

DESARROLLO DE SOFTWARE DE ANALISIS ESTRUCTURAL PARA EL ANALISIS
MATRICIAL DE PORTICOS Y ARMADURAS ISOSTATICAS PLANAS MEDIANTE EL
DISPOSITIVO PORTATIL TI-NSPIRE CX CAS

EDEL ANDRES GONZALEZ MARIÑO

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERIAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERIAS CIVIL, QUIMICA Y AMBIENTAL
PROGRAMA DE INGENIERIA CIVIL

2019

DESARROLLO DE SOFTWARE DE ANALISIS ESTRUCTURAL PARA EL ANALISIS
MATRICIAL DE PORTICOS Y ARMADURAS ISOSTATICAS PLANAS MEDIANTE EL
DISPOSITIVO PORTATIL TI-NSPIRE CX CAS

EDEL ANDRES GONZALEZ MARIÑO

Trabajo de grado como requisito para optar el titulo de ingeniero civil

DIRECTOR

NESTOR ORLANDO ROJAS RIBON

INGENIERO CIVIL

UNIVERSIDAD DE PAMPLONA

FACULTAD DE INGENIERIAS Y ARQUITECTURA

DEPARTAMENTO DE INGENIERIAS CIVIL, QUIMICA Y AMBIENTAL

PROGRAMA DE INGENIERIA CIVIL

2019

Dedicatoria

A mis padres, sin su dedicación y su
sacificacion esto no hubiera sido posible.

Agradecimientos

A la Universidad de Pamplona que me dio la oportunidad de formarme como ingeniero civil.

A mi tutor, el Ing. Nestor Rojas por su apoyo, su contribucion en mi formacion profesional y sus valiosos consejos.

A Victor Vergel, Jhair Delgado, Marcelino Maldonado, Luis Rojas, Luis Viracachá, Edgar Perez, Oscar Hernandez, y demas docentes por su contribucion en mi formacion profesional.

A Lennys Florez, Cristian Manrique y demas compañeros quienes hicieron que este largo proceso fuera entretenido.

Resumen

El objetivo de este proyecto es el desarrollo de un software de análisis estructural para el análisis matricial de pórticos y armaduras isostáticas en dos dimensiones mediante el dispositivo portátil TI-Nspire CX CAS. Con la finalidad de convertirse en una herramienta de apoyo para los estudiantes y docentes del departamento de ingeniería civil de la Universidad de Pamplona.

Para realizar el software de análisis se procedió a la revisión bibliográfica, se propuso un diseño básico que incluye tanto la apariencia de la interfaz gráfica de usuario como el desarrollo de los algunos de los métodos de análisis basados en la teoría de análisis estructural: Método matricial de rigidez, método de nudos y tres momentos, los cuales componen el software de análisis “Ac-Struc”

El diseño fue implementado en el lenguaje de programación multiplataforma Lua. Obteniendo como resultado el software de análisis estructural “Ac-Struc” para el dispositivo portátil TI-Nspire CX CAS. La verificación del software se realizó mediante la comparación entre los resultados generados por este y los ejercicios planteados en fuentes bibliográficas

Palabras clave: Analisis Estructural, Analisis Matricial, Metodo de rigidez, Metodo de nudos, Lua

Abstract

This study aims at developing a structural analysis software for matrix analysis of frames and isostatic truss in two dimensions through the portable device TI-Nspire CX CAS. With the purpose of becoming a supporting tool for students and teachers of the civil engineering department of the University of Pamplona.

In order to execute the analysis software the literature review was carried out. A basic desing was proposed, which includes both the graphical interface appearance and the development of some analysis methods based on the structural analysis thery. The “Ac-Struc” analysis software is compounded of direct stiffness method, method of joints and theorem of three moments.

The desing was implemented on cross platform programming language Lua,resulting in the structural analysis software “Ac-Struc” for the portable device effeted TI-Nspire CX CAS. The software verification was effectuated by a comparison between the generated by the above and the exercises propounded in bibligraphical sources.

Keywords: Structural analysis, Matrix analysis, Direct stiffness method, Method of joints, Lua

Contenido

Lista de figuras.....	XI
Lista de tablas	XVII
Lista de ecuaciones	XVIII
Introducción	1
Capítulo 1: Generalidades.....	2
1.1. Justificación de la investigación	2
1.2. Descripción del problema	3
1.3. Objetivos de la investigación	4
1.3.1 Objetivo general.....	4
1.3.2 Objetivos específicos	4
1.4. Marco referencial	5
1.4.1. Localización.....	5
1.4.2. Estado del arte.....	6
1.6. Bases teóricas.....	8
1.6.1. Equilibrio y compatibilidad	8
1.6.2. Linealidad y principio de superposición.....	15
1.6.3. Indeterminación estática. Grado de hiperestatismo	17
1.6.4. Análisis de estructuras isostáticas e hiperestáticas.....	24

1.6.5. Estructuras articuladas	26
Capítulo 2: LUA	29
2.1 Introducción a Lua	29
2.2 El lenguaje	30
2.2.1 Convecciones léxicas.....	30
2.2.2 Variables.....	33
2.2.3. Sentencias	35
2.2.4. Expresiones.....	40
Capítulo 3: Diseño del software.....	45
3.1 Especificación de requerimientos	45
3.2 Interfaz de usuario.....	48
3.2.1 Diseño de la interfaz de usuario.....	49
3.2.2. Color.....	51
3.2.3. Menú de opciones	52
3.2.4. Interfaz de entrada y salida de datos	52
3.3. Modelado del sistema	53
Capítulo 4: Implementación del software	57
4.1. Implementación de scripts en Lua	57
4.2. Funciones implementadas.....	58
4.2.1. Funciones de inicialización.....	59

4.2.2. Funciones de control:	60
4.2.3. Funciones de cálculo	62
4.2.4. Funciones de lectura y almacenamiento de datos	67
Capítulo 5: Fase de pruebas del software	70
5.1. Pruebas de funcionamiento de la GUI	71
5.2. Pruebas de cálculo.....	72
5.2.1. Problema 1:	72
5.2.2. Problema 2:	74
5.2.3. Problema 3:	76
5.2.4 Problema 4:	79
5.2.5. Problema 5:	81
5.2.6. Problema 6:	83
5.2.6. Problema 7:	85
5.2.8. Problema 8:	86
5.2.9. Problema 9:	87
Capítulo 6: Conclusiones y recomendaciones	88
6.1. Conclusiones	88
6.2. Recomendaciones	90
Bibliografía	91
Anexo A: Manual de usuario de Ac-Struc	94

A.1. Inicio de Ac-Struc	94
A.2. Nuevo archivo	94
A.3. Abrir Archivo	95
A.4. Método Matricial de Rigidez	96
A.4.1. Procedimiento para ingresar datos	96
A.4.2. Procedimiento para ver los resultados	105
A.5. Método de Nudos	111
A.5.1. Procedimiento para ingresar datos	111
A.5.2. Visualización de resultados	112
A.6. Tres momentos	114
A.6.1. Procedimiento para ingresar datos	114
Anexo B: Guía para el análisis de estructuras	116
B.1. Guía para el análisis de estructuras usando el método matricial de rigidez.....	116
B.1.1. Análisis detallado	116
B.1.2. Análisis usando el software Ac-struct	133
B.2. Guía para el análisis de armaduras isostáticas usando el método de nudos....	140
B.2.1. Análisis detallado	140
B.2.2. Análisis usando el software Ac-Struc.....	143

Lista de figuras

Figura 1: Universidad de Pamplona.....	5
Figura 2: Equilibrio en una estructura articulada: Equilibrio global	11
Figura 3: Equilibrio en una estructura articulada: Equilibrio de piezas y nudos	11
Figura 4: Equilibrio en una estructura reticulada: equilibrio global	12
Figura 5: Equilibrio en una estructura reticulada: equilibrio de piezas y nudos.....	12
Figura 6: Movimientos en una estructura articulada.....	14
Figura 7: Movimientos en una estructura reticulada.....	15
Figura 8: Principio de Superposición.....	17
Figura 9: Estructuras Isostáticas	19
Figura 10: Estructuras hiperestáticas	19
Figura 11: Estructura hiperestática de grado 1.....	20
Figura 12: Estructura isostática de varias piezas	21
Figura 13: Estructura hiperestática de grado 3	21
Figura 14: Estructuras exteriormente isostáticas - interiormente hiperestáticas.....	22
Figura 15: Estructuras exteriormente hiperestáticas - interiormente hiperestáticas	22
Figura 16: Esquema de resolución de estructuras isostáticas	25
Figura 17: Esquema de resolución de estructuras hiperestáticas	26
Figura 18: Estructura articulada isostática.....	27
Figura 19: Estructura articulada hiperestática.....	28
Figura 20: Estructura articulada con mecanismo parcial.....	28
Figura 21: Colores corporativos de la Universidad de Pamplona.....	51

Figura 22:Diseño menú de opciones.....	52
Figura 23: Diseño general para la entrada de datos	53
Figura 24: Diagrama de flujo funcionamiento general del software	54
Figura 25: Diagrama de flujo del método matricial de rigidez.....	55
Figura 26: Diagrama de flujo detallado de la función multiplicar matrices	56
Figura 27: Interface del editor de scripts	57
Figura 28:Fragmento de Código: Función on.construction()	59
Figura 29:Fragmento de Código: Función Nuevo Armadura	60
Figura 30: Fragmento de Código: Función Nuevo Pórtico.....	60
Figura 31: Fragmento de Código: Función Control Ventana.....	61
Figura 32: Fragmento de Código: Función Finalizar_Ejecucion_R()	62
Figura 33: Fragmento de Código: Función Ejecutar Rigidez	63
Figura 34: Fragmento de Código: Función Multiplicar Matrices.....	64
Figura 35: Fragmento de Código: Función Crear Matriz General.....	64
Figura 36: Fragmento de Código: Función Calcular Deformaciones.....	65
Figura 37: Fragmento de Código: Función Matriz Inversa	66
Figura 38: Fragmento de Código: Función Decimales()	66
Figura 39: Fragmento de Código: Función Ejecutar Armadura.....	67
Figura 40: Fragmento de Código: Función Abrir Archivo.....	68
Figura 41: Fragmento de Código: Función Leer Datos	69
Figura 42: Error no esperado en la GUI.....	71
Figura 43: Problema 1	72
Figura 44: Reacciones problema 1.....	73

Figura 45: Reacciones problema 1 usando Ac-Struc	73
Figura 46: Problema 2.....	74
Figura 47: Reacciones problema 2.....	75
Figura 48: Reacciones problema 2 usando Ac-Struc	75
Figura 49: Problema 3.....	76
Figura 50: Reacciones problema 3.....	77
Figura 51: Reacciones problema 3 usando Ac-Struc	77
Figura 52: Problema 4.....	79
Figura 53: Solución problema 4.....	80
Figura 54: Reacciones problema 4 usando Ac-Struc	80
Figura 55: Fuerzas axiales de los elementos problema 4 usando Ac-Struc	80
Figura 56: Problema 5.....	81
Figura 57: Solución problema 5.....	82
Figura 58: Solución problema 5 usando Ac-Struc	82
Figura 59: Problema 6.....	83
Figura 60: Solución problema 6.....	83
Figura 61: Reacciones problema 6 usando Ac-Struc	84
Figura 62: Fuerzas axiales de los elementos problema 6 usando Ac-Struc	84
Figura 63: Problema 7.....	85
Figura 64: Momentos problema 7 usando el software Ac-Struc.....	85
Figura 65: Problema 8.....	86
Figura 66: Momentos problema 8 usando el software Ac-Struc.....	86
Figura 67: Problema 9.....	87

Figura 68: Momentos problema 9 usando el software Ac-Struc.....	87
Figura 69: Ventana de Inicio del Software Ac-Struc	94
Figura 70: Ventana Archivo Nuevo.....	95
Figura 71: Ventana Abrir Archivo	95
Figura 72: Ventana Agregar Materiales	96
Figura 73: Agregar Sección Personalizada	97
Figura 74: Ventana Insertar Nudos.....	98
Figura 75: Ventana Insertar Elementos	99
Figura 76: Ventana Insertar Apoyos y Enlaces	100
Figura 77: Ventana Insertar Cargas Puntuales	101
Figura 78: Ventana Insertar Cargas distribuidas	102
Figura 79: Ventana Definir Combinaciones de Carga	103
Figura 80: ventana para definir los valores de la combinación	104
Figura 81: Ventana Variación de Temperatura	105
Figura 82: Ventana Desplazamientos.....	106
Figura 83: Ventana Diagrama de fuerza cortante.....	107
Figura 84: Ventana Diagrama de momento flector por elemento	108
Figura 85: Ventana de matriz K en coordenadas locales	109
Figura 86: Ventana Matriz de rigidez K de la estructura	110
Figura 87: Ventana Insertar apoyos método de nudos	111
Figura 88: Ventana de reacciones de las armaduras isostáticas	112
Figura 89: Ventana de ingresar nudos para el cálculo de vigas hiperestáticas.....	114
Figura 90: Ventana de ingresar elementos para el cálculo de vigas hiperestáticas	115

Figura 91: Estructura hiperestática	116
Figura 92: Numeración de nudos y elementos.....	117
Figura 93: Agrupación de filas y columnas en la matriz de rigidez K de la estructura	123
Figura 94: Método para ensamblar la matriz K de la estructura.....	123
Figura 95: Matriz de rigidez K de la estructura	124
Figura 96: Agrupación de filas en el vector de deformaciones.....	124
Figura 97: Método para ensamblar el vector de empotramiento de la estructura	127
Figura 98: Método de simplificación de la matriz K de la estructura y el vector de deformaciones	128
Figura 99: Parte A y Parte B de la matriz K simplificada, el vector reacción y el vector empotramiento	129
Figura 100: Nuevo pórtico, metodo matricial de rigidez.....	133
Figura 101: Sección rectangular 0.4m x 0.4m	133
Figura 102: Materiales agregados.....	134
Figura 103: Nudos insertados	134
Figura 104: Insertar elementos:	135
Figura 105: Insertar apoyos	135
Figura 106: Insertar cargas puntuales	136
Figura 107: Insertar cargas distribuidas.....	136
Figura 108: Agregar combinación	137
Figura 109: Combinaciones agregadas	137
Figura 110: Definir variación de temperatura.....	138
Figura 111: Analizar estructura.....	138

Figura 112: Resultados.....	139
Figura 113: Reacciones de la estructura	139
Figura 114: Nuevo método de los nudos	143
Figura 115:Insertar nudos armadura isostática	144
Figura 116: Insertar elemento armadura isostática	144
Figura 117: Insertar apoyos armadura isostática.....	145
Figura 118: Insertar cargas puntuales armadura isostática	145
Figura 119: Combinaciones agregadas	146
Figura 120:Analizar armadura isostática	146
Figura 121: Resultados armadura isostática	147
Figura 122: Fuerzas de los elementos de las armaduras isostáticas.....	147

Lista de tablas

Tabla 1: Requerimientos Control de las Ventanas	45
Tabla 2: Requerimientos calculo método matricial de rigidez.....	46
Tabla 3:Requerimientos calculo método de nudos	47
Tabla 4: Requerimientos de funcionalidad abrir archivo	47
Tabla 5: Requerimientos de funcionalidad guardar archivo	48
Tabla 6: Principios de diseño de las interfaces de usuario.....	50
Tabla 7: Detalles error en la GUI.....	71
Tabla 8: Tabla de datos metodo matricial de rigidez.....	117
Tabla 9: Tabla de rigideces método matricial de rigidez.....	117
Tabla 10:Tabla de reacciones de los elementos	119

Lista de ecuaciones

Ecuación 1: Sumatoria Vectorial de Fuerzas.....	8
Ecuación 2: Sumatoria Vectorial de Momentos.....	8
Ecuación 3: Sumatoria de fuerzas en el eje X.....	9
Ecuación 4: Sumatoria de fuerzas en el eje Y.....	9
Ecuación 5: Sumatoria de Momentos	9
Ecuación 6: Equilibrio de momentos con respecto al punto C	11
Ecuación 7: Grado de hiperestatismo externo	20
Ecuación 8: Grado de hiperestatismo de una estructura	23
Ecuación 9: grado de indeterminación estática de una estructura articulada.....	27
Ecuación 10: Matriz de rigidez K coordenadas locales.....	118
Ecuación 11: Vector de empotramiento en coordenadas locales	119
Ecuación 13: Matriz de transformación.....	120
Ecuación 14: Matriz de rigidez en coordenadas globales.....	121
Ecuación 15: Vector de empotramiento en coordenadas globales	122
Ecuación 16: Calculo de deformaciones de la estructura	129
Ecuación 17: Calculo de reacciones de la estructura	130
Ecuación 18: Vector de deformaciones en coordenadas locales.....	131
Ecuación 19: Vector de reacciones en coordenadas locales.....	131
Ecuación 20: Fuerzas en los extremos de cada elemento	132
Ecuación 21: Grado de indeterminación estática de la estructura	140

Introducción

La presente investigación nace a raíz de la falta de una herramienta informática de análisis estructural que brinde apoyo a los estudiantes y docentes del programa de ingeniería civil de la Universidad de Pamplona.

