población.....	53
4.5.2.	Muestra.	53
4.6.	Técnicas e Instrumentos de recolección de datos.....	53
4.6.1.	Técnicas	53
4.6.2.	Instrumentos.....	53
4.7.	Técnicas de procesamiento y análisis de datos	53
4.8.	Aspectos éticos de la Investigación.....	54
5.	CAPITULO V RESULTADOS	56
5.1.	Descripción de los Resultados.....	56
5.1.1.	Análisis y diseño de techos con tijerales metálicos tipo Howe mediante el visual C# y el API del SAP2000	56
5.1.2.	Análisis y diseño de techos con tijerales metálicos tipo Howe mediante la forma tradicional.	69
5.2.	Contrastación de hipótesis.....	81
5.2.1.	Hipótesis específica 1.	81
5.2.2.	Hipótesis específica 2	91
5.2.3.	Hipótesis específica 3	94
6.	ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	96
7.	CONCLUSIONES	97
8.	RECOMENDACIONES.....	98
9.	REFERENCIAS BIBLIOGRÁFICAS.....	99
10.	ANEXOS	100
10.1.	MATRIZ DE CONSISTENCIA.....	101
10.2.	PLANO ESTRUCTURAL DEL TIJERAL METALICO TIPO HOWE.	102
10.3.	DATA – CODIGO FUENTE DEL SOFTWARE HOWESTEEL.....	103

INDICE DE TABLAS

Tabla 1. Pendientes según zona bioclimática.	40
Tabla 2. Operacionalización de la variable independiente.	51
Tabla 3. Operacionalización de la variable dependiente.	51
Tabla 4. Datos de la población.....	54
Tabla 5. Disgregado de carga muerta.	58
Tabla 6. Disgregado de carga muerta.	70
Tabla 7. Factores de forma (C)	73
Tabla 8. Factores de forma para determinar cargas adicionales en elementos de cierre (C)...	73
Tabla 9. Valores para el Barlovento y Sotavento.	74
Tabla 10. Resumen de Metrado de cargas.	75
Tabla 11. Tiempos promedio durante el proceso de análisis y diseño.....	92
Tabla 12. Procesos en el análisis y diseño.	94

INDICE DE FIGURAS

Figura 1. Diagrama de esfuerzo-deformación característico de un acero estructural.....	22
Figura 2. Estructura metálica abovedada.	23
Figura 3. Estructura metálica entramada.	24
Figura 4. Estructura metálica triangulada.	24
Figura 5. Estructura metálica colgante.....	25
Figura 6. Estructura metálica laminar.	26
Figura 7. Acción del viento sobre una construcción con techo a dos aguas.....	29
Figura 8. Mapa eólico.	30
Figura 9. Armadura – Configuración completa.	34
Figura 10. Armadura – Configuración incompleta.	34
Figura 11. Armadura – Falla por cargas asimétricas.	34
Figura 12. Armadura – Configuración redundante.	35
Figura 13. Partes de una Armadura tipo Howe.....	36
Figura 14. Armadura tipo “A”.	36
Figura 15. Armadura de montante maestro.....	37
Figura 16. Armadura tipo Howe.	37
Figura 17. Armadura tipo Pratt.	37
Figura 18. Armadura tipo Warren.....	38
Figura 19. Armadura tipo Belga.	38
Figura 20. Armadura tipo Fink.	38
Figura 21. Armadura tipo Dientes de sierra.....	39
Figura 22. Armadura tipo Polonceau.....	39
Figura 23. Otros tipos de Armaduras.....	39
Figura 24. Pendiente de una armadura.....	40
Figura 25. Pendiente de una armadura y su equivalente en grados sexagesimales.	40
Figura 26. Componentes de .NET Framework.	42
Figura 27. Descripción de resultados - Predimensionamiento del tijeral metálico para calculo en el software HoweSteel.....	57
Figura 28. Descripción de resultados - Ingreso de datos en la pestaña “Editar Tijeral” en el software HoweSteel.	60

Figura 29. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Brida, del software HoweSteel.	60
Figura 30. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Montante, del software HoweSteel.	61
Figura 31. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Diagonal, del software HoweSteel.	61
Figura 32. Descripción de resultados - Ingreso de datos en la pestaña “Ingresos de Cargas”, en el software HoweSteel.....	62
Figura 33. Descripción de resultados - Verificación en la pestaña “Datos Finales”, en el software HoweSteel.	62
Figura 34. Descripción de resultados - Ingreso a la aplicación del SAP2000, mediante el botón “Abrir SAP2000”.....	63
Figura 35. Descripción de resultados - Espacio de trabajo del SAP2000, previo enlace con el software HoweSteel.	63
Figura 36. Descripción de resultados - Datos procesados después de ingresar en el botón “Análisis SAP2000”, del software HoweSteel.	64
Figura 37. Descripción de resultados - Verificación de los esfuerzos últimos (opcional).	64
Figura 38. Descripción de resultados - Verificación de los esfuerzos para cada Frame (Opcional).	65
Figura 39. Descripción de resultados - Obtención de los esfuerzos últimos críticos en la pestaña “Resultados de análisis” del software HoweSteel.	65
Figura 40. Análisis Convencional - Predimensionamiento del tijeral metálico para el cálculo en el software HoweTeel.	70
Figura 41. Análisis Convencional - Mapa de Isotacas, para hallar la velocidad del viento. ...	72
Figura 42. Análisis Convencional - Determinación de la carga puntual en los nudos.	75
Figura 43. Análisis Convencional - Diseño para la verificación manual de las esfuerzos últimos.....	77
Figura 44. Análisis Convencional - Verificación manual para la carga muerta (D).....	77
Figura 45. Contrastación Hipótesis - Predimensionamiento del tijeral metálico para el cálculo en el software HoweSteel.....	82
Figura 46. Contrastación Hipótesis – Ingreso de datos en la pestaña “Editar Tijeral” en el software HoweSteel.	82
Figura 47. Contrastación Hipótesis – Ingreso de datos en la pestaña “Editar Perfiles”, en el software HoweSteel.	83

Figura 48. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la brida en el software HoweSteel.....	83
Figura 49. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la Montante en el software HoweSteel	84
Figura 50. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la Diagonal en el software HoweSteel.....	84
Figura 51. Contrastación Hipótesis – Ingreso de datos en la pestaña “Ingreso de Cargas”, en el software HoweSteel.	85
Figura 52. Contrastación Hipótesis – Verificación en la pestaña “Datos Finales”, en el software HoweSteel.	86
Figura 53. Contrastación Hipótesis – Datos procesados después de ingresar en el botón “Análisis SAP2000”, del software HoweSteel.	86
Figura 54. Contrastación Hipótesis – Verificación de los esfuerzos últimos (opcional).....	87
Figura 55. Obtención de los esfuerzos críticos en la pestaña “Resultados de análisis” del software HoweSteel.	87
Figura 56. Encuesta para determinar el tiempo promedio en el predimensionamiento tradicional.	91
Figura 57. Encuesta para determinar el tiempo promedio en el análisis tradicional.	92
Figura 58. Encuesta para determinar el tiempo promedio en el diseño tradicional.	92

RESUMEN

La presente investigación nace del auge que viene ocurriendo en nuestra ciudad de Huancayo en las construcciones de techos metálicos para las diversas infraestructuras como: almacenes, campos deportivos, terminales terrestres, terminales ferroviarios y tiendas comerciales. Para poder tener una buena estructura, se debe de tener un buen análisis y diseño de la misma, por lo que tradicionalmente o convencionalmente, el proyectista o consultor realiza una serie de procesos que conlleva a utilizar diversos software para el análisis y diseño; estos procesos se vuelven cíclicas hasta encontrar un diseño optimo, generando tiempos extensos y procesos largos, entonces aquí entra a detallar esta investigación en tecnologías y procesos, con la utilización del lenguaje de programación visual C# y el API del SAP2000 se realizó una aplicación denominada HoweSteel, acorde con nuestra norma E.090, E.020 del RNE y el AISC, la cual nos permitirá optimizar tiempos y minimizar procesos en el análisis y diseño de los techos con tijerales metálicos tipo howe, dándonos los resultados de las cargas finales para su verificación y toma de decisiones por parte del consultor.

Así mismo se formula como problema general ¿Cómo es el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000?, el objetivo general: Conocer el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, y la hipótesis será: El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es conveniente.

Palabras clave: C#, tijerales

ABSTRAC

This research is born from the boom that has been occurring in our city of Huancayo in the construction of metal roofs for various infrastructures such as: warehouses, sports fields, land terminals, rail terminals and commercial stores. In order to have a good structure, you must have a good analysis and design of it, so traditionally or conventionally, the designer or consultant performs a series of processes that lead to using various software for analysis and design; These processes become cyclical until an optimal design is found, generating long times and long processes, then this research on technologies and processes goes into detail, with the use of the visual programming language C # and the SAP2000 API, an application called HoweSteel was made , in accordance with our standard E.090, E.020 of the RNE and the AISC, which will allow us to optimize times and minimize processes in the analysis and design of ceilings with metallic howe trusses, giving us the results of the final loads for their verification and decision making by the consultant. Likewise, it is formulated as a general problem: How is the analysis and design of roofs with howe metal trusses using visual C # and the SAP2000 API? The general objective: To know the analysis and design of roofs with howe metal trusses using the visual C # and the SAP2000 API, and the hypothesis will be: The analysis and design of roofs with howe metal trusses using visual C # and the SAP2000 API, is convenient

CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la realidad problemática

La ingeniería en nuestro país, es un reto para cada uno de las personas que siguen esta profesión, y la Ingeniería Civil, que es considerada una mezcla entre arte y ciencia, cada vez es mejorada con los principios y teorías de la estática, dinámica, mecánica de materiales y el respectivo análisis estructural, obteniendo un trabajo final que puede ser una estructura cuales quiera que perdure con el tiempo.

Es muy importante que una estructura considere lo mínimo (costo, peso, tiempo, trabajo y máxima eficiencia), para lo cual, las estructuras hechas con materiales de acero, es una opción óptima para lograr esas cualidades, que como profesionales y sociedad buscamos. Así mismo cabe señalar que en nuestro país, nos basamos en normas y reglamentos internacionales dando ajustes a nuestra realidad, nuestra norma E.090 “Estructuras metálicas”, contempla lo mínimo para un análisis estructural para estructura metálicas, y he ahí el reto que como profesionales debemos asumir.

En nuestro entorno es fácil observar el auge de los techos con estructuras metálicas como: almacenes, campos deportivos, viviendas , centros comerciales y centros de salud, que se espera que sean bastante buenas para soportar las cargas más críticas que pueda haber en su vida útil, con respecto a esto, podemos mencionar que el año 2019 hubo un desastre en un local comercial ubicado en el distrito de Huancan, provincia de Huancayo, el cual el techo metálico colapsó por causa de las intensas lluvias y el techo no soporto el peso, dejando 10 muertos por este accidente. Entonces podemos evidenciar que no se están considerando un buen análisis y diseño de la estructura.

Además, hay muchas coberturas metálicas, que no cumplen con las Normas Técnicas Peruanas, que están propensos a colapsar. Para lo cual, se pretende hacer uso de software

especializado y automatizar el análisis y diseño de techos metálicos con tijerales tipo HOWE, empleando una base de datos con los diversos perfiles de acero que cuenta el mercado peruano. En la etapa de análisis y diseño tradicional o convencional, se suele trabajar con un formato en Excel, que previamente el Ing. Civil lo diseña con los datos obtenidos en campo, adicional a esto, se hace el modelamiento en el programa SAP2000 u otro software especializado.

En la ciudad de Huancayo la construcción de techos de cobertura metálica viene en auge como ya lo dijimos, debido al funcionamiento, la facilidad de su transporte y colocación. Por ello, el trabajo de investigación busca determinar: optimizar tiempos y minimizar procesos en el análisis y diseño en tijerales metálicos tipo HOWE, en techos de mediana longitud, específicamente se tomará como estudio, la cobertura de un ambiente del Terminal Terrestre los Andes, ubicado en el distrito de Huancayo; la misma que permitirá realizar un análisis y diseño con el software HoweSteel, para que finalmente se pueda apreciar un diseño óptimo, que sea capaz de optimizar tiempos y minimizar procesos, basado en la norma E.090 “Estructuras metálicas”, E.020 “Cargas” del Reglamento Nacional de Edificaciones, AISC “American Institute of Steel Construction” y con la ayuda del lenguaje de programación visual C# y el API del SAP2000.

1.2. Delimitación del problema

1.2.1. Delimitación espacial

El estudio delimita espacialmente al departamento de Junín, provincia de Huancayo, distrito de Huancayo, cercado de Huancayo.

1.2.2. Delimitación temporal

El estudio se delimita temporalmente al año 2021; en el periodo de Enero a Diciembre.

1.3. Formulación del problema

1.3.1. Problema General

¿Cómo es el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000?

1.3.2. Problemas Específicos

- a) ¿Cuál es el resultado del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000?
- b) ¿Qué efectos tiene el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, en la optimización de tiempos respecto a un análisis y diseño tradicional o convencional?
- c) ¿Qué efectos tiene el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, en la minimización de procesos respecto a un análisis y diseño tradicional o convencional?

1.4. Justificación de la Investigación

1.4.1. Justificación social

La investigación final alcanzará un propósito social, el de poder ser utilizado para fines de análisis y diseño automatizado, optimizando tiempos y minimizando procesos en el análisis y diseño de techos con tijerales metálicos tipo howe. La optimización de tiempos y minimización de procesos, lo podemos hacer utilizando el software HoweSteel, creado con el lenguaje de programación C# y al API del SAP2000, esto hará la vida más fácil al consultor o proyectista.

1.4.2. Justificación teórica

La investigación tiene como finalidad conocer y evaluar los aspectos teóricos, prácticos y técnicos de la modelación de los techos con tijerales metálicos tipo howe con el lenguaje de programación C# y el API del SAP2000 a fin de optimizar tiempos y minimizar procesos para

encontrar un diseño óptimo, todo esto basado en la norma E.090 y E.020 del Reglamento Nacional de Edificaciones y del AISC.

Con el código fuente de la programación, se tiene una base, para poder ampliar los conocimientos y enfocarlo al tema BIM, la cual está en auge en estos tiempos.

1.4.3. Justificación metodológica

La investigación tiene como objetivo: Determinar el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, según el método científico, además una vez que sean demostrados su utilidad y fiabilidad de los datos, se podría emplear en otros trabajos de investigación y en otras realidades.

1.5. Objetivos

1.5.1. Objetivo General

Conocer el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000.

1.5.2. Objetivos Específicos

- a) Desarrollar el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000.
- b) Demostrar que el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, tiene efectos positivos en la optimización de tiempos, respecto a un análisis y diseño tradicional o convencional.
- c) Comparar la cantidad de procesos del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, frente a los procesos de un análisis y diseño tradicional o convencional.

CAPÍTULO II MARCO TEÓRICO

2.1. Antecedentes

2.1.1. *Antecedente Internacional*

CHAPULA. Procedimiento constructivo con vigas metálicas. Universidad Nacional Autónoma de México, para optar el título de Ingeniero Civil, 2014. Esta tesis tiene como finalidad, dar soporte y base en la materia de "Programación y Construcción de Estructuras", para facilitar los estudios de la construcción con estructuras metálicas, a su vez está dirigido a los alumnos de pre grado de Ingeniería Civil. Con estos conocimientos se podrá aprender las actividades y procesos más representativos del procedimiento constructivo con estructuras metálicas, desde la fabricación de diversos tipos de perfiles metálicos hasta su empalme o colocación final. Posee un tipo de investigación aplicada, con un diseño experimental y de generación de conocimientos. La tesis resalta el uso del acero como parte de la estructura, la cual es algo difícil de lograr con otros materiales como: ampliación vertical y horizontal, buena relación entre resistencia y peso y reutilización de materiales.

HERNÁNDEZ, LINARES Y SORIANO. Guía de análisis y diseño para vigas de alma llena y armaduras metálicas en estructuras de soporte de techos y entrepisos. Universidad de El Salvador, facultad multidisciplinaria oriental, departamento de Ingeniería y Arquitectura, para optar el título de Ingeniería Civil, 2012. El objetivo que considera esta tesis, se basa en proporcionar una base teórica para el análisis y diseño de vigas de alma llena y armaduras metálicas para techos y entrepisos, destinados a profesionales y personas inmersas en este campo de estructuras metálicas, la tesis opta por una investigación analítica, así mismo la tesis estudia y ejemplifica el diseño de vigas armadas metálicas de alma abierta mediante las metodologías de diseño manual y mediante software. Finalmente, la tesis llega a la conclusión de que existe una variedad de casos particulares de estructuras metálicas con vigas de alma abierta y armaduras, para lo cual mediante esta tesis se busca una metodología que ayude a

comprender el funcionamiento de sus elementos y ayudar a la toma de decisiones en el diseño de estas.

OVANDO. Diseño y cálculo de una pasarela de metal. Universidad San Carlos de Guatemala” para optar el título de Ingeniero Civil, 2011. La presente tesis tiene como Objetivo, el de brindar ayuda teórica y práctica en el cálculo estructural, así como las dimensiones y detalles en el proceso constructivo de una pasarela de metal. Considerando las cargas de diseño de los diferentes elementos estructurales, realizando un análisis previo y detallando los esfuerzos internos y externos que actúan, para lo cual detalla un ejemplo desde la planificación hasta su montaje final, así mismo considera las bases teóricas necesarias para el diseño final, que sea segura y no poner riesgo a la población que harán uso de la estructura. Finalmente, la conclusión de la tesis habla de lo óptimo y capacidad del acero estructural en la construcción de un puente peatonal, donde detalla la versatilidad a la hora de unir elementos (soldadura, pernos y remaches), rapidez de su montaje y la adaptabilidad de los elementos a posibles ampliaciones existentes.

VILLAR. Extensión para el modelamiento, análisis y diseño automatizado de techos retractiles en SAP2000 con paneles solares integrados a la cubierta. Universidad Nacional de Colombia, para optar el título de Magister en Construcción, línea de investigación “Estructuras”, 2017. La presente tesis tiene como objetivo, desarrollar un programa que puede ser aplicado para el modelamiento, análisis y diseño automatizado de techos retractiles con paneles solares integrados a la cubierta, utilizando el API del SAP2000 y el lenguaje de programación visual basic para la creación de una aplicación final, capaz de automatizar los procesos de modelamiento, análisis y diseño. Así mismo, la investigación proporciona una alternativa de solución capaz de aprovechar el espacio generado por las coberturas tradicionales por una estructura de techo retráctil con paneles solares aprovechando la energía limpia. Posee un tipo de investigación con enfoque cualitativo y descriptivo. La conclusión final es la

demostración de la automatización de los procedimientos del modelamiento, análisis y diseño de una estructura con techo retráctil en movimiento paralelo.

2.1.2. Antecedente Nacional

MENDOZA, M. Análisis comparativo entre coberturas reticuladas convencionales y auto portantes, caso aplicativo: centro de convenciones de la universidad nacional del altiplano-puno, para optar el título profesional de ingeniero civil. Universidad Nacional del Altiplano. Puno – Perú, 2017. El objetivo general de esta tesis, es la de evaluar y comparar el comportamiento estructural y los costos que pueden ocasionar un sistema con cobertura convencional con una de auto portante; esto será aplicado en la cobertura del Centro de Convenciones de la Universidad Nacional del Altiplano Puno, propone una solución estructural para el soporte de coberturas, con un sistema auto portante las cuales son más livianas y aprovecha los espacios. La metodología de la investigación es de tipo descriptiva. Finalmente llega a la conclusión: La cobertura auto portante resulta más económico que una cobertura reticulada, con un ahorro promedio del 2%, esto según el análisis de costos unitarios que se ha realizado.

ORIHUELA, FRANCESCO Cálculo y diseño estructural de una nave industrial aplicando la normativa AISC en la ciudad de Juliaca, provincia de San Ramón. Universidad Peruana de Ciencias Aplicadas, para optar el título de Ingeniero Civil, 2016. El objetivo general que plantea la tesis, es la de modelar, diseñar y calcular una nave industrial ubicado en la provincia de Juliaca, utilizando la norma del AISC conjuntamente con el método LRFD, para lo cual utiliza el software SAP2000, una vez completado el modelamiento, se procede a diseñar cada elemento estructural en función a las ecuaciones y por último detalla los planos respectivos de la nave industrial. El tipo de metodología utilizada es una investigación descriptiva. Finalmente llega a la conclusión de que la tesis sirve como apoyo teórico y práctico

para el análisis y diseño de los diferentes elementos estructurales que se puedan plantear, así mismo la metodología para el diseño incluye la verificación por tensión, comprensión corte, torsión, flexión y deflexión.

DIAZ, K. Evaluación y optimización en el diseño de estructuras metálicas aplicado a cubiertas según las especificaciones AISC- 2010 por el método LRFD, para optar el título profesional de ingeniero civil. Universidad Nacional de Cajamarca. Facultad de Ingeniería. Cajamarca – Perú, 2014. La tesis tiene como objetivo, realizar un evaluación y determinar el mas optimo entre una estructura compuesta por elementos espaciales, por tubos HSS y por elementos de alma llena; para que posteriormente el diseño optimo se pueda evaluar frente a esfuerzos presentes en el medio local como sismo (estática y dinámica), cargas de servicio, cargas de lluvia y cargas de viento, luego se comprobó manualmente los resultados obtenidos del Software computacional SAP2000 v16 para realizar una comprobación manual. Finalmente, la conclusión indica que los sistemas espaciales resultan ser mas livianos y de mejor comportamiento estructural, por lo que es ideal su uso tanto por su función estructural y el tema económico.

ESCOBAR, K. Optimización del diseño de estructuras de acero en cobertura mediante el método de algoritmos genéticos. Para optar el titulo profesional de ingeniero civil. Universidad Nacional de Huancavelica – Perú, 2019. La tesis tiene como objetivo general, determinar la efectividad del método de algoritmos genéticos en la optimización del diseño de estructura de acero en coberturas, reduciendo el material de acero y el costo final. La automatización se realiza mediante el lenguaje de programación fortran y el API del SAP2000 mediante algoritmos genéticos la cual busca aleatoriamente en el espacio de solución de las variables (diagonal, brida inferior, brida superior, correa interior, correa exterior), con ciertas restricciones y con una aproximación de optimizar una sección definida. La conclusión final

menciona que el método de algoritmos genéticos es eficiente en la optimización de diseño de estructuras de acero en coberturas tipo arco y reduce materiales.

2.1.3. Antecedente Local

LIMAYLLA, D. Alternativas de techo y entrepiso con estructuras metálicas en ambientes de luces grandes, para optar el Título Profesional de Ingeniero Civil. Facultad de Ingeniería. Universidad Continental. Huancayo, 2019. La tesis tiene como objetivo general: realizar y desarrollar diferentes alternativas de propuesta para un techo y entrepiso de estructura metálica, así como también vigas de soporte, todo esto aplicadas para ambientes con luces grandes y como aplicación se tendrá un edificio de 3 pisos. Las alternativas cumplen con el análisis estructural por cargas gravitacionales, cargas accidentales y sus efectos que conllevan. El análisis estructural se realizó con el software SAP2000 y ETABS, y el diseño estructural se realizó según el “Reglamento Nacional de Edificaciones”, y sus determinadas normas y reglamentos como E.020 “Cargas”, E030 “Diseño Sismo Resistente”, E.060 “Concreto Armado” y E.090 “Estructura Metálicas”, también, se tuvo como referencia el AISC. Finalmente llega a concluir: un sistema de vigacero, es una estructura mas eficiente, teniendo menos peso, admitiendo que su instalación es más fácil y rápido y no requiere apoyos intermedios en tramos largos.

2.2. Bases Teóricas o Científicas

2.2.1. Introducción al Diseño Estructural en Acero

El campo de la construcción en Perú ha estado en constante crecimiento, a la par con nuevas tecnologías y mejoras en materiales de construcción. A finales del siglo XIX en Estados Unidos, el acero se fabricó económicamente, y en el Perú se viene fabricando aproximadamente desde 1960. “La supuesta perfección de este metal, tal vez el más versátil de todos los

materiales estructurales, parece más razonable cuando se considera su gran resistencia, poco peso, facilidad de fabricación y otras propiedades convenientes” (MacCormac,2012, p.14). Todo esto hizo revolucionar los trabajos de construcción tradicionales y se han visto conveniente el reemplazo por sistemas relativamente nuevos utilizando el acero estructural (edificios, puentes, coliseos, almacenes, campos deportivos y variedad de techos).

La calidad del acero ha evolucionado y es considerado como un material de alta resistencia en comparación con otros materiales disponibles; el acero estructural moderno tiende a mejorar el material, es por ello que se fabrica aceros con fluencia $F_y = 50 \text{ Klb/plg}^2$, que es mejor que un acero con fluencia $F_y = 36 \text{ Klb/plg}^2$ comúnmente utilizado. En el mercado peruano, encontramos los perfiles: Angulos, tees, canales (U) fabricados con el estándar internacional ASTM A36, y el perfil W fabricado con el estándar internacional ASTM A992 especialmente para columnas y vigas.

2.2.1.1. Ventajas del acero como material estructural

El acero estructural, como material de una supuesta perfección, se aprecia en diferentes estructuras como edificios, almacenes, canchas deportivas, puentes y coberturas de viviendas, porque se concluye que son más versátiles de todos los materiales estructurales, gran resistencia, poco peso, elasticidad, facilidad de fabricación es el material ideal para una construcción, uniformidad, durabilidad, ductilidad, tenacidad, elasticidad y capacidad de ampliaciones de estructuras existentes.

2.2.1.2. Propiedades físico – mecánicas del acero estructural

“Si una pieza de acero estructural dúctil se somete a una fuerza de tensión, ésta comenzará a alargarse. Si se incrementa la fuerza de tensión a razón constante, la magnitud del alargamiento aumentará en forma lineal dentro de ciertos límites. El punto más alto de la porción recta con un mayor esfuerzo donde todavía es válida la ley de Hooke se denomina límite proporcional. El mayor esfuerzo que un material puede resistir sin deformarse permanentemente se llama

límite elástico. El esfuerzo en el que se presenta un incremento brusco en el alargamiento o deformación sin un incremento en el esfuerzo, se denomina esfuerzo de fluencia. La deformación que se presenta antes del esfuerzo de fluencia se denomina deformación elástica; la deformación que ocurre después del esfuerzo de fluencia, sin incremento de esfuerzo, se denomina deformación plástica, donde con muy poca tensión de deforma mucho. Después de la deformación plástica, existe un rango en el cual es necesario un esfuerzo adicional para producir deformación adicional, a esto se le denomina endurecimiento por deformación, donde se necesita más tensión para su deformación. Después del punto de endurecimiento por deformación, viene la zona de fractura en la cual la curva alcanza su esfuerzo máximo y luego disminuye poco a poco antes de la falla y su fractura correspondiente” (MacCormac,2012, p.15). En la Figura 1 se muestra un diagrama típico de esfuerzo-deformación.

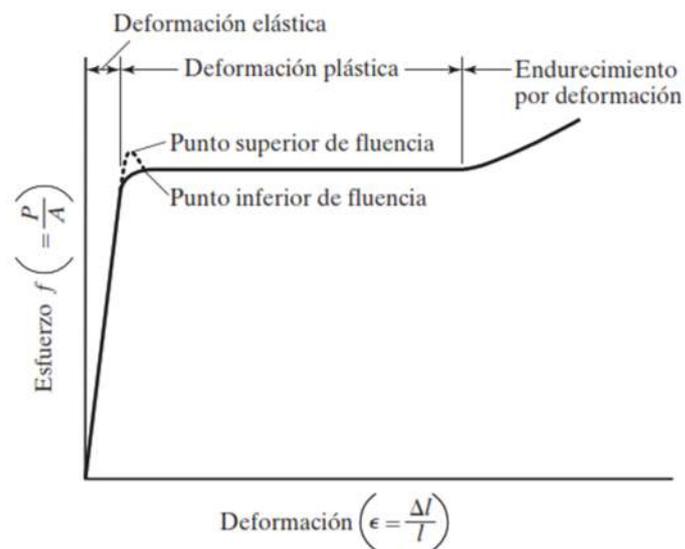


Figura 1. Diagrama de esfuerzo-deformación característico de un acero estructural.

2.2.2. Tipos de Estructuras de Acero

2.2.2.1. Estructuras abovedadas

Estos tipos de estructuras son las que están formadas por elementos de arcos y bóvedas o cupulas, que tienen como finalidad cubrir grandes espacios, dejando huecos en las estructuras y equilibrar el peso de la construcción. Su forma permite cubrir espacios mayores y los elementos reciben y transmiten fuerzas perpendiculares y transversales de tensión o compresión, como por ejemplo se pueden observar en estructuras como: catedrales, iglesias y almacenes.



Figura 2. Estructura metálica abovedada.

2.2.2.2. Estructuras entramadas

Estos tipos de estructuras es bastante utilizada en edificios y viviendas que podemos observar en cualquier ciudad relativamente grande socio/económicamente hablando, son más comunes en esos lugares. Las estructuras entramadas o también llamadas reticulares, se caracterizan por tener elementos horizontales y verticales, las cuales la conforman los cimientos, pilares, columnas y vigas; el peso es más ligero en comparación con las estructuras abovedadas. Los elementos verticales soportan esfuerzos de compresión y corte, en cambio los elementos

horizontales soportan esfuerzos de flexión y compresión. Como estas estructuras son más ligeras se pueden conseguirse edificios de gran altura.



Figura 3. Estructura metálica entramada.

2.2.2.3. Estructuras trianguladas

Estos tipos de estructuras podemos observar en puentes, coberturas y naves industriales, porque la armadura que las conforma puede soportar grandes luces y poseen poco peso y son económicos. Los elementos que los conforma soportan esfuerzos de tensión y compresión y las secciones más utilizadas son miembros de alma llena o de alma de celosía.



Figura 4. Estructura metálica triangulada.

2.2.2.4. Estructuras colgantes

Estos tipos de estructuras son típicas en diferentes casos de puentes, sus elementos que los conforman (cables o tirantes) soportan únicamente esfuerzos de tracción o tensión y tienden a estirarse, además, sirven para aguantar el peso de otros elementos. Los tirantes o cables, deben de estar anclados en cada extremo y su función es la de estabilizar la estructura.



Figura 5. Estructura metálica colgante.

2.2.2.5. Estructuras laminares

Estos tipos de estructuras podemos observarlo en diferentes construcciones de arquitectura moderna (estadios, operas, etc). Su característica principal es que el grosor de sus elementos es menor que su área y curvatura, se asemejan a una cascara de huevo, que puede soportar esfuerzos considerables antes de fallar. Su resistencia radica en la curvatura y permite ahorro en la sección de materiales y el peso, esto las hace bastante eficientes al momento de formar como una especie de paraboloides hiperbólicos.

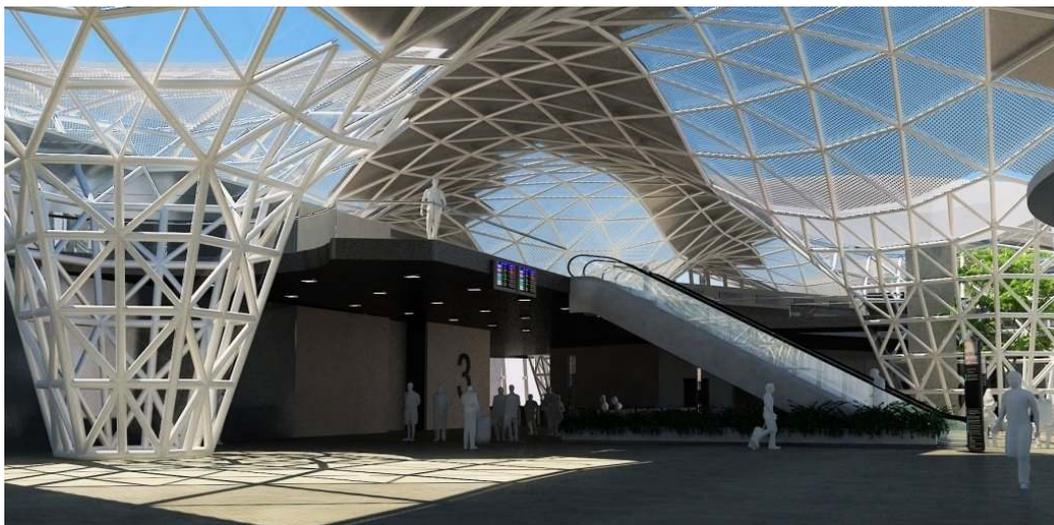


Figura 6. Estructura metálica laminar.

2.2.3. Especificaciones y Códigos de Construcción

Las construcciones en el Perú y otros países internacionales, están sujetas a guías, manuales y normas de diseño, fabricación y montaje para estructuras de acero. Estas normas y reglamentos especifican las cargas de que se utilizaran en el diseño, esfuerzos en el diseño, tipos de construcción, calidad de los materiales a utilizar y otros factores que se requiera.

Nuestro Reglamento Nacional de Edificaciones (RNE) y la norma E.090 “Estructuras metálicas”, reconoce la aplicación y las exigencias mínimas de las especificaciones del American Institute of Steel Construction con sus siglas AISC (Instituto Americano de Construcción en Acero), a raíz de no contar con especificaciones nacionales.