Gran parte de los estudiantes y docentes del programa han adquirido o desean adquirir el dispositivo portátil Ti-Nspire CX CAS desarrollado por la empresa estadounidense Texas Instruments, el cual permite la instalación de software independiente desarrollado por los usuarios del ya mencionado dispositivo, sin embargo, muchos de estos programas no se encuentran disponibles para el público en general ya es necesario la adquisición de una licencia para su uso, dificultando así el proceso de comprensión y aprendizaje de los métodos de análisis estructural.

La investigación se centrará en el desarrollo de un software de análisis estructural para el dispositivo portátil Ti-Nspire CX CAS que se desarrollara en el lenguaje de programación de código abierto llamado LUA, en el cual es sencillo y práctico desarrollar el método matricial de la rigidez, el método de tres momentos, el método de los nodos para el análisis de armaduras isostáticas y los teoremas de Castigliano, con el fin de facilitar la realización de cálculos y así despertar el interés de los usuarios en los temas relacionados tanto en el análisis y diseño estructural, como en la ingeniería de software.

Capítulo 1: Generalidades

1.1. Justificación de la investigación

Los procesos ingenieriles basados en métodos matemáticos son extensos y repetitivos, como es el caso de los métodos de análisis estructural, es por eso que en la actualidad es indispensable el uso de softwares o herramientas informáticas que disminuyan el tiempo requerido para realizar dichos análisis, debido a esto es necesario la implementación de un software que permita aumentar la confiabilidad de los datos obtenidos y además que permita una disminución considerable del tiempo empleado en la realización de estos cálculos, lo que a su vez permite profundizar más a detalle los temas analizados.

Por otra parte, desde la aparición de los primeros transistores en la década de 1950 la humanidad ha dado grandes pasos en lo que a tecnología e informática se refiere, fabricando dispositivos cada vez más potentes que además ocupan menos espacio como lo son los computadores portátiles actuales, sin embargo, en el ámbito universitario o laboral esto no es suficiente, en algunas ocasiones el uso de estos equipos puede resultar incómodo debido a su tamaño o a su peso, es por esto que el uso del dispositivo portátil Ti-Nspire CX CAS resulta más práctico y sencillo ya que permite la realización de cálculos complejos a pesar del reducido espacio que ocupa.

1.2. Descripción del problema

Los avances tecnológicos de las últimas décadas han mejorado la calidad de vida del ser humano beneficiándolo en muchos aspectos tanto cotidianos como educativos y laborales, sin embargo, es necesario destacar la aparición de los softwares relacionados con la ingeniería como lo son los softwares de análisis estructural, los cuales simplifican las tareas de análisis y diseño de obras realizando cálculos complejos, y de esta manera reducir en gran medida el tiempo requerido en esta etapa temprana de un proyecto de obras civiles.

En el mercado actual existe una gran variedad de softwares diseñados para realizar los cálculos relacionados con el análisis estructural, cálculos que pueden llegar a ser complicados y muy extensos. Los softwares de análisis estructural que son confiables son a su vez costosos, por lo que muchos de los estudiantes del programa de ingeniería civil de la Universidad de Pamplona no pueden acceder a ellos.

Además, el programa de ingeniería civil de la Universidad de Pamplona no cuenta con una licencia de uso de un software análisis estructural, ni tampoco posee un software propio el cual brinde apoyo a los estudiantes en su formación profesional ya que este permitiría la complementación de sus estudios y la realización de proyectos académicos que estén relacionados con el diseño y análisis estructural.

1.3. Objetivos de la investigación

1.3.1 Objetivo general

Desarrollar un software de análisis estructural para el análisis matricial de pórticos y armaduras isostáticas en dos dimensiones mediante el dispositivo portátil TI-Nspire CX CAS

1.3.2 Objetivos específicos

Identificar el procedimiento para el desarrollo de los métodos de análisis planteados

Diseñar la arquitectura y algoritmos necesarios para el funcionamiento del software

Implementar el software aplicando los métodos planteados mediante el lenguaje de programación LUA.

Comprobar el correcto funcionamiento de los algoritmos desarrollados detectando así errores en las sentencias

Elaborar un manual sobre el manejo básico del software y la aplicación de los métodos seleccionados

1.4. Marco referencial

1.4.1. Localización

El proyecto de investigación se realizará en la Universidad de Pamplona, la cual se encuentra ubicada en la ciudad de Pamplona, Norte de Santander, ubicada en las coordenadas $7^{\circ}23'09''\text{N}$; $72^{\circ}39'00''\text{O}$



Figura 1: Universidad de Pamplona

Fuente: Google Earth

1.4.2. Estado del arte

Carlos Mario Plata Arrieta en su trabajo de grado llamado “Aplicación del software de la Ti-Nspire CX CAS como herramienta de apoyo en el análisis geotécnico para el diseño de cimentaciones superficiales, con base en el reglamento colombiano de construcción sismo resistentes NSR-10” desarrolla un software de análisis geotécnico, el cual tenía como finalidad convertirse en una herramienta de soporte para los estudiantes del programa de ingeniería civil de la Universidad de Pamplona, haciendo uso del lenguaje de programación LUA y los métodos propuestos por Terzagui, Meyerhof, Hansen y Vesic. Con este proyecto se pudo demostrar la versatilidad del lenguaje de programación seleccionado y la confiabilidad que genera el uso de herramientas informáticas como apoyo para el diseño de proyectos afines a la ingeniería civil. (Plata Arrieta , 2015)

Manuel Jesús Meza Huaynate en su trabajo de grado: “Desarrollo del software de aplicación ESDID para el cálculo de espectros elásticos de diseños determinísticos para edificaciones” elaboro un software que analiza tres componentes principales: tratamiento de registros sísmicos, cálculo de espectros de respuesta y cálculo de los espectros elásticos de diseño, los cuales proporcionan resultados confiables ya que fueron comparados con un software comercial. (Meza Huaynate, 2019)

Arnulfo Javier Espinoza en su trabajo de grado: “MARPLA, Programa para el análisis estructural matricial de marcos planos” implementó un software que realiza el análisis estructural de marcos planos con la intención de auxiliar la formación académica de los estudiantes de

ingeniería civil en sus cursos del área de la ingeniería estructural en la Universidad Nacional Autónoma de México. (Espinoza, 1995)

Juan Regalado Ramírez y Arturo Romero Garrido en su trabajo de grado titulado:” Software Interactivo para el Diseño de Trabes Armadas con el método LRFD” desarrollaron un software programado en Visual Basic 6.0 para el diseño de trabes armadas en acero de sección I. Los resultados del software se verificaron a través de una comparación con ejercicios resueltos manualmente y paso a paso, con lo cual se verifico que los resultados obtenidos a partir del software son confiables. Juan Regalado y Arturo Romero concluyen:” Es importante recordar que el resultado de este trabajo es tan solo una herramienta de apoyo para los estudiantes de licenciatura en ingeniería civil, de modo que el software no deberá reemplazar el estudio teórico del tema” (Regalado Ramírez & Romero Garrido, 2004.)

Levy Samperio en su trabajo de grado:” Software para el Diseño de Vigas I de Concreto Presforzado” construyó una aplicación que permite el diseño de vigas tipo I de concreto presforzado. Cuyo proceso de diseño se basa en el cálculo de la resistencia a flexión y a cortante, así como las pérdidas que se presentan en el preesfuerzo. Levy Samperio concluye:” ...siempre se debe tener en cuenta que un programa de cómputo es una herramienta de apoyo, y que esta no sustituye el criterio del diseñador...” (Samperio Fuentes, 2004)

Octavio Hinojoza en su tesis titulada: “Software para la enseñanza de la dinámica estructural” desarrollo un software de dinámica estructural abarcando dos sub temas: osciladores IGL y VGL. Hinojoza concluye: “ El software educativo por sí solo no puede enseñar al alumno; es necesario siempre contar con la explicación del profesor antes y durante el tema.(...) El uso de

la tecnología dentro del salón de clase resulta ser muy benéfica para el proceso enseñanza-aprendizaje, siempre y cuando ésta sea bien dirigida y que la conjunción entre la tecnología y los temas de contenido este bien estructurada. (Hinojoza Gabriel, 2009)”

1.6. Bases teóricas

1.6.1. Equilibrio y compatibilidad

Las fuerzas (acciones y reacciones) que actúan sobre una estructura deben estar en equilibrio estático. Esto significa que deben formar un sistema de fuerzas de resultante nula y de momento resultante nulo; por tanto, deben cumplir las ecuaciones que se conocen con el nombre de ecuaciones de la estática, que, en forma vectorial, pueden escribirse como:

$$\sum_i \vec{F}_i = 0$$

Ecuación 1: Sumatoria Vectorial de Fuerzas

$$\sum_i \vec{M}_i^O = 0$$

Ecuación 2: Sumatoria Vectorial de Momentos

Donde: \vec{F}_i representa a cada una de las fuerzas que actúan sobre la estructura, \vec{M}_i^O

representa el momento de cada una de las fuerzas respecto de un punto arbitrario O, El símbolo

Σ_i representa la suma sobre todas las fuerzas i.

En el caso de estructuras planas cargadas en su plano, las anteriores ecuaciones vectoriales se reducen a tres ecuaciones escalares de la siguiente forma; donde los ejes X e Y están sobre el plano de la estructura y el eje Z es perpendicular a estos.

$$\sum_i (F_x)_i = 0$$

Ecuación 3: Sumatoria de fuerzas en el eje X

$$\sum_i (F_y)_i = 0$$

Ecuación 4: Sumatoria de fuerzas en el eje Y

$$\sum_i (M_z)_i = 0$$

Ecuación 5: Sumatoria de Momentos

En Mecánica de Estructuras se llama resolver una estructura a calcular el valor de los esfuerzos que actúan sobre cada una de las secciones de todas las piezas que componen la estructura. Cuando este cálculo puede realizarse íntegramente utilizando solo las ecuaciones de la estática, la estructura se llama isostática o estáticamente determinada. En caso contrario, se llama hiperestática o estáticamente indeterminada. (Cervera Ruiz & Blanco Diaz, 2001)

Pero la condición de equilibrio estático, y las correspondientes ecuaciones que lo expresan matemáticamente no son solo aplicables a la estructura considerada en su conjunto, sino que debe satisfacerse también para cada parte integrante de ella, siempre que se consideren

de forma explícita las fuerzas y momentos que el resto de la estructura ejerce sobre la parte considerada. De hecho, la condición de equilibrio global es una condición necesaria, pero no suficiente, de equilibrio. Para que haya realmente equilibrio en una estructura, y no existan, por ejemplo, mecanismos parciales, es necesario (y suficiente) que estén en equilibrio todas y cada una de sus partes integrantes. En particular, las piezas que forman una estructura de barras deben estar en equilibrio, siempre que se consideren las fuerzas y momentos de extremo de barra que la estructura ejerce sobre las piezas. Análogamente, los momentos que actúan en los extremos de las barras que concurren en ellos.

Estas condiciones de equilibrio deben satisfacerse bajo cualquier hipótesis cinemática que se adopte y, en particular, bajo la hipótesis habitual de que las deformaciones y movimientos que se producen en la estructura son pequeños. En este caso, las ecuaciones de equilibrio pueden plantearse sobre la geometría original, no deformada.

Así, por ejemplo, la estructura articulada plana de la Figura 2 está en equilibrio global bajo la acción de las reacciones N_1 y N_2 . Como el sistema de fuerzas actuante es coplanar, la condición de equilibrio (global) de momentos respecto al nudo B exige que las líneas de acción de las reacciones N_1 y N_2 pasen por dicho punto; la condición de equilibrio (global) de fuerzas exige que la suma de las reacciones sea igual y opuesta a la fuerza F . En la Figura 3 se observa como las condiciones de equilibrio global implican necesariamente que tanto las dos barras de la estructura como el nudo B estén también en equilibrio, considerados como elementos aislados de la estructura.

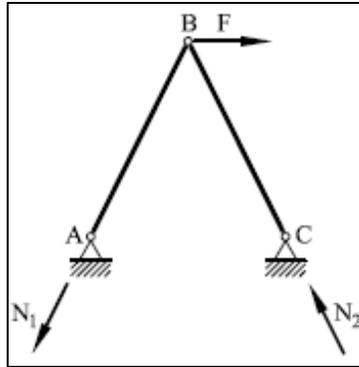


Figura 2: Equilibrio en una estructura articulada: Equilibrio global

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

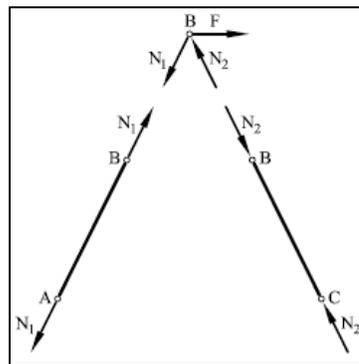


Figura 3: Equilibrio en una estructura articulada: Equilibrio de piezas y nudos

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

Análogamente, la estructura reticulada plana de la Figura 4 está en equilibrio global bajo la acción de las reacciones H, V y M . El equilibrio de fuerzas implica que las reacciones H y V sean iguales y opuestas a las correspondientes componentes de F y el equilibrio de momentos respecto al punto C implica que:

$$M = V * l - H * h$$

Ecuación 6: Equilibrio de momentos con respecto al punto C

Además, en la Figura 5 puede comprobarse que la satisfacción de las condiciones de equilibrio tanto de las barras AB y BC como del nudo B, considerados individualmente, implica que $M_b=V*L$, lo cual no se sigue necesariamente del equilibrio global.

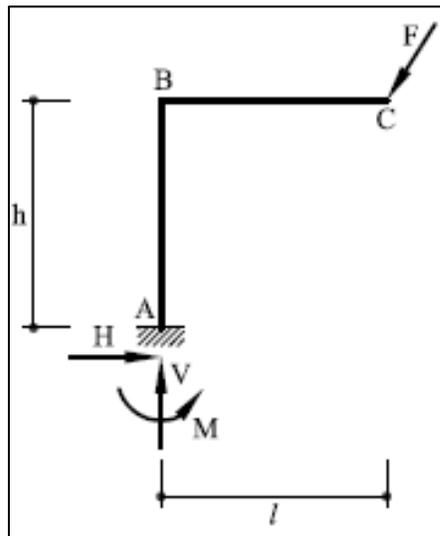


Figura 4: Equilibrio en una estructura reticulada: equilibrio global

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

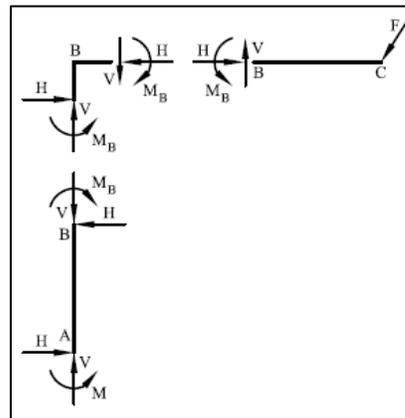


Figura 5: Equilibrio en una estructura reticulada: equilibrio de piezas y nudos

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

Pero las condiciones de equilibrio no son las únicas que se deben considerar en el comportamiento de las estructuras. De igual relevancia son las condiciones de compatibilidad sobre deformaciones y movimientos que deben satisfacerse en el proceso de deformación de la estructura bajo la acción de las cargas consideradas. Estas se traducen en que deben cumplirse los siguientes requisitos fundamentales:

- Condiciones de apoyo: la deformación debe ser tal que se cumplan las limitaciones de movimiento impuestas por los apoyos.
- Continuidad en los nudos: la deformación debe ser tal que los extremos de las diferentes barras que concurren en un nudo cumplan las limitaciones de movimiento impuestas por el correspondiente enlace.
- Continuidad en las barras: la deformación debe ser tal que se mantenga la continuidad de las piezas consideradas como elementos estructurales, es decir, que no se produzcan en ellas huecos ni solapamientos.