Para el material acero estructural, la norma E.090, aprueba el uso del material que cumpla con los estándares de la norma ASTM, según las siguientes especificaciones:

- ✓ Acero estructural, ASTM A36 (AASHTO M270 Grado36).
- ✓ Tubos redondos de acero negro y galvanizado, soldados y sin costura, ASTM A53, Gr. B.
- ✓ Acero de alta resistencia y baja aleación, ASTM A242.

- ✓ Tubos estructurales de acero al carbono, doblados en frío, soldados y sin costura, ASTM A500.
- ✓ Tubos estructurales de acero al carbono, doblados en caliente, soldados y sin costura, ASTM A501.
- ✓ Planchas de acero aleado, templado y revenido, de alta resistencia, adecuadas para soldadura, ASTM A514 (AASHTO M270 Grado 100 y 100W).
- ✓ Acero al Carbono – Manganeso, de alta resistencia, de calidad estructural, ASTM A529.
- ✓ Planchas y flejes de acero al carbono, laminadas en caliente, de calidad estructural, ASTM A570, Gr. 275, 310 y 345.
- ✓ Acero de alta resistencia y baja aleación al niobio - vanadio, de calidad estructural, ASTM A572 (AASHTO M270 Grado 50).
- ✓ Acero estructural de alta resistencia y baja aleación, con un límite de fluencia mínimo de 345 MPa, de hasta 100 mm de espesor, ASTM A588 (AASHTO M270 Grado 50W).
- ✓ Planchas y flejes de acero de alta resistencia y baja aleación, laminadas en caliente y laminadas en frío, con resistencia mejorada a la corrosión atmosférica, ASTM A606.
- ✓ Planchas y flejes de acero de alta resistencia y baja aleación, con Niobio o Vanadio o ambos, laminadas en caliente y laminadas en frío, ASTM A607.
- ✓ Tubos estructurales de alta resistencia y baja aleación, soldados y sin costura, formados en caliente, ASTM A618.
- ✓ Planchas de acero estructural de baja aleación, templado y revenido, con límite de fluencia mínimo de 485MPa, de hasta 100 mm de espesor, ASTM A852 (AASHTO M270 Grado 70W).
- ✓ Acero estructural para puentes, ASTM A709 Grado 36, 50, 50W, 70W, 100 y 100W.

Las normas, las especificaciones y la toma de decisiones del proyectista, harán que se pueda obtener un resultado más favorable en la protección de los usuarios de las estructuras metálicas finales, evitando desastres por alguna falla de las mismas.

2.2.4. Cargas y Combinaciones de Carga

Una carga, es una fuerza que resulta del peso de los materiales que se utiliza en una construcción, ocupantes y efecto del medio ambiente; las cuales permanecen en la estructura o adheridos a ellas.

Las cargas que actúan en una estructura durante el periodo de su vida útil, deberán ser estimadas en magnitud y ubicación lo más preciso posible; una vez obtenido el valor de las cargas, se procederá a analizar las diferentes combinaciones más desfavorables que puedan suceder en un momento dado.

2.2.4.1. Carga Muerta

Son cargas de magnitud constante que permanentemente están instaladas en un mismo lugar. Estas cargas son debido al peso propio de los varios componentes del edificio y que incluyen el peso de la estructura. Para un edificio con estructura de acero, son cargas muertas las estructuras en sí. Ejemplo: muros, pisos, techos, plomería y los accesorios.

2.2.4.2. Carga Viva

Son aquellas cargas que pueden cambiar de lugar y magnitud. Son causadas cuando una estructura se ocupa, se usa o se le hace el mantenimiento respectivo. Ejemplo: ocupantes, materiales, equipos muebles y otros elementos móviles soportados por la edificación.

2.2.4.3. Carga por Nieve

Son aquellas causadas por la acumulación de la nieve en el techo. En las zonas que son más propensas a esta carga climatológica, deberán de tener una atención especial en la selección adecuada de los materiales para las cubiertas y las pendientes de los techos.

2.2.4.4. Carga por Viento

Son aquellas cargas que son causadas por un fenómeno climático (el viento), las estructuras en general están sujetas a la acción del viento y en especial las de más de 2 o 3 pisos de altura o en aquellas en las zonas donde la velocidad del viento varía significativamente o en las que debido a su forma, son más vulnerable a los efectos aerodinámicos. Se considera que en las estructuras de acero, su peso es relativamente bajo frente a otros materiales y además quedan expuestas a la acción del viento, entonces se puede afirmar que las cargas de viento es más significativa que los esfuerzos debido a las cargas de sismo.

Las cargas de viento son diferentes en cada localidad geográfica y estas pueden variar según: las alturas que llegan a construirse sobre el nivel del terreno, los tipos de espacios o terreno que rodean a los edificios, incluyendo otras estructuras y algunos otros factores. Las fuerzas de viento actúan como presiones sobre las superficies verticales a barlovento, como presiones o succiones sobre superficies inclinadas a barlovento (dependiendo de la pendiente) y como succiones sobre superficies planas y superficies verticales o inclinadas a sotavento (debido a la creación de presiones negativas o vacías).

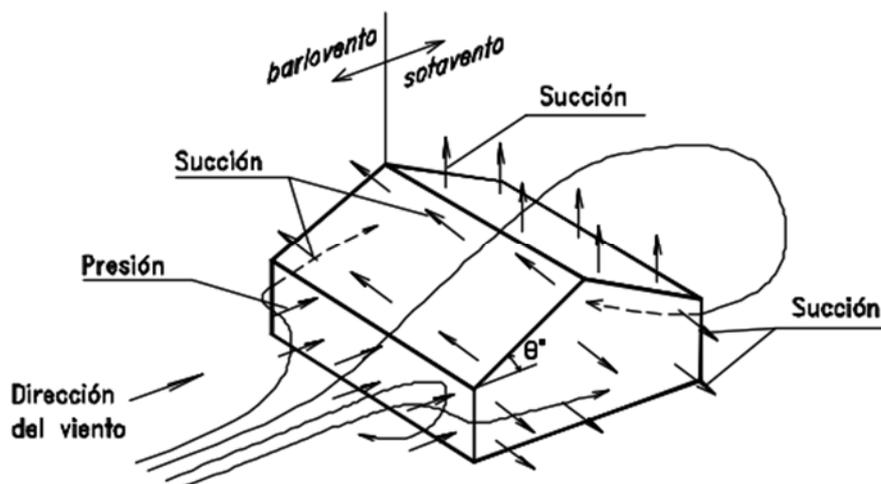


Figura 7. Acción del viento sobre una construcción con techo a dos aguas.

Se deben de considerar diversos aspectos en la selección de una velocidad de viento, sobre la cual se basan las cargas de diseño para edificios u otras estructuras. Según la Norma E.020 del RNE, se utilizará el mapa eólico para establecer las velocidades básicas del viento en la zona donde se ubica la estructura; sin embargo, se debe tener en cuenta la variabilidad debida a las construcciones locales (topográficas, climáticas). Si hubiera mediciones confiables en la zona en cuestión, podrá adoptarse la velocidad proveniente del estudio.

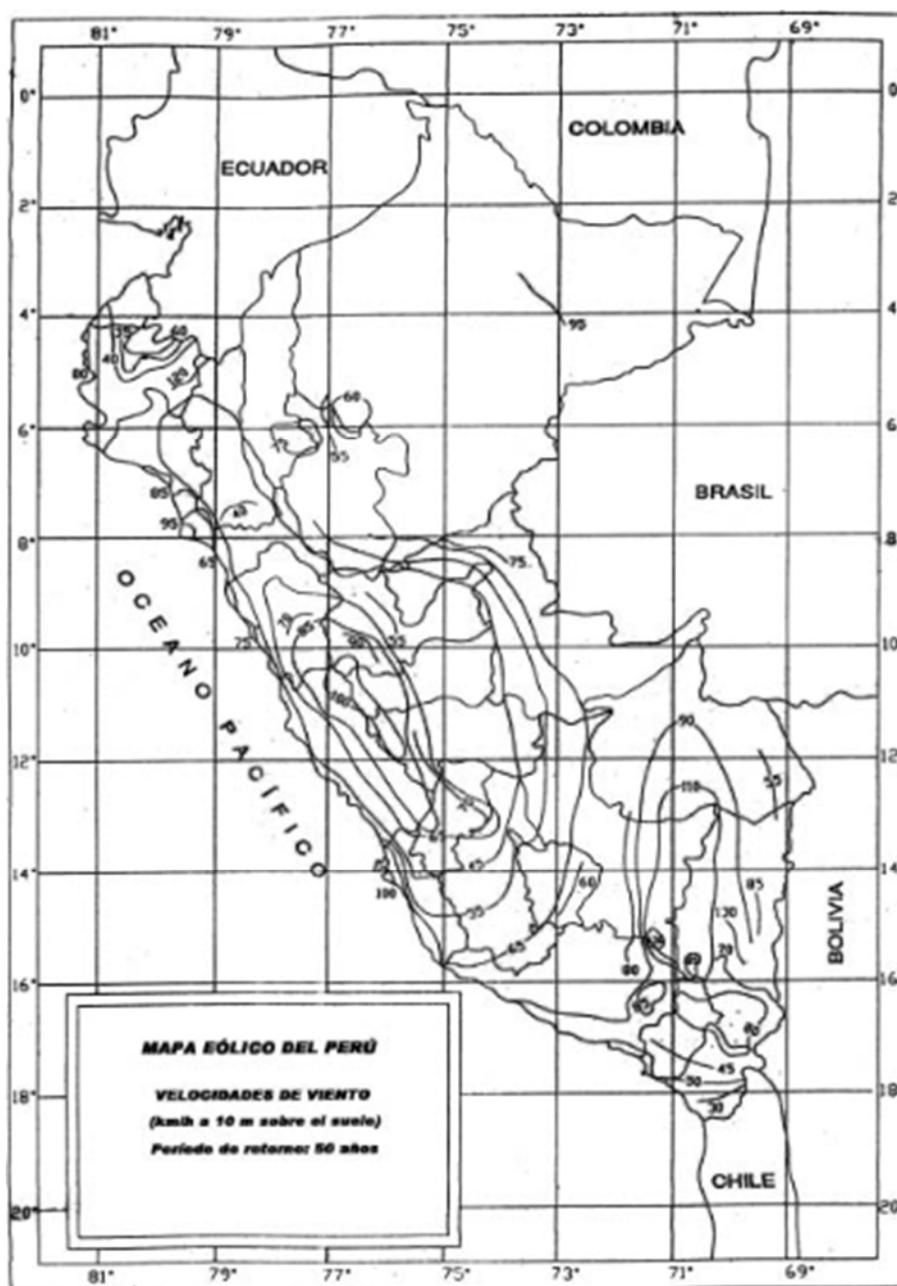


Figura 8. Mapa eólico.

El RNE y la Norma E.090, hace referencia a las cargas de diseño mínimas que establece la Norma E.020 “Cargas” del RNE.

Así mismo, en la Norma E.090, especifica las combinaciones de carga que se deben de utilizar cuando se aplica el método de diseño LRFD, de la cual detallaremos en el ítem 2.2.5 de nuestra investigación. A continuación, se detallan las combinaciones de carga según la Norma E.090:

$$1.4D \quad (1)$$

$$1.2D + 1.6L + 0.5(L_r \text{ ó } S \text{ ó } R) \quad (2)$$

$$1.2D + 1.6(L_r \text{ ó } S \text{ ó } R) + (0.5L \text{ ó } 0.8W) \quad (3)$$

$$1.2D + 1.3W + 0.5L + 0.5(L_r \text{ ó } S \text{ ó } R) \quad (4)$$

$$1.2D \pm 1.0E + 0.5L + 0.2S \quad (5)$$

$$0.9D \pm (1.3W \text{ ó } 1.0E) \quad (6)$$

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 4800 Pa.

2.2.4.5. Cargas de Sismo

La gran mayoría de lugares del mundo están situadas en “áreas sísmicas”, y es necesario considerar en el diseño un análisis sísmico. Durante un sismo, ocurre una aceleración de la superficie del terreno, las cuales pueden ser horizontales y verticales, entonces el análisis debe de ser considerado en respuesta a los efectos esperados de un sismo. Cuando el suelo se mueve, la inercia de la masa de la estructura tiende a resistir tal movimiento y se afirma que las fuerzas debido a la aceleración horizontal se incrementan con las distancias entre el suelo y cada piso de la casa o edificio.

2.2.5. Métodos de diseño de Miembros de Acero Estructural

La especificación AISC proporciona dos métodos aceptables para diseñar miembros de acero estructural y sus conectores, estos son:

- ✓ Diseño con factores de carga y resistencia (LRFD: Load and Resistance Factor Design).
- ✓ Diseño por esfuerzos permisibles (ASD: Allowable Strength Design).

Ambos procedimientos se basan en el principio del diseño de estados límites, el cual proporciona las fronteras de la utilidad estructural. El término estado límite se usa para describir una condición en la que una estructura o parte de ella deja de cumplir su función prescrita.

Existen 2 tipos de estados límites:

- a. Estado límite de resistencia.

Definen la capacidad de sustentar una carga, incluyendo la fluencia excesiva, la fractura, el pandeo, la fatiga y el movimiento bruto de cuerpo rígido.

- b. Estado límite de servicio.

Definen el comportamiento, incluyendo la deflexión, el agrietamiento, los deslizamientos, la vibración y el deterioro. Todos los estados límite deben evitarse.

El LRFD es un enfoque; el ASD es otro. Ambos métodos tienen como objetivo obtener un margen numérico entre la resistencia y la carga que conduzca a una probabilidad aceptablemente pequeña de una respuesta estructural inaceptable.

Hay dos diferencias notorias entre el método LRFD y el ASD. La primera tiene que ver con el método que se usa para calcular las cargas de diseño y la segunda diferencia tiene que ver con el uso de los factores de resistencia (ϕ en el método LRFD) y los factores de seguridad (Ω en el método ASD). Para la presente investigación, se usará el método LRFD.

2.2.6. Techos

Los techos son un conjunto de elementos que forman una estructura destinados a cubrir o proteger de los agentes climatológicos (lluvia, granizo, viento y nieve), un área específico, y dándole un tratamiento de evacuación cuando estas se acumulan.

Los techos se pueden clasificar en provisionales y permanentes, considerando los materiales que se utilicen al realizar su armado. Los techos pueden ser de diferente material como: paja, plástico, teja, madera, piedras, hasta materiales de ladrillo, asbesto, acrílico, concreto y armaduras de acero. En cuanto a su forma tenemos: curvos, planos inclinados, planos horizontales. Los planos inclinados se manufacturan con una gran variedad de materiales y pueden ser de varias aguas.

2.2.6.1. Configuración de Miembros Estructurales

La construcción de un techo metálico para una determinada estructura en la cual la separación entre columnas o muros sean mayores a 40 pies, además entre la separación no exista soportes intermedios, es ideal recurrir a sistemas de estructura capas de soportar las cargas proyectadas y que sean económicas a comparación de una viga simplemente apoyada. La configuración de miembros estructurales que es ideal para este propósito se denomina armadura para techo. Entonces una armadura es el conjunto de elementos que unidos entre ellos nos ofrece una configuración estructural óptima, generalmente están soportada solo en sus extremos y están formadas por una serie de elementos o miembros rectos arreglados y conectados unos a otro que forman la armadura, de tal manera que las cargas que actúan en la armadura, generen esfuerzos que son transmitidos de un miembro a otro ya sean axiales o longitudinales en cada uno de ellos; esto es, esfuerzo de tensión o compresión. La figura geométrica por excelencia que es capaz de soportar esfuerzos sin modificarse geoméricamente es el triángulo; en consecuencia, una armadura optima e ideal está compuesta esencialmente de un sistema de triángulos y que verificadas sus elementos que los componen, deben de tener una ubicación que nos ofrezca solamente esfuerzos de tensión.

2.2.6.2. Tipos de configuración

- a) **Configuración completa.** Es un tipo de armadura que se compone del número mínimo de elemento o miembros necesarios para formar una estructura completamente de triángulos unidos entre ellos.

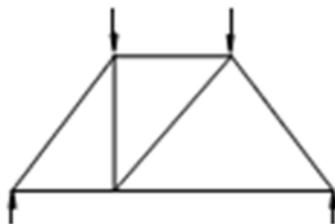


Figura 9. Armadura – Configuración completa.

- b) **Configuración incompleta.** En este tipo de armadura, el entramado o las uniones de sus elementos no forman total o parcialmente un triángulo.

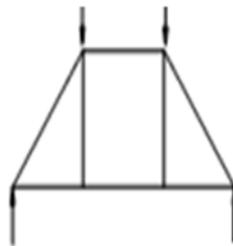


Figura 10. Armadura – Configuración incompleta.

Estos tipos de armaduras pueden fallar cuando existen cargas que son asimétricas, o que son perpendiculares a cada elemento, entonces una configuración incompleta se considera que es inestable y siempre debe evitarse.

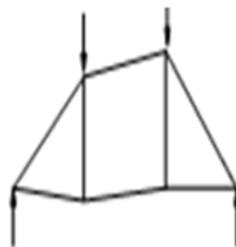


Figura 11. Armadura – Falla por cargas asimétricas.

c) **Configuración redundante.** En estos tipos de armaduras, los elementos que forman un entramado, son mayores que lo requerido para formar el número mínimo de triángulos. Como se muestra en la figura 12, se aprecia dos diagonales en el tablero central; entonces una de las diagonales se llama miembro redundante. Sin embargo, en la práctica estas dos diagonales, formadas de varillas, se usan frecuentemente; como las varillas son capaces de resistir únicamente fuerzas de tensión, de las dos varillas diagonales en el tablero, solamente una de ellas actuara a la vez.

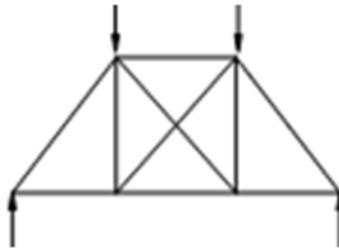


Figura 12. Armadura – Configuración redundante.

2.2.6.3. Partes de una armadura

Una armadura, esta compuesta de varios elementos que están unidos entre ellos por medio de soldadura o empernado, estos elementos son: la brida superior e inferior y por los miembros del alma (montantes y diagonales). La brida superior es la línea de elementos o miembros que se encuentran ubicados en la parte más alta de la armadura que se extiende de un apoyo a otro pasando por la cumbrera. La brida inferior es la línea de elementos o miembros que están ubicados en la parte más baja que va de un apoyo a otro. Los elementos o miembros del alma que conforman la armadura, son las que unen los elementos tanto la brida superior como la brida inferior, y dependiendo de su ubicación se llaman montante o diagonales. En base al tipo de los esfuerzos, los miembros a compresión de una armadura se llaman puntales mientras que aquellos miembros que están sometidos a esfuerzos de tensión se llaman tirantes. La junta en

el apoyo de una armadura triangular se llama junta de talón, y la junta en el pico más alto se le llama cumbre. Los puntos en donde se unen los miembros del alma a las cuerdas reciben el nombre de nudos. Las partes descritas se aprecian en la siguiente figura:

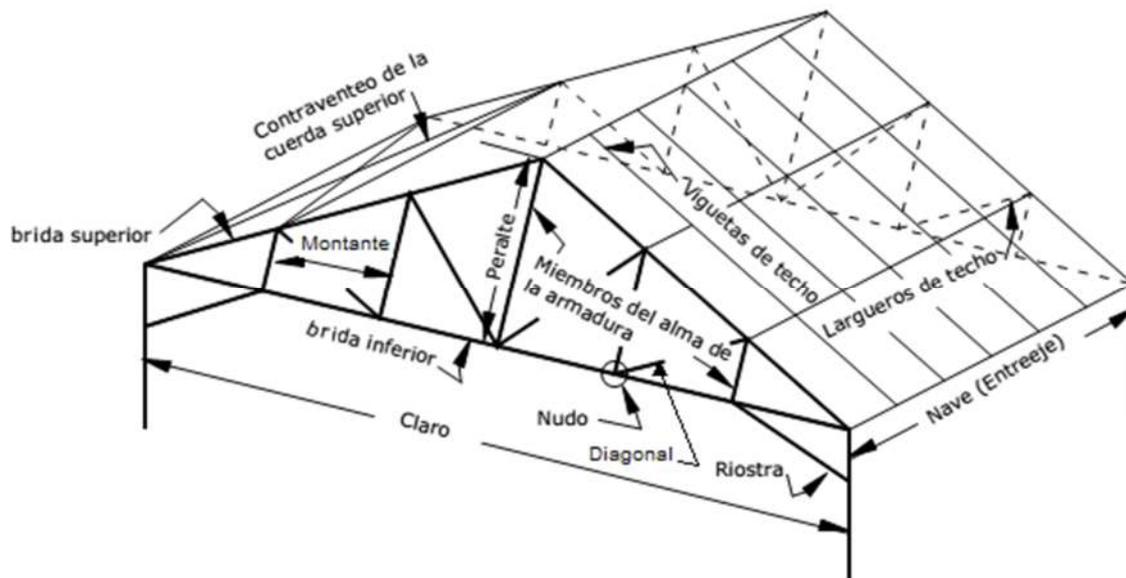


Figura 13. Partes de una Armadura tipo Howe.

2.2.6.4. Tipos de armadura

Las armaduras se pueden clasificar según la forma en que se combinan los diferentes sistemas de triangulación:

- a) **Armadura tipo "A"**. Es considerada la armadura más sencilla que existe y se muestra a continuación:

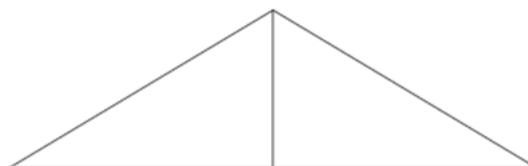


Figura 14. Armadura tipo "A".

- b) **Armadura de montante maestro.** Al igual que la armadura tipo “A” es una de las más sencillas.

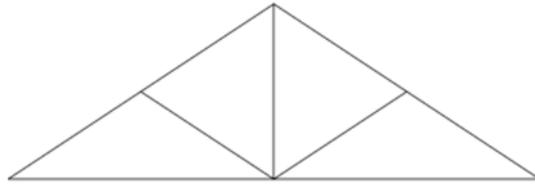


Figura 15. Armadura de montante maestro.

- c) **Armadura tipo Howe.** Es un tipo de armadura usada comúnmente para techos a dos aguas, pueden ser empleadas para salvar claros o luz de hasta 30 metros, sus diagonales trabajan a compresión y las rectas o montantes a tensión. Para la presente investigación nos centraremos exclusivamente en este tipo de armadura.

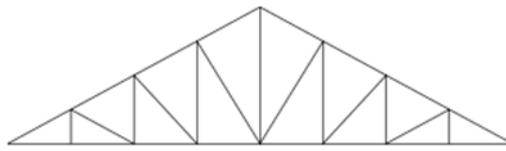


Figura 16. Armadura tipo Howe.

- d) **Armadura tipo Pratt.** También muy común en su uso, se adaptan mejor a construcción en acero que de madera, es muy parecida a la armadura tipo howe, la diferencia radica en la dirección de los miembros diagonales al alma. Pueden ser utilizadas económicamente en techos planos para claros entre 12 y 38 metros.

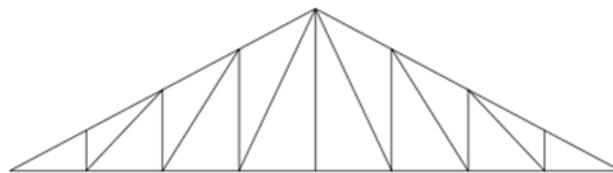


Figura 17. Armadura tipo Pratt.

- e) **Armadura tipo Warren.** Así como las dos predecesoras, bastante utilizada en techos metálicos, generalmente más satisfactoria que la Pratt. Pueden ser utilizadas económicamente en techos planos para claros entre 12 y 38 metros, este tipo de

armadura se pueden utilizar en techos completamente planos para claros que no exceda de 9 ó 12 metros.

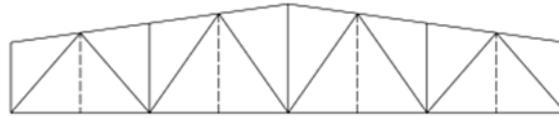


Figura 18. Armadura tipo Warren.

- f) **Armadura tipo Belga.** Se caracterizan por tener diagonales perpendiculares a la brida superior y la brida inferior en tensión. Pueden salvar claros hasta 30 metros.

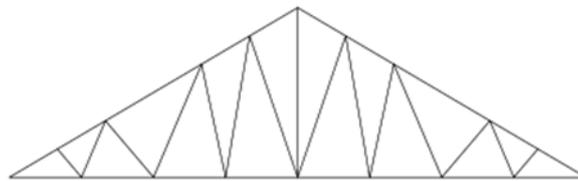


Figura 19. Armadura tipo Belga.

- g) **Armadura tipo Fink.** La estructura Fink ha sido utilizada para claros del orden de 36.5 metros. Es un techo que la hace más económica por tener la mayoría de sus miembros en fuerza de tensión. Las armaduras Fink pueden ser divididas en un gran número de triángulos y coincidir caso con cualquier espaciamiento de largueros. Su uso para techos con pendientes fuerte entre 12.7 y 15.2 cm por metro son muy populares.



Figura 20. Armadura tipo Fink.

- h) **Armadura tipo Dientes de sierra.** Se usan principalmente para los talleres, su propósito es ayudar en la distribución de la luz natural sobre áreas de piso cubiertas. Ayuda a tener claros de hasta 15 metros. Este tipo de armadura es de forma asimétrica, así como también lo son sus cargas.

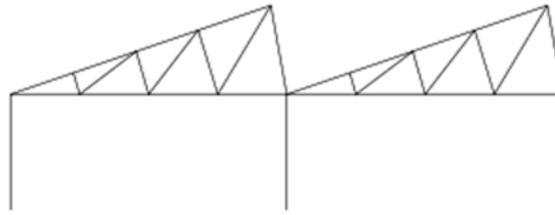


Figura 21. Armadura tipo Dientes de sierra.

i) **Armadura Polonceau.** Se pueden emplear para salvar claros de hasta 24 metros.

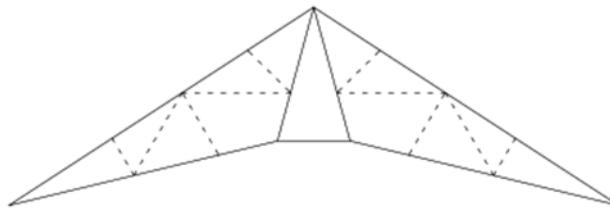


Figura 22. Armadura tipo Polonceau.

j) **Otras Armaduras.** Se menciona también que se puede tener otras formas de las armaduras anteriormente mencionadas, como se muestra a continuación:

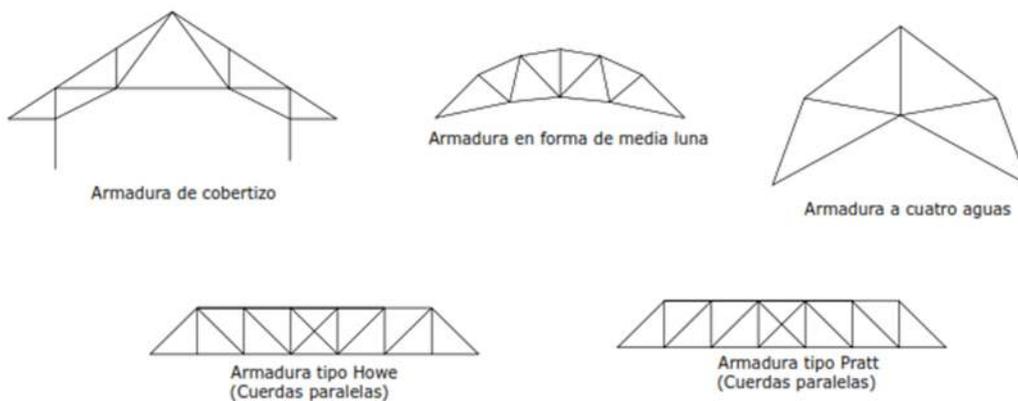


Figura 23. Otros tipos de Armaduras.

2.2.6.5. Inclinación de las Armaduras

La inclinación de un techo se puede expresar por la pendiente que genera la longitud de la montante con la longitud de la brida inferior, así mismo se puede expresar por su equivalente en grados sexagesimales.

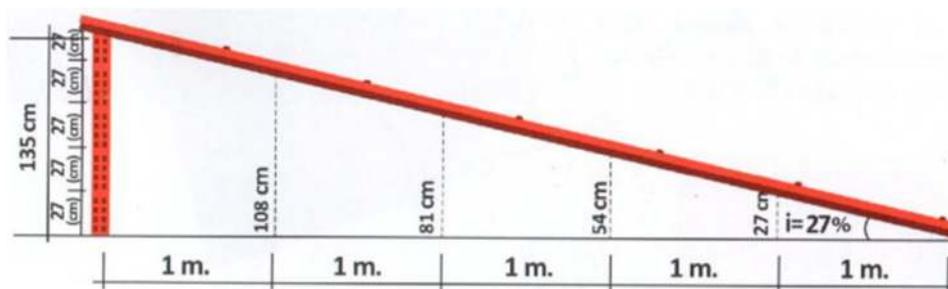


Figura 24. Pendiente de una armadura.

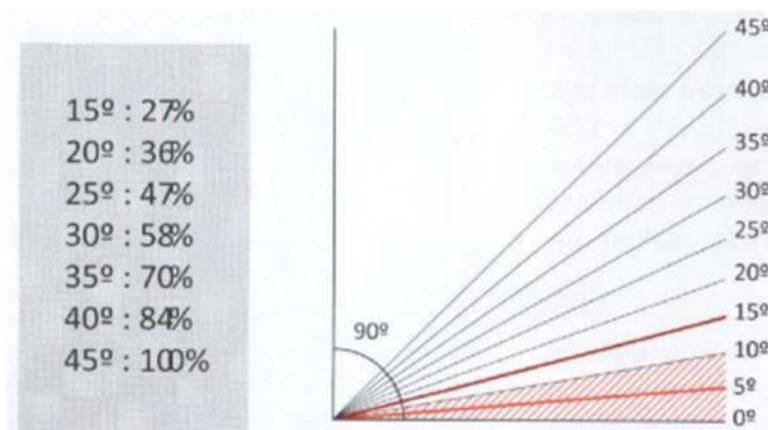


Figura 25. Pendiente de una armadura y su equivalente en grados sexagesimales.

Para la inclinación o pendientes de las armaduras metálicas de los techos, nuestra Norma E.090 del RNE, no especifica claramente este tema. Es por ello que se ha recurrido a un documento legal y técnico que provee datos teóricos para su aplicación de las pendientes mínimas acorde a la realidad de nuestro país. Para ello se recurrirá a la Norma Técnica de Infraestructura Educativa NTIE 001-2017, donde nos proporciona los criterios generales de diseño y para nuestra investigación tomaremos los datos de las pendientes de los techos según la zona climática.

Tabla 1. Pendientes según zona bioclimática.

ZONA CLIMATICA	CARACTERISTICA DE LA ZONA CLIMATICA	ALTURA (msnm)	PENDIENTE (%)
Zona 01	Desértico marino	0 - 400	0 - 10
Zona 02	Desértico	400 - 2000	5 - 15
Zona 03	Interandino bajo	2000 - 3000	20 - 40
Zona 04	Mesoandino	3000 - 4000	40 - 70
Zona 05	Altoandino	4000 - 4800	40 - 70

Zona 06	Nevado	Más de 4800	40 - 70
Zona 07	Ceja de montaña	2000 - 3000	> 80
Zona 08	Subtropical húmedo	400 - 2000	> 80
Zona 09	Tropical húmedo	80 - 1000	> 80

Fuente: Norma Técnica de Infraestructura Educativa NTIE 001-2017.

2.2.7. Lenguaje de programación Visual C#

C# es un lenguaje de programación bastante completo y potente que tiene como propietario a la gran Microsoft conocida a nivel Mundial, con similitudes y mejoras con los siguientes lenguajes de programación: C, C++ y Java.

C# está orientado a objetos POO, por lo que los conceptos de: encapsulación, herencia y polimorfismo; los maneja perfectamente. Está orientado a objetos porque el programador asemeja la vida real con la creación de un objeto.

C# es un lenguaje de programación bastante conocido y utilizado, es por ello que muchos programas de ingeniería usados en nuestro medio, facilitan sus funciones o métodos de sus programas por medio de las API (Interfaz de programación de aplicaciones) a diversos lenguajes de programación, y uno de ellos es el C#. Se mencionan algunos de los programas usados en nuestro medio y orientados a la ing. civil que facilitan sus APIs con el manejo del lenguaje de programación C#: SAP2000, ETABS, CSiBridge, AutoCAD, Revit y ArcGis.

2.2.7.1. .Net Framework

Es una plataforma de desarrollo, que tiene el lado común en la creación de aplicaciones orientados a: web, móvil y escritorio; donde todas estas comparten librerías, bibliotecas, herramientas y componentes al ser utilizadas por algún lenguaje de programación que pertenece o acepta el .Net.

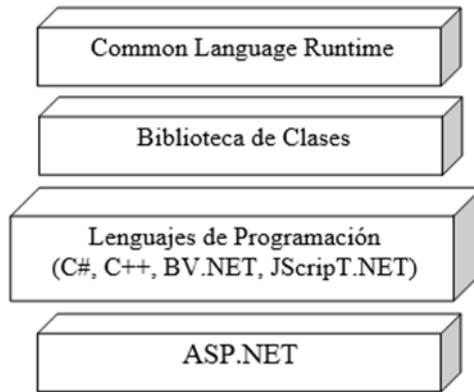


Figura 26. Componentes de .NET Framework.

2.2.7.2. Clases

Las clases en programación, hace referencia a todo el contenido y sus características comunes que debe de tener un objeto o grupo de objetos, por medio de la escritura de código.

Una clase también denominada trozos, partes o módulos, hace que la programación sea más ordena y modularizada para su mejor entendimiento y trabajabilidad.

2.2.7.3. Objetos

Un objeto en programación, es la representación de la realidad de lo que se desea analizar y automatizar por medio de código en una computadora. Por ejemplo: Un auto, una mesa, una persona, un techo para una casa, etc.

Estos objetos manejan sus propiedades(atributos) y comportamientos, ejemplo:

Objeto: Auto.

Propiedades: Alto, largo, color, peso.

Comportamiento: Frenar, girar, acelerar, arrancar.

2.2.7.4. Tipos de datos

El lenguaje de programación C#, manejan dos tipos de datos:

- ✓ **Datos de valor.** Estos tipos de datos son los que contienen valores reales. Ejemplo los tipos primitivos como char, int y float.

- ✓ **Datos de referencia.** Estos tipos de datos son los que se almacenan en un lugar de la memoria y se deben de hacer referencia a esa dirección de memoria para poder acceder al dato. Ejemplo un array.

Todos los tipos de valor y referencia derivan en última instancia de un tipo base llamado object.

2.2.7.5. Funciones o métodos.

Todo el código que lleva un programa grande, suelen ser complejos, difíciles de comprender y de depurar, debe de ser necesario dividirlo y organizarlo para su mejor manejo, a esas divisiones de código en bloques se les denomina funciones o métodos.

Las funciones o métodos son un fragmento de código que puede ser invocado y que puede o no devolver un valor al código que lo invocó en un principio.

2.2.7.6. Variables

El uso de variables en un programa, es como no tener materia prima para hacer un producto, las variables se utilizan en cálculos, controles, dibujos, etc.

Así mismo, las variables pueden ser definidas como constantes. Las constantes tienen valores que no pueden cambiar durante la ejecución del código. Por ejemplo, el valor de pi es una buena muestra de una constante porque el valor no cambia a medida que el código se ejecuta.

2.2.7.7. Atributos

Los atributos permiten al programador vincular información a las clases (cualquier tipo de información). El atributo o propiedades, podemos mencionar, por ejemplo: el color, la medida, fuente, marca, etc. Las posibilidades de usar un atributo son infinitas y ésta es la razón por la que Microsoft incluye en sus librerías tantos atributos predefinidos con el .NET Framework.

2.2.8. API del SAP2000 V16.0

Una Interfaz de Programación de Aplicaciones (API), es un conjunto de protocolos, librerías y funciones que un determinado programa o aplicación brinda para su uso, comunicándose con otra aplicación o programa.

Las API permiten la comunicación entre productos y servicios mediante un código de programación se hace uso de las funciones y librerías, sin necesidad de saber cómo están implementados esas funciones. Todo esto conlleva a simplificar los procesos y el código fuente, y además permite ahorrar tiempo y dinero. Así mismo, la API, proporciona: flexibilidad, simplifican el diseño y el código fuente, simplifican la administración y el uso de las aplicaciones, también, nos ofrece la facilidad de inventar nuevas aplicaciones a partir de lo que ya se tiene.

La interfaz de programación de aplicaciones abiertas (OAPI) del CSI es una herramienta poderosa que permite a los usuarios automatizar muchos de los procesos necesarios para construir, analizar y diseñar modelos y obtener resultados. También permite a los usuarios vincular SAP2000 con software de terceros, proporcionando una ruta para el intercambio bidireccional de información del modelo con otros programas.

La mayoría de los lenguajes de programación principales se pueden usar para acceder a SAP2000 a través de OAPI. Esto incluye al C# que es el lenguaje usado para esta tesis.

2.3. Marco Conceptual

- **Estructuras:** Una estructura tiene el enfoque sistémico, es decir, está conformada por un conjunto de elementos o componentes que unidos en conjunto y en forma ordenada cumplen un mismo fin para las que fueron creadas.
- **Análisis estructural:** El análisis estructural en un conjunto de procesos donde se hace uso de las diferentes ecuaciones de la resistencia de materiales para luego determinar los esfuerzos internos, tensiones y deformaciones que actúan sobre una determinada estructura.
- **Tijeral metálico:** Son vigas que nacen de columnas, muros o vigas maestras, y en el otro extremo llegan, también, a una estructura similar en una construcción civil.

- **El acero:** El acero es un tipo de material producto de la aleación del hierro, en la cual se adiciona el carbono y otros elementos con el fin de mejorar sus propiedades físico-químicas. Existen varios tipos de acero que están definidos depende al tipo de uso que se le darán.
- **Techos:** Se denomina cubierta o techo a la superficie entramada que cierra una edificación por su parte superior, destinada a proteger su interior de los agentes climatológicos dando una adecuada evacuación, a la vez que se asegura del agua de lluvia, proporcionando al mismo tiempo un aislamiento térmico acústico al conjunto así obtenido.
- **Armaduras:** Una armadura es una construcción reticulada conformada generalmente por triángulos formados por elementos rectos y que se utiliza para soportar cargas. Las armaduras pueden ser planas o espaciales. Ejemplos típicos de armaduras son: puentes, cerchas, torres de transmisión, cúpulas de estadios.
- **Brida superior o cuerda superior:** Es un elemento o miembro de una armadura, que se ubica en la parte más alta o superior de los elementos, formado una línea que se extienden de un apoyo a otro apoyo.
- **Brida inferior o cuerda inferior:** Es un elemento o miembro de una armadura, que se ubica en la parte más baja o inferior de los elementos, formado una línea que se extienden de un apoyo a otro apoyo.
- **Montante:** Son los elementos o miembros que conectan la brida superior con la brida inferior.
- **Diagonal:** Son los miembros que conectan las juntas o nudos de la brida superior con la brida inferior y las montantes.
- **Nudos:** Son los puntos en las cuales, las montantes y las diagonales se juntan con la cuerda superior e inferior.

- **Panel:** parte de una armadura que se localiza justo en el medio de las dos juntas consecutivas de la brida superior.
- **Larguero:** Es la viga que se origina en una armadura y termina en otra, tomando descanso en la cuerda superior.
- **Tensores o tirantes:** Son los miembros que se someten a tensión, basándose en la clase de los esfuerzos.
- **Predimensionamiento:** Se define por predimensionado o predimensionamiento al conjunto de técnicas que permiten calcular elementos de ingeniería de manera sintetizada. El objetivo de esta reducción es el de encontrar unas magnitudes orientativas en cuanto a dimensiones o características del elemento que puedan servir para afinar un proceso de diseño que, finalmente, habrá de ser ratificado por un cálculo exhaustivo.
- **Cargas:** Son las acción y fenómenos externos que una determinada estructura tiene que soportar y deben de ser incluida en el análisis estructural. Estas cargas pueden ser el resultado del peso de los materiales de construcción, los ocupantes, el del mobiliario, del equipamiento, efectos del medio ambiente, movimientos diferenciales y cambios dimensionales restringidos.
- **Carga Muerta:** Las cargas muertas son las que poseen constante magnitud y su ubicación es fija dentro de la estructura y permanece ahí hasta que cumpla la vida útil. Generalmente las cargas muertas es el propio peso de los materiales que forman la estructura.
- **Cargas Vivas:** Las cargas vivas son las cargas gravitacionales que pueden cambiar de lugar y magnitud, es temporal y no tienen carácter permanente.
- **Cargas de Nieve:** Las precipitaciones de hielo y nieve varían con el clima y cantidad retenida por una cubierta depende de su proporción y el tipo de superficie. Las cubiertas

de madera, asbesto o similares retendrán mayor cantidad que las tejas planas o las metálicas.

- **Cargas debidas al Viento:** Las cargas de viento varían con la localidad geográfica, las alturas sobre el nivel del terreno, los tipos de terreno que rodean a los edificios, incluyendo otras estructuras y algunos otros factores. Las fuerzas de viento actúan como presiones sobre las superficies verticales a barlovento, como presiones o succiones sobre superficies inclinadas a barlovento (dependiendo de la pendiente) y como succiones sobre superficies planas y superficies verticales o inclinadas a sotavento (debido a la creación de presiones negativas o vacías).
- **Cargas de Sismo:** Los terremotos producen movimientos horizontales y verticales. Los movimientos horizontales son los que generan en la estructura los efectos más significativos. Cuando el suelo se mueve, la inercia de la masa de la estructura tiende a resistir tal movimiento.
- **Otras Cargas Vivas:** Existen otros tipo de cargas vivas que se debe de considerar, como lo son: presiones del suelo (como ejercidas por la presión lateral de la tierra en muros o las ejercidas verticalmente contra las cimentaciones), las presiones hidrostáticas (como la presión del agua sobre cortinas de presas, las fuerzas de inercia de grandes cantidades de agua durante un sismo y las subpresiones sobre tanques y estructuras de cimentación).
- **Diseño con Factores de Carga y Resistencia (LRFD):** Este método de diseño denominado también diseño por estados límites o resistencia última, consiste en determinar en primer término, las acciones (cargas o momentos) que se presentan en las secciones críticas de un miembro estructural o estructura bajo el efecto de las acciones de diseño o cargas factorizadas.

- **Diseño por Esfuerzos Permisibles (ASD):** Este método de diseño denominado también diseño elástico, consiste en determinar, en primer término, los esfuerzos que se presentan en las secciones críticas de un miembro estructural bajo la acción de las cargas de servicio o de trabajo, considerando un comportamiento elástico del material. Se considera que un miembro está diseñado correctamente cuando los esfuerzos de trabajo, ocasionados por las cargas de servicio que obran en el miembro no exceden los esfuerzos permisibles.
- **Lenguaje de Programación:** Un lenguaje de programación es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de una computadora, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa.
- **Visual C#:** Es un lenguaje de programación orientado a objetos que está basado en el lenguaje C++, creado por la Empresa Microsoft como parte de su plataforma .Net, que después fue aprobado como un estándar por la ECMA (ECMA – 334) e ISO (ISO/IEC 23270). El lenguaje de programación C# es uno de los más comunes y mayor utilizado en el mundo, por sus características completas, es ideal para trabajar con cualquier API de un determinado programa.
- **Clase:** A nivel de programación una clase es el desarrollo de un tipo personalizado de variables, métodos y eventos, que se caracteriza por tener un determinado fin y ser parte de otras clases y del código principal. Es la definición de la estructura y el comportamiento de un objeto.

- **Objeto:** Es una entidad prevista de un conjunto de propiedades o atributos (variables) y métodos que.
- **Métodos:** Viene a ser un bloque de código, que manejan el comportamiento o funcionalidad de un objeto, reaccionan a eventos.
- **API del SAP2000:** El programa SAP2000, la cual es una de las más utilizadas para análisis estructural, ofrece una API bastante legible, dinámico y sobre todo entendible para su uso y poder desarrollar aplicaciones puntuales.

CAPÍTULO III HIPÓTESIS

3.1.1. *Hipótesis General*

El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es conveniente.

3.1.2. *Hipótesis Específicas*

- a) El resultado del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es adecuado.
- b) El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a mejorar los tiempos necesarios para un diseño óptimo de un tijeral metálico tipo howe.
- c) El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a minimizar procesos que existen en un análisis y diseño tradicional o convencional.

3.1.3. *Variables*

3.1.3.1. **Variable Independiente**

Análisis estructural de techos con tijerales metálicos: Cuenta con tres dimensiones predimensionamiento estructural, factores de carga, diseño y análisis por el método LRFD, todos independientes cuantitativas ya que estos no están sujetos a ningún cambio, es decir, son autónomas.

3.1.3.2. **Variable dependiente**

Lenguaje de programación visual C# y la API del SAP2000: El Sap2000 es un programa desarrollado por la empresa Computers and Structures Inc., para el modelamiento, análisis y diseño de estructuras; la cual nos ofrece una interfaz de programación para aplicaciones (Application Programming Interface API); que conjuntamente con el lenguaje de programación

C#, nos permite interactuar y desarrollar una aplicación netamente puntual, en nuestro caso es obtener del programa los esfuerzos últimos para el diseño final.

3.1.3.3. Operacionalización de las Variables

Tabla 2. Operacionalización de la variable independiente.

Variable	Dimensiones	Indicadores
Variable Independiente: Análisis estructural de techos con tijerales metálicos	Predimensionamiento Factores de carga (Carga muerta, viva, viento y otros) Diseño y análisis por el método LRFD	Carga muerta total (D) Carga viva total (Lr) Carga viento total (W1, W2) Carga de lluvia total (B) Combinación de carga (Envolvente) Diseño por tracción (Pnf) Diseño por compresión (Pn)

Fuente: elaboración propia.

Tabla 3. Operacionalización de la variable dependiente.

Variable	Dimensiones	Indicadores
Variable dependiente: Programación aplicativa	El lenguaje de programación visual C# API del SAP2000	Clases y métodos (Cls_TijeralHowe.cs) Fomulario Principal (Form_Principal.cs) Tabla de esfuerzos últimos (Pn)

Fuente: elaboración propia.

CAPÍTULO IV METODOLOGÍA

4.1. Método de Investigación

Los métodos a utilizar será el Inductivo - Deductivo, según Bernal, (2010), basado en la lógica y relacionándolo con el estudio de hechos particulares, siguiendo los pasos de la observación y la experimentación para obtener nuevos conocimientos, se interesan por identificar las cualidades y características del hecho y al mismo tiempo manipular las variables. Para el presente estudio, se predimensionara en base a lo observado, un tijeral metálico tipo Howe, para luego procesar los datos de forma automatizada utilizando el C# y el API del SAP2000 manipulando las variables según la norma E.090, E.020, AISC, y todo el proceso de programación, para luego obtener resultados finales y realizar el diseño correspondiente.

4.2. Tipo de Investigación

El tipo de investigación por la naturaleza del estudio será tecnológico, según Sánchez y Reyes, (2016), “está dirigida a descubrir y conocer que técnicas son más eficaces o apropiadas, para operar, es decir producir cambios o conservar los progresos alcanzados” para el presente estudio, el resultado final, es un proyecto de nuevas tecnologías, que nos permite analizar y diseñar techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000 y con ello alcanzar mejoras en tiempos y procesos de forma automatizada.

4.3. Nivel de Investigación

El estudio por el nivel de profundidad será descriptivo; según Sabino (2008), manifiesta “el nivel descriptivo expone las características y/o cualidades del hecho, tal y como se observa. Para el estudio se recogerá información, características y cualidades de una armadura tipo Howe para realizar un análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000.

4.4. Diseño de Investigación

El diseño metodológico por la naturaleza del estudio será el no experimental - transversal; según Hernández (2010), manifiesta que los diseños no experimentales – transversales recolectan información en un momento determinado de tiempo.

4.5. Población y Muestra

4.5.1. Población.

La población estará constituida por los techos metálicos tipo HOWE del cercado de la ciudad de Huancayo: Terminal terrestre Los Andes, Terminal Ferroviario, Vivienda y Comercio "Mega Muebles"

4.5.2. Muestra.

La muestra estará constituida por el techo metálico tipo HOWE del Terminal terrestre Los Andes, del cercado de la ciudad de Huancayo. La muestra fue elegida considerando el tipo de muestreo no probabilístico por conveniencia.

4.6. Técnicas e Instrumentos de recolección de datos

4.6.1. Técnicas

Las técnicas de recopilación de datos serán la observación, el análisis de contenido y las libretas de campo.

4.6.2. Instrumentos

Los instrumentos serán las fichas de observación - técnicas y los software respectivos.

4.7. Técnicas de procesamiento y análisis de datos

En este punto, el análisis de los datos se llegó a utilizar la siguiente técnica de investigación: Se trabajará cuadros con los datos obtenidos mediante la observación y se realizará un tipo de análisis no estadístico. Para el orden y manejo de los datos se usó el **Excel**, que permitirán procesar datos obtenidos con los instrumentos de recolección.

Tabla 4. Datos de la población.

ITEM	MATERIAL DE COBERTURA	TIPO COBERTURA	UBICACION	ESTABLECIMIENTO
1	Metálica	Howe	Cercado de Huancayo	Terminal terrestre Los Andes
2	Metálica	Howe	Cercado de Huancayo	Terminal Ferroviario
3	Metálica	Howe	Cercado de Huancayo	Vivienda
4	Metálica	Howe	Cercado de Huancayo	Comercio "Mega Muebles"

Fuente: Elaboración propia.

4.8. Aspectos éticos de la Investigación

Para el presente desarrollo del aspecto ético de la investigación se adoptará los artículos 19° y 20° del Capítulo III “Ética de Investigación”, del Reglamento de Investigación de la Universidad Peruana Los Andes, así mismo se cumplirán con los incisos respectivos.

Artículo 19° incisos:

- e) La confidencialidad de los datos será importante, ya que, la muestra que se ha considerado, el propietario nos indicó que los datos, solo se utilizará en la investigación.
- f) Beneficiencia y no maleficiencia, durante la investigación se optó por no causar daño el área en estudio.
- g) Responsabilidad, se asume bastante responsabilidad al realizar la presente investigación, para ofrecer buenos alcances, tanto individual, institucional y social.
- h) Veracidad, respecto a la confiabilidad y veracidad de los procesos, se ha tenido bastante cuidado para poder demostrar nuestros objetivos en base al problema planteado, así mismo, se han verificado los resultados finales.

Artículo 20° incisos:

- a) La presente investigación cumple con la originalidad, es coherente con las líneas de investigación Institucional.
- b) La presente investigación asegura la validez, la fiabilidad y credibilidad de los métodos usados, así como las normas que rigen en la actualidad.

- c) Se ha asumido, en todo momento la responsabilidad de la investigación, siendo conscientes de las consecuencias individuales, sociales y académicas que se derivan de la misma.
- d) No se ha percibido subvención alguna o realizado ningún contrato de investigación que especifiquen condiciones inconscientes con su juicio científico, con la Visión y Misión de la Universidad Peruana Los Andes, o que permitan a los patrocinadores vetar o retrasar la publicación académica, porque no están de acuerdo con los resultados.
- e) Así mismo, se garantiza la confidencialidad y anonimato de las personas involucradas en la investigación.
- f) Se cumple con reportar la investigación de manera abierta, completa y oportuna a la comunidad científica.
- g) Se optó por tratar con sigilo la información obtenida y no utilizarla para el lucro personal ilícito o para otros propósitos distintos de los fines de la investigación.
- h) Se ha tenido bastante cuidado en cumplir con las normas institucionales, nacionales e internacionales que regulen la investigación.
- i) Se asume la acción de revelar los conflictos de intereses que puedan presentarse en el transcurso del desarrollo de la investigación, evaluación y asesoría.
- j) Se consideró evitar la falsificación de datos, plagio de investigación y la no participación en el diseño e la tesis.
- k) Se optó por no publicar el trabajo de investigación de acuerdo al Reglamento de Propiedad Intelectual de la Universidad Peruana Los Andes,

CAPITULO V RESULTADOS

5.1. Descripción de los Resultados

5.1.1. *Análisis y diseño de techos con tijerales metálicos tipo Howe mediante el visual C# y el API del SAP2000*

5.1.1.1. Planteamiento de caso.

Se desea analizar y diseñar un techo metálico tipo howe, en la ciudad de Huancayo con las siguientes consideraciones:

- Luz del tijeral = 16.80 m.
- Separación entre tijerales = 5.00 m.
- Altura total (desde el nivel 0.00) = 10.00 m.
- Cantidad de paneles del tijeral = 6

5.1.1.2. Predimensionamiento

Se realizará el predimensionamiento de la altura del tijeral, donde:

$$h = \frac{L}{8}$$

$$h = \frac{16.8}{8}$$

$$h = 2.1 \text{ m.}$$

Como se trata de una estructura ubicada en la ciudad de Huancayo, en la zona 04 “Mesoandina” según la tabla 01, la pendiente de la armadura tipo howe será entre 40% y 70% entonces:

$$40\% < S < 70\%$$

$$S = \frac{h}{l}$$

$$h = S \times l$$

$$h = 40\% \times \left(\frac{16.8}{2}\right)$$

$$h = 3.36 \cong h = 3.50 \text{ m.}$$

Por consiguiente, tomamos el valor mayor de $h = 3.50$ m.

La armadura tendrá las siguientes dimensiones, según el predimensionamiento y los datos previos:

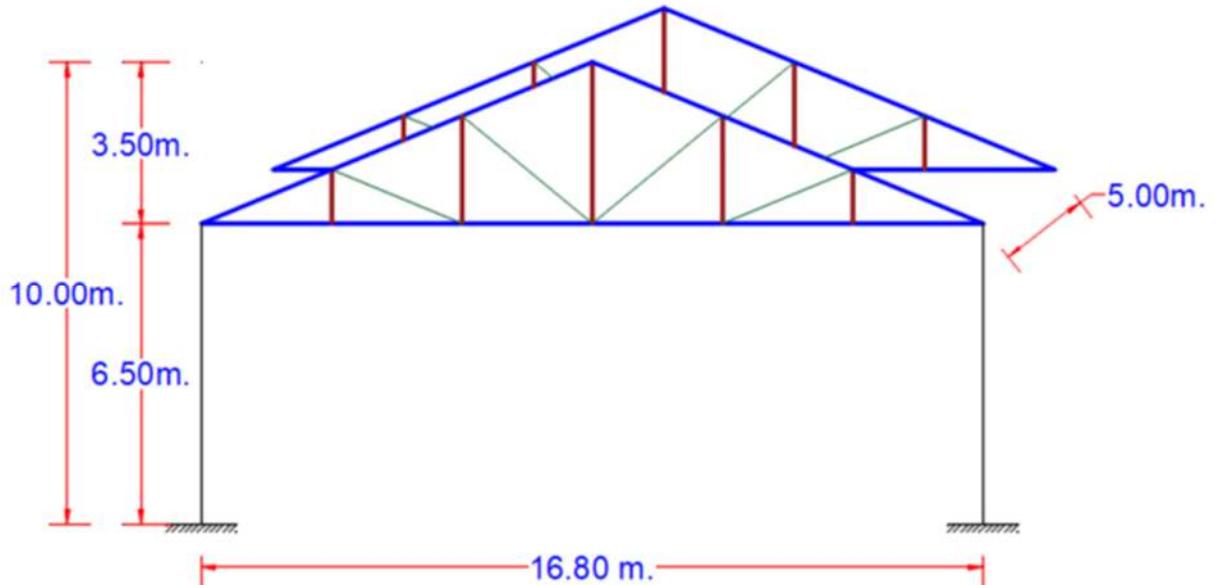


Figura 27. Descripción de resultados - Predimensionamiento del tejado metálico para cálculo en el software HoweSteel.

5.1.1.3. Medrado de cargas del tejado

Para el Medrado de cargas, se hace referencia a la norma E.020 “Cargas” de nuestro Reglamento Nacional de Edificaciones.

5.1.1.3.1. Carga Muerta (CM)

La carga muerta a considerar, será el peso de los materiales y de la armadura fijo y constante, como se detalla a continuación:

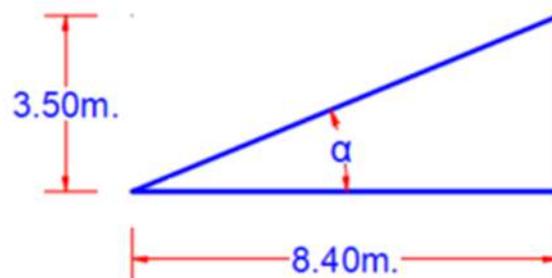
Tabla 5. Disgregado de carga muerta.

Descripción	Peso (Kg/m ²)	Distancia @ Tijeral (ml)	Sub Total (Kg/ml)
Peso de Tijeral + Viguetas	16.00	5.00	80.00
Plancha del Techado	3.00	5.00	15.00
Peso de Luminaria	4.00	5.00	20.00
Sistema contra Incendio	8.00	5.00	40.00
Peso de Extras	4.00	5.00	20.00
		TOTAL CM	175.00

Fuente: elaboración propia.

5.1.1.3.2. Carga Viva (L_r)

Según la norma E.020 del RNE, la carga mínima para techos con inclinación mayor a 3° , con respecto a la horizontal será de 1.0 KPa (100 Kg/m²), si la inclinación del techo es mayor a 3° , entonces, el valor se reducirá en 0.05 KPa (5 Kg/m²) por cada grado de pendiente por encima de los 3° , hasta un mínimo de 0.50 KPa (50 Kg/m²).



$$\tan \alpha = \frac{3.5}{8.4}$$

$$\alpha = \text{Arctang} \left(\frac{3.5}{8.4} \right)$$

$$\alpha = 22.6^\circ$$

$$\Rightarrow L_r = 100 - 5(22.6 - 3)$$

$$L_r = 2$$

$$\Rightarrow \text{Carga Viva } L_r = 50 \text{ Kg/m}^2$$

5.1.1.3.3. Carga de lluvia o granizo (R)

La carga de lluvia, granizo o acumulación de agua, será considerado de acuerdo a la ubicación geográfica y climatológica; para nuestro caso la estructura estará ubicada en la ciudad de Huancayo y se considerarán las siguientes características:

$$\text{Peso x m}^3 = 1000.00 \text{ Kg/m}^3$$

$$\text{Altura o capa de agua} = 0.05 \text{ m.}$$

$$\text{Ancho tributario} = 5.00 \text{ m}$$

$$\text{Carga Lluvia R} = 1000 \times 0.05 \times 5.00$$

$$\text{Carga Lluvia R} = 250.00 \text{ Kg/ml}$$

5.1.1.3.4. Carga Viento

Para la carga de viento, los datos o parámetros serán automatizados dentro del programa HoweSteel, la estructura será cerrada, y la ubicación es en la ciudad de Huancayo del departamento de Junín.

Todo el procedimiento para el cálculo de la carga de viento, se ha tenido en cuenta la Norma E.090 del RNE.

5.1.1.4. Ingreso de datos en el programa HoweSteel

El programa HoweSteel, se ha diseñado para automatizar el tiempo y minimizar los procesos en el análisis y diseño de un tijeral metálico tipo Howe, para lo cual se deberá primero de parametrizar la cobertura tipo Howe, predimensionar y realizar el Metrado de cargas que es propia del consultor o proyectista.

5.1.1.4.1. Ingreso de datos para el tijeral tipo Howe

Se procede a ingresar los datos que se requiere para el tijeral tipo Howe, en el programa HoweSteel; se ingresará la longitud o luz del tijeral (L), la altura del tijeral (H) previo predimensionamiento, el número de paneles que tendrá el tijeral (n) y la separación entre tijaerales.

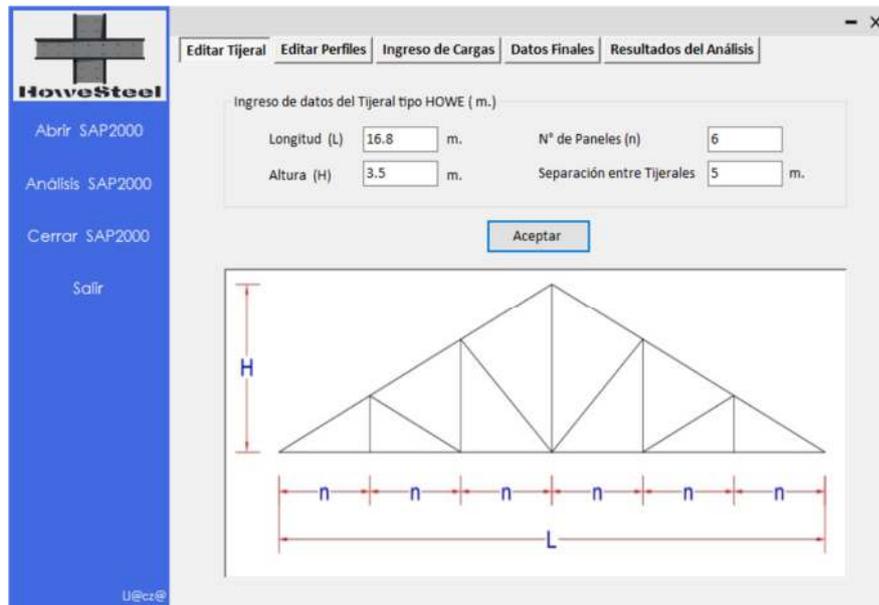


Figura 28. Descripción de resultados - Ingreso de datos en la pestaña “Editar Tijeral” en el software HoweSteel.

5.1.1.4.2. Ingreso de datos para el perfil de acero

El ingreso del perfil de acero, puede ingresarse previamente antes de la verificación por tracción y por compresión, por lo tanto, si no cumple el área de la sección elegida, se volvería a elegir una nueva sección con los datos obtenidos en la etapa de verificación por tracción y por compresión.

- Brida superior e inferior

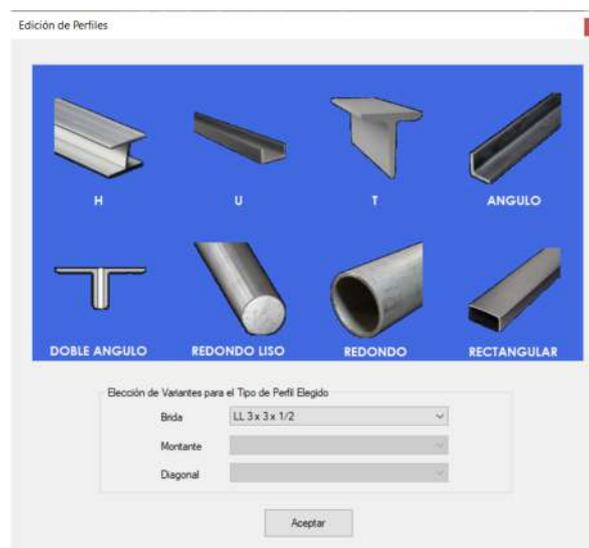


Figura 29. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Brida, del software HoweSteel.

➤ Montante

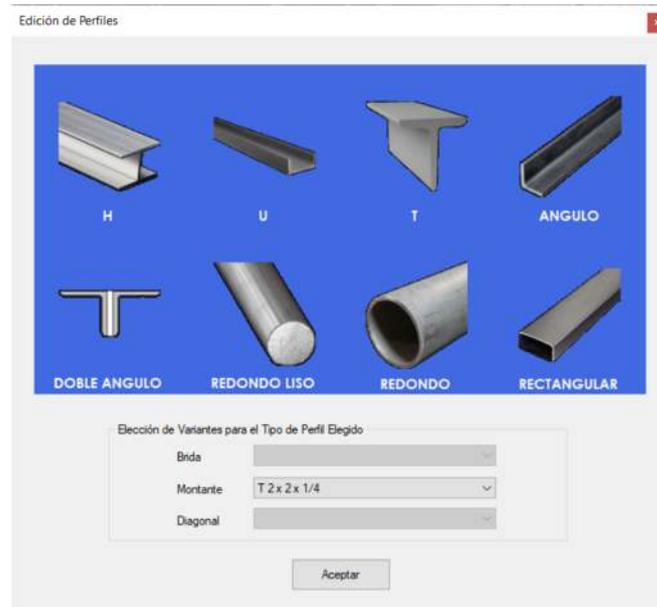


Figura 30. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Montante, del software HoweSteel.

➤ Diagonal

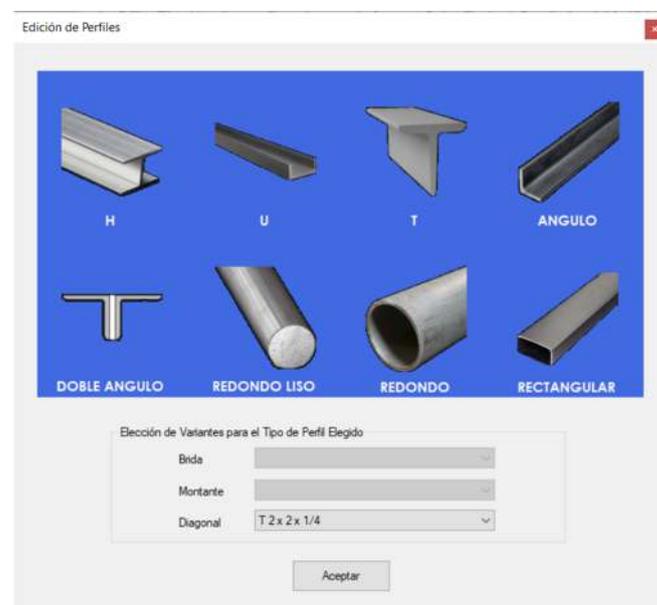


Figura 31. Descripción de resultados - Ingreso de datos en “Editar perfiles” para la Diagonal, del software HoweSteel.

5.1.1.4.3. Ingreso de datos de las cargas para el tijeral tipo Howe

Se procederá a ingresar los datos para las cargas que intervienen en nuestra armadura. Para la Carga Viva (Lr), el programa lo procesará automáticamente, así como también para la Carga de Viento.