Estas condiciones de compatibilidad deben satisfacerse bajo cualquier hipótesis cinemática que se adopte y, en particular, bajo la hipótesis de pequeños movimientos.

Así, en la Figura 6 se muestra la deformada de la estructura articulada considerada anteriormente. Se observa que: (a) los puntos A y C son fijos, como corresponde a la condición de apoyo fijo especificada en ellos, pero las barras AB y BC pueden girar alrededor de dichos

puntos, respectivamente; (b) los desplazamientos del nudo B son únicos y, por lo tanto, idénticos para los correspondientes extremos de las barras que concurren en él; dado que el nudo B es una articulación, puede darse un giro relativo entre las barras AB y BC; (c) las barras AB y BC trabajan a tracción y compresión, respectivamente, y sufre, por tanto, un alargamiento y un acortamiento que pueden calcularse en función del valor de los axiles respectivos. Las consideraciones de compatibilidad anteriores determinan de forma única la deformación de la estructura.

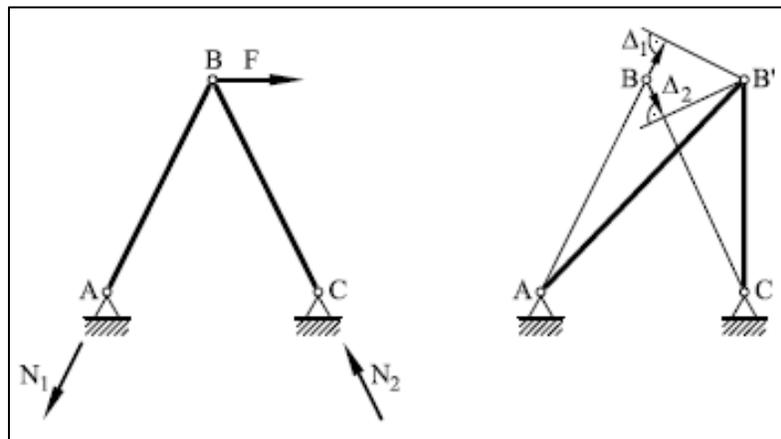


Figura 6: Movimientos en una estructura articulada

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

En la Figura 7 se muestra la deformada de la estructura reticulada considerada anteriormente. Se observa que: (a) los movimientos del punto A son nulos, desplazamientos y giro, como corresponde a la condición de empotramiento; (b) los movimientos del nudo B son únicos y, por lo tanto, idénticos para los correspondientes extremos de las barras que concurren en él; dado que el nudo B es rígido, no puede darse un giro relativo entre las barras AB y BC; (c) las barras AB y BC trabajan a flexión compuesta; sus rebanadas sufren unos acortamientos y

giros diferenciales de flexión que pueden calcularse en función del valor de los esfuerzos respectivos. Las consideraciones de compatibilidad anteriores determinan, de nuevo, la deformación de la estructura. (Mecánica de estructuras Libro 2 Métodos de análisis, 2004)

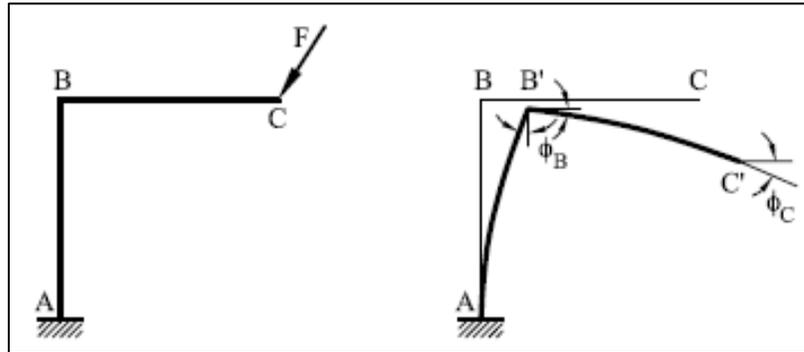


Figura 7: Movimientos en una estructura reticulada

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

1.6.2. Linealidad y principio de superposición

Además de los principios básicos de equilibrio y compatibilidad, gran parte de los métodos que se emplean en mecánica de estructuras se basan en el principio de linealidad. El problema de determinar los esfuerzos y movimientos que se producen en una estructura por acción de las cargas es lineal, esto es, la respuesta estructural es una función lineal de la sollicitación si se admiten dos hipótesis adicionales:

- Linealidad geométrica, es decir, que los movimientos (desplazamientos y giros) que se producen son pequeños. Se entiende que los desplazamientos son pequeños

comparados con las dimensiones geométricas de la estructura como espesor o luz; los giros (en radianes) son pequeños comparados con la unidad.

- Linealidad material, es decir, que la relación entre tensiones y deformaciones es elástica y lineal, o sea, los materiales de la estructura cumplen la ley de Hooke generalizada.

Es necesario precisar que ninguna de estas hipótesis son principios fundamentales, sino que se consideran porque su adopción se traduce en dos consecuencias importantes:

- Desde el punto de vista formal, en la garantía de que la solución del problema estructural, que satisface a la vez las condiciones de equilibrio y de compatibilidad, existe y es única, independientemente del procedimiento empleado para hallarla
- Desde el punto de vista metodológico, en la posibilidad de adoptar importantes simplificaciones en el planteamiento del problema estructura, tanto en la imposición de las condiciones de equilibrio como de compatibilidad.

La consecuencia directa de las hipótesis de linealidad es el principio de superposición, que establece que los efectos que un sistema de fuerzas origina sobre una estructura son iguales a la suma de los efectos que originan cada una de las fuerzas del sistema actuando por separado. Alternativamente, se puede enunciar diciendo que los efectos que un sistema de fuerzas origina

sobre una estructura no dependen del orden de aplicación de las fuerzas del sistema sobre la estructura.

Este principio, muy utilizado en mecánica de estructuras se ilustra en la Figura 8. Es fácil ver que, sin las hipótesis de linealidad geométrica y material, el principio de superposición no sería válido.

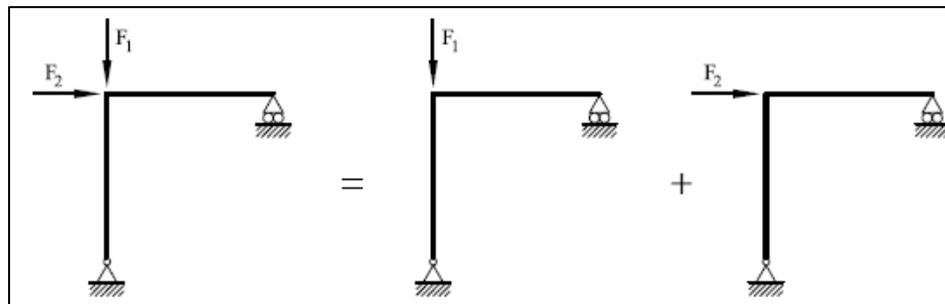


Figura 8: Principio de Superposición

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

1.6.3. Indeterminación estática. Grado de hiperestatismo

1.6.3.1. Concepto de indeterminación estática

Cuando es posible determinar (totalmente) las leyes que actúan sobre todas las barras que forman una estructura utilizando solamente consideraciones de equilibrio de fuerzas y de momentos, sobre la estructura en su globalidad o sobre sus partes integrantes, la estructura está estáticamente determinada y se llama isostática. En general, las estructuras de barras están

estáticamente indeterminadas. Se llaman entonces hiperestáticas y para resolverlas es necesario imponer, adicionalmente, condiciones de compatibilidad sobre sus movimientos.

Si la estructura es articulada, sus barras trabajan a esfuerzo axial y resolver la estructura consiste en hallar los valores de los axiles que actúan sobre las distintas barras. Si la estructura es isostática, esto podrá hacerse aplicando exclusivamente consideraciones de equilibrio de fuerzas en los nudos, si la estructura es hiperestática, será necesario considerar explícitamente los movimientos de estos que, a su vez, deberán ser compatibles con los alargamientos (o acortamientos) que sufran las diferentes barras concurrentes, por efecto de esfuerzo axial.

Si la estructura es reticulada, sus barras trabajan, en general, a flexión compuesta y torsión, y resolver, la estructura consiste en determinar las leyes de momentos flectores, esfuerzos cortantes y axiles y, en su caso, momentos torsores que actúan sobre las distintas barras. Si la estructura es isostática, esto podrá hacerse aplicando, exclusivamente consideraciones de equilibrio de fuerzas en los nudos. Si la estructura es hiperestática, será necesario considerar en la resolución los movimientos (desplazamientos y giros) de los nudos que, a su vez, deberán ser compatibles con las deformaciones que sufran las diferentes barras concurrentes en ellos. La multiplicidad de esfuerzos que actúan, en este tipo de estructura hace que este proceso sea más complejo que en las estructuras articuladas.

En la Figura 9 se muestran dos ejemplos de estructuras isostáticas. En ambos casos, las incógnitas que implican las reacciones exteriores para poder determinar los esfuerzos que actúan sobre una sección cualquiera de la estructura pueden hallarse utilizando solo la Ecuación 3, la Ecuación 3: Sumatoria de fuerzas en el eje X

y la Ecuación 4: Sumatoria de fuerzas en el eje Y

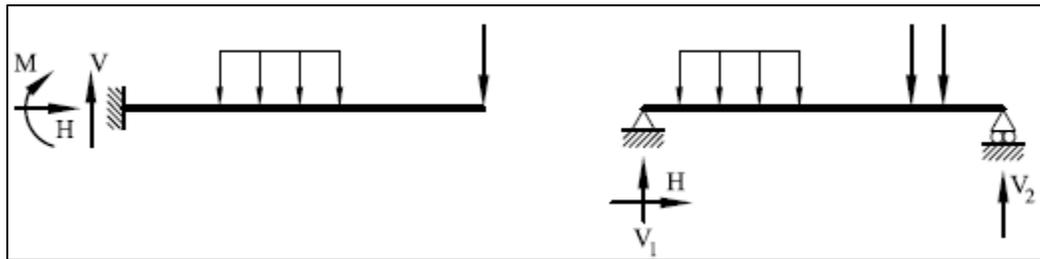


Figura 9: Estructuras Isostáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

Por el contrario, en la Figura 10 se muestran dos ejemplos de estructuras hiperestáticas. En ambos casos, las incógnitas que implican las reacciones exteriores no pueden determinarse utilizando solo la Ecuación 3, la Ecuación 3: Sumatoria de fuerzas en el eje X

y la Ecuación 4: Sumatoria de fuerzas en el eje Y

, ya que su número es superior al de ecuaciones.

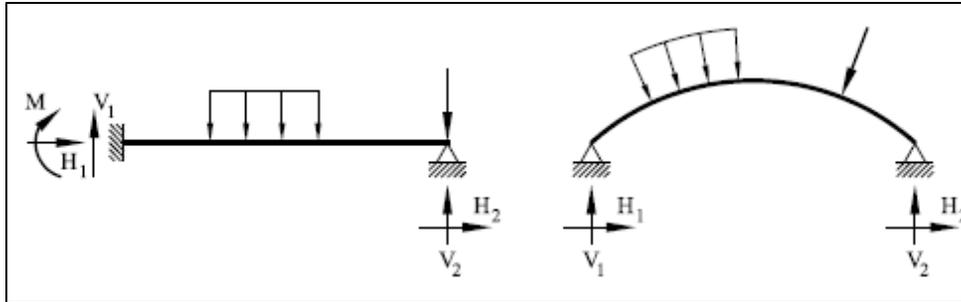


Figura 10: Estructuras hiperestáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

1.6.3.2. Hiperestatismo externo

En el caso de estructuras reticuladas formadas por una única pieza basta determinar las reacciones exteriores para poder calcular esfuerzos en cualquier sección de la pieza. En consecuencia, puede definirse el grado de indeterminación externa o grado de hiperestaticidad externo de la estructura como la diferencia entre el número de reacciones exteriores incógnita y el número de ecuaciones de la estática (seis en el espacio, tres en el plano), es decir:

$$h_{est} = n - e$$

Ecuación 7: Grado de hiperestatismo externo

Donde h_{est} es el grado de hiperestatismo externo, n es el número de reacciones incógnita y e es el número de ecuaciones de la estática.

El grado de hiperestatismo externo de las estructuras de la Figura 9 es $h_{est} = 0$, y esta es, por tanto, isostática. El grado de hiperestatismo externo de la estructura de la Figura 11 es $h_{est} = 1$, y se dice que esta es hiperestática de grado 1.

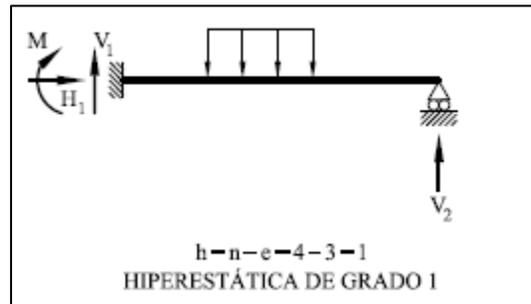


Figura 11: Estructura hiperestática de grado 1

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

El mismo concepto de hiperestatismo externo puede aplicarse a estructuras formadas por varias piezas; por ejemplo, el grado de hiperestatismo externo de la estructura de la Figura 12 es $h_{est} = 0$, y la estructura es isostática. Por el contrario, el grado de hiperestatismo externo de la estructura de la Figura 13 es $h_{est} = 3$

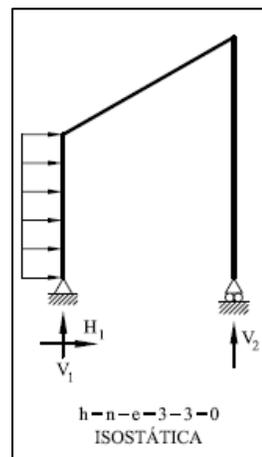


Figura 12: Estructura isostática de varias piezas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

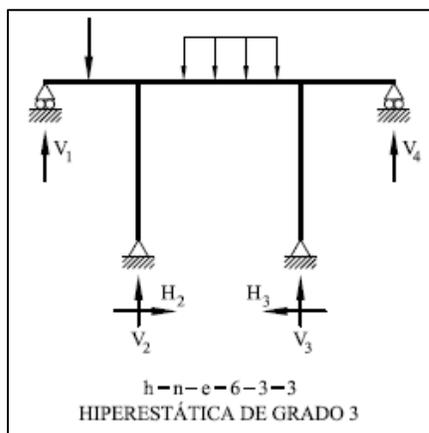


Figura 13: Estructura hiperestática de grado 3

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

1.6.3.3. Hiperestatismo interno

Las estructuras reticuladas formadas por varias piezas pueden ser externamente isostáticas y, sin embargo, ser estáticamente indeterminadas. Es decir que, a pesar de conocerse sus reacciones exteriores, es imposible determinar los esfuerzos en todas las secciones. Para poder resolverlas hay que suprimir enlaces internos y sustituirlos por sus reacciones internas equivalentes. Estas incógnitas adicionales que aparecen al liberar enlaces internos determina el grado de hiperestatismo interno. Así, en la Figura 14 se muestran dos ejemplos de estructuras que, si bien son externamente isostáticas, son, sin embargo, internamente hiperestáticas. Las dos estructuras de la Figura 15 son externa e internamente hiperestáticas.

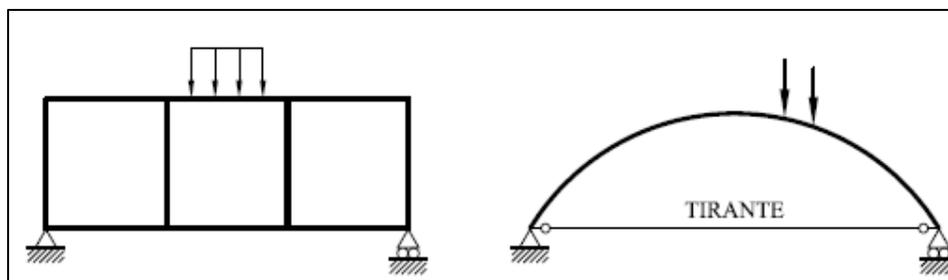


Figura 14: Estructuras exteriormente isostáticas - interiormente hiperestáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

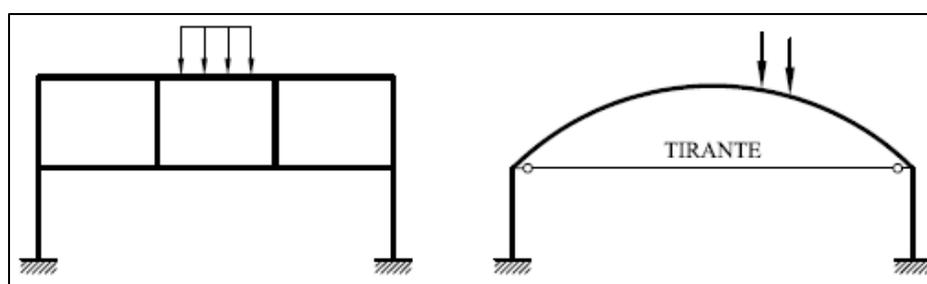


Figura 15: Estructuras exteriormente hiperestáticas - interiormente hiperestáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

Es, por tanto, necesario definir el grado de hiperestatismo de una estructura con independencia de si el origen de este es la indeterminación de las reacciones en los apoyos o en los enlaces internos.

1.6.3.4. Grado de hiperestatismo

De una forma general, se define el grado de indeterminación estática o grado de hiperestatismo de una estructura como el número mínimo de fuerzas (o momentos) que es necesario conocer para determinar completamente las leyes de esfuerzos. El grado de hiperestatismo se puede calcular como la diferencia:

$$h = n - p * e$$

Ecuación 8: Grado de hiperestatismo de una estructura

donde n es el número de reacciones (externas o internas) que hay que conocer para poder determinar los esfuerzos actuantes en cualquier sección de la estructura, e es el número de ecuaciones de la estática y p es el número de partes en que queda dividida la estructura al hacer los cortes necesarios para poder determinar los esfuerzos. Cada una de las partes resultantes al dividir la estructura debe ser isostática, y debe, por tanto, satisfacer independientemente las ecuaciones de equilibrio estático.

Una vez calculado el valor h , para: $h=0$ hay el mismo número de incógnitas que de ecuaciones, la estructura es isostática, $h>0$ hay más incógnitas que ecuaciones, la estructura es hiperestática de grado h , $h<0$ hay menos incógnitas que ecuaciones, la estructura es un mecanismo de grado h . (Cervera Ruiz & Blanco Diaz, 2004)

1.6.4. Análisis de estructuras isostáticas e hiperestáticas

A partir de los esfuerzos se pueden calcular las distribuciones de tensiones que actúan sobre las secciones de las diferentes piezas de las estructuras.

Los esfuerzos causan la deformación de las rebanadas y esto da lugar a la deformación de las piezas y, por ende, a los movimientos de la estructura.

Este procedimiento de resolución se muestra en el esquema de la

. Se observa que el proceso secuencial consiste en, a partir de la geometría de la estructura y del a definición de las acciones:

1. Calculo de reacciones, estableciendo las condiciones de equilibrio de la estructura,
2. Calculo de esfuerzos, a partir de condiciones de equilibrio de las piezas,
3. Calculo de la deformación de las piezas, a partir de las ecuaciones constitutivas (de comportamiento) de las barras
4. Calculo de movimientos en la estructura, a partir de condiciones de compatibilidad en apoyos y enlaces.

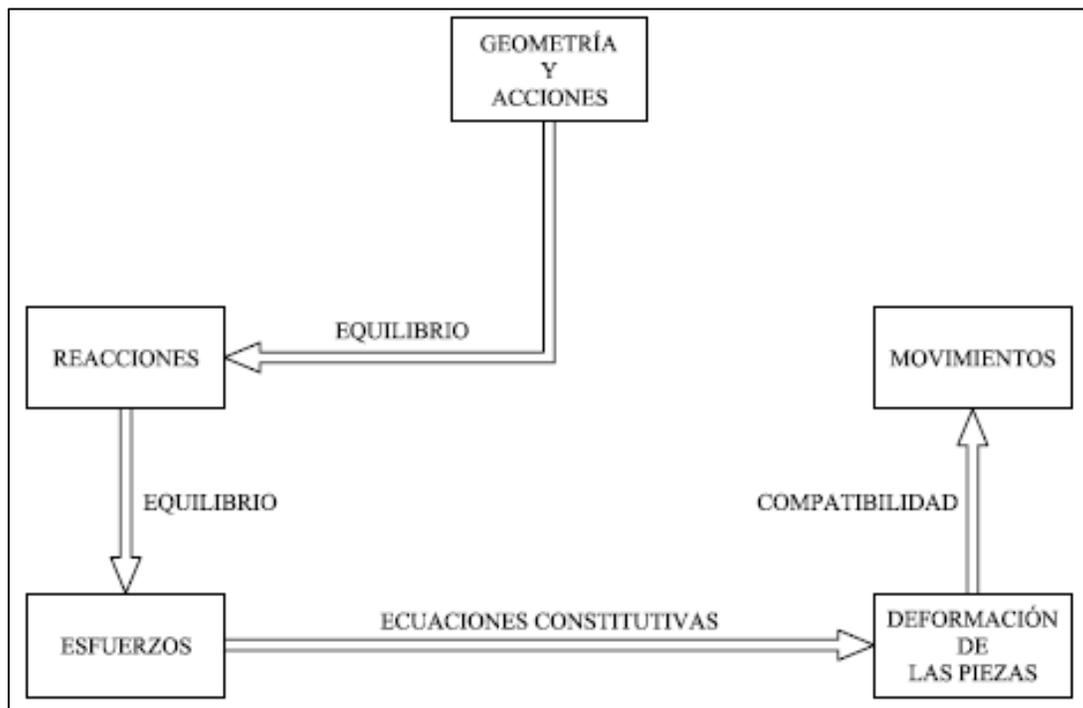


Figura 16: Esquema de resolución de estructuras isostáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

No es posible aplicar directamente este proceso a la resolución de estructuras hiperestáticas, en las que es necesario considerar conjuntamente las condiciones de equilibrio y compatibilidad, dado que el grado de indeterminación estática y cinemática que tienen estas estructuras. En el esquema de la Figura 17 se muestra como la imposibilidad de resolver a priori la indeterminación estática, o bien la cinemática, produce un “bucle” cerrado en el cual es imposible proceder de forma secuencial. (Cervera Ruiz & Blanco Diaz, 2004)

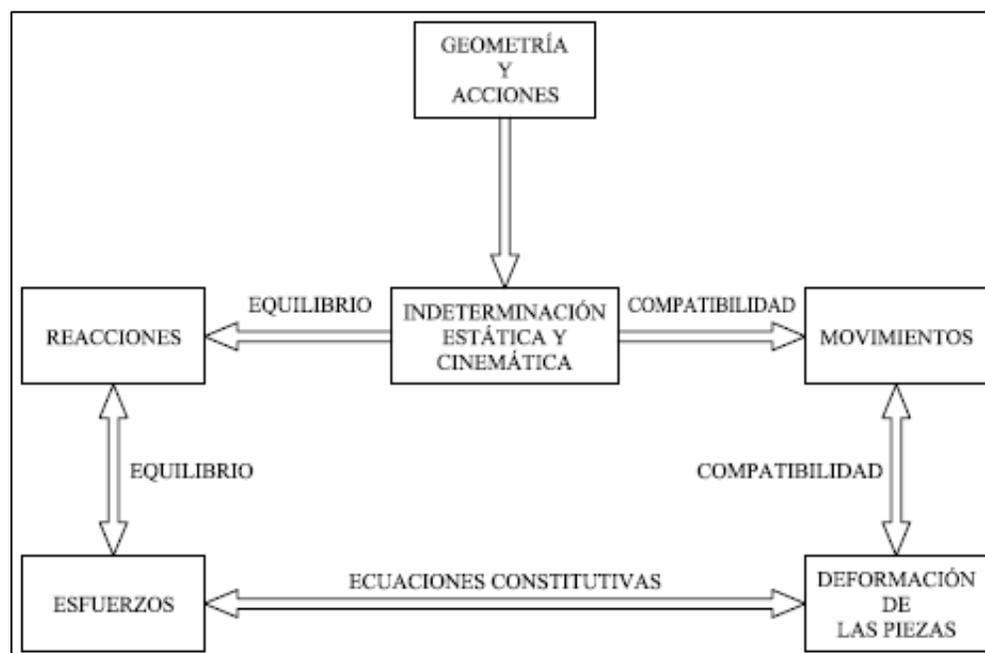


Figura 17: Esquema de resolución de estructuras hiperestáticas

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

1.6.5. Estructuras articuladas

Se denomina estructura articulada, al sistema, cuyo esquema de cálculo está compuesto por barras unidas entre sí por articulaciones o rótulas ideales. En la práctica constructiva, estas rótulas no se realizan y los nudos son rígidos (por ser más sencillo), pero la suposición de rótulas ideales en los nudos permite aminorar la dificultad del cálculo. Para ello, las cargas deben ser aplicadas en los nudos. En los esquemas de cálculo de las armaduras, las rótulas no se muestran, pero esto no implica que no existan.

La característica especial del estado esfuerzo-deformación de las armaduras, viene a estar dado por el trabajo de cada barra en tracción o compresión, por ello, su característica principal es la fuerza axial o normal, que originará alargamiento o acortamiento de la misma. (Villarreal Castro, Analisis estructural, 2009)

1.6.5.1. Estructuras articuladas isostáticas

Si la estructura articulada en cuestión es isostática, puede resolverse utilizando únicamente las ecuaciones de la estática. Esto quiere decir que el número de incógnitas del problema, o sea, el número de reacciones exteriores, n_r , más el número de axiles a determinar, n_b (igual al número de barras), es igual al número de ecuaciones de la estática que se pueden plantear, o sea, para un problema plano, el doble del número de nudos $2n_n$, ya que se pueden plantear 2 ecuaciones de equilibrio de fuerzas para cada nudo. Esto es:

$$nr + nb = 2 * nn$$

Ecuación 9: grado de indeterminación estática de una estructura articulada

En la Figura 18 se muestra una estructura articulada isostática ($nr=3$, $nb=5$, $nn=4$), mientras que la mostrada en la Figura 19 es hiperestática de grado 1 ($nr=3$, $nb=6$, $nn=4$).

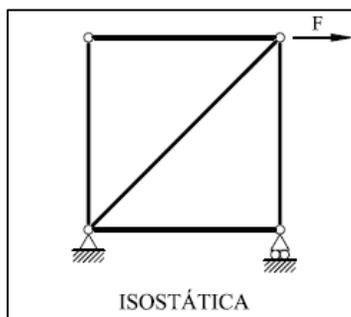


Figura 18: Estructura articulada isostática

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

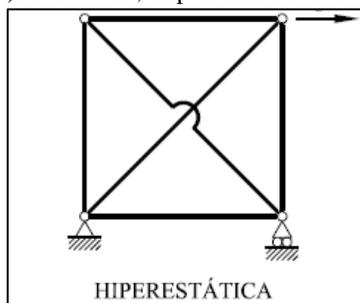


Figura 19: Estructura articulada hiperestática

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

Es importante destacar que la condición expresada en la Ecuación 9 es necesaria pero no suficiente para que la estructura sea isostática. Así, la “estructura” de la Figura 20 satisface la condición ($nr=3$, $nb=9$, $nn=6$), pero es, sin embargo, un mecanismo en su parte izquierda.

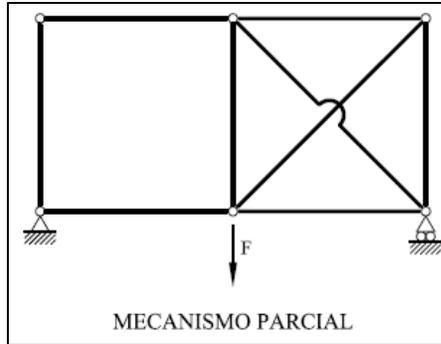


Figura 20: Estructura articulada con mecanismo parcial

Fuente: Cervera Ruiz, M., & Blanco Diaz, E. (2004). Mecánica de estructuras Libro 2 Métodos de análisis (Segunda ed.). Barcelona, España: Edicions UPC.

En cualquier caso, la condición expresada en la Ecuación 9 refleja el procedimiento para resolver estructuras isostáticas: siendo las incógnitas las n_r reacciones exteriores más los n_b axiles en las barras, se plantean en función de estas las necesarias ecuaciones de equilibrio de fuerzas en los nudos, $2n_n$ o $3n_n$ según el problema sea plano o espacial, respectivamente.

(Cervera Ruiz & Blanco Diaz, 2004)

Capítulo 2: LUA

2.1 Introducción a Lua

“Lua es un lenguaje de programación extensible diseñado para una programación procedimental general con utilidades para la descripción de datos. También ofrece un buen soporte para la programación orientada a objetos, programación funcional y programación orientada a datos. Se pretende que Lua sea usado como un lenguaje de script potente y ligero para cualquier programa que lo necesite. Lua está implementado como una biblioteca escrita en C limpio (esto es, en el subconjunto común de ANSI C y C++).

Siendo un lenguaje de extensión, Lua no tiene noción de programa principal (main): sólo funciona embebido en un cliente anfitrión, denominado programa contenedor o simplemente anfitrión (host). Éste puede invocar funciones para ejecutar un trozo de código Lua, puede escribir y leer variables de Lua y puede registrar funciones C para que sean llamadas por el código Lua. A través del uso de funciones C, Lua puede ser aumentado para abarcar un amplio rango de diferentes dominios, creando entonces lenguajes de programación personalizados que comparten el mismo marco sintáctico. La distribución de Lua incluye un programa anfitrión de muestra denominado Lua, que usa la biblioteca de Lua para ofrecer un intérprete de Lua completo e independiente.” (Lerusalimschy, de Figueiredo, & Celes, 2019)

2.2 El lenguaje

2.2.1 Convecciones léxicas

Los nombres (también llamados identificadores) en Lua pueden ser cualquier tira de caracteres (string) sólo con letras, dígitos y caracteres de subrayado (underscore), no comenzando por un dígito. Esto coincide con la definición de los nombres en la mayoría de los lenguajes. (La definición de letra depende de la implementación local actual a través del sistema locale: cualquier carácter considerado alfabético en el sistema local puede ser usado en un identificador.) Los identificadores se usan para nombrar variables y campos de tablas. Las siguientes palabras clave (keywords) están reservadas y no pueden usarse como nombres

<code>and</code>	<code>end</code>	<code>in</code>	<code>repeat</code>	<code>while</code>
<code>break</code>	<code>false</code>	<code>local</code>	<code>return</code>	
<code>do</code>	<code>for</code>	<code>nil</code>	<code>then</code>	
<code>else</code>	<code>function</code>	<code>not</code>	<code>true</code>	
<code>elseif</code>	<code>if</code>	<code>or</code>	<code>until</code>	

En Lua las letras mayúsculas y las minúsculas se consideran diferentes: `and` es una palabra reservada, pero `And` y `AND` son dos nombres diferentes válidos. Como convención, los nombres que comienzan por un subrayado seguido por letras en mayúsculas (como `_VERSION`) están reservados para uso como variables globales internas de Lua. Los siguientes strings denotan otros elementos:

<code>+</code>	<code>-</code>	<code>*</code>	<code>/</code>	<code>%</code>	<code>^</code>	<code>#</code>
----------------	----------------	----------------	----------------	----------------	----------------	----------------

==	>=	=	{	}	,	...
~=	<	(}	;	.	
<=	>)	[:	..	