The screenshot shows the 'Ingreso de Cargas' tab in the HoweSteel software. The interface is divided into several sections for data entry:

- Carga Muerta (Dead Load):** Includes input fields for 'Peso de Tijeral + Vigüeta' (16 Kg/m²), 'Plancha del Techado' (3 Kg/m²), 'Peso de Luminaria' (4 Kg/m²), 'Sist. Contra Incendio' (8 Kg/m²), and 'Peso Extras' (4 Kg/m²).
- Carga de Lluvia o Granizo (Rain or Hail Load):** Includes 'Peso por m²' (1000 Kg/m²) and 'Altura Capa de Lluvia o Granizo' (0.05 m).
- Carga de Viento (Wind Load):** Includes 'Altura Total desde Nivel 0.00' (10 m), 'Región dentro del Mapa Eólico' (Junín), 'Velocidad Según la Región' (70 Km/h), and a checkbox for 'Carga Adicional en Elemento de Cierre'.

An 'Aceptar' (Accept) button is located at the bottom of the main input area.

Figura 32. Descripción de resultados - Ingreso de datos en la pestaña "Ingresos de Cargas", en el software HoweSteel.

5.1.1.4.4. Verificación del ingreso de datos finales

En esta pestaña verificamos los datos antes de procesarlo con el SAP2000, caso que se desee realizar algún cambio, se volverá a ingresar el dato errado.

The screenshot shows the 'Datos Finales' tab in the HoweSteel software, providing a summary of the input data for the analysis:

- Datos del Tijeral Para el Análisis (Truss Data for Analysis):** Longitud (16.8 m), Altura (3.5 m), Cantidad Segmentos (6), Separación Entre Tijerales (5 m).
- Datos de los Perfiles a Usar Para Análisis (Profile Data for Analysis):** Brida (LL 3 x 3 x 1/2), Montante (T 2 x 2 x 1/4), Diagonal (T 2 x 2 x 1/4).
- Datos de los Tipos de Carga Para el Análisis (Load Type Data for Analysis):** Carga Muerta (35 Kg/m²), Carga Viva (50 Kg/m²), Carga de Lluvia o Granizo (50 Kg/m²).
- Carga de Viento (Wind Load):** Barlovento W1 (17.15 Kg/m²), Barlovento W2 (-7.35 Kg/m²), Sotavento W1 (-14.7 Kg/m²), Sotavento W2 (-14.7 Kg/m²).

Figura 33. Descripción de resultados - Verificación en la pestaña "Datos Finales", en el software HoweSteel.

5.1.1.5. Proceso de análisis en el SAP2000

5.1.1.5.1. Abrir el programa SAP2000 desde el programa HoweSteel.

Dentro del código fuente, se hace referencia al ejecutable del SAP2000, para nuestro proyecto se ha utilizado la versión 16.0.0 del SAP2000.

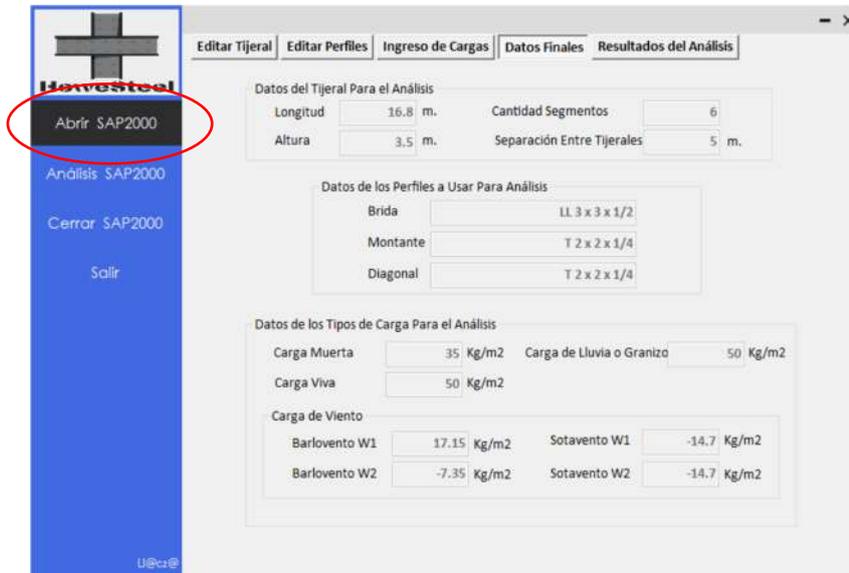


Figura 34. Descripción de resultados - Ingreso a la aplicación del SAP2000, mediante el botón “Abrir SAP2000”.

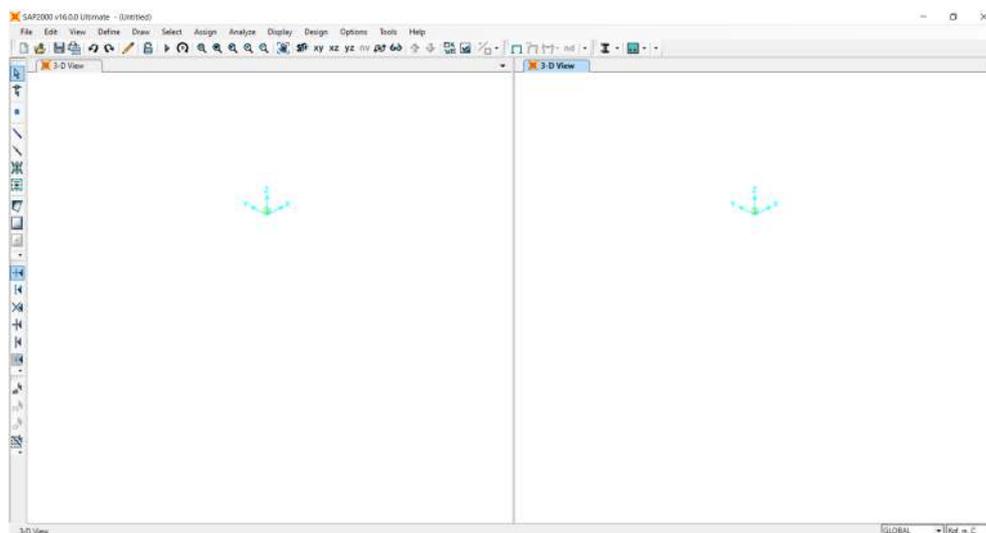


Figura 35. Descripción de resultados - Espacio de trabajo del SAP2000, previo enlace con el software HoweSteel.

5.1.1.5.2. Análisis del tijeral tipo Howe en el SAP2000

Por medio del API y del lenguaje visual C#, se obtiene como resultado de presionar el botón “Análisis SAP2000”:

- Dibujo de los frame para el tijeral con los datos previamente ingresados.

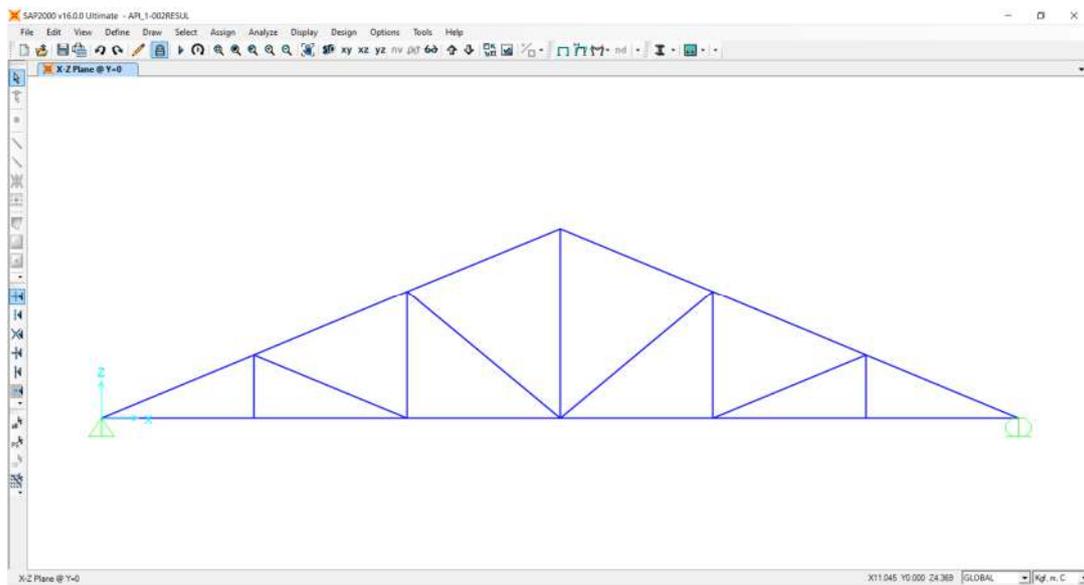


Figura 36. Descripción de resultados - Datos procesados después de ingresar en el botón “Análisis SAP2000”, del software HoweSteel.

- El resultado para la verificación de los esfuerzos, ya se obtiene automáticamente.

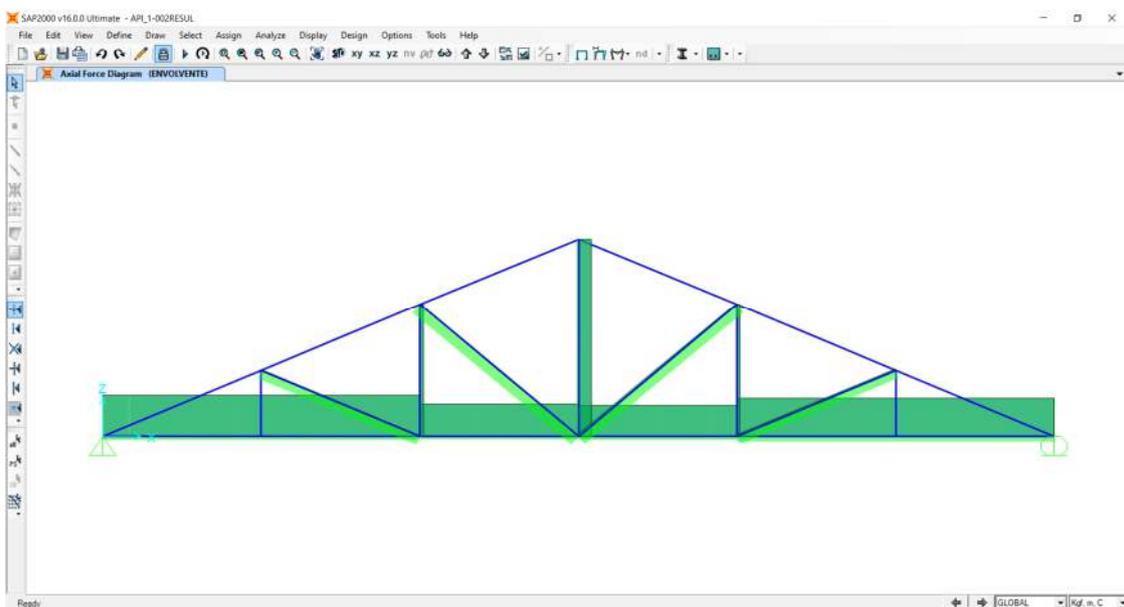


Figura 37. Descripción de resultados - Verificación de los esfuerzos últimos (opcional).

- Se puede verificar los esfuerzos internos para cada Frame del tijeral.

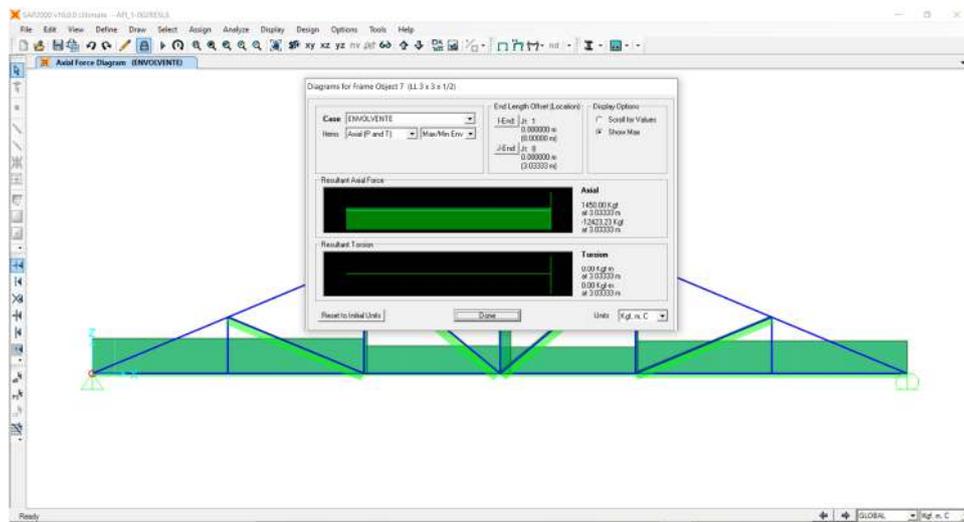


Figura 38. Descripción de resultados - Verificación de los esfuerzos para cada Frame (Opcional).

5.1.1.6. Obtención de los esfuerzos últimos

Una vez que ya haya sido procesado el análisis en el SAP2000, automáticamente podemos obtener detalles de los resultados de dicho análisis para todos los componentes del tijeral, cabe resaltar que ya se obtiene los esfuerzos últimos críticos para cada componente. Con estos datos, el consultor o proyectista tendrá todo lo necesario para el análisis por tracción y por compresión.

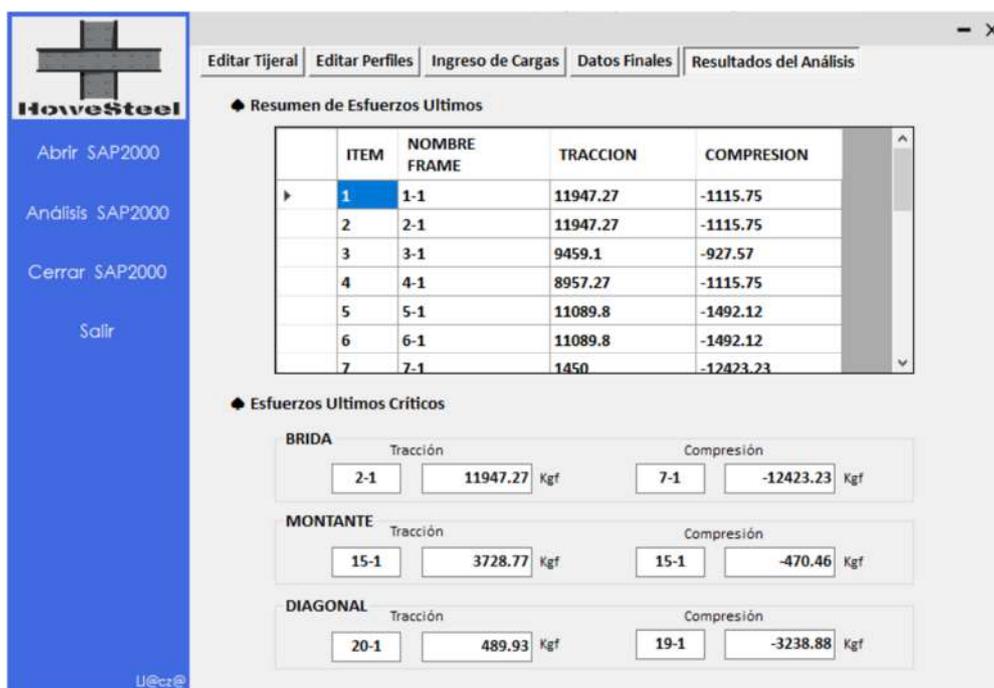


Figura 39. Descripción de resultados - Obtención de los esfuerzos últimos críticos en la pestaña "Resultados de análisis" del software HoweSteel.

5.1.1.7. Diseño por Tracción

De los resultados del análisis del programa HoweSteel se tiene:

$$P_u = 11\,947.27 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\emptyset = 0.90$$

$$L = 2.8 \text{ m.}$$

Límite de fluencia por tracción:

$$P_{nf} = \emptyset \cdot F_y \cdot A_g$$

$$A_g = \frac{11\,947.27}{0.9 (2\,530)} = 5.247 \text{ cm}^2 = 0.81 \text{ plg}^2$$

Donde:

P_{nf} = Límite de fluencia por tracción.

\emptyset = Factor de resistencia.

F_y = Fluencia del acero

A_g = Área bruta de la sección.

Selección de sección a utilizar: 2L 2 x 2 x 1/8

Características geométricas de la sección:

$$A_g = 0.982 \text{ plg}^2 = 6.34 \text{ cm}^2$$

$$r_x = 0.62 \text{ plg} = 1.58 \text{ cm}$$

$$P_{nf} = 0.9 \times 2\,530 \times 6.34 = 14\,425.87 \text{ Kg} \dots \dots OK$$

Verificación por esbeltez

$$\frac{KL}{r} < 300 \quad \text{donde } K = 1$$

$$\frac{(1)(280)}{1.58} < 300$$

$$177.22 < 300 \dots \dots OK$$

5.1.1.8. Diseño por Compresión

De los resultados del análisis del programa HoweSteel se tiene:

$$P_u = 12\,423.23 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\phi = 0.90$$

$$L = 3.03 \text{ m.}$$

Selección de sección a utilizar: la sección 2L 2 x 2 x 1/8, elegido en el diseño por tracción no siempre cumple con el A_g (área bruta) necesaria para el diseño por compresión, normalmente se elige otra sección con un área mayor, por lo tanto, la otra sección elegida será la 2L 3 x 3 x 1/4

Características geométricas de la sección:

$$A_g = 2.87 \text{ plg}^2 = 18.52 \text{ cm}^2$$

$$b = 3.00 \text{ plg} = 7.62 \text{ cm}$$

$$t = 0.25 \text{ plg} = 0.64 \text{ cm}$$

$$E = 29\,000 \text{ Klb/plg}^2$$

$$F_y = 36 \text{ Klb/plg}^2$$

$$r_x = 0.93 \text{ plg} = 2.35 \text{ cm}$$

$$r_y(0) = 1.25 \text{ plg} = 3.18 \text{ cm}$$

Verificación si el elemento es esbelto

$$\frac{b}{t} < \lambda r \quad \text{donde según tabla } \lambda r = 0.45 \sqrt{\frac{E}{F_y}}$$

$$\frac{3.00}{0.25} < 0.45 \sqrt{\frac{29\,000}{36}}$$

$$12 < 12.77 \dots \dots OK$$

Relación de esbeltez

$$\frac{KL}{r} < 200$$

$$\frac{(1)(303)}{2.35} < 200$$

$$128.94 < 200 \dots \dots OK$$

Cálculo del esfuerzo critico o de pandeo a flexión

$$\frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}}$$

$$128.94 \leq 133.68 \dots \dots OK$$

Aplicando la ecuación: $F_{cr} = \left(0.658^{\frac{F_y}{F_e}}\right) F_y$

Por Euler:

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

$$F_e = \frac{\pi^2 (29\ 000)}{128.94^2} = 17.22 \text{ Klb/plg}^2$$

$$\Rightarrow F_{cr} = \left(0.658^{\frac{36}{20.20}}\right) \times 36 = 15.00 \text{ Klb/plg}^2$$

Para LRFD, $\phi_c = 0.90$

$$\Rightarrow \phi_c F_{cr} = 13.50 \text{ Klb/plg}^2$$

Cálculo de la resistencia de diseño por pandeo

$$\phi_c P_n = \phi_c F_{cr} \times A_g$$

$$\phi_c P_n = 13.50 \times 2.87$$

$$\phi_c P_n = 38.75 \text{ Klb}$$

$$\Rightarrow \phi_c P_n = 17,592.50 \text{ Kg} \dots \dots OK$$

5.1.2. *Análisis y diseño de techos con tijerales metálicos tipo Howe mediante la forma tradicional.*

Para poder constatar con los resultados del análisis y diseño mediante el programa HoweSteel que se realizó en el punto anterior, se procederá a realizar el análisis y diseño de forma manual y tradicional, con el cual se viene realizando hoy en día. Para este análisis y diseño se tomará en cuenta, los mismos datos, pero procesados de forma tradicional.

5.1.2.1. **Planteamiento de caso.**

Se desea analizar y diseñar un techo metálico tipo howe, en la ciudad de Huancayo con las siguientes consideraciones:

- Luz del tijeral = 16.80 m.
- Separación entre tijerales = 5.00 m.
- Altura total (desde el nivel 0.00) = 10.00 m.
- Cantidad de paneles del tijeral = 6

5.1.2.2. **Predimensionamiento**

Se realizará el predimensionamiento de la altura del tijeral, donde:

$$h = \frac{L}{8}$$

$$h = \frac{16.8}{8}$$

$$h = 2.1 \text{ m.}$$

Como se trata de una estructura ubicada en la ciudad de Huancayo, en la zona 04 “Mesoandina” según la tabla 01, la pendiente de la armadura tipo howe será entre 40% y 70% entonces:

$$40\% < S < 70\%$$

$$S = \frac{h}{l}$$

$$h = S \times l$$

$$h = 40\% \times \left(\frac{16.8}{2}\right)$$

$$h = 3.36 \cong h = 3.50 \text{ m.}$$

Por consiguiente, tomamos el valor mayor de $h = 3.50$ m.

La armadura tendrá las siguientes dimensiones, según el predimensionamiento y los datos previos:

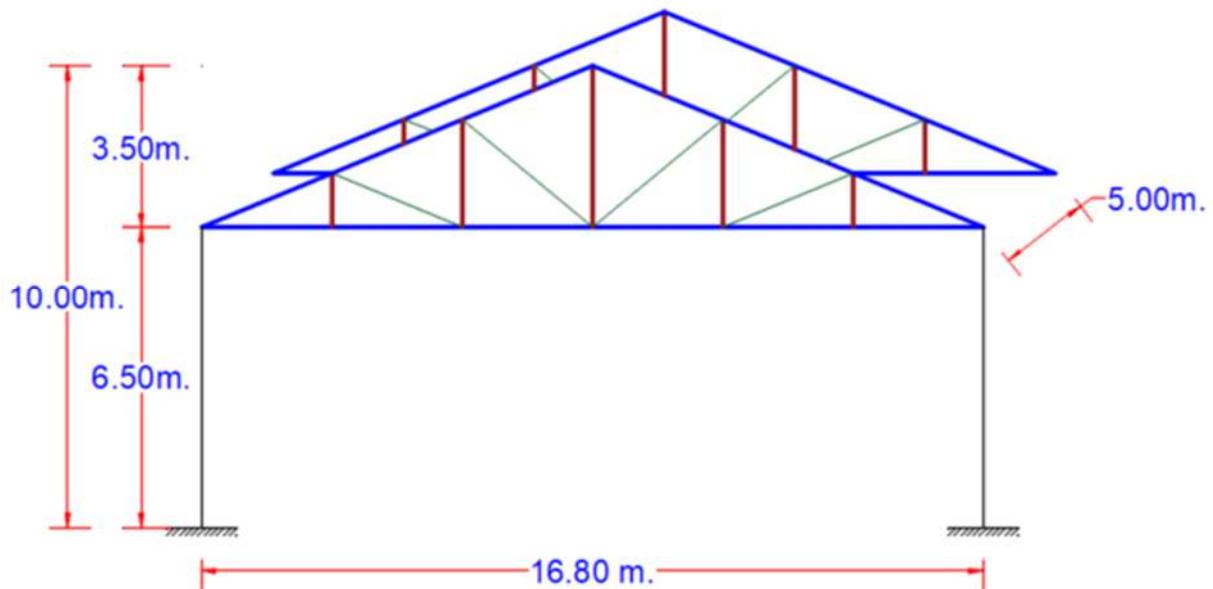


Figura 40. Análisis Convencional - Predimensionamiento del tijeral metálico para el cálculo en el software HoweTeel.

5.1.2.3. Metrado de cargas del tijeral

Para el Metrado de cargas, se hace referencia a la norma E.020 “Cargas” de nuestro Reglamento Nacional de Edificaciones.

5.1.2.3.1. Carga Muerta (CM)

La carga muerta a considerar, será el peso de los materiales y de la armadura fijo y constante, como se detalla a continuación:

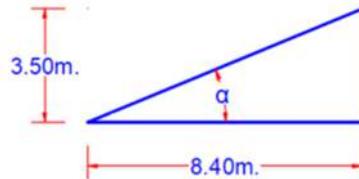
Tabla 6. Disgregado de carga muerta.

Descripción	Peso (Kg/m ²)	Distancia @ Tijeral (ml)	Sub Total (Kg/ml)
Peso de Tijeral + Viguetas	16.00	5.00	80.00
Plancha del Techado	3.00	5.00	15.00
Peso de Luminaria	4.00	5.00	20.00
Sistema contra Incendio	8.00	5.00	40.00
Peso de Extras	4.00	5.00	20.00
		TOTAL CM	175.00

Fuente: elaboración propia.

5.1.2.3.2. Carga Viva (L_r)

Según la norma E.020 del RNE, la carga mínima para techos con inclinación mayor a 3° , con respecto a la horizontal será de 1.0 KPa (100 Kg/m²), si la inclinación del techo es mayor a 3° , entonces, el valor se reducirá en 0.05 KPa (5 Kg/m²) por cada grado de pendiente por encima de los 3° , hasta un mínimo de 0.50 KPa (50 Kg/m²).



$$\tan \alpha = \frac{3.5}{8.4}$$

$$\alpha = \text{Arctang} \left(\frac{3.5}{8.4} \right)$$

$$\alpha = 22.6^\circ$$

$$\Rightarrow L_r = 100 - 5(22.6 - 3)$$

$$L_r = 2$$

$$\Rightarrow \text{Carga Viva } L_r = 50 \text{ Kg/m}^2$$

5.1.2.3.3. Carga de lluvia o granizo (R)

La carga de lluvia, granizo o acumulación de agua, será considerado de acuerdo a la ubicación geográfica y climatológica; para nuestro caso la estructura estará ubicada en la ciudad de Huancayo y se considerarán las siguientes características:

$$\text{Peso x m}^3 = 1000.00 \text{ Kg/m}^3$$

$$\text{Altura o capa de agua} = 0.05 \text{ m.}$$

$$\text{Ancho tributario} = 5.00 \text{ m}$$

$$\text{Carga Lluvia } R = 1000 \times 0.05 \times 5.00$$

$$\text{Carga Lluvia } R = 250.00 \text{ Kg/ml}$$

5.1.2.3.4. Carga Viento

Todo el procedimiento para el cálculo de la carga de viento, se ha tenido en cuenta la Norma E.090 del RNE.

Para la obtención de la velocidad de diseño se ha empleado el mapa eólico del anexo 02 de la Norma E.020 del RNE.

Departamento: Junín.

Provincia: Huancayo

Distrito: Huancayo

Para alturas $h \leq 10$ (desde el nivel 0.00 hasta la cumbrera del tejado)

$\Rightarrow V_h = 70 \text{ Km/h}$

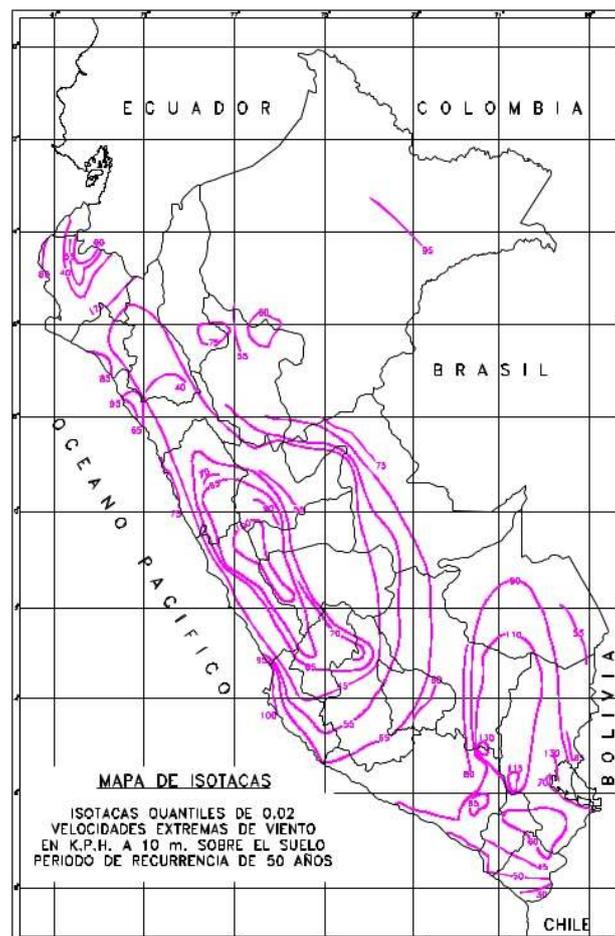


Figura 41. Análisis Convencional - Mapa de Isotacas, para hallar la velocidad del viento.

Para hallar el factor de forma adimensional C, utilizamos la tabla 04 de la norma E.020 del RNE.

Tabla 7. Factores de forma (C)

CONSTRUCCIÓN	BARLOVENTO	SOTAVENTO
Superficies verticales de edificios	+0,8	-0,6
Anuncios, muros aislados, elementos con una dimensión corta en la dirección del viento	+1,5	
Tanques de agua, chimeneas y otros de sección circular o elíptica	+0,7	
Tanques de agua, chimeneas, y otros de sección cuadrada o rectangular	+2,0	
Arcos y cubiertas cilíndricas con un ángulo de inclinación que no exceda 45°	±0,8	-0,5
Superficies inclinadas a 15° o menos	+0,3-0,7	-0,6
Superficies inclinadas entre 15° y 60°	+0,7-0,3	-0,6
Superficies inclinadas entre 60° y la vertical	+0,8	-0,6
Superficies verticales ó inclinadas (planas ó curvas) paralelas a la dirección del viento	-0,7	-0,7

* El signo positivo indica presión y el negativo succión.

Fuente: Norma E.020 del RNE.

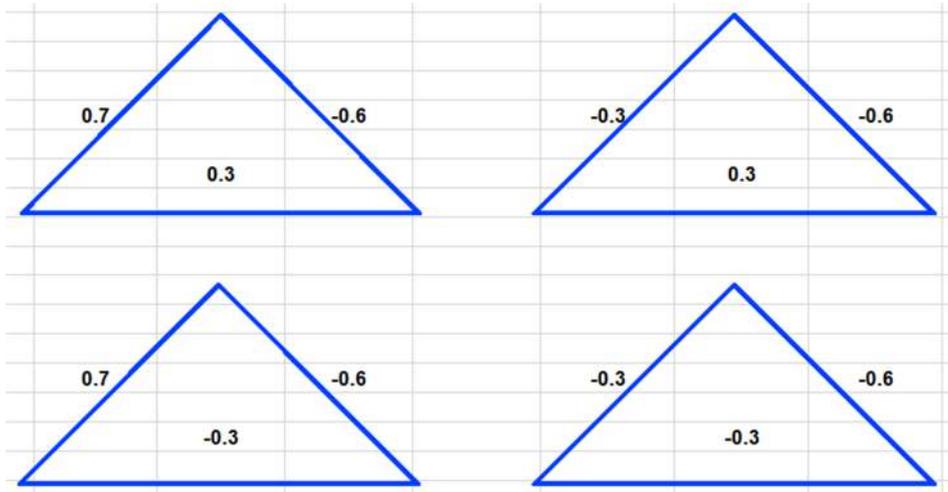
Se determina los factores de forma para determinar las cargas adicionales en elementos de cierre (C), según la tabla 05 de la norma E.020 del RNE.

Tabla 8. Factores de forma para determinar cargas adicionales en elementos de cierre (C).

ABERTURAS		
Uniforme en lados a barlovento y sotavento	Principales en lado a barlovento	Principales en lado a sotavento o en los costados
±0,3	+0,8	-0,6

Fuente: Norma E.020 del RNE.

Entonces, se determina los factores considerados para el diseño:



Se realiza las combinaciones de los factores a usar, quedando de la siguiente manera:

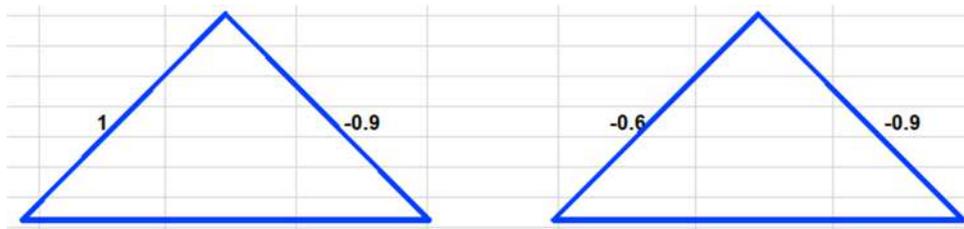


Tabla 9. Valores para el Barlovento y Sotavento.

ANGULO	BARLOVENTO	SOTAVENTO	BARLOVENTO	SOTAVENTO	BARLOVENTO	SOTAVENTO	BARLOVENTO	SOTAVENTO
> 15°	0.7	-0.6	0.7	-0.6	-0.3	-0.6	-0.3	-0.6
	0.3	0.3	-0.3	-0.3	0.3	0.3	-0.3	-0.3
	1.0	-0.3	0.4	-0.9	0.0	-0.3	-0.6	-0.9

Fuente: Elaboración propia.