Los strings literales pueden ser delimitados por comillas simples (apóstrofes) o dobles, y pueden contener las siguientes secuencias de escape de C:

\a' (pitido, bell)

\b' (retroceso, backspace)

\f' (salto de página, form feed)

\n' (nueva línea, newline)

\r' (retorno de carro, carriage return)

\t' (tabulador horizontal, horizontal tab)

\v' (tabulador vertical, vertical tab)

\|' (barra inversa, backslash)

\"' (comilla doble, quotation mark o double quote)

\'' (apóstrofe, apostrophe o single quote)

Además, una '\newline' (esto es, una barra inversa seguida por un salto de línea real) produce un salto de línea en el string. Un carácter en un string puede también especificarse por su valor numérico usando la secuencia de escape '\ddd', donde ddd es una secuencia de tres dígitos decimales. (Tenga presente que si la secuencia numérica de escape está seguida de un dígito debe ser expresada usando exactamente tres dígitos.) Los strings en Lua pueden contener cualquier valor de 8 bits, incluyendo el carácter cero, el cual puede ser especificado mediante '\0'.

Para poner una comilla (simple) doble, una barra inversa, un retorno de carro o un carácter cero dentro de un string literal encerrado por comillas (simples) dobles se debe usar una secuencia de escape. Cualquier otro carácter puede ser incluido en el literal. (Algunos caracteres de control pueden causar problemas con el sistema de ficheros, pero Lua no tiene problemas con ellos.)

Los strings literales pueden definirse usando un formato largo, encerrados en corchetes largos. Definimos un corchete largo de abrir de nivel n como un corchete de abrir seguido de n signos igual (=) seguidos de otro corchete de abrir. Así, un corchete largo de abrir de nivel 0 se escribe `[`, un corchete largo de abrir de nivel 1 se escribe `[=`, y así sucesivamente. Los corchetes largos de cerrar se definen de manera similar; por ejemplo, un corchete largo de cerrar de nivel 4 se expresa `]====`. Un string largo comienza en un corchete largo de abrir de cualquier nivel y termina en el primer corchete largo de cerrar del mismo nivel. Los strings literales delimitados de esta manera pueden extenderse por varias líneas, las secuencias de escape no son interpretadas y se ignoran los corchetes largos de cualquier otro nivel. Por tanto, pueden contener cualquier cosa excepto un corchete de cerrar del mismo nivel o caracteres cero.

Por conveniencia, cuando un corchete largo de abrir es seguido inmediatamente de un carácter de nueva línea, éste no es incluido en el string. Por ejemplo, usando el código de caracteres ASCII (en el cual 'a' se codifica como 97, el carácter de nueva línea se codifica como 10, y 'l' se codifica como 49), los cinco literales siguientes denotan el mismo string:

```

a = 'alo\n123"'
a = "alo\n123\""
a = '\971o\10\04923"'
a = [[alo
123"]]
a = [==[
alo
123"]==]

```

Las constantes numéricas pueden contener una parte decimal opcional y también un exponente opcional. Lua también acepta constantes enteras hexadecimales, escritas anteponiendo el prefijo 0x. Algunos ejemplos de constantes numéricas válidas son

```

3      3.0      3.1416      314.16e-2      0.31416E1      0xff      0x56

```

Los comentarios comienzan con un doble guion (--) en cualquier lugar fuera de un string. Si el texto que sigue inmediatamente después de -- no es un corchete largo de abrir el comentario es corto y llega hasta el final de línea. En otro caso tenemos un comentario largo, que alcanza hasta el correspondiente corchete largo de cerrar. Los comentarios largos se usan frecuentemente para deshabilitar temporalmente trozos de código

2.2.2 Variables

Las variables son lugares donde se almacenan valores. Existen tres tipos de variables en Lua: globales, locales y campos de tabla. Un único nombre puede denotar una variable local o una global (o un argumento formal de una función, el cual es una forma particular de una variable local):

```
var ::= nombre
```

Lua asume que las variables son globales, a no ser que sean declaradas explícitamente como locales. Las variables locales tienen un ámbito (scope) definido léxicamente: pueden ser accedidas libremente desde dentro de las funciones definidas en su mismo ámbito. Antes de la primera asignación el valor de una variable es nil. Los corchetes se usan para indexar una tabla:

```
var ::= prefixexp '[' exp ']'
```

La primera expresión (prefixexp) debe dar como resultado un valor tabla; la segunda expresión (exp) identifica una entrada específica en esta tabla. La expresión que denota la tabla que es indexada tienen una sintaxis restringida;

La sintaxis var.nombre es otra manera de expresar var[hombre] y se usa para denotar campos de tablas:

```
var ::= prefixexp '.' nombre
```

2.2.3. Sentencias

Lua soporta un conjunto casi convencional de sentencias, similar a los de Pascal o C. Este conjunto incluye la asignación, estructuras de control de flujo, llamadas a funciones, constructores de tablas y declaraciones de variables.

2.2.3.1 Chunks

La unidad de ejecución en Lua se denomina chunk, el cual es simplemente un conjunto de sentencias que se ejecutan secuencialmente. Cada sentencia puede llevar opcionalmente al final un punto y coma:

```
chunk ::= {sentencia [';']}
```

No existen sentencias vacías en Lua y por tanto ';' no es legal. Lua maneja cada chunk como el cuerpo de una función anónima con un número variable de argumentos. Los chunks pueden definir variables locales, recibir argumentos y retornar valores.

Un chunk puede ser almacenado en un fichero o en un string dentro de un programa anfitrión. Cuando se ejecuta un chunk primero se precompila, creándose instrucciones para una máquina virtual, y es entonces cuando el código compilado es ejecutado por un intérprete de la máquina virtual.

Los chunks pueden también estar precompilados en forma binaria. Las formas fuente y compilada de los programas son intercambiables; Lua detecta automáticamente el tipo de fichero y actúa de manera acorde.

2.2.3.2. Bloques

Un bloque es una lista de sentencias; sintácticamente un bloque es lo mismo que un chunk:

bloque ::= chunk

Un bloque puede ser delimitado explícitamente para producir una sentencia simple:

*sentencia ::= **do** bloque **end***

*Los bloques explícitos son útiles para controlar el ámbito de las declaraciones de variable. También se utilizan a veces para añadir sentencias **return** o **break** en medio de otro bloque*

2.2.3.3. La asignación

Lua permite asignaciones múltiples. Por tanto, la sintaxis de una asignación define una lista de variables a la izquierda y una lista de expresiones a la derecha. Los elementos de ambas listas están separados por comas:

```

sentencia ::= varlist '=' explist
varlist  ::= var {',' var}
explist  ::= exp {',' exp}

```

Antes de una asignación la lista de expresiones se ajusta a la longitud de la lista de variables. Si existen más valores de los necesarios el exceso se descarta. Si existen menos valores de los necesarios la lista se extiende con tantos valores nil como se necesiten. Si la lista de expresiones finaliza con una llamada a una función entonces todos los valores devueltos en la llamada pueden entrar en la lista de valores antes del ajuste (excepto cuando se encierra entre paréntesis).

La sentencia de asignación primero evalúa todas sus expresiones y sólo después se hace la asignación. Entonces, el código

```

i = 3
i, b[i] = i+1, 20

```

Asigna 20 a b[3], sin afectar a b[4] debido a que i en b[i] se evalúa (a 3) antes de que se le asigne el valor 4. Similarmente, la línea mostrada a continuación intercambia los valores de x e y.

```

x, y = y, x

```

2.2.3.4. Estructuras de control

Las estructuras de control if, while y repeat tienen el significado habitual y la sintaxis familiar:

```
sentencia ::= while exp do bloque end
```

```
sentencia ::= repeat bloque until exp
```

```
sentencia ::= if exp then bloque {elseif exp then bloque} [else  
bloque] end
```

La condición de una expresión de una estructura de control puede retornar cualquier valor. Tanto false como nil se consideran falsos. Todos los valores diferentes de nil y false se consideran verdaderos (en particular, el número 0 y el string vacío son también verdaderos).

En el bucle repeat–until el bloque interno no acaba en la palabra clave until sino detrás de la condición. De esta manera la condición puede referirse a variables locales declaradas dentro del bloque del bucle.

La orden return se usa para devolver valores desde una función o un chunk (el cual es justamente una función). Las funciones y los chunks pueden retornar más de un valor, por lo que la sintaxis para return es

```
sentencia ::= return [explist]
```

La orden `break` se usa para terminar la ejecución de los bucles `while`, `repeat` y `for`, saltando a la sentencia que sigue después del bucle, un `break` finaliza el bucle más interno que esté activo.

`sentencia ::= break`

Las órdenes `return` y `break` pueden aparecer sólo como última sentencia dentro de un bloque. Si se necesita realmente un `return` o un `break` en medio de un bloque se debe usar un bloque más interno explícitamente, como en '`do return end`' y '`do break end`', debido a que así `return` y `break` son las últimas sentencias en su propio bloque.

2.2.3.5. La sentencia `for`

La sentencia `for` tiene dos formas: una numérica y otra genérica. La forma numérica del bucle `for` repite un bloque mientras una variable de control sigue una progresión aritmética.

Tiene la sintaxis siguiente:

```
sentencia ::= for nombre '=' exp1 ',' exp2 [',' exp3] do bloque
end
```

El bloque se repite para los valores de nombre comenzando en `exp1` hasta que sobrepasa `exp2` usando como paso `exp3`.

La sentencia for genérica trabaja con funciones, denominadas iteradores. En cada iteración se invoca a la función iterador que produce un nuevo valor, parándose la iteración cuando el nuevo valor es nil. El bucle for genérico tiene la siguiente sintaxis:

```
sentencia ::= for lista_de_nombres in explist do bloque end
lista_de_nombres ::= nombre {',' nombre}
```

2.2.3.6. Declaraciones locales

Las variables locales pueden ser declaradas en cualquier lugar dentro de un bloque. Esas declaraciones pueden incluir una asignación inicial:

```
sentencia ::= local lista_de_nombres ['=' explist]
```

Si está presente, una asignación inicial tiene la misma semántica que una asignación múltiple. En otro caso todas las variables son inicializadas con nil. Un chunk es también un bloque, así que las variables locales pueden ser declaradas en un chunk fuera de cualquier bloque explícito. El ámbito de esas variables se extiende hasta el final del chunk.

2.2.4. Expresiones

2.2.4.1. Operadores aritméticos

Lua tiene los operadores aritméticos comunes: los binarios + (adición), - (substracción), * (multiplicación), / (división), % (módulo) y ^ (exponenciación); y el unario - (negación). Si los operandos son números o strings que se convierten a números, entonces todas las operaciones tienen el significado corriente. El exponenciación trabaja con cualquier exponente. Por ejemplo, $x^{-0.5}$ calcula la inversa de la raíz cuadrada de x . El módulo se define como:

```
a % b == a - math.floor(a/b) * b
```

Esto es, es el resto de la división que redondea el cociente hacia menos infinito.

2.2.4.2. Operadores relacionales

Devuelven siempre un resultado false o true. Los operadores relacionales en Lua son

```
==      ~=      <      >      <=      >=
```

La igualdad (==) primero compara el tipo de los operandos. Si son diferentes entonces el resultado es false. En otro caso se comparan los valores de los operandos. Los números y los strings se comparan de la manera usual. Los objetos (tablas, userdata, procesos y funciones) se comparan por referencia: dos objetos se consideran iguales sólo si son el mismo objeto. Cada vez que se crea un nuevo objeto (una tabla, userdata, proceso o función) este nuevo objeto es diferente de todos los demás objetos preexistentes. El operador ~= es exactamente la negación de la igualdad (==).

Las reglas de conversión no se aplican en las comparaciones de igualdad. De este modo `0==0` es false, y `t[0]` y `t[0]` denotan diferentes entradas en una tabla.

El orden de los operadores funciona de la siguiente manera. Si ambos argumentos son números entonces se comparan como tales. En otro caso, si ambos argumentos son strings sus valores se comparan de acuerdo al sistema local.

2.2.4.3. Operadores lógicos

Los operadores lógicos en Lua son `and`, `or` y `not`. Como las estructuras de control todos los operadores lógicos consideran false y nil como falso y todo lo demás como verdadero.

El operador negación `not` siempre retorna false o true. El operador conjunción `and` retorna su primer operando si su valor es false o nil; en caso contrario `and` retorna su segundo operando. El operador disyunción `or` retorna su primer operando si su valor es diferente de nil y false; en caso contrario `or` retorna su segundo argumento. Tanto `and` como `or` usan evaluación de cortocircuito; esto es, su segundo operando se evalúa sólo si es necesario. He aquí varios ejemplos, donde '-->' indica el resultado de la expresión precedente.

```
10 or 20           --> 10
10 or error()     --> 10
nil or "a"        --> "a"
nil and 10        --> nil
false and error() --> false
false and nil     --> false
false or nil      --> nil
```

10 and 20

--> 20

2.2.4.4. Concatenación

El operador de concatenación de strings en Lua se denota mediante dos puntos seguidos ('..'). Si ambos operandos son strings o números entonces se convierten a strings.

2.2.4.5. El operador longitud

El operador longitud se denota mediante #. La longitud de un string es su número de bytes (significado normal de la longitud de un string cuando cada carácter ocupa un byte).

La longitud de una tabla t se define como un índice entero n tal que $t[n]$ no es nil y $t[n+1]$ es nil; además, si $t[1]$ es nil entonces n puede ser cero. Para un array regular, con valores no nil desde 1 hasta un n dado, la longitud es exactamente n , el índice es su último valor. Si el array tiene "agujeros" (esto es, valores nil entre otros valores que no lo son), entonces $\#t$ puede ser cualquiera de los índices que preceden a un valor nil (esto es, Lua puede considerar ese valor nil como el final del array).

2.2.4.6. Precedencia de los operadores

La precedencia de los operadores en Lua sigue lo expuesto a continuación de menor a mayor prioridad:

```

or
and
<      >      <=     >=     ~=     ==
..
+      -
*      /      %
not    #      - (unario)
^

```

Como de costumbre, el uso de paréntesis puede ser usado para cambiar las precedencias de una expresión. Los operadores de concatenación ('.') y exponenciación ('^') son asociativos correctos. Todos los demás operadores binarios quedan asociativos.” (Lerusalimschy, de Figueiredo, & Celes, 2019)

Capítulo 3: Diseño del software

3.1 Especificación de requerimientos

Ian Sommerville en su libro: Ingeniería de Software define los requerimientos de software como: “Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información.” (p. 83)

Por este motivo se definen a continuación los requerimientos del software de análisis estructural propuesto; En la Tabla 1 se muestran los requerimientos básicos que refieren al desplazamiento del usuario a través de las diferentes funcionalidades del sistema a desarrollar.

Tabla 1: Requerimientos Control de las Ventanas

Ac-Struc/ Software de Control/ Control Ventana	
Función	Muestra la ventana solicitada por el usuario.
Descripción	La función controla la visualización de ventanas en pantalla, mostrando únicamente aquella que ha sido solicitada por el usuario.
Entradas	Nombre de la ventana solicitada
Fuente	Menú
Salidas	Ventana Solicitada
Acción	Almacena en una variable global el nombre de la ventana solicitada por el usuario y posteriormente muestra en pantalla la ventana correspondiente

En la Tabla 2 se muestran los requerimientos relacionados con la funcionalidad de cálculo de estructuras isostáticas e hiperestáticas haciendo uso del método matricial de rigidez, el cual es una de las funcionalidades más importantes que deben ser incluidas en el software, ya que este es uno de los métodos principales para el análisis matricial de estructuras

Tabla 2: Requerimientos calculo método matricial de rigidez

Ac-Struc/ Software de Cálculo/ Calculo Matriz Rigidez	
Función	Analizar la estructura usando el método matricial de rigidez
Descripción	La función analiza la estructura isostática o hiperestática dada por el usuario y muestra el resultado de los cálculos realizados
Entradas	Materiales, posición de los nudos en el espacio, elementos, apoyos, cargas aplicadas en la estructura, combinaciones de carga y variación de temperatura
Fuente	Formularios de ingreso de datos
Salidas	Deformación de la estructura, reacciones, diagramas de fuerza cortante, diagramas de momento flector, matrices de rigidez de los elementos
Acción	<p>Crear matrices de rigidez de cada elemento para posteriormente crear matriz de rigidez de la estructura.</p> <p>Calcular los vectores de empotramiento, reacciones y deformaciones.</p> <p>Encontrar los datos desconocidos.</p> <p>Calcular las ecuaciones de fuerza cortante y momento flector</p> <p>Almacenar todos los datos en variables globales</p>

En la Tabla 3 se muestran los requerimientos relacionados con la funcionalidad de cálculo de armaduras isostáticas haciendo uso del método matricial de nudos, el cual es el método principal que usara el software para el análisis de estructuras articuladas.

Tabla 3: Requerimientos calculo método de nudos

Ac-Struc/ Software de Cálculo/ Calculo Armadura isostática	
Función	Analizar armadura por el método de nudos
Descripción	Analiza una armadura isostática haciendo uso del método de nudos
Entradas	Posición de los nudos en el espacio, elementos, apoyos, cargas aplicadas en la estructura y combinaciones de carga
Fuente	Formularios de ingreso de datos
Salidas	Reacciones en los apoyos y fuerzas internas de los elementos
Acción	Crear las matrices de la estructura y resolver las incógnitas
Precondición	Evaluar la posibilidad de analizar la estructura de manera matricial

En la Tabla 4 se definen los requerimientos relacionados con la funcionalidad de guardar un archivo, esta permite al usuario abrir archivos almacenados con anterioridad y de esta forma disminuir el tiempo requerido introduciendo los datos de una estructura analizada con anterioridad.

Tabla 4: Requerimientos de funcionalidad abrir archivo

Ac-Struc/ Software de Almacenamiento/ Abrir Archivo	
Función	Abrir archivo
Descripción	La función abre un archivo almacenado el cual contiene la información de la estructura.
Entradas	Información codificada de la estructura
Fuente	Memoria interna
Salidas	Información decodificada de la estructura
Acción	Extraer los datos desde la memoria interna del dispositivo y decodificarlos para extraer la información almacenada para posteriormente mostrar en el dispositivo la estructura solicitada
Precondición	Ninguna

En la Tabla 5 se definen los requerimientos relacionados con la funcionalidad guardar, la cual permite al usuario almacenar la estructura analizada para de esta forma, si es necesario, verificar los resultados en otro instante de tiempo.

Tabla 5: Requerimientos de funcionalidad guardar archivo

Ac-Struc/ Software de Almacenamiento/ Guardar Archivo	
Función	Guardar Archivo
Descripción	La función guarda un archivo en la memoria interna del dispositivo
Entradas	Información de la estructura
Fuente	Variables globales
Salidas	Información codificada de la estructura
Acción	Lee las variables globales y codificarlas para ser almacenadas en la memoria interna del dispositivo
Precondición	Ejecutar el análisis de la estructura con anterioridad para verificar la validez de los datos

3.2 Interfaz de usuario

“La Interfaz Gráfica de Usuario (GUI por su nombre en inglés, Graphical User Interface) es parte fundamental de cualquier aplicación; al comenzar a trabajar con una computadora el usuario comienza a interactuar con la Interfaz, ya sea la del sistema operativo, la de un software en particular o la de cualquier sitio web. Es donde comienza la interacción hombre computadora. El diseño de la GUI no se lo debe considerar como una tarea secundaria y sin importancia; por el contrario, el equipo de desarrollo debe contar con integrantes especializados en el tema.

En ocasiones, a partir de la GUI se puede determinar si una aplicación será utilizada o no para resolver los problemas para los cuales fue diseñada. En el actual mundo informatizado se debe ofrecer al usuario una Interfaz que lo ayude a concretar las tareas de manera rápida, sencilla y satisfactoria. Es la Interfaz la responsable de ofrecer una interacción fluida y agradable.” (Albornoz, Berón, & Montejano)

3.2.1 Diseño de la interfaz de usuario

“Un diseño cuidadoso de la interfaz de usuario es parte fundamental del proceso de diseño general del software. Si un sistema software debe alcanzar su potencial máximo, es fundamental que su interfaz de usuario sea diseñada para ajustarse a las habilidades, experiencia y expectativas de sus usuarios previstos. Un buen diseño de la interfaz de usuario es crítico para la confiabilidad del sistema. Muchos de los llamados “errores de usuario” son causados por el hecho de que las interfaces de usuario no consideran las habilidades de los usuarios reales y su entorno de trabajo. Una interfaz de usuario mal diseñada significa que los usuarios probablemente no podrán acceder a algunas características del sistema, cometerán errores y sentirán que el sistema les dificulta en vez de ayudarlos a conseguir cualquier objetivo para el que utilizan el sistema.

Cuando se toman decisiones en el diseño de las interfaces de usuario, deben tenerse en cuenta las capacidades físicas y mentales de las personas que utilizaran el software. Algunos factores importantes que se deben considerarse son los siguientes:” (Sommerville, Ingeniería de Software, 2005)

1. *“Las personas tienen una memoria limitada a corto plazo; podemos recordar instantáneamente alrededor de siete elementos de información” (Miller, 1955). “Por lo tanto, si a los usuarios se les presenta demasiada información al mismo tiempo, es posible que no puedan asimilarla.*

2. *Todos cometemos errores, especialmente cuando tenemos que manejar demasiada información o estamos estresados. Cuando los sistemas fallan y emiten mensajes de aviso y alarmas, a menudo aumentan el estrés de los usuarios, incrementando así la posibilidad de que cometa errores.*

3. *Poseemos un amplio rango de capacidades físicas. Unas personas ven y escuchan mejor que otras, otras son daltónicas, y otras son mejores en manipulaciones físicas. No se debe diseñar para las propias capacidades y suponer que todos los otros usuarios serán capaces de adaptarse.*

4. *Tenemos diferentes preferencias de interacción. A algunas personas les gusta trabajar con imágenes, a otras con texto. La manipulación directa es natural para algunas personas, pero otras prefieren un estilo de interacción basado en emitir comandos al sistema*

Estos factores humanos son la base para los principios de diseño que se muestran en la Tabla 6. Estos principios generales se aplican a todos los diseños de interfaces de usuario y normalmente se deben instalar como directrices de diseño más detalladas para organizaciones o tipos de sistema específicos.” (Sommerville, Ingeniería de Software, 2005)

Tabla 6: Principios de diseño de las interfaces de usuario

Principio	Descripción
-----------	-------------

Principio	Descripción
Familiaridad del usuario	La interfaz debe utilizar términos y conceptos obtenidos de la experiencia de las personas que más utilizan el sistema.
Uniformidad	Siempre que sea posible, la interfaz debe ser uniforme en el sentido de que las operaciones comparables se activen de la misma forma.
Mínima sorpresa	El comportamiento del sistema no debe provocar sorpresa a los usuarios.
Recuperabilidad	La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores.
Guía de usuario	Cuando ocurran errores, la interfaz debe proporcionar retroalimentación significativa y características de ayuda sensible al contexto.
Diversidad de usuario	La interfaz debe proporcionar características de interacción apropiadas para los diferentes tipos de usuarios del sistema

Fuente: Sommerville, I. (2005). *Ingeniería de Software* (Septima ed.). Madrid, España: Pearson Educacion. (p. 330)

3.2.2. Color

El software está diseñado para ser usado principalmente por estudiantes y docentes de la Universidad de Pamplona, por tal motivo, los colores seleccionados son el color blanco, el color negro y los colores corporativos de la Universidad de Pamplona, los cuales se muestran en la Figura 21.

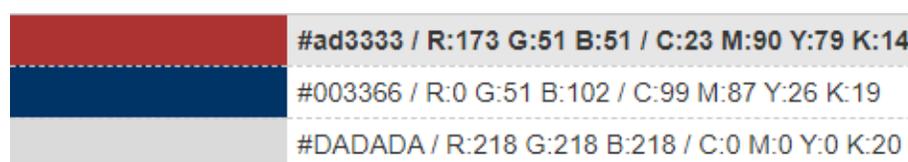


Figura 21: Colores corporativos de la Universidad de Pamplona

Fuente: Universidad de Pamplona. (s.f.). Recuperado el 19 de noviembre de 2019, de Imagen Corporativa Universidad de Pamplona: http://www.unipamplona.edu.co/unipamplona/portallG/home_1/recursos/corporativo/15022011/descargas_unipamplona.jsp

3.2.3. Menú de opciones

El menú de opciones permite al usuario desplazarse a través de las diferentes funcionalidades disponibles en el software. El desplazamiento entre las diferentes ventanas solo se puede realizar por medio de este menú ya que de esta manera el usuario puede acceder fácilmente a cualquier opción que desee. El menú está diseñado para que el ingreso de datos se realice de manera secuencial, facilitando así su uso. En la Figura 22 se muestra el diseño del menú de opciones.

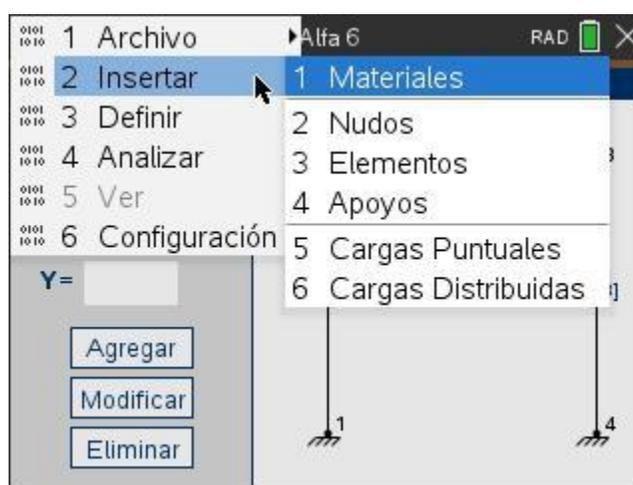


Figura 22: Diseño menú de opciones

3.2.4. Interfaz de entrada y salida de datos

La interfaz de usuario posee un diseño simplificado el cual permite a los usuarios entender claramente los datos solicitados por el software o los resultados mostrados en pantalla, en la Figura 23 se muestra el diseño general del formulario de entrada de datos del software.



Figura 23: Diseño general para la entrada de datos

3.3. Modelado del sistema

“Los modelos se usan durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema. El aspecto más importante de un modelo del sistema es que deja fuera los detalles.(...) Un modelo es una abstracción del sistema a estudiar, y no una representación alternativa de dicho sistema.” (Sommerville, Ingeniería de Software, 2011)

“Los modelos dirigidos por datos muestran la secuencia de acciones involucradas en el procesamiento de datos de entrada, así como la generación de una salida asociada. Son particularmente útiles durante el análisis de requerimientos, pues sirven para mostrar procesamiento “extremo a extremo” en un sistema. Esto es, exhiben toda la secuencia de acciones que ocurren desde una entrada a procesar hasta la salida correspondiente, que es la respuesta del sistema.” (Sommerville, Ingeniería de Software, 2011)

En la Figura 24 se muestra el diagrama de flujo que representa gráficamente el funcionamiento general del software el cual muestra de manera simplificada la forma en la que el software interactúa con el usuario, en el diagrama no se detallan procesos complejos como lo son la traducción de los datos introducidos por el usuario, ya que estos no son legibles directamente por los algoritmos de cálculo, de la misma forma, no se detalla el cálculo de las deformaciones de la estructura.

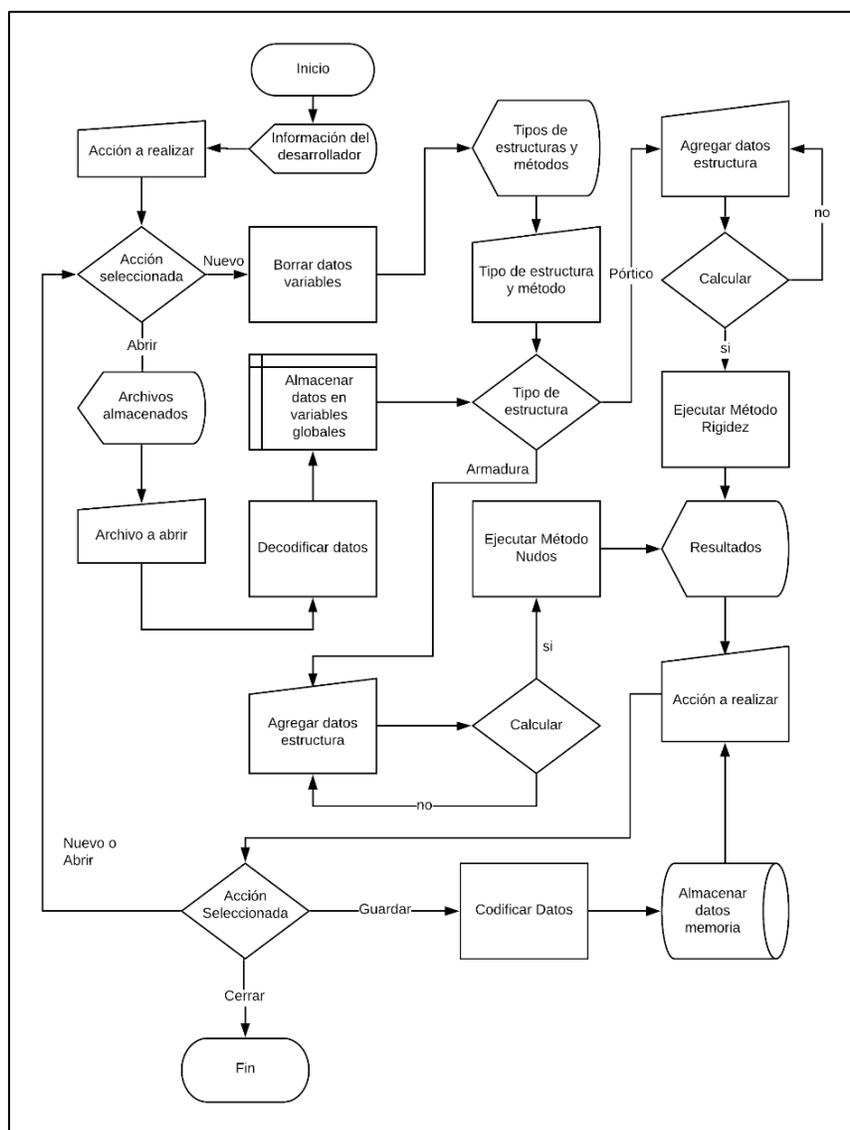


Figura 24: Diagrama de flujo funcionamiento general del software

En la

se muestra el diagrama de flujo que representa gráficamente el procedimiento para el análisis de estructuras isostáticas e hiperestáticas usando el método matricial de rigidez, en este diagrama aborda el método de una manera superficial, por lo que no se detalla el procedimiento de ningún proceso de cálculo.