Se obtiene la carga exterior del viento (Barlovento y Sotavento), para nuestro caso el ambiente es cerrado.

$$Ph = 0.005 * C * Vh^2 \text{ Kg/m}^2$$

	Barlovento		Sotavento
Ph	17.15	-7.35	-14.70

Se determina la carga puntual en los nudos

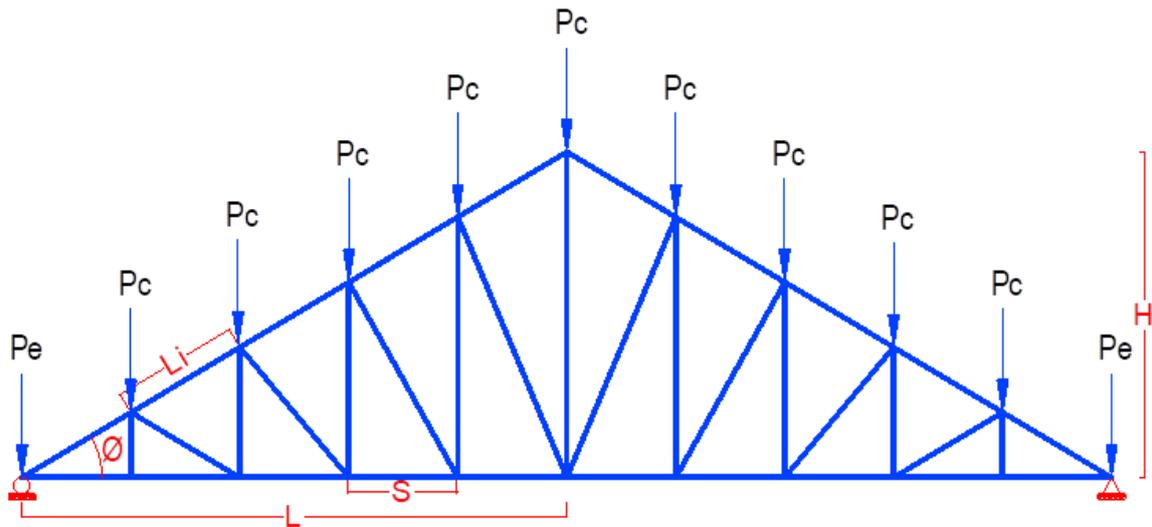


Figura 42. Análisis Convencional - Determinación de la carga puntual en los nudos.

$$Li = \frac{S}{\cos(\theta)}$$

$$Pc = W * Ls * Li$$

$$Pe = \frac{W * Ls * Li}{2}$$

Donde:

Pc = Carga puntual en el nudo central (Kg)

Pe = Carga puntual en el nudo de esquina (Kg)

Li = Longitud inclinada (m)

Ls = Longitud de separación entre tijerales (m)

W = Presión del viento (Kg/m²)

S = Separación entre montantes

Se obtiene un cuadro resumen de Metrado de cargas:

Tabla 10. Resumen de Metrado de cargas.

RELACION LONGITUD - MONTANTE	LONGITUD INCLINADA	CARGA MUERTA	CARGA VIVA	CARGA LLUVIA O GRANIZO	PRESION DEL VIENTO - NUDOS CENTRALES		
					BARLOVENTO		SOTAVENTO
					CARGA PUNTUAL	CARGA PUNTUAL	CARGA PUNTUAL
Ls	Li	D	L	R	W1	W2	(W1, W2)
2.8	3.03	530.25	757.50	757.50	259.82	-111.35	-222.71

Fuente: Elaboración propia.

Se hace el cálculo de los esfuerzos últimos máximos, se realiza el modelamiento y simulación en el programa SAP2000 con los siguientes pasos:

- Configuración de las unidades de medida.
- Se crea un nuevo modelo con la selección de un “Grid Only”.
- Se ingresan los parámetros para la grilla.
- Se define la norma a utilizar.
- Se define los materiales.
- Se define los tipos de cargas.
- Se define las combinaciones de carga.
- Se procede a dibujar en el espacio de trabajo.
- Se define las restricciones para la armadura.
- Se procede a ingresar los esfuerzos o cargas en el modelo de armadura creada.
- Se liberan los frame “Assing frame releases”.
- Se guarda el modelo o archivo.
- Se quita la selección de la carga Modal para la ejecución del modelamiento.
- Se ejecuta el modelamiento respectivo.
- Para el diagrama de los esfuerzos se selección el tipo de combinación a utilizar en el análisis.
- Se procede a seleccionar los esfuerzos últimos más críticos para utilizar en al análisis por tracción y compresión.

Así mismo para verificar los datos de los esfuerzos últimos del programa SAP2000, se realizó una verificación manual, como se muestra a continuación:

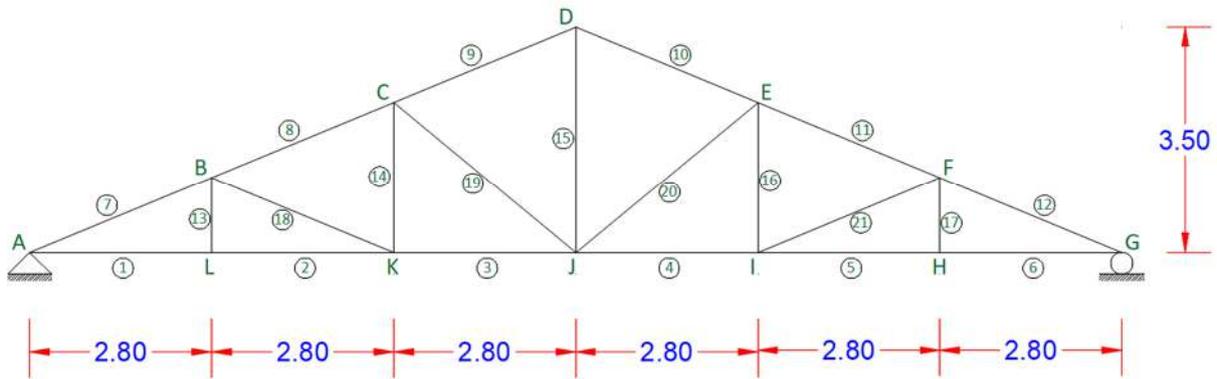


Figura 43. Análisis Convencional - Diseño para la verificación manual de las esfuerzos últimos.

Para la carga muerta (DEAD)

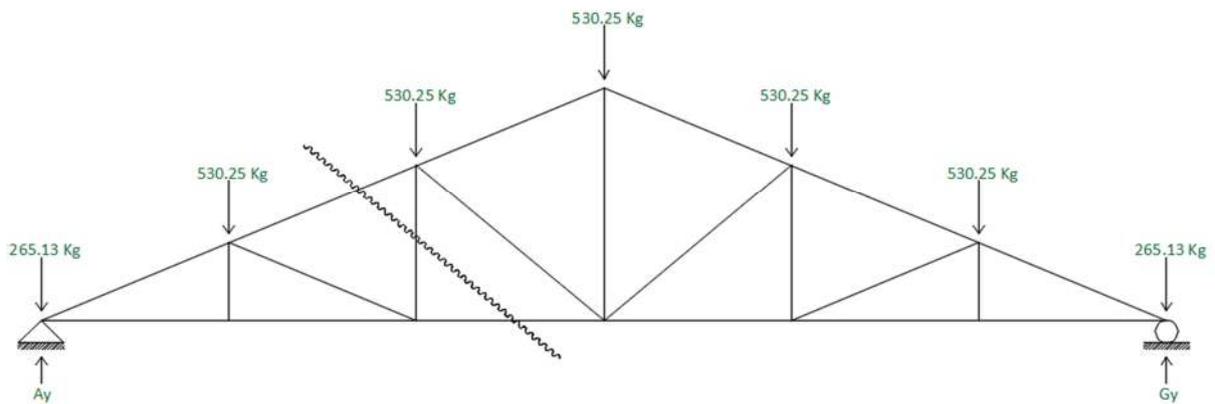


Figura 44. Análisis Convencional - Verificación manual para la carga muerta (D)

$$\Sigma M_G = 0$$

$$A_Y(16.80) - 265.13(16.80) - 530.25(14.00) - 530.25(11.20) - 530.25(8.40) - 530.25(5.60) - 530.25(2.80) = 0$$

$$16.80 A_Y = 26\ 724.68$$

$$A_Y = 1\ 590.76$$

$$\Sigma F_Y = 0$$

$$1\ 590.76 - 2(265.13) - 5(530.25) + G_Y = 0$$

$$G_Y = 1\ 590.75$$

$$\Sigma M_K = 0$$

$$1\,590.75(5.60) - 265.13(5.60) - 530.25(2.80) - F_{CB} \operatorname{sen}(22.62)(2.80) -$$

$$F_{CB} \operatorname{cos}(22.62)(1.1667) = 0$$

$$5\,938.83 = 2.154 F_{CB}$$

$$F_{CB} = 2\,757.12 \text{ Kg (Compresión)}$$

$$\Sigma M_C = 0$$

$$1\,590.75(5.60) - 265.13(5.60) - 530.25(2.80) - F_{KJ}(2.3334) = 0$$

$$5\,938.83 = 2.3334 F_{KJ}$$

$$F_{KJ} = 2\,545.14 \text{ Kg (Tensión)}$$

$$\Sigma F_Y = 0$$

$$1\,590.76 - 265.13 - 530.25 - 2\,757.12(\operatorname{sen} \theta) + F_{CK} = 0$$

$$F_{CK} = 265.06 \text{ Kg (Tensión)}$$

$$F_{BA} = 12\,423.23 \text{ Kg (Compresión)}$$

$$F_{AL} = 11\,947.27 \text{ Kg (Tensión)}$$

5.1.2.1. Diseño por Tracción

De los resultados del análisis se tiene:

$$P_u = 11\,947.27 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\phi = 0.90$$

$$L = 2.8 \text{ m.}$$

Límite de fluencia por tracción:

$$P_{nf} = \phi \cdot F_y \cdot A_g$$

$$A_g = \frac{11\,947.27}{0.9 (2\,530)} = 5.247 \text{ cm}^2 = 0.81 \text{ plg}^2$$

Donde:

P_{nf} = Limite de fluencia por tracción.

ϕ = Factor de resistencia.

F_y = Fluencia del acero

A_g = Area bruta de la sección.

Selección de sección a utilizar: 2L 2 x 2 x 1/8

Características geométricas de la sección:

$$A_g = 0.982 \text{ plg}^2 = 6.34 \text{ cm}^2$$

$$r_x = 0.62 \text{ plg} = 1.58 \text{ cm}$$

$$P_{nf} = 0.9 \times 2\,530 \times 6.34 = 14,425.87 \text{ Kg} \dots \dots OK$$

Verificación por esbeltez

$$\frac{KL}{r} < 300 \quad \text{donde } K = 1$$

$$\frac{(1)(280)}{1.58} < 300$$

$$177.22 < 300 \dots \dots OK$$

5.1.2.2. Diseño por Compresión

De los resultados del análisis se tiene:

$$P_u = 12,423.23 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\phi = 0.90$$

$$L = 3.03 \text{ m.}$$

Selección de sección a utilizar: la sección 2L 2 x 2 x 1/8, elegido en el diseño por tracción no siempre cumple con el A_g (área bruta) necesaria para el diseño por compresión, normalmente se elige otra sección con un área mayor, por lo tanto, la otra sección elegida será la 2L 3 x 3 x 1/4

Características geométricas de la sección:

$$A_g = 2.87 \text{ plg}^2 = 18.52 \text{ cm}^2$$

$$b = 3.00 \text{ plg} = 7.62 \text{ cm}$$

$$t = 0.25 \text{ plg} = 0.64 \text{ cm}$$

$$E = 29\,000 \text{ Klb/plg}^2$$

$$F_y = 36 \text{ Klb/plg}^2$$

$$r_x = 0.93 \text{ plg} = 2.35 \text{ cm}$$

$$r_y(0) = 1.25 \text{ plg} = 3.18 \text{ cm}$$

Verificación si el elemento es esbelto

$$\frac{b}{t} < \lambda r \quad \text{donde según tabla } \lambda r = 0.45 \sqrt{\frac{E}{F_y}}$$

$$\frac{3.00}{0.25} < 0.45 \sqrt{\frac{29\,000}{36}}$$

$$12 < 12.77 \dots \dots OK$$

Relación de esbeltez

$$\frac{KL}{r} < 200$$

$$\frac{(1)(303)}{2.35} < 200$$

$$128.94 < 200 \dots \dots OK$$

Cálculo del esfuerzo crítico o de pandeo a flexión

$$\frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}}$$

$$128.94 \leq 133.68 \dots \dots OK$$

$$\text{Aplicando la ecuación: } F_{cr} = \left(0.658^{\frac{F_y}{F_e}}\right) F_y$$

Por Euler:

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

$$F_e = \frac{\pi^2 (29\,000)}{128.94^2} = 17.22 \text{ Klb/plg}^2$$

$$\Rightarrow F_{cr} = \left(0.658^{\frac{36}{20.20}}\right) \times 36 = 15.00 \text{ Klb/plg}^2$$

Para LRFD, $\phi_c = 0.90$

$$\Rightarrow \phi_c F_{cr} = 13.50 \text{ Klb/plg}^2$$

Cálculo de la resistencia de diseño por pandeo

$$\phi_c P_n = \phi_c F_{cr} \times A_g$$

$$\phi_c P_n = 13.50 \times 2.87$$

$$\phi_c P_n = 38.75 \text{ Klb}$$

$$\Rightarrow \phi_c P_n = 17,592.50 \text{ Kg} \dots \dots OK$$

5.2. Contrastación de hipótesis

5.2.1. Hipótesis específica 1: El resultado del análisis y diseño de techos con tijerales

metálicos tipo howe mediante el visual C# y el API del SAP2000, es conveniente.

Para la contrastación de hipótesis, se procederá a realizar el análisis y diseño del techo metálico tipo Howe del Terminal Terrestre Los Andes utilizando el software HoweSteel, creado con el visual C# y el API del SAP2000.

Las medidas tomadas en campo se aprecian en la siguiente figura:

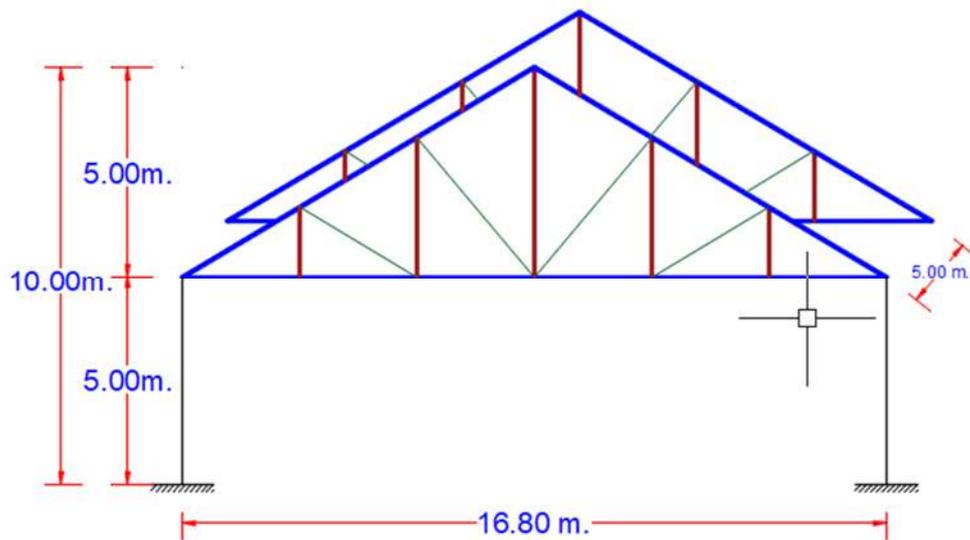


Figura 45. Contrastación Hipótesis - Predimensionamiento del tejeral metálico para el cálculo en el software HoweSteel.

Se procede con el ingreso de los datos respectivos al software HoweSteel para su diseño y análisis.

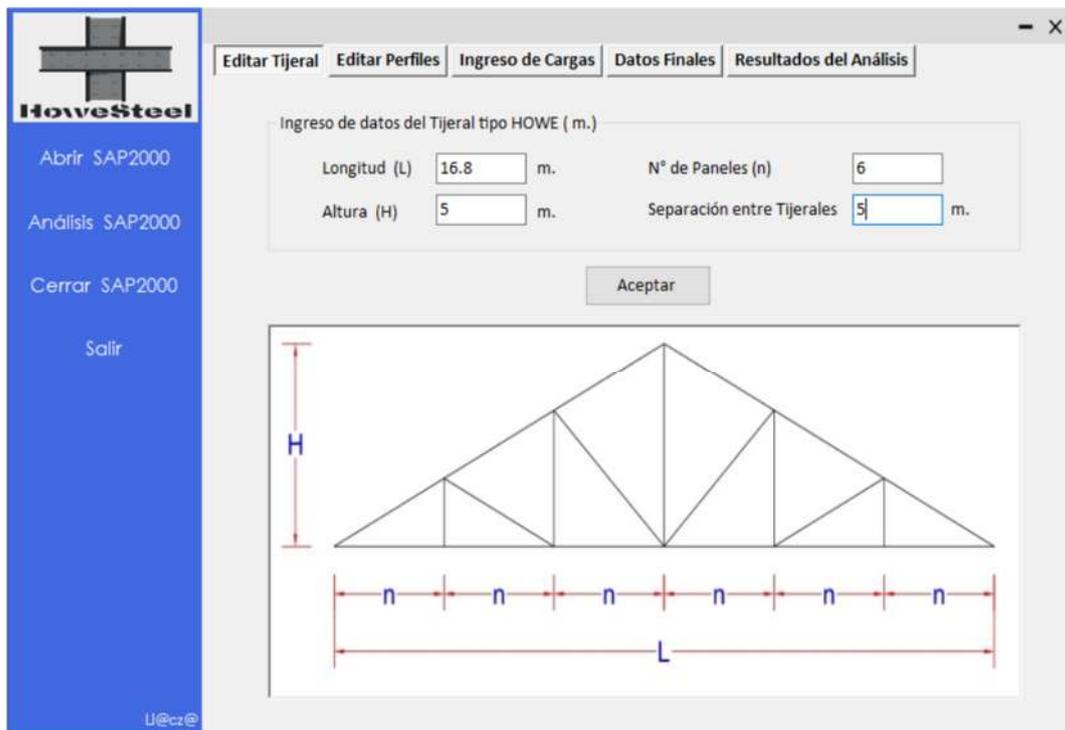


Figura 46. Contrastación Hipótesis – Ingreso de datos en la pestaña “Editar Tijeral” en el software HoweSteel.

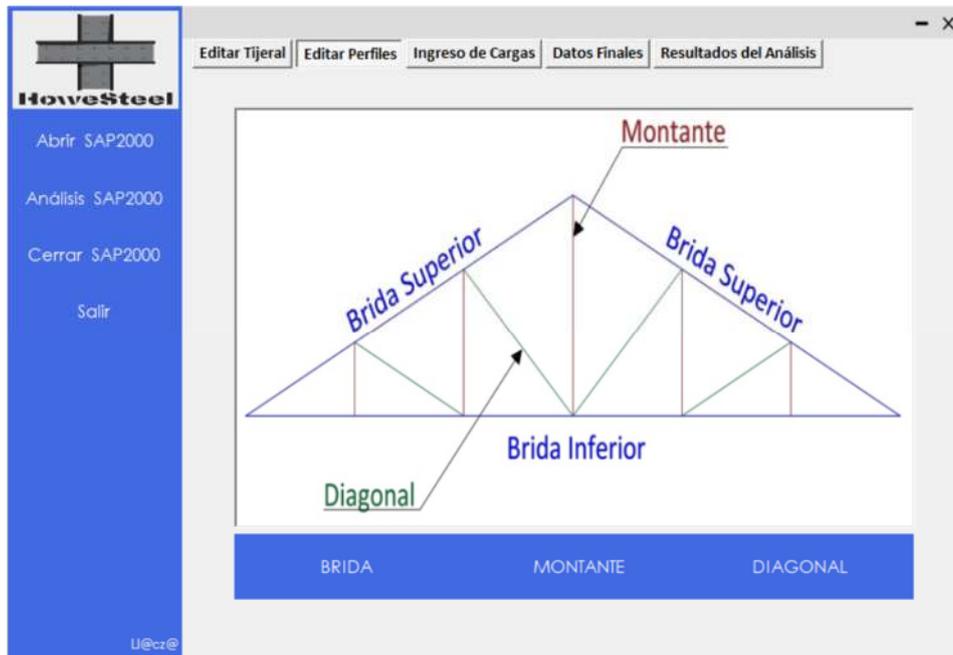


Figura 47. Contrastación Hipótesis – Ingreso de datos en la pestaña “Editar Perfiles”, en el software HoweSteel.

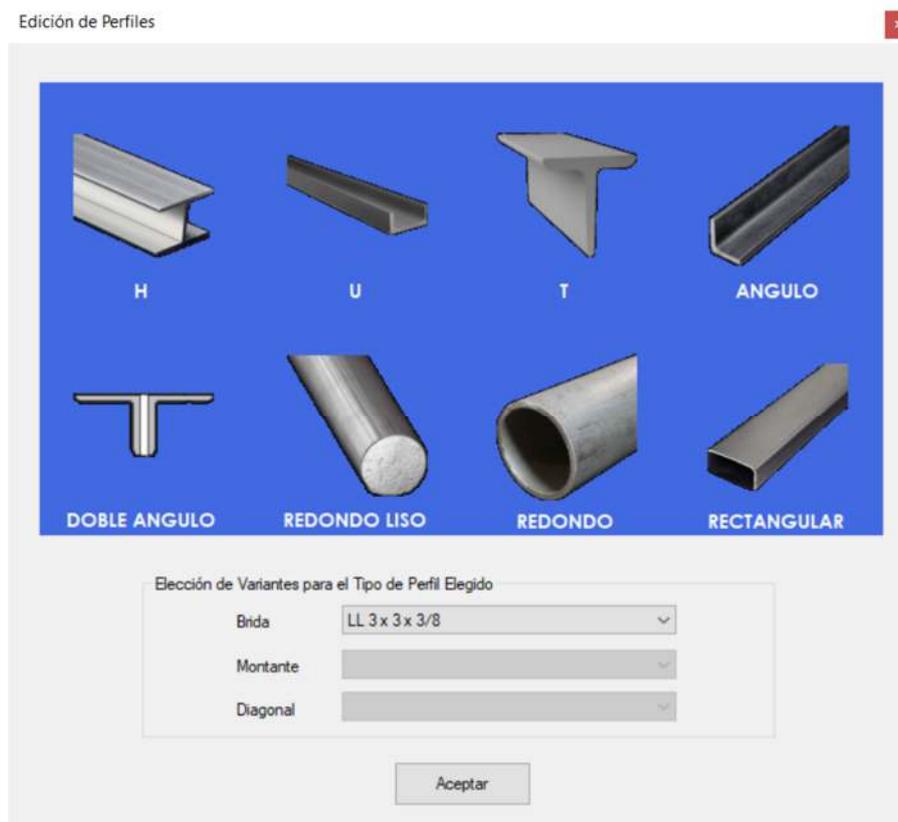


Figura 48. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la brida en el software HoweSteel

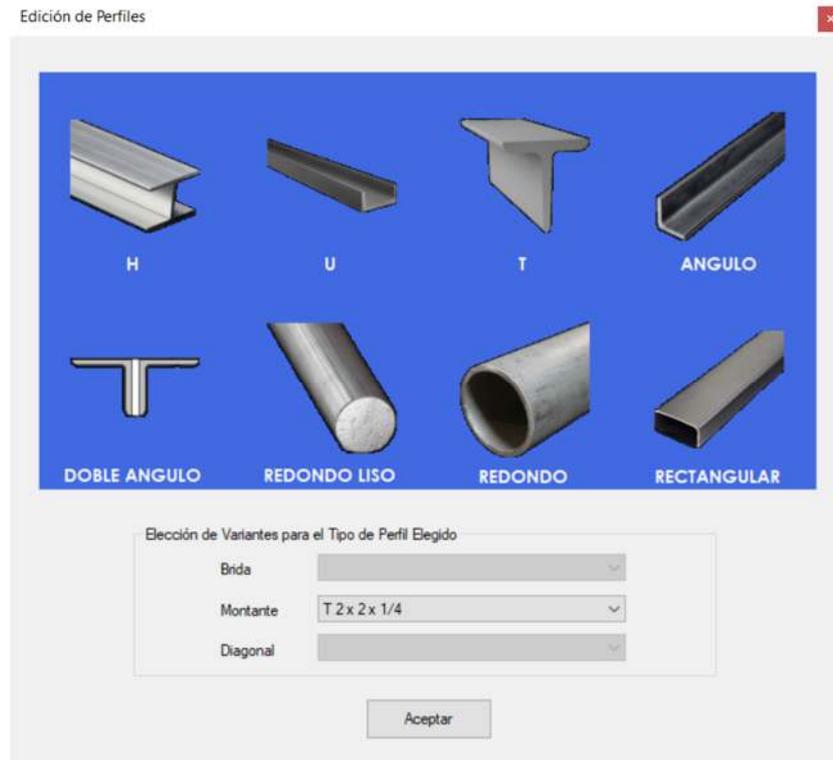


Figura 49. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la Montante en el software HoweSteel

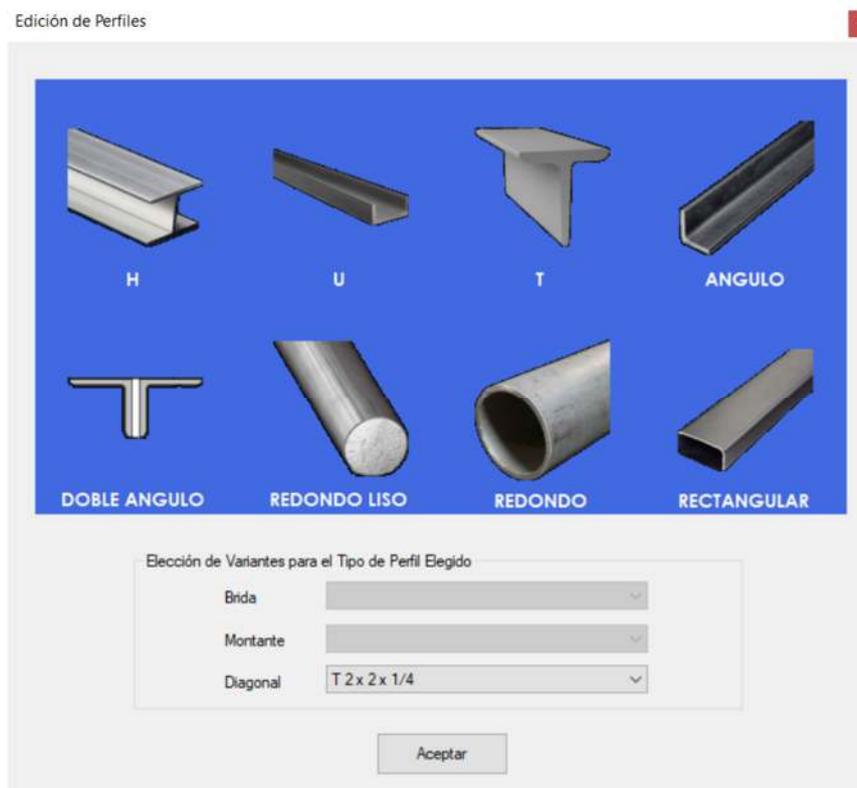


Figura 50. Contrastación Hipótesis – Ingreso de datos en “Edición de perfiles” para la Diagonal en el software HoweSteel.

Para el ingreso de Cargas, el programa recibe los datos de las cargas que interviene en el análisis, como podemos apreciar en la Figura 51, para la carga Muerta, se ingresan los valores ya preestablecidos; para la carga de lluvia o granizo, se ha considerado una capa de lluvia o granizo de 0.05 con un peso de 1000 kg/m² en caso extremo y para la carga de viento dependerá de la ubicación y localización del techo metálico tipo Howe.

Figura 51. Contrastación Hipótesis – Ingreso de datos en la pestaña “Ingreso de Cargas”, en el software HoweSteel.

En la pestaña Datos finales del programa HoweSteel, se verifican todos los datos ingresados, si hubiera algún error, se puede hacer los cambios respectivos, ingresando los datos correctos nuevamente. En la figura 52, se aprecia los datos que serán analizados para su posterior diseño; todos los procesos para las cargas muerta, carga viva y carga de viento se han procesado de acuerdo a la RNE y la norma E.090 y la E.020.

Editar Tijeral **Editar Perfiles** **Ingreso de Cargas** **Datos Finales** **Resultados del Análisis**

Datos del Tijeral Para el Análisis

Longitud	16.8 m.	Cantidad Segmentos	6
Altura	5 m.	Separación Entre Tijerales	5 m.

Datos de los Perfiles a Usar Para Análisis

Brida	LL 3 x 3 x 3/8
Montante	T 2 x 2 x 1/4
Diagonal	T 2 x 2 x 1/4

Datos de los Tipos de Carga Para el Análisis

Carga Muerta	35 Kg/m ²	Carga de Lluvia o Granizo	50 Kg/m ²
Carga Viva	50 Kg/m ²		

Carga de Viento

Barlovento W1	17.15 Kg/m ²	Sotavento W1	-14.7 Kg/m ²
Barlovento W2	-7.35 Kg/m ²	Sotavento W2	-14.7 Kg/m ²

Figura 52. Contratación Hipótesis – Verificación en la pestaña “Datos Finales”, en el software HoweSteel.

En la figura 53, apreciamos que la automatización es el adecuado, ya que al procesar los datos, obtenemos el tijeral en el SAP2000, con todas las características según nuestro ingreso de datos y en base a la NTP E.090 y la E.020.

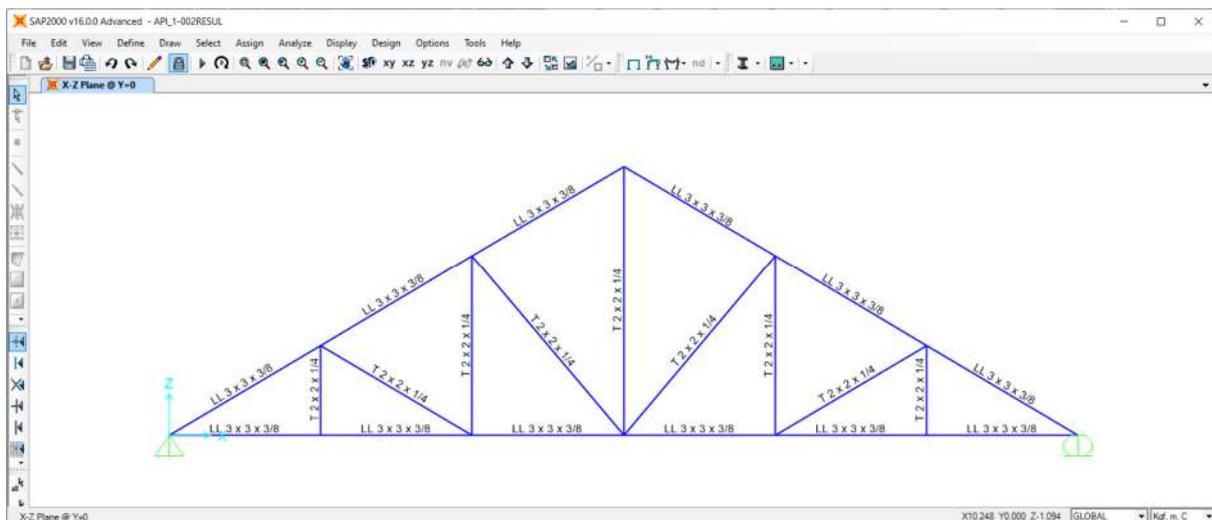


Figura 53. Contratación Hipótesis – Datos procesados después de ingresar en el botón “Análisis SAP2000”, del software HoweSteel.

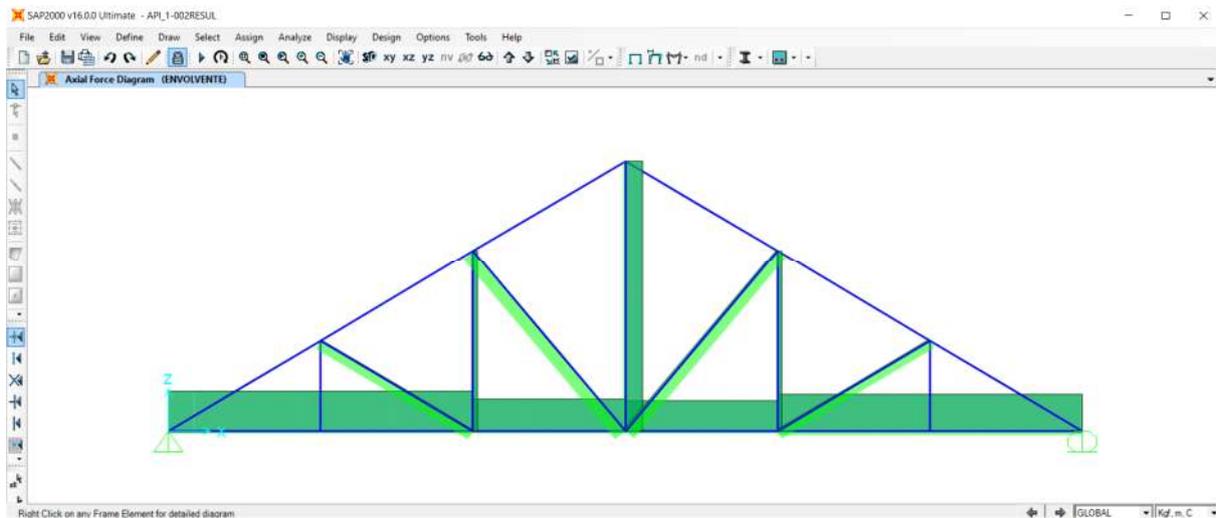


Figura 54. Contrastación Hipótesis – Verificación de los esfuerzos últimos (opcional).