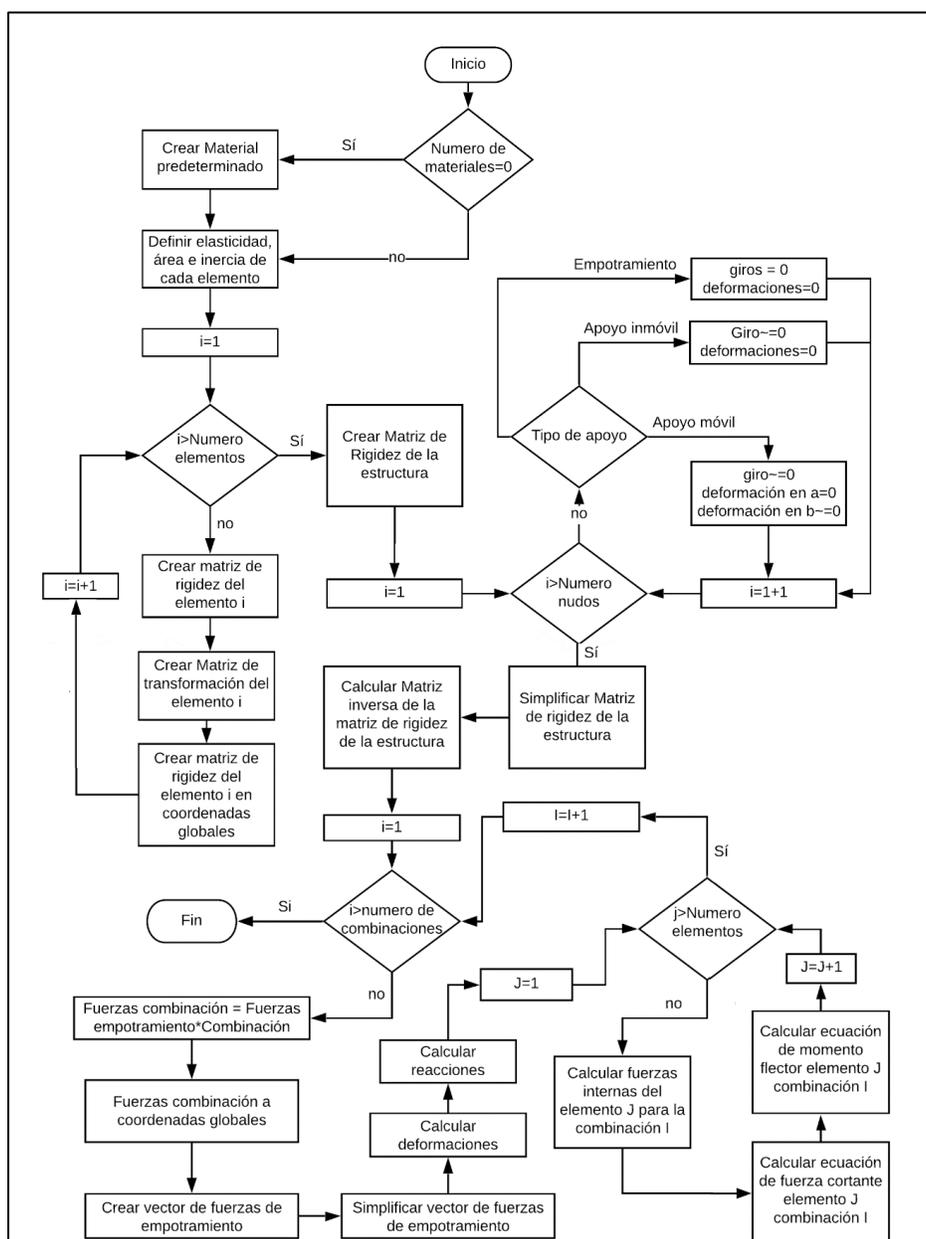


Figura 25: Diagrama de flujo del método matricial de rigidez

Un paso importante en la etapa el diseño de un software es el análisis de funciones que desempeñaran una tarea importante, en la Figura 26 se muestra el diagrama de flujo detallado de la función que realizara la tarea de multiplicar dos matrices, procedimiento que es esencial para el análisis matricial de estructuras.

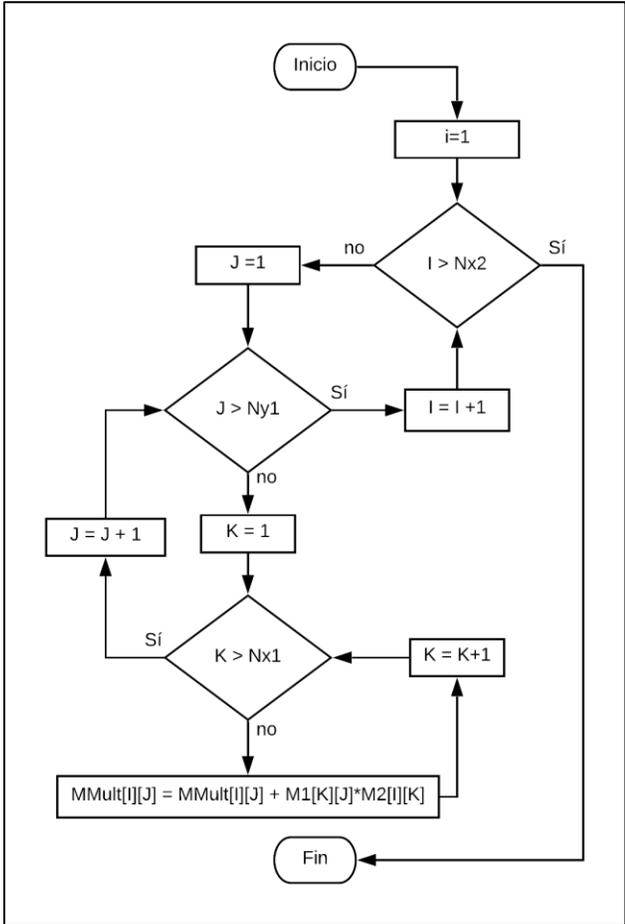


Figura 26: Diagrama de flujo detallado de la función multiplicar matrices

Dónde: N_{x1} corresponde al número de columnas de la Matriz 1, N_{y1} equivale al número de filas de la Matriz 1, N_{x2} es el número de columnas de la Matriz 2, M_1 y M_2 son las matrices las cuales serán multiplicadas y $MMult$ es el resultado de operar M_1 y M_2

Capítulo 4: Implementación del software

4.1. Implementación de scripts en Lua

“El editor de scripts le permite crear y entregar simulaciones vinculadas dinámicamente, utilidades potentes y flexibles, y otros contenidos educativos para explorar conceptos matemáticos y científicos. Cuando un documento que contiene un script es abierto, el script se ejecuta automáticamente según lo programado.”

La ventana del editor de script se abre cuando una nueva aplicación de script es insertada o cuando aplicación de script existente es editada en un documento de TI-Nspire o Publish View” (Texas Instruments). En la Figura 27 se muestra la ventana del editor de scripts

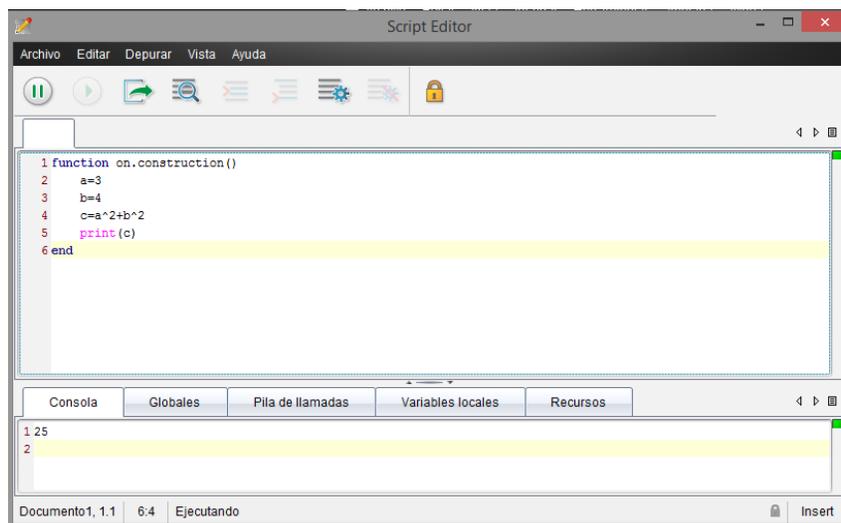


Figura 27: Interface del editor de scripts

“Para insertar un nuevo script es necesario seguir los siguientes pasos:

- 1. Abrir el documento donde se desea insertar el script*
- 2. En la barra de menú presionar insertar> Editor de Scripts> Insertar Script*
- 3. Digitar el título del script (el número máximo de caracteres es 32)*
- 4. Escribir el texto en las líneas del script*
- 5. Cuando el script este completo, presionar set script para ejecutarlo.”*
(Texas Instruments)

4.2. Funciones implementadas

Para una mejor comprensión del software cada una de las funciones implementadas fueron agrupadas en cuatro tipos principales: Funciones de inicialización, Funciones de control, Funciones de Cálculo y Funciones de Lectura y almacenamiento de datos. Cada una tiene una función diferente, las cuales, serán definidas a continuación, además se mostrarán algunos ejemplos de cada clasificación usando funciones reales implementadas en el software planteado.

4.2.1. Funciones de inicialización

Las funciones de inicialización se definen como aquellas que como su nombre lo indica: inician procesos esenciales como la creación de variables y parámetros del sistema.

4.2.1.1. Función *on.construction*

Esta función tiene como objetivo establecer valores iniciales a variables las cuales son fundamentales para la inicialización correcta del software, esta función se ejecuta una única vez tras iniciar el software.(Figura 28)

```

89 function on.construction()
90     platform.window.setBackgroundColor(0xDADADA)
91     Logo_Upa=image.new(_R.IMG.logo_upa)
92     Ventana="Inicio"
93     Seleccion={}
94     Seleccion.Abrir=1
95     Seleccion.PreEstructura=1
96     Seleccion.Estructura=nil
97     Seleccion.Metodo=1
98     Alfabeto={"", "", "", "", "", "", "", "", "", "", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0"
99     , "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t"
100     , "u", "v", "w", "x", "y", "z", " "}
101     Menu_Nuevo()
102 end

```

Figura 28:Fragmento de Código: Función *on.construction()*

4.2.1.2. Función *Nuevo Armadura*

Esta función se encarga de establecer las variables iniciales que son necesarias para analizar matricialmente una armadura isostática.(Figura 29)

```

137 function Nuevo_Armadura()
138     Menu_Armaduras()
139     Save=false
140     Tipo_Archivo=2
141     Ventana="Insertar Apoyos"
142     AVentana=platform.window:width()
143     Tipo_Carga={"Unitaria","Viva","Muerta","Sismica","Viento","Auxiliar"}
144     Caja={}
145     for i=1,8 do
146         Caja[i]=D2Editor.newRichText()
147     end

```

Figura 29: Fragmento de Código: Función Nuevo Armadura

4.2.1.3. Función Nuevo Pórtico

Esta función se encarga de establecer las variables iniciales que son necesarias para ejecutar los algoritmos encargados de analizar la estructura haciendo uso del método matricial de rigidez. (Figura 30)

```

217 function Nuevo_Portico()
218     Menu_Porticos()
219
220     Tipo_Archivo=1
221     Barra_Carga=0
222
223     Ventana="Definir Materiales"
224     AVentana=platform.window:width()
225     Caja={}
226     for i=1,6 do
227         Caja[i]=D2Editor.newRichText()
228     end
229
230     Tipo_Carga={"Unitaria","Viva","Muerta","Sismica","Viento","Auxiliar"}
231     Tipo_Seccion={"Rectangular","Circular","' T '", "' I '", "' L '", "Personalizado"}

```

Figura 30: Fragmento de Código: Función Nuevo Pórtico

4.2.2. Funciones de control:

Se definen como aquellas funciones que modifican los parámetros que controlan la interfaz gráfica, es decir, son el puente que conectan a el usuario y la máquina, ya que traduce las

acciones del usuario a un lenguaje entendible por la máquina, para de esta manera brindar una mejor experiencia de uso

4.2.2.1. Función Control Ventana

Esta función es la encargada de gestionar cada una de las ventanas, carga la ventana que el usuario desea ver evitando así que se desplieguen múltiples ventanas al mismo tiempo (Figura 31)

```

1006 function Control_Ventana(gc)
1007   if Ventana=="Inicio" then Ventana_Inicio(gc)
1008   elseif Ventana=="Nuevo" then Ventana_Nuevo(gc)
1009   elseif Ventana=="Abrir" then Ventana_Abrir(gc)
1010   elseif Ventana=="Guardar" then Ventana_Guardar(gc)
1011   elseif Ventana=="Cargando" then Ventana_Cargando(gc)
1012   elseif Ventana=="Definir Materiales" then Ventana_Def_Materiales(gc)
1013   elseif Ventana=="Agregar Materiales" then Ventana_A_Materiales(gc)
1014   elseif Ventana=="Insertar Nudos" then Ventana_I_Nudos(gc)
1015   elseif Ventana=="Insertar Elementos" then Ventana_I_Elementos(gc)
1016   elseif Ventana=="Insertar Apoyos" then
1017     if Tipo_Archivo==1 then Ventana_I_Apoyos(gc)
1018     elseif Tipo_Archivo==2 then Ventana_I_Apoyos_Ar(gc)
1019     end
1020   elseif Ventana=="Insertar Cargas Puntuales" then Ventana_I_CargasPuntuales(gc)
1021   elseif Ventana=="Insertar Cargas Distribuidas" then Ventana_I_CargasDistribuidas(gc)
1022   elseif Ventana=="Definir Combinaciones" then Ventana_D_Combinaciones(gc)
1023   elseif Ventana=="Agregar Combinaciones" then Ventana_A_Combinaciones(gc)
1024   elseif Ventana=="Definir Temperatura" then Ventana_D_Temperatura(gc)
1025   elseif Ventana=="Diagrama Cortante G" then Ventana_Diag_Cortante_G(gc)
1026   elseif Ventana=="Diagrama Momento G" then Ventana_Diag_Momento_G(gc)
1027   elseif Ventana=="Diagrama Cortante" then Ventana_Diag_Cortante(gc)
1028   elseif Ventana=="Diagrama Momento" then Ventana_Diag_Momento(gc)
1029   elseif Ventana=="Reacciones Armadura" then Ventana_Rea_Armadura(gc)
1030   elseif Ventana=="Fuerzas Elementos" then Ventana_Fuerza_Armadura(gc)
1031   elseif Ventana=="Matriz Locales" then Ventana_Matriz_Locales(gc)
1032   elseif Ventana=="Matriz Globales" then Ventana_Matriz_Globales(gc)
1033   elseif Ventana=="Matriz K" then Ventana_Matriz_Estructura(gc)
1034   end
1035 end

```

Figura 31: Fragmento de Código: Función Control Ventana

4.2.2.2. Función Finalizar Ejecución Rigidez

Esta función se ejecuta tras realizar todos los cálculos necesarios para analizar una estructura usando el método matricial de rigidez, se encarga de habilitar la opción “Ver” del

menú, además habilita la función de “Almacenar Como...” la cual permite al usuario guardar la estructura para que de esta forma pueda ser abierta en otro momento. (Figura 32)

```

4033 function Finalizar_Ejecucion_R()
4034     Ventana="Diagrama Cortante G"
4035     toolpalette.enable("Archivo", "Almacenar Como...", true)
4036     toolpalette.enable("Ver", "Diagrama Cortante", true)
4037     toolpalette.enable("Ver", "Diagrama Momento", true)
4038     toolpalette.enable("Ver", "Diagrama Cortante Elemento", true)
4039     toolpalette.enable("Ver", "Diagrama Momento Elemento", true)
4040     toolpalette.enable("Ver", "Deformaciones", true)
4041     toolpalette.enable("Ver", "Matriz K Coordenadas Locales", true)
4042     toolpalette.enable("Ver", "Matriz K Coordenadas Globales", true)
4043     toolpalette.enable("Ver", "Matriz K Estructura", true)
4044     Save=true
4045     Mat_T, Mat_T_I=nil, nil
4046 end

```

Figura 32: Fragmento de Código: Función Finalizar_Ejecucion_R()

4.2.3. Funciones de cálculo

El objetivo de este tipo de funciones es la realización de cálculos matemáticos, estos pueden ir desde operaciones aritméticas hasta operaciones con matrices y son las encargadas de completar los cálculos necesarios para realizar el análisis de la estructura

4.2.3.1. Función Ejecutar Rigidez

Esta función se encarga de ejecutar cada una de las funciones necesarias para analizar la estructura por el método matricial de rigidez, la función mostrada en la Figura 33 fue implementada basándose en el diagrama de flujo mostrado en la Figura 25

```

3376 function Ejecutar_Rigidez()
3377     Numero.MG=Numero.Nudos*3
3378     Calcular_L_A()
3379     Crear_Combinacion_Defecto()
3380     Propiedades_Elementos()
3381     Calcular_Rigidez()
3382     Crear_Matrices()
3383     Crear_Matriz_General()
3384     Crear_Vector_Def_Reac()
3385     Vector_Combinacion_D()
3386     Vector_Combinacion_P()
3387     Simplificar_MG_1()
3388     Simplificar_MG_2()
3389     Matriz_Inver=Mat_Inversa(Matriz_Simp_1,Numero.Deformaciones)
3390     Calcular_Deformaciones()
3391     Calcular_Reacciones()
3392     Vector_Deformacion_Elemental()
3393     Armar_Vector_Fuerzas_Local()
3394     Vector_Combinado()
3395     Ecuaciones_Cortante_Rigidez()
3396     Ecuaciones_Momento_Rigidez()
3397     Cambiar_Ventana()
3398     Finalizar_Ejecucion_R()
3399     Ventana="Diagrama Cortante G"
3400 end