Resumen de Esfuerzos Ultimos

ITEM	NOMBRE FRAME	TRACCION	COMPRESION
1	1-1	9194.97	-660.36
2	2-1	9194.97	-660.36
3	3-1	7305.93	-508.11
4	4-1	6899.93	-660.36
5	5-1	8382.97	-964.86
6	6-1	8382.97	-964.86
7	7-1	1139.32	-9901.92

Esfuerzos Ultimos Críticos

BRIDA

Tracción	Compresión
1-1: 9194.97 Kgf	7-1: -9901.92 Kgf

MONTANTE

Tracción	Compresión
15-1: 4014.38 Kgf	15-1: -543.75 Kgf

DIAGONAL

Tracción	Compresión
20-1: 473.42 Kgf	19-1: -2936.97 Kgf

Figura 55. Obtención de los esfuerzos críticos en la pestaña “Resultados de análisis” del software HoweSteel.

5.2.1.1. Diseño por Tracción

De los resultados del análisis del programa HoweSteel se tiene:

$$P_u = 9,194.97 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\emptyset = 0.90$$

$$L = 2.8 \text{ m.}$$

Límite de fluencia por tracción:

$$P_{nf} = \emptyset \cdot F_y \cdot A_g$$

$$A_g = \frac{9\,194.97}{0.9 (2\,530)} = 4.04 \text{ cm}^2 = 0.63 \text{ plg}^2$$

Donde:

P_{nf} = Límite de fluencia por tracción.

\emptyset = Factor de resistencia.

F_y = Fluencia del acero

A_g = Área bruta de la sección.

Selección de sección a utilizar: 2L 2 x 2 x 1/8

Características geométricas de la sección:

$$A_g = 0.982 \text{ plg}^2 = 6.34 \text{ cm}^2$$

$$r_x = 0.62 \text{ plg} = 1.58 \text{ cm}$$

$$P_{nf} = 0.9 \times 2\,530 \times 6.34 = 14,425.87 \text{ Kg} \dots \dots OK$$

Verificación por esbeltez

$$\frac{KL}{r} < 300 \quad \text{donde } K = 1$$

$$\frac{(1)(280)}{1.58} < 300$$

$$177.22 < 300 \dots \dots OK$$

5.2.1.2. Diseño por Compresión

De los resultados del análisis del programa HoweSteel se tiene:

$$P_u = 9,901.92 \text{ Kg.}$$

$$F_y = 2\,530 \text{ Kg/cm}^2$$

$$\phi = 0.90$$

$$L = 3.26 \text{ m.}$$

Selección de sección a utilizar: la sección 2L 2 x 2 x 1/8, elegido en el diseño por tracción no siempre cumple con el A_g (área bruta) necesaria para el diseño por compresión, normalmente se elige otra sección con un área mayor, por lo tanto, la otra sección elegida será la 2L 4 x 4 x 3/8

Características geométricas de la sección:

$$A_g = 5.71 \text{ plg}^2 = 36.84 \text{ cm}^2$$

$$b = 4.00 \text{ plg} = 10.16 \text{ cm}$$

$$t = 0.375 \text{ plg} = 0.95 \text{ cm}$$

$$E = 29\,000 \text{ Klb/plg}^2$$

$$F_y = 36 \text{ Klb/plg}^2$$

$$r_x = 1.23 \text{ plg} = 3.12 \text{ cm}$$

$$r_y(0) = 1.67 \text{ plg} = 4.24 \text{ cm}$$

Verificación si el elemento es esbelto

$$\frac{b}{t} < \lambda r \quad \text{donde según tabla } \lambda r = 0.45 \sqrt{\frac{E}{F_y}}$$

$$\frac{4.00}{0.375} < 0.45 \sqrt{\frac{29\,000}{36}}$$

$$10.67 < 12.77 \dots \dots OK$$

Relación de esbeltez

$$\frac{KL}{r} < 200$$

$$\frac{(1)(326)}{3.12} < 200$$

$$104.49 < 200 \dots \dots OK$$

Cálculo del esfuerzo crítico o de pandeo a flexión

$$\frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}}$$

$$104.49 \leq 133.68 \dots \dots OK$$

Aplicando la ecuación: $F_{cr} = \left(0.658 \frac{F_y}{F_e}\right) F_y$

Por Euler:

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

$$F_e = \frac{\pi^2 (29\,000)}{104.49^2} = 26.21 \text{ Klb/plg}^2$$

$$\Rightarrow F_{cr} = \left(0.658 \frac{36}{26.21}\right) \times 36 = 20.26 \text{ Klb/plg}^2$$

Para LRFD, $\phi_c = 0.90$

$$\Rightarrow \phi_c F_{cr} = 18.23 \text{ Klb/plg}^2$$

Cálculo de la resistencia de diseño por pandeo

$$\phi_c P_n = \phi_c F_{cr} \times A_g$$

$$\phi_c P_n = 18.23 \times 5.71$$

$$\phi_c P_n = 104.09 \text{ Klb}$$

$$\Rightarrow \phi_c P_n = 47,256.86 \text{ Kg} \dots \dots OK$$

Con el resultado final podemos, obtenemos la resistencia ultima que nos ofrece la sección elegida (2L 4 x 4 x 3/8), la cual cumple con el análisis del tijeral metálico tipo Howe planteado.

Entonces podemos mencionar que con un análisis y diseño de tijaerales metálicos tipo Howe mediante el visual C# y el API del SAP2000 es conveniente, porque es óptimo en los tiempos y minimiza los procesos; por lo tanto es de gran ayuda al proyectista.

5.2.2. Hipótesis específica 2: El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a mejorar los tiempos necesarios para un diseño óptimo de un tijeral metálico tipo Howe.

Según nuestra hipótesis específica, los tiempos que se pueden optimizar en el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API SAP2000, dependerán de los procesos que se lleven a cabo, para lo cual se ha hecho una encuesta a 20 personas y se ha determinado el tiempo promedio que conlleva a realizar un análisis y diseño con el software HoweSteel y de la manera tradicional, el cual se puede apreciar en la figura 56, figura 57 y figura 58.

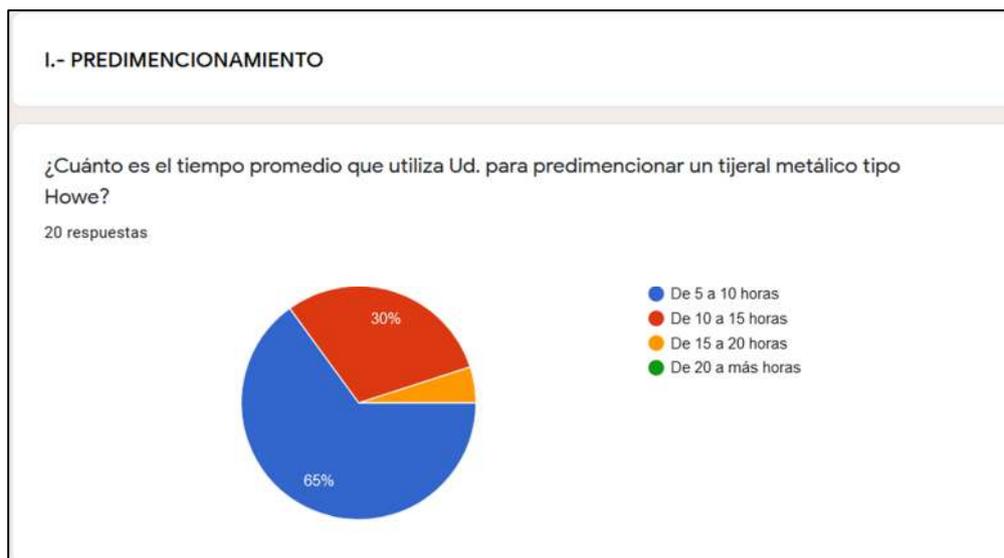


Figura 56. Encuesta para determinar el tiempo promedio en el predimensionamiento tradicional.

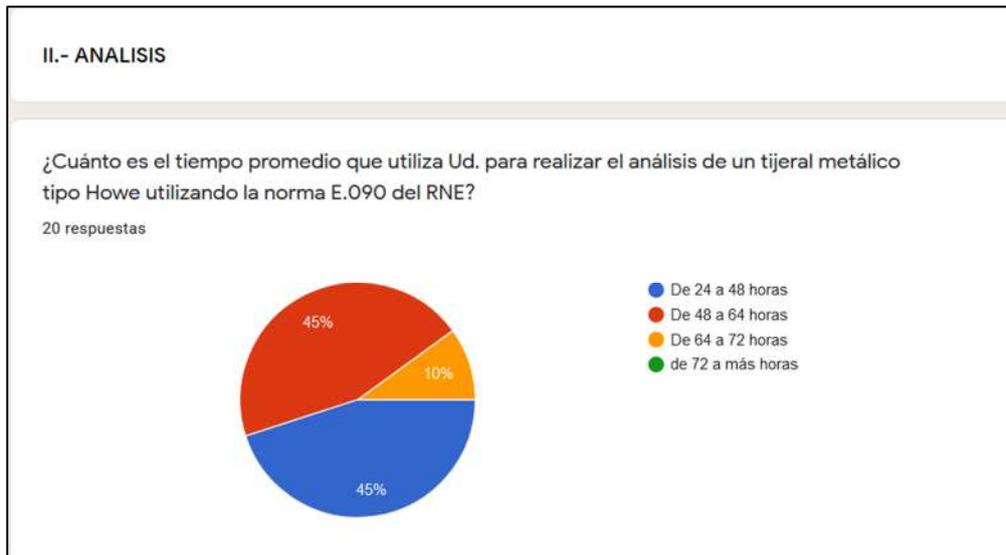


Figura 57. Encuesta para determinar el tiempo promedio en el análisis tradicional.

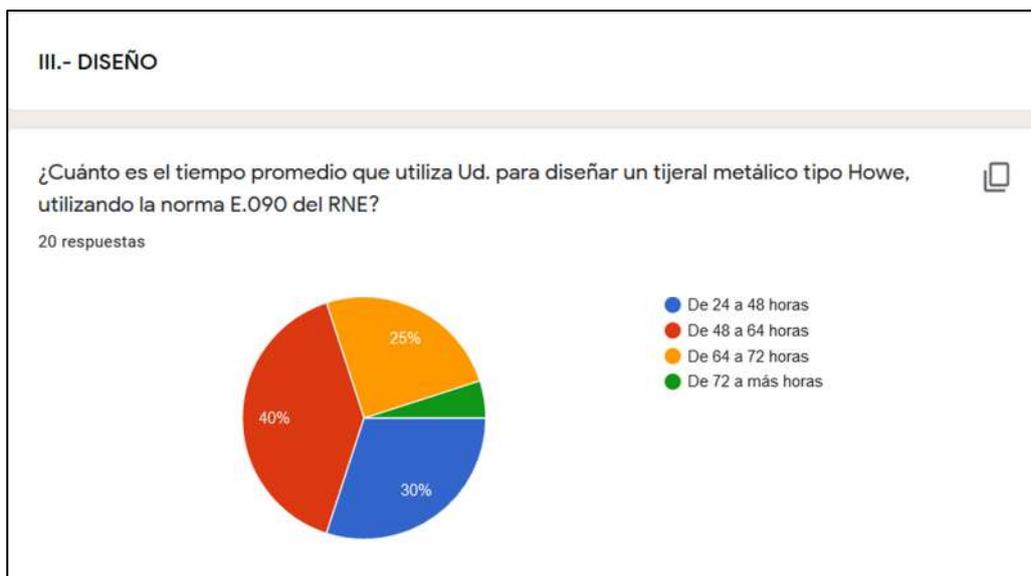


Figura 58. Encuesta para determinar el tiempo promedio en el diseño tradicional.

Tabla 11. Tiempos promedio durante el proceso de análisis y diseño.

TIPO DE ANALISIS Y DISEÑO	PROCESOS EN EL ANALISIS	TIEMPOS PROMEDIO DURANTE EL PROCESO
	Predimensionamiento. Etapa en la cual se determina: el peralte del tijeral, cantidad de paneles, distancia entre montantes y distancia entre tijerales.	Predimensionamiento = de 5 a 10 horas.

Mediante la forma convencional	<p>Análisis. Etapa en la cual se determina: carga viva, carga muerta, carga viento (barlovento y sotavento), carga de lluvia y granizo, dibujo en CAD, dibujo en SAP2000, preparar el espacio de trabajo en el SAP2000 y el análisis en el SAP2000.</p> <p>Diseño. Etapa en la cual se determina: diseño por tracción de un perfil de acero y diseño por compresión de un perfil de acero.</p> <p>Predimensionamiento. Etapa en la cual se determina: el peralte del tijeral, cantidad de paneles, distancia entre montantes y distancia entre tijerales.</p>	<p>Análisis = de 48 a 64 horas.</p> <p>Diseño = de 48 a 64 horas.</p> <p>Predimensionamiento = de 5 a 10 horas.</p>
Mediante el C# y el API del SAP2000	<p>Análisis. Etapa en la cual se determina: carga viva, carga muerta, carga viento (barlovento y sotavento), carga de lluvia y granizo, dibujo en CAD, dibujo en SAP2000, preparar el espacio de trabajo en el SAP2000 y el análisis en el SAP2000.</p> <p>Diseño. Etapa en la cual se determina: diseño por tracción de un perfil de acero y diseño por compresión de un perfil de acero.</p>	<p>Análisis = 0.10 a 0.30 horas.</p> <p>Diseño = de 2.00 a 3.00 horas.</p>

Fuente: Elaboración propia.

Según la tabla 11, podemos indicar que el enunciado en la hipótesis específica “El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a mejorar los tiempos necesarios para un diseño óptimo de un tijeral metálico tipo howe”, es cierta.

5.2.3. Hipótesis específica 3: El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a minimizar procesos que existen en un análisis y diseño tradicional o convencional.

Según nuestra hipótesis específica, los procesos que se han empleado para el análisis y diseño de techos con tijerales metálicos tipo howe se han reducido en comparación con un análisis y diseño tradicional, esto lo podemos apreciar desde la Figuras 45 a la Figura 55; así mismo, podemos apreciar en la siguiente tabla los procesos que se incluye con el software HoweSteel y de la forma tradicional:

Tabla 12. Procesos en el análisis y diseño.

TIPO DE ANALISIS Y DISEÑO	PROCESOS EN EL ANALISIS Y DISEÑO
Mediante la forma tradicional.	<ul style="list-style-type: none"> • Obtención de datos. • Predimensionamiento: peralte del tijeral, cantidad de paneles, distancia entre montantes y distancia entre tijerales. (Manual u hoja Excel) • Análisis y metrado de carga viva. • Análisis y metrado de carga muerta. • Análisis metrado de carga viento (barlovento y sotavento). • Análisis y metrado de carga lluvia o granizo. • Determinación de la carga puntual en los nudos. • Dibujo en CAD. • Preparar el espacio de trabajo en el SAP2000. • Definición de tipo de cargas en el SAP2000. • Definición de los tipos de combinaciones en el SAP2000. • Definición del tipo de material en el SAP2000 • Definición de las características y dimensiones del tipo de perfil en el SAP2000. • Dibujo de los objetos frame en el SAP2000. • Incluir las restricciones del objeto frame. • Cargar los esfuerzos puntuales de carga viva, muerta, viento y lluvia o granizo en el SAP2000. • Liberación de los momentos en los nudos en el SAP2000. • Análisis utilizando el SAP2000. • Verificación de resultados. • Diseño por tracción. • Diseño por compresión.

<p>Mediante el software HoweSteel, creado con el visual C# y el API del SAP2000.</p>	<ul style="list-style-type: none"> • Obtención de datos. • Predimensionamiento. • Análisis de cargas efectuanes. (automatizado) • Diseño utilizando el API del SAP2000. (automatizado) • Verificación de resultados. (automatizado) • Diseño por tracción. • Diseño por compresión.
--	--

Fuente: Elaboración propia.

Según la tabla 12, podemos indicar que el enunciado en la hipótesis específica “El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a minimizar procesos que existen en un análisis y diseño tradicional o convencional”, es adecuado en la minimización de procesos, ya que en un análisis y diseño de forma tradicional se aprecia 21 procesos y con un análisis y diseño utilizando el software HoweSteel, creado con el visual C# y el API del SAP2000 obtenemos 7 procesos.

Para terminar, se adjunta en los anexos el plano EST.01, donde se puede apreciar tres imágenes del análisis y diseño que se realizó en la contratación de hipótesis, donde el proyectista puede determinar el tijeral o armadura tipo Howe más óptima para una luz de 16.80m.

ANÁLISIS Y DISCUSIÓN DE RESULTADOS

El análisis y diseño de techos con tijerales metálicos tipo howe, trabajado de la forma tradicional o convencional, viene utilizando en la ejecución de sus procesos, diferentes software y trabajos de gabinete, las cuales son complejos y extensos; de estos procesos tradicionales obtenemos como resultado un análisis y diseño confiable y correcto.

Según el RNE, las combinaciones de cargas serán de acuerdo a la norma E.090 “Estructuras Metálicas”, donde manifiesta que la resistencia requerida de la estructura y sus elementos deben ser determinada para la adecuada combinación crítica de cargas factorizadas, así mismo menciona que para la aplicación del método LRFD se deben de investigar las combinaciones de carga según se manifiesta en el numeral 1.4.1. Cargas, factores de carga y combinaciones de cargas. Este punto se encuentra automatizado dentro del código fuente del programa HoweSteel.

Las diferentes secciones que se pueden elegir en el diseño del tijeral metálico, están determinadas bajo la norma ASTM A36, la cual ya viene definido en el SAP2000 y esta a su vez se encuentra automatizado dentro del código fuente del programa HoweSteel.

El análisis y diseño de techos con tijerales metálicos tipo howe, realizado de forma automatizada utilizando el software HoweSteel creado con el visual C# y el API del SAP200, también resulta confiable y correcto, y más aún, con el auge de la TIC y el método BIM en las construcciones, se puede adaptar fácilmente a lo que el proyectista busca, optimizar tiempos, minimizar procesos y un buen análisis y diseño.

CONCLUSIONES

- Por medio del presente proyecto, podemos apreciar que la tecnología es fundamental en este siglo XXI, es así que, al utilizar un análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es conveniente, para las empresas consultoras, proyectistas y todo profesional inmerso a las construcciones metálicas, utilizar el software HoweSteel, porque automatizamos los procesos de un análisis y diseño tradicional y obtenemos resultados rápidos y precisos.
- Se puede apreciar en los resultados obtenidos, y la contrastación de hipótesis, que al realizar un análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, podemos reducir los tiempos en los diferentes procesos, más aún, cuando el proyectista requiera reestructurar o rediseñar, el programa HoweSteel le proporcionara la herramienta necesaria para efectuar los análisis con el menor tiempo posible, esto mejora los trabajos del proyectista para beneficio del usuario final.
- Así mismo utilizando el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000 y comparándolo con el análisis y diseño de forma tradicional, podemos afirmar que: un análisis y diseño de forma tradicional nos da como resultado 21 procesos y utilizando el software HoweSteel nos da 7 procesos (tabla. 12), la cual nuestra hipótesis es válida.
- El análisis y diseño de techos con tijerales metálicos tipo howe con el visual C# y el API del SAP2000, ofrece una automatización al análisis y diseño tradicional, para beneficio del proyectista y de los usuarios finales.
- Producto de la programación del C# y el API del SAP2000, se ha obtenido el software denominado HoweSteel, el cual, nos sirve para el procesamiento de datos y análisis, es dinámico, versátil y fácil de utilizar.

RECOMENDACIONES

- Se recomienda utilizar el software HoweSteel, para el análisis y diseño de techos con tijerales metálicos tipo howe, ya que es el aplicativo final de la programación entre el visual C# y el API del SAP2000, y como herramienta tecnológica, es adecuada para el diseño de estos tipos de estructuras o armaduras.
- En la etapa de estructuración y diseño, nos podemos enfrentar a un proceso ciclico, el cual salir de ese proceso, hará que la estructura final sea la más optima; he aquí que los tiempos se amplían para el proyectista, entonces podemos recomendar utilizar el software HoweSteel, para que los tiempos puedan ser mínimos.
- Se recomienda que el proyectista, consultoría o profesional inmerso a la construcción metálica, específicamente en el análisis y diseño de tijerales metálicos tipo Howe, utilice el programa HoweSteel, para poder minimizar los procesos de una manera clara, rápida y confiable.
- Se recomienda que para la utilización del software HoweSteel, se instale la versión 16.0.0 del SAP2000 o superior.
- Se recomienda verificar los datos antes del procesamiento en el programa HoweSteel, aunque el programa ya tiene incluido validación de datos reales.
- Se recomienda verificar bien, las fuerza ultimas de diseño, tanto por tracción y por compresión, en la pestaña “Resultado de Análisis” del software HoweSteel.

REFERENCIAS BIBLIOGRÁFICAS

1. AISC (American Institute of Steel Constructions); 2011; Manual of Steel Construction; Fourteenth Edición; USA.
2. ASOCIACION LATINOAMERICANA DEL ACERO; Especificación para construcciones de acero; 2016.
3. COMPUTERS & STRUCTURES INC.; 2016; Csi OAPI Documentation.
4. COMPUTERS & STRUCTURES INC.; Tutorial SAP2000 Integrated Solution for Structural Analysis and Design; 2016; 73pp.
5. CORPORACION ACEROS AREQUIPA; Catálogo de productos y servicios; Peru; 2021.24pp.
6. HARRY PARKER; 1991; Diseño Simplificado de Armaduras de Techo para Arquitectos y Constructores; Segunda Edición; Mexico D.F.; Editorial Limusa S.A.
7. HERNÁNDEZ, R; FERNÁNDEZ, R; BAPTISTA, L Metodología de la investigación, México, 6ta edición. 2014.
8. ICG; Análisis y diseño de estructuras con SAP2000; Lima; Editorial ICG; 2016. 146pp.
9. INGENIERIA Y GESTION. Reglamento Nacional de Edificaciones; Lima; Editorial MACRO. 2016. 797pp.
10. JACK C. McCORMAC – STEPHEN F. CSERNAK; (2012); Diseño de Estructuras de Acero; USA; Quinta Edición; Editorial Alfaomega Grupo Editor S.A.; 736pp.
11. ÑAUPAS, VALDIVIA, PALACIOS Y ROMERO, Metodologia de la investigación cuantitativa – cualitativa y redacción de tesis; 5ª.Edición. Bogotá. Ediciones de la U. 2018.
12. ORIHUELA, FRANCESCO. Cálculo y diseño estructural de una nave industrial aplicando la normativa AISC en la ciudad de Juliaca, provincia de San Ramón. Universidad Peruana de Ciencias Aplicadas, para optar el título de Ingeniero Civil. 2016.
13. ROB MILES; Programación C# Yellow Book; edición 8.2 en Español, 2016. 342pp.
14. SÁNCHEZ, H Y REYES, C. Boletín de Investigación de la Universidad Ricardo Palma, Lima. 2016.
15. ZAPATA BAGLIETTO, LUIS F.; s/f; Diseño Estructural en Acero; Lima; 399pp.

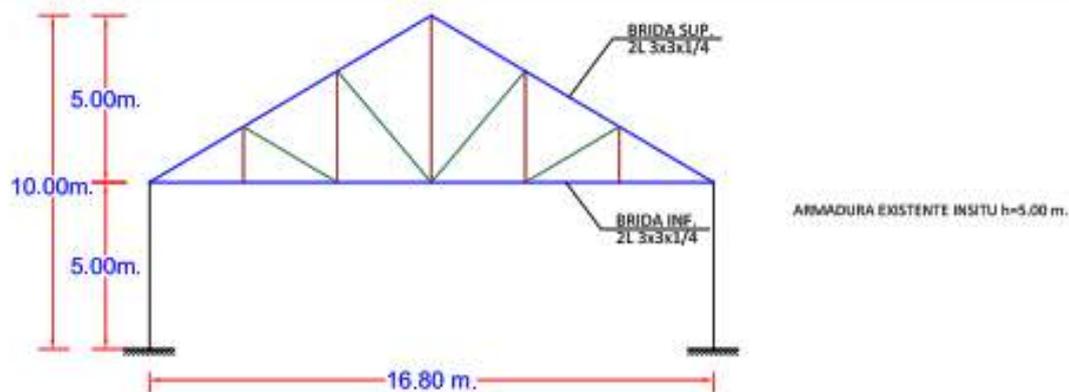
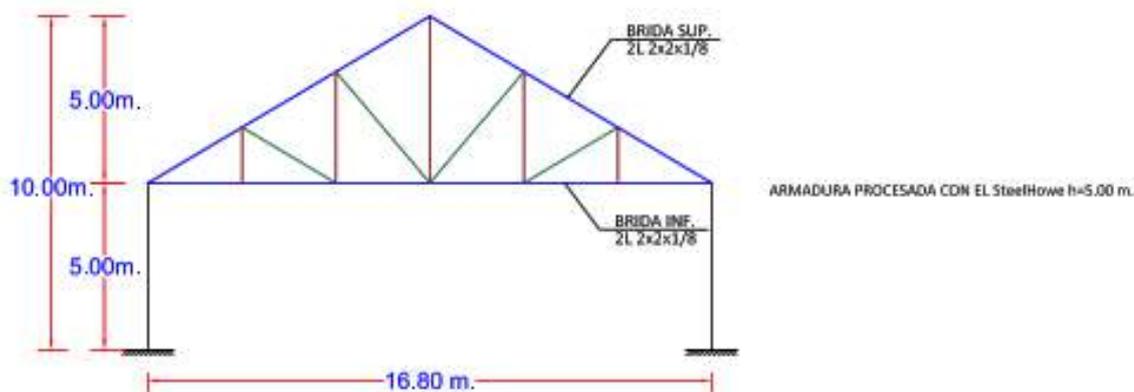
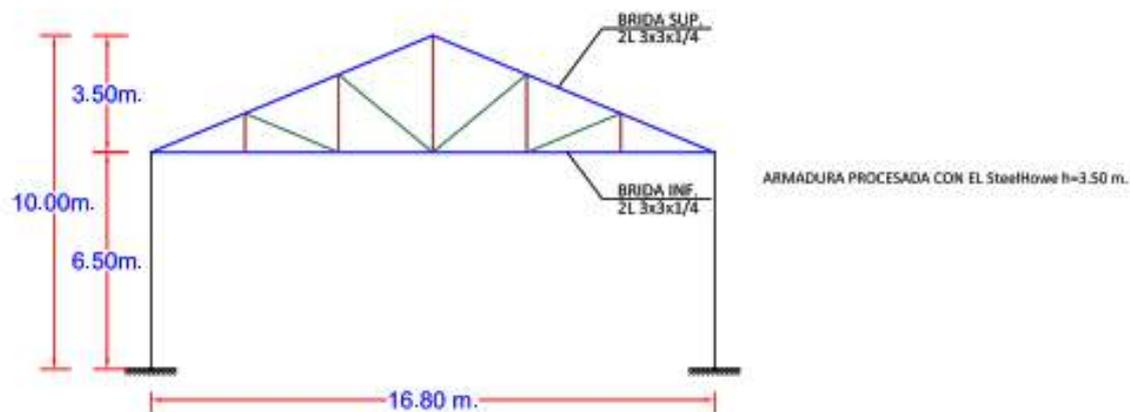
ANEXOS

10.1. MATRIZ DE CONSISTENCIA

ANÁLISIS Y DISEÑO DE TECHOS CON TIJERALES METÁLICOS TIPO HOWE MEDIANTE EL VISUAL C# Y API DEL SAP2000

PROBLEMA	OBJETIVOS	HIPOTESIS	VARIABLES	METODOLOGIA
<p>Problema general</p> <p>¿Cómo es el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000?</p> <p>10.1.1. Problemas específicos</p> <p>a) ¿Cuál es el resultado del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000?</p> <p>b) ¿Qué efectos tiene el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, en la optimización de tiempos respecto a un análisis y diseño tradicional o convencional?</p> <p>c) ¿Qué efectos tiene el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, en la minimización de procesos respecto a un análisis y diseño tradicional o convencional?</p>	<p>Objetivo general</p> <p>Conocer el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000.</p> <p>10.1.2. Objetivos Específicos</p> <p>a) Desarrollar el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000.</p> <p>b) Demostrar que el análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, tiene efectos positivos en la optimización de tiempos, respecto a un análisis y diseño tradicional o convencional.</p> <p>c) Comparar la cantidad de procesos del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, frente a los procesos de un análisis y diseño tradicional o convencional.</p>	<p>Hipótesis General</p> <p>El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es conveniente.</p> <p>Hipótesis Específicas</p> <p>a) El resultado del análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, es adecuado.</p> <p>b) El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a mejorar los tiempos necesarios para un diseño óptimo de un tijeral metálico tipo howe.</p> <p>c) El análisis y diseño de techos con tijerales metálicos tipo howe mediante el visual C# y el API del SAP2000, te ayuda a minimizar procesos que existen en un análisis y diseño tradicional o convencional.</p>	<p>Variable Independiente</p> <p>El análisis estructural de techos con tijerales metálicos</p> <p>Variable dependiente:</p> <p>Lenguaje de programación visual C# y el API del SAP2000.</p>	<p>Tipo de estudio</p> <p>Por la naturaleza de estudio el tipo Tecnológico</p> <p>Nivel de investigación</p> <p>Descriptivo</p> <p>Método</p> <p>Inductivo – Deductivo</p> <p>Diseño metodológico</p> <p>No experimental - transversal:</p> <p>Población:</p> <p>Cercado de la Ciudad de Huancayo</p> <p>Muestra:</p> <p>Distrito de Huancayo</p>

10.2. PLANO ESTRUCTURAL DEL TIJERAL METALICO TIPO HOWE.



PLANO:			
PLANO ESTRUCTURAL ARMADURA TIPO HOWE			
UBICADOR:	PROYECTISTA:	Fecha:	Escala:
Departamento:	JUNIN	Edwin LACZA ROJAS	15-06-2021
Provincia:	HUANCAYO		1:200
Ciudad:	HUANCAYO	CENTRO DE ESTUDIOS:	Nº de Plano:
Lugar:	CERCADO DE HYO.	UNIVERSIDAD PERUANA LOS ANDES	EST.01

10.3. DATA – CODIGO FUENTE DEL SOFTWARE HOWESTEEL.

CODIGO Frm Principal

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace SapHOWE
{
    public partial class Form_Principal : Form
    {
        SAP2000v16.SapObject MiSapObjeto;
        SAP2000v16.cSapModel MiSapModelo;
        // ////////////
        public int flag_CompoTijeral;//VARIABLE PARA EL MANEJO DE LOS BOTONES DE LA
        ELECCION DE LA PARTE DE LA ARMADURA (BRIDA,MONTANTE Y DIAGONAL) EN EL FORM1
        public struct ElecCompoTijeral
        {
            public int DatoElecCompoTijeral; //OBTIENE EL DATO AL PULSAR EL BOTON BRIDA,
            MONTANTE O DIAGONAL
        }
        public Form_Principal()
        {
            InitializeComponent();
            MiSapObjeto = new SAP2000v16.SapObject();
            MiSapModelo = MiSapObjeto.SapModel;
        }
        // //////////////////////////////////////
        [DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
        private extern static void ReleaseCapture();
        [DllImport("user32.DLL", EntryPoint = "SendMessage")]
        private extern static void SendMessage(System.IntPtr hwnd, int wmsg, int wparam, int
        lparam);
        // //////////////////////////////////////
    }
}

```

```

private void iconCerrar_Click(object sender, EventArgs e)
{
    Application.Exit();
}
private void iconMinimizar_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}
private void BarraTitulo_MouseDown(object sender, MouseEventArgs e)
{
    ReleaseCapture();
    SendMessage(this.Handle, 0x112, 0xf012, 0);
}
private void btn_Salir_Click(object sender, EventArgs e)
{
    Application.Exit();
}
private void btn_Brida_Click(object sender, EventArgs e)
{
    flag_CompoTijeral = 1;// BANDERA PARA DIFERENCIAR EL COMPONENTE ELEGIDO
1=BRIDA
    //PASA EL DATO DE LA ELECCION DEL TIPO DE COMPO AL FORM PERFILES MEDIANTE
UNA ESTRUTURA Y EL CONSTRUCTOR
    ElecCompoTijeral infoCompoTijeral;
    infoCompoTijeral.DatoElecCompoTijeral = 1;
    Frm_Perfiles FrmPerfiles = new Frm_Perfiles(infoCompoTijeral);// INSTANCIA EL
FORMULARIO PERFILES TANTO PARA EL DELEGADO Y PARA LA ESTRUCTURA DE DATOS
    // RECIBE EL DATO DEL FORM PERFILES PARA PASAR A LOS TEXTBOX DE LOS
COMPONENTES DEL TIJERAL CON EL METODO RECIBIRNOMBRE
    FrmPerfiles.PasadoNombPerfil += new
Frm_Perfiles.PasarNombPerfil(RecibirNombre);
    FrmPerfiles.Show();
}
private void btn_Montante_Click(object sender, EventArgs e)
{
    flag_CompoTijeral = 2;// BANDERA PARA DIFERENCIAR EL COMPONENTE ELEGIDO
2=MONTANTE
    ElecCompoTijeral infoCompoTijeral;
    infoCompoTijeral.DatoElecCompoTijeral = 2;
    Frm_Perfiles FrmPerfiles = new Frm_Perfiles(infoCompoTijeral);// INSTANCIA EL
FORMULARIO PERFILES TANTO PARA EL DELEGADO Y PARA LA ESTRUCTURA DE DATOS

```

```

        // RECIBE EL DATO DEL FORM PERFILES PARA PASAR A LOS TEXTBOX DE LOS
        COMPONENTES DEL TIJERAL CON EL METODO RECIBIRNOMBRE
        FrmPerfiles.PasadoNombPerfil += new
        Frm_Perfiles.PasarNombPerfil(RecibirNombre);
        FrmPerfiles.Show();
    }

    private void btn_Diagonal_Click(object sender, EventArgs e)
    {
        flag_CompoTijeral = 3;// BANDERA PARA DIFERENCIAR EL COMPONENTE ELEGIDO
        3=DIAGONAL
        ElecCompoTijeral infoCompoTijeral;
        infoCompoTijeral.DatoElecCompoTijeral = 3;
        Frm_Perfiles FrmPerfiles = new Frm_Perfiles(infoCompoTijeral);// INSTANCIA EL
        FORMULARIO PERFILES TANTO PARA EL DELEGADO Y PARA LA ESTRUCTURA DE DATOS
        // RECIBE EL DATO DEL FORM PERFILES PARA PASAR A LOS TEXTBOX DE LOS
        COMPONENTES DEL TIJERAL CON EL METODO RECIBIRNOMBRE
        FrmPerfiles.PasadoNombPerfil += new
        Frm_Perfiles.PasarNombPerfil(RecibirNombre);
        FrmPerfiles.Show();
    }

    //METODO QUE RECIBE EL DATO DEL PERFIL SELECCIONADO DEL FORM PERFIL POR
    MEDIO DE DELEGADOS Y EVENTOS
    public void RecibirNombre(string NomPerfil)
    {
        switch (flag_CompoTijeral)
        {
            case 1:
            {
                text_FinBrida.Text = NomPerfil;
            }
            break;
            case 2:
            {
                text_FinMontante.Text = NomPerfil;
            }
            break;
            case 3:
            {
                text_FinDiagonal.Text = NomPerfil;
            }
        }
    }

```

```

        break;
    }
}
private void btn_AbrirSap_Click(object sender, EventArgs e)
{
    Cls_TijeralHowe Abrir = new Cls_TijeralHowe();
    Abrir.AbrirSAP(MiSapObjeto, MiSapModelo);
}
private void btn_CerrarSap_Click(object sender, EventArgs e)
{
    Cls_TijeralHowe Cerrar = new Cls_TijeralHowe();
    Cerrar.CerrarSAP(MiSapObjeto, MiSapModelo);
}
double NumSegmentos;
private void btn_AnalisisSap_Click(object sender, EventArgs e)
{
    double LongTijeral, Flecha, SepEntreTijeral;
    string NombSeccBrida, NombSeccMontante, NombSeccDiagonal;
    double TotalCM, TotalCV, TotalCLG, TotalBarloW1, TotalBarloW2, TotalSotaW1,
TotalSotaW2;

    LongTijeral = Convert.ToDouble(text_FinLongitud.Text);
    NumSegmentos = Convert.ToDouble(text_FinNumSegmen.Text);
    Flecha = Convert.ToDouble(text_FinFlecha.Text);
    SepEntreTijeral = Convert.ToDouble(text_FinSepTijeral.Text);
    NombSeccBrida = text_FinBrida.Text;
    NombSeccMontante = text_FinMontante.Text;
    NombSeccDiagonal = text_FinDiagonal.Text;
    TotalCM = Convert.ToDouble(text_FinCM.Text);
    TotalCV = Convert.ToDouble(text_FinCV.Text);
    TotalCLG = Convert.ToDouble(text_FinCLG.Text);
    TotalBarloW1 = Convert.ToDouble(text_FinBarloventoW1.Text);
    TotalBarloW2 = Convert.ToDouble(text_FinBarloventoW2.Text);
    TotalSotaW1 = Convert.ToDouble(text_FinSotaventoW1.Text);
    TotalSotaW2 = Convert.ToDouble(text_FinSotaventoW2.Text);

    Cls_TijeralHowe AnalizarSAP = new Cls_TijeralHowe();
    AnalizarSAP.AgregarNuevoMaterial(MiSapObjeto, MiSapModelo);
    AnalizarSAP.CrearTipoCargasYComb(MiSapObjeto, MiSapModelo);
    AnalizarSAP.CrearSecciones(MiSapObjeto, MiSapModelo);
}

```

```

        AnalizarSAP.DibujarObjFrame(MiSapObjeto, MiSapModelo, LongTijeral, Flecha,
        NumSegmentos,NombSeccBrida,NombSeccMontante,NombSeccDiagonal);
        AnalizarSAP.AsignaCargas(MiSapObjeto,
        MiSapModelo,NumSegmentos,TotalCM,TotalCV,TotalCLG,TotalBarloW1
        ,TotalBarloW2,TotalSotaW1,TotalSotaW2,SepEntreTijeral);
        AnalizarSAP.AnalizarSinMomentos(MiSapObjeto, MiSapModelo);
        AnalizarSAP.GuardarYAnalizar(MiSapObjeto, MiSapModelo);
        ResultadosAnalisis());
    }

```

```

public void ResultadosAnalisis()
{
    long ret;
    int NumberResults;

    string[] Obj;
    double[] ObjSta;
    string[] Elm;
    double[] ElmSta;
    string[] LoadCase;
    string[] StepType;
    double[] StepNum;
    double[] P;
    double[] V2;
    double[] V3;
    double[] T;
    double[] M2;
    double[] M3;

    int CantFrame;
    double[] EsfuerzoPresion;
    double[] EsfuerzoSuccion;
    int[] Item;
    string[] NombFrame;

    CantFrame = MiSapModelo.FrameObj.Count();
    EsfuerzoPresion = new double[CantFrame];
    EsfuerzoSuccion = new double[CantFrame];
    Item = new int [CantFrame];
    NombFrame = new string[CantFrame];

```

```

NumberResults = 0;
Obj = new string[1];
ObjSta = new double[1];
Elm = new string[1];
ElmSta = new double[1];
LoadCase = new string[1];
StepType = new string[20];
StepNum = new double[1];
P = new double[20];
V2 = new double[1];
V3 = new double[1];
M2 = new double[1];
M3 = new double[1];
T = new double[1];
ret = MiSapModelo.Results.Setup.DeselectAllCasesAndCombosForOutput();
for (int i = 1; i <= CantFrame; i++)
{
    ret = MiSapModelo.Results.Setup.DeselectAllCasesAndCombosForOutput();
    ret = MiSapModelo.Results.Setup.SetComboSelectedForOutput("ENVOLVENTE",
Convert.ToBoolean(-1));
    ret = MiSapModelo.Results.FrameForce(Convert.ToString(i),
SAP2000v16.eltemTypeElm.ObjectElm, NumberResults, ref Obj, ref ObjSta, ref Elm, ref
ElmSta, ref LoadCase, ref StepType, ref StepNum, ref P, ref V2, ref V3, ref T, ref M2, ref M3);
    P.CopyTo(P,0);
    StepType.CopyTo(StepType, 0);
    Elm.CopyTo(Elm, 0);
    EsfuerzoPresion[i-1] = P[0];
    Item[i-1] = i;
    NombFrame[i - 1] = Elm[0];
    for (int j=0; j < 20; j++)
    {
        if(StepType[j]=="Min")
        {
            EsfuerzoSuccion[i - 1] = P[j];
            j = 20;
        }
    }
}

int n;

```

```

for (int i = 0; i < CantFrame; i++)
{
    n = dataGridView_Resultados.Rows.Add();
    dataGridView_Resultados.Rows[n].Cells[0].Value = Item[i];
    dataGridView_Resultados.Rows[n].Cells[1].Value = NombFrame[i];
    dataGridView_Resultados.Rows[n].Cells[2].Value
=Math.Round(EsfuerzoPresion[i],2);
    dataGridView_Resultados.Rows[n].Cells[3].Value = Math.Round(EsfuerzoSuccion[i],
2);

}
double TraccCriticaBrida=0,CompCriticaBrida=0;
string NombTraccBrida="",NombCompBrida="";
double TraccCriticaMontante = 0, CompCriticaMontante = 0;
string NombTraccMontante = "", NombCompMontante = "";
double TraccCriticaDiagonal = 0, CompCriticaDiagonal = 0;
string NombTraccDiagonal = "", NombCompDiagonal = "";
//DATOS ESFUERZOS ULTIMOS BRIDA
for (int i = 0; i < NumSegmentos*2;i++)
{
    if (TraccCriticaBrida<EsfuerzoPresion[i])
    {
        TraccCriticaBrida = EsfuerzoPresion[i];
        NombTraccBrida = NombFrame[i];
    }
    if (CompCriticaBrida >EsfuerzoSuccion [i])
    {
        CompCriticaBrida = EsfuerzoSuccion[i];
        NombCompBrida = NombFrame[i];
    }
}
text_TraccBrida.Text = Convert.ToString(Math.Round(TraccCriticaBrida,2));
text_NombTraccBrida.Text = NombTraccBrida;
text_CompBrida.Text = Convert.ToString(Math.Round(CompCriticaBrida, 2));
text_NombCompBrida.Text = NombCompBrida;

//DATOS ESFUERZOS ULTIMOS MONTANTE
for (int i = Convert.ToInt32(NumSegmentos*2); i <
(NumSegmentos*2)+(NumSegmentos-1); i++)
{

```

```

        if (TraccCriticaMontante < EsfuerzoPresion[i])
        {
            TraccCriticaMontante = EsfuerzoPresion[i];
            NombTraccMontante = NombFrame[i];
        }
        if (CompCriticaMontante > EsfuerzoSuccion[i])
        {
            CompCriticaMontante = EsfuerzoSuccion[i];
            NombCompMontante = NombFrame[i];
        }
    }
    text_TraccMontante.Text = Convert.ToString(Math.Round(TraccCriticaMontante, 2));
    text_NombTraccMontante.Text = NombTraccMontante;
    text_CompMontante.Text = Convert.ToString(Math.Round(CompCriticaMontante,
2));
    text_NombCompMontante.Text = NombCompMontante;
    //DATOS ESFUERZOS ULTIMOS DIAGONAL
    for (int i = Convert.ToInt32((NumSegmentos * 2)+(NumSegmentos-1)); i < CantFrame;
i++)
    {
        if (TraccCriticaDiagonal < EsfuerzoPresion[i])
        {
            TraccCriticaDiagonal = EsfuerzoPresion[i];
            NombTraccDiagonal = NombFrame[i];
        }
        if (CompCriticaDiagonal > EsfuerzoSuccion[i])
        {
            CompCriticaDiagonal = EsfuerzoSuccion[i];
            NombCompDiagonal = NombFrame[i];
        }
    }
    text_TraccDiagonal.Text = Convert.ToString(Math.Round(TraccCriticaDiagonal, 2));
    text_NombTraccDiagonal.Text = NombTraccDiagonal;
    text_CompDiagonal.Text = Convert.ToString(Math.Round(CompCriticaDiagonal, 2));
    text_NombCompDiagonal.Text = NombCompDiagonal;
}

private void btn_OkDatosTijeral_Click(object sender, EventArgs e)
{
    text_FinLongitud.Text = text_Longitud.Text;
    text_FinFlecha.Text = text_Flecha.Text;
}

```

```
text_FinNumSegmen.Text = text_NumSegmentos.Text;
text_FinSepTijeral.Text = text_SeparaTijerales.Text;
}

private void btn_OkRegionEolico_Click(object sender, EventArgs e)
{
    comboBox_VelMapaEolico.Items.Clear();
    switch (comboBox_MapasEolico.Text)
    {
        case ("Amazonas"):
            {
                comboBox_VelMapaEolico.Items.Add("75");
            }
            break;
        case ("Ancash"):
            {
                comboBox_VelMapaEolico.Items.Add("70");
                comboBox_VelMapaEolico.Items.Add("75");
                comboBox_VelMapaEolico.Items.Add("85");
            }
            break;
        case ("Apurímac"):
            {
                comboBox_VelMapaEolico.Items.Add("60");
                comboBox_VelMapaEolico.Items.Add("65");
            }
            break;
        case ("Arequipa"):
            {
                comboBox_VelMapaEolico.Items.Add("65");
                comboBox_VelMapaEolico.Items.Add("80");
                comboBox_VelMapaEolico.Items.Add("85");
                comboBox_VelMapaEolico.Items.Add("90");
            }
            break;
        case ("Ayacucho"):
            {
                comboBox_VelMapaEolico.Items.Add("35");
                comboBox_VelMapaEolico.Items.Add("45");
                comboBox_VelMapaEolico.Items.Add("65");
            }
    }
}
```

```
    break;
case ("Cajamarca"):
    {
        comboBox_VelMapaEolico.Items.Add("40");
        comboBox_VelMapaEolico.Items.Add("45");
    }
    break;
case ("Callao"):
    {
        comboBox_VelMapaEolico.Items.Add("45");
    }
    break;
case ("Cuzco"):
    {
        comboBox_VelMapaEolico.Items.Add("30");
        comboBox_VelMapaEolico.Items.Add("35");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("80");
        comboBox_VelMapaEolico.Items.Add("90");
        comboBox_VelMapaEolico.Items.Add("110");
    }
    break;
case ("Huancavelica"):
    {
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("70");
        comboBox_VelMapaEolico.Items.Add("85");
    }
    break;
case ("Huánuco"):
    {
        comboBox_VelMapaEolico.Items.Add("55");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("85");
        comboBox_VelMapaEolico.Items.Add("90");
        comboBox_VelMapaEolico.Items.Add("100");
    }
    break;
case ("Ica"):
    {
        comboBox_VelMapaEolico.Items.Add("35");
```

```
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("65");
    }
    break;
case ("Junín"):
    {
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("70");
        comboBox_VelMapaEolico.Items.Add("85");
        comboBox_VelMapaEolico.Items.Add("100");
    }
    break;
case ("La Libertad"):
    {
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("95");
    }
    break;
case ("Lambayeque"):
    {
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("85");
    }
    break;
case ("Lima"):
    {
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("85");
        comboBox_VelMapaEolico.Items.Add("100");
    }
    break;
case ("Loreto"):
    {
        comboBox_VelMapaEolico.Items.Add("60");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("95");
    }
    break;
case ("Madre de Dios"):
```

```
{
    comboBox_VelMapaEolico.Items.Add("55");
    comboBox_VelMapaEolico.Items.Add("90");
    comboBox_VelMapaEolico.Items.Add("110");
}
break;
case ("Moquegua"):
{
    comboBox_VelMapaEolico.Items.Add("45");
    comboBox_VelMapaEolico.Items.Add("50");
    comboBox_VelMapaEolico.Items.Add("80");
}
break;
case ("Pasco"):
{
    comboBox_VelMapaEolico.Items.Add("65");
    comboBox_VelMapaEolico.Items.Add("85");
    comboBox_VelMapaEolico.Items.Add("100");
}
break;
case ("Piura"):
{
    comboBox_VelMapaEolico.Items.Add("35");
    comboBox_VelMapaEolico.Items.Add("40");
    comboBox_VelMapaEolico.Items.Add("60");
    comboBox_VelMapaEolico.Items.Add("80");
    comboBox_VelMapaEolico.Items.Add("120");
}
break;
case ("Puno"):
{
    comboBox_VelMapaEolico.Items.Add("35");
    comboBox_VelMapaEolico.Items.Add("70");
    comboBox_VelMapaEolico.Items.Add("80");
    comboBox_VelMapaEolico.Items.Add("85");
    comboBox_VelMapaEolico.Items.Add("90");
    comboBox_VelMapaEolico.Items.Add("110");
    comboBox_VelMapaEolico.Items.Add("130");
}
break;
case ("San Martín"):
```

```

    {
        comboBox_VelMapaEolico.Items.Add("35");
        comboBox_VelMapaEolico.Items.Add("55");
        comboBox_VelMapaEolico.Items.Add("60");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("75");
    }
    break;
case ("Tacna"):
    {
        comboBox_VelMapaEolico.Items.Add("30");
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("50");
        comboBox_VelMapaEolico.Items.Add("80");
    }
    break;
case ("Tumbes"):
    {
        comboBox_VelMapaEolico.Items.Add("35");
    }
    break;
case ("Ucayali"):
    {
        comboBox_VelMapaEolico.Items.Add("35");
        comboBox_VelMapaEolico.Items.Add("45");
        comboBox_VelMapaEolico.Items.Add("65");
        comboBox_VelMapaEolico.Items.Add("75");
    }
    break;
}
}

private void btn_OkCargas_Click(object sender, EventArgs e)
{
    double Long_Viga, Altura_Viga;
    double AnguloInclinacion;
    double CM_PesoVigueta, CM_PesoMatCobertura,
    CM_PesoLuminaria,CM_PesoContraIncen,CM_Extras, CM_Total;
    double CV_Total;
    double CLG_PesoPorM3, CLG_AlturaCapaGranizo, CLG_Total;

```

```

    double CW_VelMapaEolico, CW_AlturaDesdeTerreno, CW_VelDiseño,
    CW_FactForma_BarW1, CW_FactForma_BarW2, CW_FactForma_SotaW1W2
    ,CW_TotalBarW1,CW_TotalBarW2,CW_TotalSotaW1,CW_TotalSotaW2;
    ////////////CM//////////

    try
    {
        CM_PesoVigueta = Convert.ToDouble(text_CMPesoViguetas.Text);
        CM_PesoMatCobertura = Convert.ToDouble(text_CMPesoMatCober.Text);
        CM_PesoLuminaria = Convert.ToDouble(text_CMPesoLuminaria.Text);
        CM_PesoContraIncen = Convert.ToDouble(text_CMPesoContraIncen.Text);
        CM_Extras = Convert.ToDouble(text_CMPesoExtras.Text);
    }
    catch(Exception ex)
    {
        MessageBox.Show("DATOS INVALIDOS!!! ...ASEGURATE DE LLENAR BIEN LOS
    DATOS. "+ex.Message);
        CM_PesoVigueta = 0;
        CM_PesoMatCobertura = 0;
        CM_PesoLuminaria = 0;
        CM_PesoContraIncen = 0;
        CM_Extras = 0;
    }

    CM_Total = (CM_PesoVigueta + CM_PesoMatCobertura+
    CM_PesoLuminaria+CM_PesoContraIncen+CM_Extras);
    text_FinCM.Text = Convert.ToString(CM_Total);
    ////////////CV//////////

    try
    {
        Long_Viga = Convert.ToDouble(text_FinLongitud.Text);
        Altura_Viga = Convert.ToDouble(text_FinFlecha.Text);
    }
    catch(Exception ex)
    {
        MessageBox.Show("DATOS INVALIDOS!!! ...ASEGURATE DE LLENAR BIEN LOS
    DATOS. "+ex.Message);
        Long_Viga = 0;
        Altura_Viga = 0;
    }

```

```

AnguloInclinacion = Math.Atan(Altura_Viga / (Long_Viga / 2)) * (180 / Math.PI);
if (AnguloInclinacion <= 3)
{
    CV_Total = 100;
}
else
{
    CV_Total = (100 - (5 * (AnguloInclinacion - 3)));
    if (CV_Total < 50)
    {
        CV_Total = 50;
    }
}
text_FinCV.Text = Convert.ToString(Math.Round(CV_Total, 2));
//////////LLUVIA O GRANIZO//////////
try
{
    CLG_PesoPorM3 = Convert.ToDouble(text_CLGPesoGranizo.Text);
    CLG_AlturaCapaGranizo = Convert.ToDouble(text_CLGAlturaCapaGranizo.Text);
}
catch(Exception ex)
{
    MessageBox.Show("DATOS INVALIDOS!!! ...ASEGURATE DE LLENAR BIEN LOS
DATOS. "+ex.Message);
    CLG_PesoPorM3 = 0;
    CLG_AlturaCapaGranizo = 0;
}

CLG_Total = CLG_PesoPorM3 * CLG_AlturaCapaGranizo;
text_FinCLG.Text = Convert.ToString(CLG_Total);
//////////CARGA VIENTO//////////
if (checkBox_CargaCierre.Checked == true)
{
    CW_FactForma_BarW1 = 1;
    CW_FactForma_BarW2 = -0.6;
    CW_FactForma_SotaW1W2 = -0.9;
}
else
{
    CW_FactForma_BarW1 = 0.7;
    CW_FactForma_BarW2 = -0.3;
}

```

```

    CW_FactForma_SotaW1W2 = -0.6;
}
try
{
    CW_AlturaDesdeTerreno = Convert.ToDouble(text_AlturaDesdeTerreno.Text);
    CW_VelMapaEolico = Convert.ToDouble(comboBox_VelMapaEolico.Text);
}
catch(Exception ex)
{
    MessageBox.Show("DATOS INVALIDOS!!! ...ASEGURATE DE LLENAR BIEN LOS
DATOS "+ex.Message);
    CW_AlturaDesdeTerreno = 0;
    CW_VelMapaEolico = 0;
}

if (CW_AlturaDesdeTerreno <= 10)
{
    CW_VelDiseño = CW_VelMapaEolico;
}
else
{
    CW_VelDiseño = CW_VelMapaEolico * Math.Pow((CW_AlturaDesdeTerreno / 10),
0.22);
    if (CW_VelDiseño < 75)
    {
        CW_VelDiseño = 75;
    }
}
    CW_TotalBarW1 = Math.Round((0.005 * CW_FactForma_BarW1 *
Math.Pow(CW_VelDiseño, 2)),2);
    CW_TotalBarW2 = Math.Round((0.005 * CW_FactForma_BarW2 *
Math.Pow(CW_VelDiseño, 2)),2);
    CW_TotalSotaW1 = Math.Round((0.005 * CW_FactForma_SotaW1W2 *
Math.Pow(CW_VelDiseño, 2)),2);
    CW_TotalSotaW2 = Math.Round((0.005 * CW_FactForma_SotaW1W2 *
Math.Pow(CW_VelDiseño, 2)),2);
    text_FinBarloventoW1.Text = Convert.ToString(CW_TotalBarW1);
    text_FinBarloventoW2.Text = Convert.ToString(CW_TotalBarW2);
    text_FinSotaventoW1.Text = Convert.ToString(CW_TotalSotaW1);
    text_FinSotaventoW2.Text = Convert.ToString(CW_TotalSotaW2);
}

```

```

    }
}

```

CODIGO Frm Perfiles.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;

namespace SapHOWE
{
    public partial class Frm_Perfiles : Form
    {
        public int flag_CompotijeralFP;// VARIABLE QUE RECIBE LA INF DEL TIPO DE
        COMPONENTE ELEJIDO EN EL FORM PERFILES (BRIDA, MONTANTE O DIAGONAL)

        //DECLARACION DE DELEGADO Y EVENTO PARA PaSAR INF DEL TIPO DE PERFIL DE CADA
        COMPONENTE AL FORM PRINCIPAL
        public delegate void PasarNombPerfil(string NombPerfil);
        public event PasarNombPerfil PasadoNombPerfil;
        ///////////////////////////////////////////////////////////////////

        //DECLARACION DE VARIABLES PARA EL MANEJO DE ARCHIVOS PARA EL COMBOBOX
        string linea;
        string[] palabras = new string[1];
        char[] separador = { ',' };
        StreamReader flujoEntrada;
        ///////////////////////////////////////////////////////////////////
        public Frm_Perfiles(Form_Principal.ElecCompotijeral info)
        {
            InitializeComponent();
            flag_CompotijeralFP = info.DatoElecCompotijeral;
            switch (flag_CompotijeralFP)
            {

```

```

    case 1:
    {
        comboBoxBrida.Enabled = true;
    }
    break;
    case 2:
    {
        comboBoxMontante.Enabled = true;
    }
    break;
    case 3:
    {
        comboBoxDiagonal.Enabled = true;
    }
    break;
}
}
// //////////////////////////////////////
[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
private extern static void ReleaseCapture();
[DllImport("user32.DLL", EntryPoint = "SendMessage")]
private extern static void SendMessage(System.IntPtr hwnd, int wmsg, int wparam, int
lparam);
// //////////////////////////////////////
private void iconCerrar_Click(object sender, EventArgs e)
{
    this.Close();
}
private void iconMinimizar_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}
private void btn_PerfilH_Click(object sender, EventArgs e)
{
    flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_H.txt");
    CargarDatosPerfComboBox();
    flujoEntrada.Close();
}
private void btn_PerfilU_Click(object sender, EventArgs e)
{

```

```

        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_U.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }
    private void btn_PerfilT_Click(object sender, EventArgs e)
    {
        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_T.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }
    private void btn_PerfilAngulo_Click(object sender, EventArgs e)
    {
        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_L.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }
    private void btn_Perfil2L_Click(object sender, EventArgs e)
    {
        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_2L.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }
    private void btn_PerfilRLiso_Click(object sender, EventArgs e)
    {
        flujoEntrada =
File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_RLiso.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }
    private void btn_PerfilCir_Click(object sender, EventArgs e)
    {
        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_Cir.txt");
        CargarDatosPerfComboBox();
        flujoEntrada.Close();
    }

    private void btn_PerfilRec_Click(object sender, EventArgs e)
    {
        flujoEntrada =
File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_Rec.txt");

```

```
CargarDatosPerfComboBox();
flujoEntrada.Close();
}
//METODO PARA PASAR LA LISTA DEL ARCHIVO CREADO A LOS COMBOBOX
private void CargarDatosPerfComboBox()
{
    comboBoxBrida.Items.Clear();
    comboBoxMontante.Items.Clear();
    comboBoxDiagonal.Items.Clear();
    switch (flag_CompoTijeraFP)
    {
        case 1:
            {
                linea = flujoEntrada.ReadLine();
                while (linea != null)
                {
                    palabras = linea.Split(separador);
                    comboBoxBrida.Items.Add(palabras[0]);
                    linea = flujoEntrada.ReadLine();
                }
            }
            break;
        case 2:
            {
                linea = flujoEntrada.ReadLine();
                while (linea != null)
                {
                    palabras = linea.Split(separador);
                    comboBoxMontante.Items.Add(palabras[0]);
                    linea = flujoEntrada.ReadLine();
                }
            }
            break;
        case 3:
            {
                linea = flujoEntrada.ReadLine();
                while (linea != null)
                {
                    palabras = linea.Split(separador);
                    comboBoxDiagonal.Items.Add(palabras[0]);
                    linea = flujoEntrada.ReadLine();
                }
            }
    }
}
```



```

namespace SapHOWE
{
    class Cls_TijeralHowe
    {
        ///VARIABLES GLOBALES PARA LEER LOS NOMBRES DE LOS FRAMES DE LA BRIDA
        SUPERIOR
        string[] Nombre_Frame_BridalZq;
        string[] Nombre_Frame_BridaDere;
        double Li;
        public Cls_TijeralHowe()
        {

        }

        public void AbrirSAP(SAP2000v16.SapObject MiSapObjeto,SAP2000v16.cSapModel
        MiSapModelo)
        {
            long ret;
            MiSapObjeto.ApplicationStart(SAP2000v16.eUnits.kgf_m_C,true,"");
            ret = MiSapModelo.InitializeNewModel(SAP2000v16.eUnits.kgf_m_C);
            ret = MiSapModelo.File.NewBlank();
        }

        public void CerrarSAP(SAP2000v16.SapObject MiSapObjeto, SAP2000v16.cSapModel
        MiSapModelo)
        {
            MiSapObjeto.ApplicationExit(false);
            MiSapModelo = null;
            MiSapObjeto = null;
        }

        public void AgregarNuevoMaterial(SAP2000v16.SapObject MiSapObjeto,
        SAP2000v16.cSapModel MiSapModelo)
        {
            long ret;
            ///AGREGA NUEVO MATERIAL "A36" PARA LOS DIFERENTES PERFILES EXCEPTO
            REDONDO Y RECTANGULAR
            ret = MiSapModelo.PropMaterial.AddMaterial("A36",
            SAP2000v16.eMatType.MATERIAL_STEEL, "United States", "ASTM A36", "Grade 36");
            ///MODIFICA LOS DATOS DE LA PROPIEDAD DEL MATERIAL
            ret = MiSapModelo.PropMaterial.SetMPIsotropic("A36", 2.100E+10, 0.3, 1.170E-05);
            ///MODIFICA EL PESO DEL MATERIAL
            ret = MiSapModelo.PropMaterial.SetWeightAndMass("A36", 1, 7850);
        }
    }
}

```

```

//MODIFICA OTROS CARACTERISTICAS DEL MATERIAL STEEL
ret = MiSapModelo.PropMaterial.SetOSteel_1("A36", 25300000, 40800000,
37965760, 44855842, 1, 1, 0.02, 0.14, 0.2, -0.1);

//AGREGA NUEVO MATERIAL "A500GrB42" PARA LOS PERFILES REDONDO
ret = MiSapModelo.PropMaterial.AddMaterial("A500GrB42",
SAP2000v16.eMatType.MATERIAL_STEEL, "United States", "ASTM A500", "Grade B, Fy 42
(HSS Round)");
//MODIFICA LOS DATOS DE LA PROPIEDAD DEL MATERIAL
ret = MiSapModelo.PropMaterial.SetMPIsotropic("A500GrB42", 2.100E+10, 0.3,
1.170E-05);
//MODIFICA EL PESO DEL MATERIAL
ret = MiSapModelo.PropMaterial.SetWeightAndMass("A500GrB42", 1, 7850);
//MODIFICA OTROS CARACTERISTICAS DEL MATERIAL STEEL
ret = MiSapModelo.PropMaterial.SetOSteel_1("A500GrB42", 29570715, 40778038,
32481817, 44855842, 1, 1, 0.02, 0.14, 0.2, -0.1);

//AGREGA NUEVO MATERIAL "A500GrB46" PARA LOS PERFILES RECTANGULARES
ret = MiSapModelo.PropMaterial.AddMaterial("A500GrB46",
SAP2000v16.eMatType.MATERIAL_STEEL, "United States", "ASTM A500", "Grade B, Fy 46
(HSS Rect.)");
//MODIFICA LOS DATOS DE LA PROPIEDAD DEL MATERIAL
ret = MiSapModelo.PropMaterial.SetMPIsotropic("A500GrB46", 2.100E+10, 0.3,
1.170E-05);
//MODIFICA EL PESO DEL MATERIAL
ret = MiSapModelo.PropMaterial.SetWeightAndMass("A500GrB46", 1, 7850);
//MODIFICA OTROS CARACTERISTICAS DEL MATERIAL STEEL
ret = MiSapModelo.PropMaterial.SetOSteel_1("A500GrB46", 32119914, 40778038,
35575323, 44855842, 1, 1, 0.018, 0.12, 0.18, -0.1);
}
public void CrearTipoCargasYComb(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo)
{
    long ret;
    bool temp = true;
    ret = MiSapModelo.LoadPatterns.Add("D",
SAP2000v16.eLoadPatternType.LTYPE_DEAD, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("L",
SAP2000v16.eLoadPatternType.LTYPE_LIVE, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("Lr",
SAP2000v16.eLoadPatternType.LTYPE_LIVE, 0, temp);

```

```

    ret = MiSapModelo.LoadPatterns.Add("W1",
SAP2000v16.eLoadPatternType.LTYPE_WIND, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("W2",
SAP2000v16.eLoadPatternType.LTYPE_WIND, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("S",
SAP2000v16.eLoadPatternType.LTYPE_SNOW, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("E",
SAP2000v16.eLoadPatternType.LTYPE_QUAKE, 0, temp);
    ret = MiSapModelo.LoadPatterns.Add("R",
SAP2000v16.eLoadPatternType.LTYPE_OTHER, 0, temp);

//CREA LAS COMBINACIONES NECESARIAS SEGUN LA LRFD Y LA NORMA E090
//(1) 1.4D
ret = MiSapModelo.RespCombo.Add("1.4D", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.4D", SAP2000v16.eCType.LoadCase,
"D", 1.4);
//(2) 1.2D+1.6L+0.5Lr
ret = MiSapModelo.RespCombo.Add("1.2D+1.6L+0.5Lr", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5Lr",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5Lr",
SAP2000v16.eCType.LoadCase, "L", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5Lr",
SAP2000v16.eCType.LoadCase, "Lr", 0.5);
//(3) 1.2D+1.6L+0.5S
ret = MiSapModelo.RespCombo.Add("1.2D+1.6L+0.5S", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5S",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5S",
SAP2000v16.eCType.LoadCase, "L", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5S",
SAP2000v16.eCType.LoadCase, "S", 0.5);
//(4) 1.2D+1.6L+0.5R
ret = MiSapModelo.RespCombo.Add("1.2D+1.6L+0.5R", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5R",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5R",
SAP2000v16.eCType.LoadCase, "L", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6L+0.5R",
SAP2000v16.eCType.LoadCase, "R", 0.5);
//(5) 1.2D+1.6Lr+0.5L

```

```

ret = MiSapModelo.RespCombo.Add("1.2D+1.6Lr+0.5L", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.5L",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.5L",
SAP2000v16.eCType.LoadCase, "Lr", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.5L",
SAP2000v16.eCType.LoadCase, "L", 0.5);
//(6) 1.2D+1.6Lr+0.8W1
ret = MiSapModelo.RespCombo.Add("1.2D+1.6Lr+0.8W1", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W1",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W1",
SAP2000v16.eCType.LoadCase, "Lr", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W1",
SAP2000v16.eCType.LoadCase, "W1", 0.8);
//(7) 1.2D+1.6Lr+0.8W2
ret = MiSapModelo.RespCombo.Add("1.2D+1.6Lr+0.8W2", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W2",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W2",
SAP2000v16.eCType.LoadCase, "Lr", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6Lr+0.8W2",
SAP2000v16.eCType.LoadCase, "W2", 0.8);
//(8) 1.2D+1.6S+0.5L
ret = MiSapModelo.RespCombo.Add("1.2D+1.6S+0.5L", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.5L",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.5L",
SAP2000v16.eCType.LoadCase, "S", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.5L",
SAP2000v16.eCType.LoadCase, "L", 0.5);
//(9) 1.2D+1.6S+0.8W1
ret = MiSapModelo.RespCombo.Add("1.2D+1.6S+0.8W1", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W1",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W1",
SAP2000v16.eCType.LoadCase, "S", 1.6);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W1",
SAP2000v16.eCType.LoadCase, "W1", 0.8);
//(10) 1.2D+1.6S+0.8W2
ret = MiSapModelo.RespCombo.Add("1.2D+1.6S+0.8W2", 0);

```

```

    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W2",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W2",
SAP2000v16.eCType.LoadCase, "S", 1.6);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6S+0.8W2",
SAP2000v16.eCType.LoadCase, "W2", 0.8);
    //(11) 1.2D+1.6R+0.5L
    ret = MiSapModelo.RespCombo.Add("1.2D+1.6R+0.5L", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.5L",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.5L",
SAP2000v16.eCType.LoadCase, "R", 1.6);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.5L",
SAP2000v16.eCType.LoadCase, "L", 0.5);
    //(12) 1.2D+1.6R+0.8W1
    ret = MiSapModelo.RespCombo.Add("1.2D+1.6R+0.8W1", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W1",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W1",
SAP2000v16.eCType.LoadCase, "R", 1.6);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W1",
SAP2000v16.eCType.LoadCase, "W1", 0.8);
    //(13) 1.2D+1.6R+0.8W2
    ret = MiSapModelo.RespCombo.Add("1.2D+1.6R+0.8W2", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W2",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W2",
SAP2000v16.eCType.LoadCase, "R", 1.6);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.6R+0.8W2",
SAP2000v16.eCType.LoadCase, "W2", 0.8);
    //(14) 1.2D+1.3W1+0.5L+0.5Lr
    ret = MiSapModelo.RespCombo.Add("1.2D+1.3W1+0.5L+0.5Lr", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "W1", 1.3);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "L", 0.5);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "Lr", 0.5);
    //(15) 1.2D+1.3W2+0.5L+0.5Lr

```

```

ret = MiSapModelo.RespCombo.Add("1.2D+1.3W2+0.5L+0.5Lr", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "W2", 1.3);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "L", 0.5);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5Lr",
SAP2000v16.eCType.LoadCase, "Lr", 0.5);
//(16) 1.2D+1.3W1+0.5L+0.5S
ret = MiSapModelo.RespCombo.Add("1.2D+1.3W1+0.5L+0.5S", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "W1", 1.3);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "L", 0.5);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "S", 0.5);
//(17) 1.2D+1.3W2+0.5L+0.5S
ret = MiSapModelo.RespCombo.Add("1.2D+1.3W2+0.5L+0.5S", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5Sr",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "W2", 1.3);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "L", 0.5);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5S",
SAP2000v16.eCType.LoadCase, "S", 0.5);
//(18) 1.2D+1.3W1+0.5L+0.5R
ret = MiSapModelo.RespCombo.Add("1.2D+1.3W1+0.5L+0.5R", 0);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "D", 1.2);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "W1", 1.3);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "L", 0.5);
ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W1+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "R", 0.5);
//(19) 1.2D+1.3W2+0.5L+0.5R
ret = MiSapModelo.RespCombo.Add("1.2D+1.3W2+0.5L+0.5R", 0);

```

```

    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "W2", 1.3);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "L", 0.5);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.3W2+0.5L+0.5R",
SAP2000v16.eCType.LoadCase, "R", 0.5);
    //(20) 1.2D+1.0E+0.5L+0.2S
    ret = MiSapModelo.RespCombo.Add("1.2D+1.0E+0.5L+0.2S", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "E", 1.0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "L", 0.5);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D+1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "S", 0.2);
    //(21) 1.2D-1.0E+0.5L+0.2S
    ret = MiSapModelo.RespCombo.Add("1.2D-1.0E+0.5L+0.2S", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D-1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "D", 1.2);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D-1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "E", -1.0);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D-1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "L", 0.5);
    ret = MiSapModelo.RespCombo.SetCaseList("1.2D-1.0E+0.5L+0.2S",
SAP2000v16.eCType.LoadCase, "S", 0.2);
    //(22) 0.9D+1.3W1
    ret = MiSapModelo.RespCombo.Add("0.9D+1.3W1", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.3W1",
SAP2000v16.eCType.LoadCase, "D", 0.9);
    ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.3W1",
SAP2000v16.eCType.LoadCase, "W1", 1.3);
    //(23) 0.9D+1.3W2
    ret = MiSapModelo.RespCombo.Add("0.9D+1.3W2", 0);
    ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.3W2",
SAP2000v16.eCType.LoadCase, "D", 0.9);
    ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.3W2",
SAP2000v16.eCType.LoadCase, "W2", 1.3);
    //(24) 0.9D-1.3W1

```

```

ret = MiSapModelo.RespCombo.Add("0.9D-1.3W1", 0);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.3W1",
SAP2000v16.eCType.LoadCase, "D", 0.9);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.3W1",
SAP2000v16.eCType.LoadCase, "W1", -1.3);
//(25) 0.9D-1.3W2
ret = MiSapModelo.RespCombo.Add("0.9D-1.3W2", 0);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.3W2",
SAP2000v16.eCType.LoadCase, "D", 0.9);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.3W2",
SAP2000v16.eCType.LoadCase, "W2", -1.3);
//(26) 0.9D+1.0E
ret = MiSapModelo.RespCombo.Add("0.9D+1.0E", 0);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.0E",
SAP2000v16.eCType.LoadCase, "D", 0.9);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D+1.0E",
SAP2000v16.eCType.LoadCase, "E", 1.0);
//(27) 0.9D-1.0E
ret = MiSapModelo.RespCombo.Add("0.9D-1.0E", 0);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.0E",
SAP2000v16.eCType.LoadCase, "D", 0.9);
ret = MiSapModelo.RespCombo.SetCaseList("0.9D-1.0E",
SAP2000v16.eCType.LoadCase, "E", -1.0);
//(28) 1.0D+1.0L+1.0R deformada
ret = MiSapModelo.RespCombo.Add("DEFORMADA", 0);
ret = MiSapModelo.RespCombo.SetCaseList("DEFORMADA",
SAP2000v16.eCType.LoadCase, "D", 1.0);
ret = MiSapModelo.RespCombo.SetCaseList("DEFORMADA",
SAP2000v16.eCType.LoadCase, "L", 1.0);
ret = MiSapModelo.RespCombo.SetCaseList("DEFORMADA",
SAP2000v16.eCType.LoadCase, "Lr", 1.0);
//ENVOLVENTE
ret = MiSapModelo.RespCombo.Add("ENVOLVENTE", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.4D", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6L+0.5Lr", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6L+0.5S", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6L+0.5R", 1);

```

```
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6Lr+0.5L", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6Lr+0.8W1", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6Lr+0.8W2", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6S+0.5L", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6S+0.8W1", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6S+0.8W2", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6R+0.5L", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6R+0.8W1", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.6R+0.8W2", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W1+0.5L+0.5Lr", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W2+0.5L+0.5Lr", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W1+0.5L+0.5S", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W2+0.5L+0.5S", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W1+0.5L+0.5R", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.3W2+0.5L+0.5R", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D+1.0E+0.5L+0.2S", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "1.2D-1.0E+0.5L+0.2S", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D+1.3W1", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D+1.3W2", 1);
ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D-1.3W1", 1);
```

```

        ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D-1.3W2", 1);
        ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D+1.0E", 1);
        ret = MiSapModelo.RespCombo.SetCaseList("ENVOLVENTE",
SAP2000v16.eCType.LoadCombo, "0.9D-1.0E", 1);
        ret = MiSapModelo.LoadCases.Delete("DEAD");
        ret = MiSapModelo.LoadPatterns.Delete("DEAD");
    }
    public void CrearSecciones(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo)
    {
        long ret;
        double t3, t2, tf, tw, t2b, tfb, dis;
        string nomb_seccion;
        string linea;
        string[] Datos_Perfil_H = new string[7];
        string[] Datos_Perfil_U = new string[5];
        string[] Datos_Perfil_T = new string[5];
        string[] Datos_Perfil_L = new string[5];
        string[] Datos_Perfil_2L = new string[6];
        string[] Datos_Perfil_RLiso = new string[2];
        string[] Datos_Perfil_Cir = new string[3];
        string[] Datos_Perfil_Rec = new string[5];
        char[] separador = { ',' };
        StreamReader flujoEntrada;
        flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_H.txt");
        linea = flujoEntrada.ReadLine();
        while (linea != null)
        {
            Datos_Perfil_H = linea.Split(separador);
            Datos_Perfil_H[0].Trim();
            Datos_Perfil_H[1].Trim();
            Datos_Perfil_H[2].Trim();
            Datos_Perfil_H[3].Trim();
            Datos_Perfil_H[4].Trim();
            Datos_Perfil_H[5].Trim();
            Datos_Perfil_H[6].Trim();
            nomb_seccion = Datos_Perfil_H[0];
            t3 = Convert.ToDouble(Datos_Perfil_H[1]);
            t2 = Convert.ToDouble(Datos_Perfil_H[2]);

```

```

    tf = Convert.ToDouble(Datos_Perfil_H[3]);
    tw = Convert.ToDouble(Datos_Perfil_H[4]);
    t2b = Convert.ToDouble(Datos_Perfil_H[4]);
    tfb = Convert.ToDouble(Datos_Perfil_H[4]);
    ret = MiSapModelo.PropFrame.SetISection(nomb_seccion, "A36", t3, t2, tf, tw, t2b,
tfb, -1);
    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_U.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_U = linea.Split(separador);
    Datos_Perfil_U[0].Trim();
    Datos_Perfil_U[1].Trim();
    Datos_Perfil_U[2].Trim();
    Datos_Perfil_U[3].Trim();
    Datos_Perfil_U[4].Trim();
    nomb_seccion = Datos_Perfil_U[0];
    t3 = Convert.ToDouble(Datos_Perfil_U[1]);
    t2 = Convert.ToDouble(Datos_Perfil_U[2]);
    tf = Convert.ToDouble(Datos_Perfil_U[3]);
    tw = Convert.ToDouble(Datos_Perfil_U[4]);
    ret = MiSapModelo.PropFrame.SetChannel(nomb_seccion, "A36", t3, t2, tf, tw, -1);

    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_T.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_T = linea.Split(separador);
    Datos_Perfil_T[0].Trim();
    Datos_Perfil_T[1].Trim();
    Datos_Perfil_T[2].Trim();
    Datos_Perfil_T[3].Trim();
    Datos_Perfil_T[4].Trim();
}

```

```

nomb_seccion = Datos_Perfil_T[0];
t3 = Convert.ToDouble(Datos_Perfil_T[1]);
t2 = Convert.ToDouble(Datos_Perfil_T[2]);
tf = Convert.ToDouble(Datos_Perfil_T[3]);
tw = Convert.ToDouble(Datos_Perfil_T[4]);
ret = MiSapModelo.PropFrame.SetTee(nomb_seccion, "A36", t3, t2, tf, tw, -1);

    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_L.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_L = linea.Split(separador);
    Datos_Perfil_L[0].Trim();
    Datos_Perfil_L[1].Trim();
    Datos_Perfil_L[2].Trim();
    Datos_Perfil_L[3].Trim();
    Datos_Perfil_L[4].Trim();
    nomb_seccion = Datos_Perfil_L[0];
    t3 = Convert.ToDouble(Datos_Perfil_L[1]);
    t2 = Convert.ToDouble(Datos_Perfil_L[2]);
    tf = Convert.ToDouble(Datos_Perfil_L[3]);
    tw = Convert.ToDouble(Datos_Perfil_L[4]);
    ret = MiSapModelo.PropFrame.SetAngle(nomb_seccion, "A36", t3, t2, tf, tw, -1);

    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_2L.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_2L = linea.Split(separador);
    Datos_Perfil_2L[0].Trim();
    Datos_Perfil_2L[1].Trim();
    Datos_Perfil_2L[2].Trim();
    Datos_Perfil_2L[3].Trim();

```

```

    Datos_Perfil_2L[4].Trim();
    Datos_Perfil_2L[5].Trim();
    nomb_seccion = Datos_Perfil_2L[0];
    t3 = Convert.ToDouble(Datos_Perfil_2L[1]);
    t2 = Convert.ToDouble(Datos_Perfil_2L[2]);
    tf = Convert.ToDouble(Datos_Perfil_2L[3]);
    tw = Convert.ToDouble(Datos_Perfil_2L[4]);
    dis = Convert.ToDouble(Datos_Perfil_2L[5]);
    ret = MiSapModelo.PropFrame.SetDbAngle(nomb_seccion, "A36", t3, t2, tf, tw,
dis, -1);
    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada =
File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_RLiso.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_RLiso = linea.Split(separador);
    Datos_Perfil_RLiso[0].Trim();
    Datos_Perfil_RLiso[1].Trim();
    nomb_seccion = Datos_Perfil_RLiso[0];
    t3 = Convert.ToDouble(Datos_Perfil_RLiso[1]);
    ret = MiSapModelo.PropFrame.SetCircle(nomb_seccion, "A36", t3, -1);

    linea = flujoEntrada.ReadLine();
}
flujoEntrada.Close();

flujoEntrada = File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_Cir.txt");
linea = flujoEntrada.ReadLine();
while (linea != null)
{
    Datos_Perfil_Cir = linea.Split(separador);
    Datos_Perfil_Cir[0].Trim();
    Datos_Perfil_Cir[1].Trim();
    Datos_Perfil_Cir[2].Trim();
    nomb_seccion = Datos_Perfil_Cir[0];
    t3 = Convert.ToDouble(Datos_Perfil_Cir[1]);
    tw = Convert.ToDouble(Datos_Perfil_Cir[2]);

```

```

        ret = MiSapModelo.PropFrame.SetPipe(nomb_seccion, "A500GrB42", t3, tw, 1);
        linea = flujoEntrada.ReadLine();
    }
    flujoEntrada.Close();

    flujoEntrada =
File.OpenText(@"D:\SapHOWE\Componentes\PerfilesACERO_Rec.txt");
    linea = flujoEntrada.ReadLine();
    while (linea != null)
    {
        Datos_Perfil_Rec = linea.Split(separador);
        Datos_Perfil_Rec[0].Trim();
        Datos_Perfil_Rec[1].Trim();
        Datos_Perfil_Rec[2].Trim();
        Datos_Perfil_Rec[3].Trim();
        Datos_Perfil_Rec[4].Trim();
        nomb_seccion = Datos_Perfil_Rec[0];
        t3 = Convert.ToDouble(Datos_Perfil_Rec[1]);
        t2 = Convert.ToDouble(Datos_Perfil_Rec[2]);
        tf = Convert.ToDouble(Datos_Perfil_Rec[3]);
        tw = Convert.ToDouble(Datos_Perfil_Rec[4]);
        ret = MiSapModelo.PropFrame.SetTube(nomb_seccion, "A500GrB46", t3, t2, tf, tw,
-1);
        linea = flujoEntrada.ReadLine();
    }
    flujoEntrada.Close();
}

public void DibujarObjFrame(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo,double Longitud,double Flecha,double
NumSegmentos,string NombSeccBrida
    ,string NombSeccMontante,string NombSeccDiagonal)
    {
        long ret;
        bool temp_bool;
        double xi = 0, xj = 0, zi = 0, zj = 0, xFrameLong, zFrameLong, DivisionesZ;
        xFrameLong = Longitud / NumSegmentos;
        DivisionesZ = NumSegmentos / 2;
        zFrameLong = Flecha / DivisionesZ;
        Li = Math.Round((Math.Sqrt(Math.Pow(xFrameLong, 2) + Math.Pow(zFrameLong, 2))),
2);

```

```

    DibujaBridaInferior(MiSapObjeto,MiSapModelo, xi, xj, zi, zj, NumSegmentos,
xFrameLong, Flecha,NombSeccBrida);
    DibujaBridaSuperior(MiSapObjeto, MiSapModelo, xi, xj, zi, zj, DivisionesZ,
xFrameLong, zFrameLong,NombSeccBrida);
    DibujaMontantes(MiSapObjeto, MiSapModelo, xi, xj, zi, zj, DivisionesZ, xFrameLong,
zFrameLong,NombSeccMontante);
    DibujaDiagonales(MiSapObjeto, MiSapModelo, xi, xj, zi, zj, DivisionesZ, xFrameLong,
zFrameLong,NombSeccDiagonal);

    // REFRESCA LA VENTANA PARA VISUALIZAR EL OBJETO FRAME
    //////////////////////////////////////
    temp_bool = false;
    ret = MiSapModelo.View.RefreshView(0, temp_bool);
}
public void DibujaBridaInferior(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo,double xi, double xj, double zi, double zj
, double NumSegmentos, double xFrameLong, double Flecha,string NombSeccBrida)
{
    long ret;
    bool[] Restriccion1;
    bool[] Restriccion2;

    //////////////////////////////////

    // bool[] ii;
    // bool[] jj;
    // double[] StartValue;
    //double[] EndValue;

    //ii = new bool[5];
    // jj = new bool[5];
    //StartValue = new double[5];
    //EndValue = new double[5];

    //ii[0] = true;
    //StartValue[0] = 10;
    //////////////////////////////////

    //DIBUJA BRIDA INFERIOR
    //////////////////////////////////
    ret = MiSapModelo.GroupDef.SetGroup("BridaInf");
    for (int i = 0; i < NumSegmentos; i++)

```

```

{
    xj = xj + xFrameLong;
    ret = MiSapModelo.FrameObj.AddByCoord(xi, 0, zi, xj, 0, zj, "", NombSeccBrida);
    ret = MiSapModelo.FrameObj.SetGroupAssign(Convert.ToString(i+1), "BridaInf");
    xi = xj;
}

//ASIGNACION DE LAS RESTRICCIONES FIJO Y MOVIL
////////////////////////////////////
Restriccion1 = new bool[6];
for (int i = 0; i < 3; i++)
{
    Restriccion1[i] = true;
}
for (int i = 3; i < 6; i++)
{
    Restriccion1[i] = false;
}
Restriccion2 = new bool[6];
for (int i = 0; i < 6; i++)
{
    if (i == 2)
    {
        Restriccion2[i] = true;
    }
    else
    {
        Restriccion2[i] = false;
    }
}

ret = MiSapModelo.PointObj.SetRestraint("1", ref Restriccion1);
ret = MiSapModelo.PointObj.SetRestraint(Convert.ToString(NumSegmentos + 1), ref
Restriccion2);
}
private void DibujaBridaSuperior(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo, double xi, double xj, double zi, double zj, double
DivisionesZ
, double xFrameLong, double zFrameLong, string NombSeccBrida)
{
    string Cadena_Temp1;

```

```

long ret;
Nombre_Frame_BridalZq = new string[Convert.ToInt32(DivisionesZ)];
Nombre_Frame_BridaDere = new string[Convert.ToInt32(DivisionesZ)];

Cadena_Temp1 = "";

//DIBUJA BRIDA SUPERIOR IZQUIERDA
////////////////////
for (int i = 0; i < DivisionesZ; i++)
{
    xj = xj + xFrameLong;
    zj = zj + zFrameLong;
    ret = MiSapModelo.FrameObj.AddByCoord(xi, 0, zi, xj, 0, zj, ref Cadena_Temp1,
NombSeccBrida);
    ret = MiSapModelo.FrameObj.SetLocalAxes(Cadena_Temp1, -90);
    Nombre_Frame_BridalZq[i] = Cadena_Temp1;
    /* if (xi == 0)
    {
        // Li = 8;
        Li = Math.Round((Math.Sqrt(Math.Pow(xj, 2) + Math.Pow(zj, 2))), 2);
    }*/
    xi = xj;
    zi = zj;
}

//DIBUJA BRIDA SUPERIOR DERECHA
////////////////////
for (int i = 0; i < DivisionesZ; i++)
{
    xj = xj + xFrameLong;
    zj = zj - zFrameLong;
    ret = MiSapModelo.FrameObj.AddByCoord(xi, 0, zi, xj, 0, zj, ref Cadena_Temp1,
NombSeccBrida);
    ret = MiSapModelo.FrameObj.SetLocalAxes(Cadena_Temp1, -90);
    Nombre_Frame_BridaDere[i] = Cadena_Temp1;
    xi = xj;
    zi = zj;
}
}

```

```

private void DibujaMontantes(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo, double xi, double xj, double zi, double zj, double
DivisionesZ
, double xFrameLong, double zFrameLong,string NombSeccMontante)
{
    long ret;
    //DIBUJA MONTANTES IZQUIERDA
    //////////////////////////////////////
    for (int i = 0; i < DivisionesZ; i++)
    {
        xj = xj + xFrameLong;
        zj = zj + zFrameLong;
        ret = MiSapModelo.FrameObj.AddByCoord(xj, 0, zj, xj, 0, 0, "",
NombSeccMontante);
        xi = xj;
        zi = zj;
    }
    //DIBUJA MONTANTES DERECHA
    //////////////////////////////////////
    for (int i=0;i<DivisionesZ-1;i++)
    {
        xj = xj + xFrameLong;
        zj = zj - zFrameLong;
        ret = MiSapModelo.FrameObj.AddByCoord(xj,0,zj,xj,0,0,"",NombSeccMontante);
    }
    //DIBUJA MONTANTES DERECHA
    //////////////////////////////////////
    /*for (int i = 0; i < DivisionesZ; i++)
    {
        xj = xj + xFrameLong;
        zj = zj - zFrameLong;
        ret = MiSapModelo.FrameObj.AddByCoord(xi, 0, zi, xi, 0, 0, "",
NombSeccMontante);
        xi = xj;
        zi = zj;
    }*/
}

private void DibujaDiagonales(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo, double xi, double xj, double zi, double zj, double
DivisionesZ
, double xFrameLong, double zFrameLong,string NombSeccDiagonal)

```

```

{
    long ret;
    //DIBUJA DIAGONALES IZQUIERDA
    //////////////////////////////////////
    for (int i = 0; i < DivisionesZ; i++)
    {
        xj = xj + xFrameLong;
        zj = zj + zFrameLong;
        if (i < DivisionesZ - 1)
        {
            ret = MiSapModelo.FrameObj.AddByCoord(xj, 0, zj, xj + xFrameLong, 0, 0, "",
NombSeccDiagonal);
        }
        xi = xj;
        zi = zj;
    }
    //DIBUJA DIAGONALES DERECHA
    //////////////////////////////////////
    for (int i = 0; i < DivisionesZ; i++)
    {
        xj = xj + xFrameLong;
        zj = zj - zFrameLong;
        if (i != 0)
        {
            ret = MiSapModelo.FrameObj.AddByCoord(xi, 0, zi, xi - xFrameLong, 0, 0, "",
NombSeccDiagonal);
        }
        xi = xj;
        zi = zj;
    }
}

public void AsignaCargas(SAP2000v16.SapObject MiSapObjeto, SAP2000v16.cSapModel
MiSapModelo,double NumSegmentos,double TotalCM,double TotalCV,double
TotalCLG,double TotalBarloW1
    ,double TotalBarloW2,double TotalSotaW1,double TotalSotaW2,double
SepEntreTijeral)
{
    int ret;
    string[] Nombre_Punto;
    string Cadena_Temp1 = "", Cadena_Temp2 = "";
    double DivisionesZ;

```

```

double[] valorCarga;
DivisionesZ = NumSegmentos / 2;
Nombre_Punto = new string[2];
valorCarga = new double[6];
//////////CARGA MUERTA//////////
//ASIGNA CARGA MUERTA A LA BRIDA SUPERIOR IZQUIERDA
for (int i = 0; i < DivisionesZ; i++)
{
    if (i == 0)
    {
        valorCarga[2] = -((TotalCM * Li) / 2) * SepEntreTijeral;
    }
    else
    {
        valorCarga[2] = -(TotalCM * Li) * SepEntreTijeral;
    }
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridalZq[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

    ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "D", ref valorCarga,
false);
}
//ASIGNA CARGAS MUERTA A LA BRIDA SUPERIOR DERECHA
for (int i = 0; i < DivisionesZ; i++)
{
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

    ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "D", ref valorCarga,
false);
    if (i == DivisionesZ - 1)
    {
        valorCarga[2] = -((TotalCM * Li) / 2) * SepEntreTijeral;
        ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
        Nombre_Punto[0] = Cadena_Temp1;
        Nombre_Punto[1] = Cadena_Temp2;
    }
}

```

```

        ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[1], "D", ref
valorCarga, false);
    }
    else
    {
        valorCarga[2] = -(TotalCM * Li) * SepEntreTijeral;
    }
}
//////////CARGA VIVA//////////
//ASIGNA CARGA VIVA A LA BRIDA SUPERIOR IZQUIERDA
for (int i = 0; i < DivisionesZ; i++)
{
    if (i == 0)
    {
        valorCarga[2] = -((TotalCV * Li) / 2) * SepEntreTijeral;
    }
    else
    {
        valorCarga[2] = -(TotalCV * Li) * SepEntreTijeral;
    }
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridalZq[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

    ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "Lr", ref valorCarga,
false);
}
//ASIGNA CARGAS VIVA A LA BRIDA SUPERIOR DERECHA
for (int i = 0; i < DivisionesZ; i++)
{
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

    ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "Lr", ref valorCarga,
false);
    if (i == DivisionesZ - 1)
    {
        valorCarga[2] = -((TotalCV * Li) / 2) * SepEntreTijeral;
    }
}

```

```

        ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
        Nombre_Punto[0] = Cadena_Temp1;
        Nombre_Punto[1] = Cadena_Temp2;
        ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[1], "Lr", ref
valorCarga, false);
    }
    else
    {
        valorCarga[2] = -(TotalCV * Li) * SepEntreTijeral;
    }
}
////////////////////////////////CARGA LLUVIA O GRANIZO////////////////////////////////
//ASIGNA CARGA LLUVIA O GRANIZO A LA BRIDA SUPERIOR IZQUIERDA
for (int i = 0; i < DivisionesZ; i++)
{
    if (i == 0)
    {
        valorCarga[2] = -((TotalCLG * Li) / 2) * SepEntreTijeral;
    }
    else
    {
        valorCarga[2] = -(TotalCLG * Li) * SepEntreTijeral;
    }
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_Bridalq[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

    ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "R", ref valorCarga,
false);
}
//ASIGNA CARGAS LLUVIA O GRANIZO A LA BRIDA SUPERIOR DERECHA
for (int i = 0; i < DivisionesZ; i++)
{
    ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
    Nombre_Punto[0] = Cadena_Temp1;
    Nombre_Punto[1] = Cadena_Temp2;

```

```

        ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[0], "R", ref valorCarga,
false);
        if (i == DivisionesZ - 1)
        {
            valorCarga[2] = -((TotalCLG * Li) / 2) * SepEntreTijeral;
            ret = MiSapModelo.FrameObj.GetPoints(Nombre_Frame_BridaDere[i], ref
Cadena_Temp1, ref Cadena_Temp2);
            Nombre_Punto[0] = Cadena_Temp1;
            Nombre_Punto[1] = Cadena_Temp2;
            ret = MiSapModelo.PointObj.SetLoadForce(Nombre_Punto[1], "R", ref
valorCarga, false);
        }
        else
        {
            valorCarga[2] = -(TotalCLG * Li) * SepEntreTijeral;
        }
    }
    //////////////////////////////////CARGA VIENTO////////////////////////////////
    //ASIGNA CARGA VIENTO W1 A LA BRIDA SUPERIOR IZQUIERDA
    for (int i = 0; i < DivisionesZ; i++)
    {
        if (i == 0)
        {
            ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W1", 1,
3, 0, -((TotalBarloW1 * Li * SepEntreTijeral) / 2), "Local");
        }
        if (i > 0 && i < DivisionesZ-1)
        {
            ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W1", 1,
3, 0, -(TotalBarloW1 * Li * SepEntreTijeral), "Local");
        }
        if (i == DivisionesZ - 1)
        {
            ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W1", 1,
3, 0, -(TotalBarloW1 * Li * SepEntreTijeral), "Local");
            ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W1", 1,
3, Li, -(TotalBarloW1 * Li * SepEntreTijeral), "Local",true,false);
        }
    }
    //ASIGNA CARGAS VIENTO W1 A LA BRIDA SUPERIOR DERECHA
    for (int i = 0; i < DivisionesZ; i++)

```

```

{
    if (i == 0)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W1",
1, 3, 0, -(TotalSotaW1 * Li * SepEntreTijeral), "Local");
    }
    if (i > 0 && i < DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W1",
1, 3, 0, -(TotalSotaW1 * Li * SepEntreTijeral), "Local");
    }
    if (i == DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W1",
1, 3, 0, -(TotalSotaW1 * Li * SepEntreTijeral), "Local");
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W1",
1, 3, Li, -(TotalSotaW1 * Li * SepEntreTijeral) / 2, "Local",true,false);
    }
}
//ASIGNA CARGA VIENTO W2 A LA BRIDA SUPERIOR IZQUIERDA
for (int i = 0; i < DivisionesZ; i++)
{
    if (i == 0)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W2", 1,
3, 0, -((TotalBarloW2 * Li * SepEntreTijeral) / 2), "Local");
    }
    if (i > 0 && i < DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W2", 1,
3, 0, -(TotalBarloW2 * Li * SepEntreTijeral), "Local");
    }
    if (i == DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W2", 1,
3, 0, -(TotalBarloW2 * Li * SepEntreTijeral), "Local");
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_Bridalzq[i], "W2", 1,
3, Li, -(TotalBarloW2 * Li * SepEntreTijeral), "Local",true,false);
    }
}
//ASIGNA CARGAS VIENTO W2 A LA BRIDA SUPERIOR DERECHA

```

```

for (int i = 0; i < DivisionesZ; i++)
{
    if (i == 0)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W2",
1, 3, 0, -(TotalSotaW2 * Li * SepEntreTijeral), "Local");
    }
    if (i > 0 && i < DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W2",
1, 3, 0, -(TotalSotaW2 * Li * SepEntreTijeral), "Local");
    }
    if (i == DivisionesZ - 1)
    {
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W2",
1, 3, 0, -(TotalSotaW2 * Li * SepEntreTijeral), "Local");
        ret = MiSapModelo.FrameObj.SetLoadPoint(Nombre_Frame_BridaDere[i], "W2",
1, 3, Li, -(TotalSotaW2 * Li * SepEntreTijeral) / 2, "Local",true,false);
    }
}
}

public void AnalizarSinMomentos(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo)
{
    long ret;
    bool[] ii;
    bool[] jj;
    double[] InicioEvaluacion;
    double[] FinalEvaluacion;

    ii = new bool[6];
    jj = new bool[6];
    InicioEvaluacion = new double[6];
    FinalEvaluacion = new double[6];

    for (int i = 0; i <= 5; i++)
    {
        if (i == 4 || i == 5)
        {
            ii[i] = true;
            jj[i] = true;

```

```

        InicioEvaluacion[i] = 0;
        FinalEvaluacion[i] = 0;
    }
    else
    {
        ii[i] = false;
        jj[i] = false;
        InicioEvaluacion[i] = 0;
        FinalEvaluacion[i] = 0;
    }
}
ret = MiSapModelo.FrameObj.SetSelected("ALL", true,
SAP2000v16.eltemType.Group);
ret = MiSapModelo.FrameObj.SetReleases("", ref ii, ref jj, ref InicioEvaluacion, ref
FinalEvaluacion, SAP2000v16.eltemType.SelectedObjects);
ret = MiSapModelo.SelectObj.ClearSelection();
}
public void GuardarYAnalizar(SAP2000v16.SapObject MiSapObjeto,
SAP2000v16.cSapModel MiSapModelo)
{
    long ret;
    ret = MiSapModelo.File.Save("D:\\CARPETA TRABAJO VISUAL C#\\PROYECTOS
SAP2000 Y C#\\API_1-002RESUL.sdb");
    ret = MiSapModelo.Analyze.SetRunCaseFlag("MODAL", false);
    ret = MiSapModelo.Analyze.RunAnalysis();
}
}
}

```