```

Figura 33: Fragmento de Código: Función Ejecutar Rigidez

4.2.3.2. Función Multiplicar Matrices

Esta función tiene como propósito multiplicar dos matrices, el funcionamiento de esta función fue analizado a profundidad durante la fase de diseño, como resultado se obtuvo el diagrama de flujo detallado mostrado en la Figura 26 y posteriormente fue implementado como se muestra en la Figura 34

```

3519 function Multiplicar_Matrices (Matriz_1,Matriz_2,Nx1,Ny1,Nx2)
3520     local Mmult={}
3521     for i=1,Nx2 do
3522         Mmult[i]={}
3523         for j=1,Ny1 do
3524             Mmult[i][j]=0
3525             for k=1,Nx1 do
3526                 Mmult[i][j]=Mmult[i][j]+Matriz_1[k][j]*Matriz_2[i][k]
3527             end
3528         end
3529     end
3530     return Mmult
3531 end

```

Figura 34: Fragmento de Código: Función Multiplicar Matrices

4.2.3.4. Función Crear Matriz General

Esta función se encarga de crear la matriz de rigidez K de la estructura a partir de las matrices de rigidez K de cada elemento. (Figura 35)

```

3544 function Crear_Matriz_General()
3545     Matriz_General={}
3546     for i=1,Numero.MG do
3547         Matriz_General[i]={}
3548         for j=1,Numero.MG do
3549             Matriz_General[i][j]=0
3550         end
3551     end
3552     for i=1,Numero.Elementos do
3553         local Aux1,Aux2=Dat_elemento.Ni[i],Dat_elemento.Nf[i]
3554         local Aux3,Aux4
3555         for j=1,6 do
3556             for k=1,6 do
3557                 if j<=3 then Aux3=Aux1*3+j-3
3558                     else Aux3=Aux2*3+j-6
3559                 end
3560                 if k<=3 then Aux4=Aux1*3+k-3
3561                     else Aux4=Aux2*3+k-6
3562                 end
3563                 Matriz_General[Aux3][Aux4]=Matriz_General[Aux3][Aux4]+Mat_Global[i][j][k]
3564             end
3565         end
3566     end
3567 end

```

Figura 35: Fragmento de Código: Función Crear Matriz General

4.2.3.5. Función Calcular Deformaciones

Esta función es la encargada de calcular las deformaciones de la estructura haciendo uso de la teoría del método matricial de rigidez. (Figura 36)

```

3826 function Calcular_Deformaciones()
3827     Vector.Def_Sol={}
3828     for i=1,Numero.Combinaciones do
3829         Vector.Def_Sol[i]={}
3830         for j=1,Numero.Deformaciones do
3831             Vector.Def_Sol[i][j]=0
3832         end
3833     end
3834     for i=1,Numero.Combinaciones do
3835         local Vect_Aux={}
3836         for j=1,Numero.Deformaciones do
3837             Vect_Aux[j]=Vector_F_Simp_1[i][j]-Vector_E_Simp_1[i][j]
3838         end
3839         Vector.Def_Sol[i]=Multiplicar_Vector(Matriz_Inver,Vect_Aux,Numero.Deformaciones,Numero.Deformaciones)
3840     end
3841 end

```

Figura 36: Fragmento de Código: Función Calcular Deformaciones

4.2.3.6. Función Matriz Inversa

Esta función se encarga de recibir una matriz y devolver la matriz inversa de esta. Para esta función se hizo uso del CAS del dispositivo portátil, el cual permite realizar este tipo de cálculos de manera más sencilla.(Figura 37)

```

3805 function Mat_Inversa(Matriz,N)
3806     local Minv={}
3807     math.eval("Mi:=NewMat("..N..","..N..")")
3808     for i=1,N do
3809         Minv[i]={}
3810         for j=1,N do
3811             Minv[i][j]=0
3812             var.storeAt("Mi", Matriz[i][j],i,j)
3813         end
3814     end
3815     math.eval("Minv:=Mi^(-1)")
3816     for i=1,N do
3817         for j=1,N do
3818             Minv[i][j] = var.recallAt("Minv", i, j)
3819         end
3820     end
3821     math.eval("DelVar Mi")
3822     math.eval("DelVar Minv")
3823     return(Minv)
3824 end

```

Figura 37: Fragmento de Código: Función Matriz Inversa

4.2.3.7. Función Decimales

Esta función se encarga de redondear el valor “Dato”, lo cual reduce la cantidad de que decimales mostrados. (Figura 38)

```

4162 function Decimales(Dato,Numero_D)
4163     return tonumber(string.format("%.Numero_D..f",Dato))
4164 end

```

Figura 38: Fragmento de Código: Función Decimales()

4.2.3.8. Función Ejecutar Armadura

Esta función se encarga de ejecutar cada una de las funciones necesarias para analizar una armadura isostática por medio del método de nudos. (Figura 39)

```

4058 function Ejecutar_Armadura()
4059     Numero.MG=Numero.Nudos*2
4060     Numero.Reacciones=3
4061     Numero.Apoyos=2
4062     Crear_Combinacion_Defecto()
4063     local Continuar=Verificar_Armadura()
4064     if Continuar==true then
4065         Crear_Fuerzas_Comb_Armaduras()
4066         Calcular_L_A()
4067         Ensamblar_Matriz()
4068         Resolver_Fuerzas()
4069         Finalizar_Ejecucion_A()
4070     else
4071     end
4072 end

```

Figura 39: Fragmento de Código: Función Ejecutar Armadura

4.2.4. Funciones de lectura y almacenamiento de datos

Este tipo de funciones se encargan de almacenar los datos introducidos por el usuario en variables, además se encargan tanto de abrir como guardar archivos. Este tipo de función destaca de las otras ya que encriptan o des encriptan los datos.

4.2.4.1. Función Abrir Archivo

Esta función es aquella que le permite al usuario abrir un archivo que había sido guardado con anterioridad, de esta forma no es necesario que ingrese nuevamente los datos cada vez que encienda el dispositivo portátil. (Figura 40)

```

4486 function Abrir_Archivo()
4487     local Aux1,Aux2=0,0
4488     for i=1,Seleccion.Abrir do
4489         if Ty[i]==1 then Aux1=Aux1+1
4490             elseif Ty[i]==2 then Aux2=Aux2+1
4491         end
4492     end
4493     if Ty[Seleccion.Abrir]==1 then
4494         Reiniciar_Sistema()
4495         Nuevo_Portico()
4496         Abrir_Portico(Aux1)
4497     elseif Ty[Seleccion.Abrir]==2 then
4498         Reiniciar_Sistema()
4499         Nuevo_Armadura()
4500         Abrir_Armadura(Aux2)
4501     end
4502     Seleccion.Abrir=1
4503     Ty,N_arc,Nombre_Archivo=nil,nil,nil
4504 end

```

Figura 40: Fragmento de Código: Función Abrir Archivo

4.2.4.2. Función Leer Datos

Esta función se encarga de leer y decodificar los datos almacenados en la memoria interna, los datos decodificados son mostrados posteriormente por una función de control y de esta manera el usuario puede ver los archivos que se encuentran almacenados en el dispositivo.

(Figura 41)

```
4460 function Leer_Datos()
4461     Seleccion.Abrir=1
4462     local Nom={}
4463     Nom[1]=var.recall("n1")
4464     Nom[2]=var.recall("n2")
4465     Nom[3]=var.recall("n3")
4466     Nom[4]=var.recall("n4")
4467     Ty=var.recall("tipo")
4468     Ty2=var.recall("tipo2")
4469     N_arc=table.getn(Ty)
4470     Nombre_Archivo={}
4471     for i=1,N_arc do
4472         Nombre_Archivo[i]=" "
4473         for j=1,4 do
4474             for k=1,4 do
4475                 local Aux=string.sub(Nom[j][i],k*2-1,k*2)
4476                 Aux=tonumber(Aux)
4477                 Nombre_Archivo[i]=Nombre_Archivo[i]..Alfabeto[Aux]
4478             end
4479         end
4480     end
4481     Nom=nil
4482 end
```

Figura 41: Fragmento de Código: Función Leer Datos

Capítulo 5: Fase de pruebas del software

“Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa.

El proceso de prueba tiene dos metas distintas:

- 1. Demostrar al desarrollador y al cliente que el software cumple con los requerimientos. Para el software personalizado, esto significa que en el documento de requerimientos debe haber, por lo menos, una prueba por cada requerimiento. Para los productos de software genérico, esto quiere decir que tiene que haber pruebas para todas las características del sistema, junto con combinaciones de dichas características que se incorporarán en la liberación del producto.*
- 2. Encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación. Tales situaciones son consecuencia de defectos del software. La prueba de defectos tiene la finalidad de erradicar el comportamiento indeseable del sistema, como caídas del sistema, interacciones indeseadas con otros sistemas, cálculos incorrectos y corrupción de datos.” (Sommerville, Ingeniería de Software, 2011)*

5.1. Pruebas de funcionamiento de la GUI

Durante la realización de las pruebas de funcionamiento de la interfaz gráfica de usuario se logró identificar un fallo en una de las funciones de control, la cual se encontraba incompleta, en la Figura 42 se muestra el error obtenido durante la etapa de pruebas de la interfaz gráfica del usuario. Así mismo, en la Tabla 7 se especifica a detalle las causas, las consecuencias y la solución a este error.

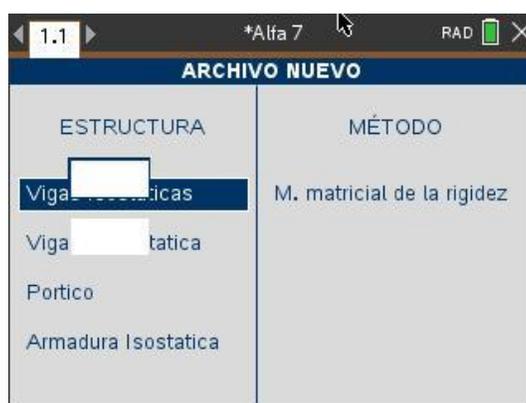


Figura 42: Error no esperado en la GUI

Tabla 7: Detalles error en la GUI

Nombre de la falla	Aparición de cuadros en lugares inesperados
Descripción	Al momento de navegar entre las diferentes opciones del software, en las ventanas aparecen rectángulos indeseados, los cuales permiten el ingreso de datos
Acción realizada	Ingresar datos de los nudos y seguidamente acceder a la opción de archivo nuevo
Fallo	Al navegar entre las ventanas y las diferentes opciones, el software debe realizar el proceso de ocultar los cuadros de texto, acción que no se realiza al momento de crear un nuevo archivo
Solución	Agregar la función Ocultar_Cajas() dentro de la función Cambiar_Ventana(), esto le permitirá al software ocultar los cuadros de texto cuando no sean necesarios

5.2. Pruebas de cálculo

5.2.1. Problema 1:

Para el marco que se muestra en la Figura 43, calcule las reacciones en los empotramientos A y C . Obsérvese que la columna inclinada recibe una carga distribuida uniforme de $4T/m$ ortogonal a su eje. Considere para ambos miembros el módulo de elasticidad del concreto con $E = 2213594.362 \text{ T/m}^2$ y una sección transversal rectangular de base $b = 20c \text{ m}$ y altura $h = 30c \text{ m}$.¹

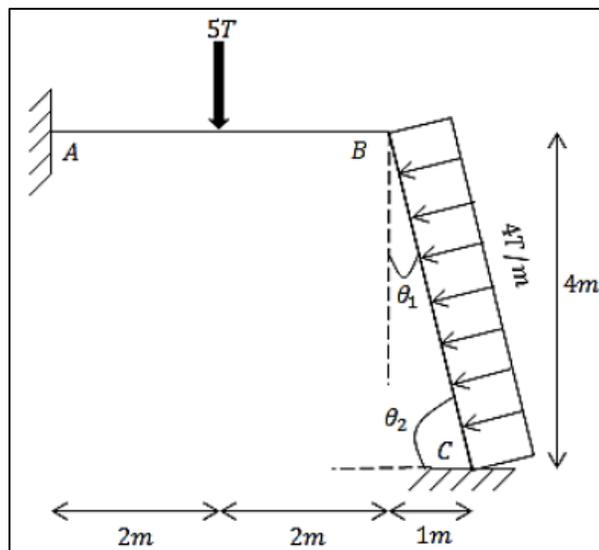


Figura 43: Problema 1

Fuente: Ortiz Soto, D. (2015). *Análisis de estructuras Problemas resueltos* (Segunda ed.). México (p. 222)

¹ Tomado de Ortiz Soto, D. (2015). *Análisis de estructuras Problemas resueltos* (Segunda ed.). México (p. 222-230). Ejercicio 2.14

5.2.1.1. Resultados obtenidos de la fuente bibliográfica

$$R_{xA} = 8.6294 T; \quad R_{yA} = 1.944 T; \quad M_A = 1.787 T \cdot m$$

$$R_{xC} = 7.3706 T; \quad R_{yC} = 7.0560 T; \quad M_C = -6.5486 T \cdot m$$

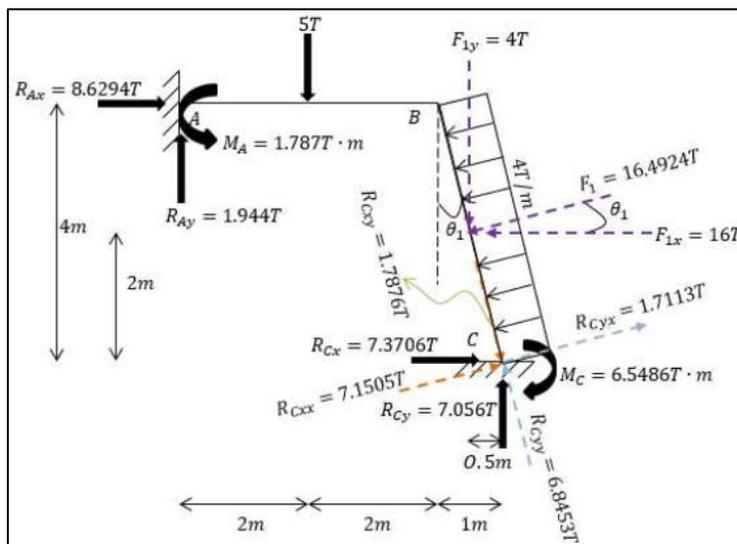


Figura 44: Reacciones problema 1

Fuente: Ortiz Soto, D. (2015). Analisis de estructuras Problemas resueltos (Segunda ed.). México (p. 230)

5.2.1.2. Resultados obtenidos mediante el software

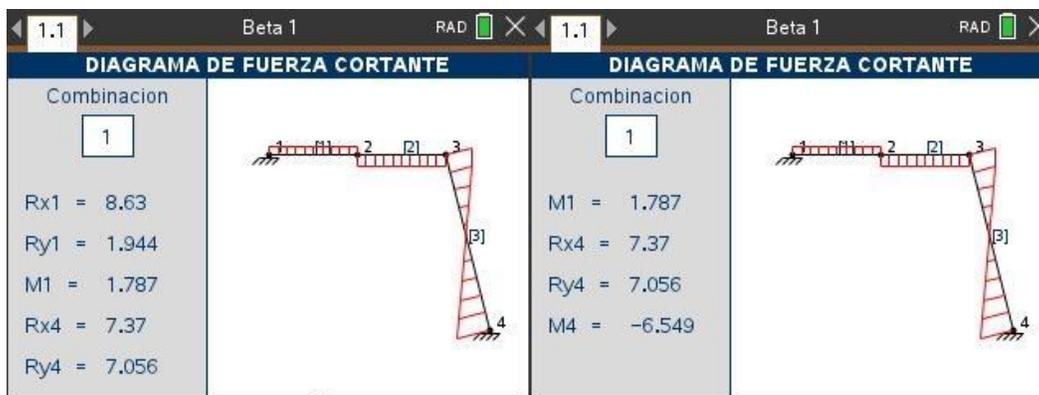


Figura 45: Reacciones problema 1 usando Ac-Struc

5.2.2. Problema 2:

En la Figura 46 se representa un marco con una columna de doble altura. Calcule las reacciones en los empotramientos A, F y H. Considere para todos los miembros una sección transversal rectangular de base $b = 20\text{cm}$ y altura $h = 50\text{cm}$ y $E = 21707947.03 \text{ kN/m}^2$.²

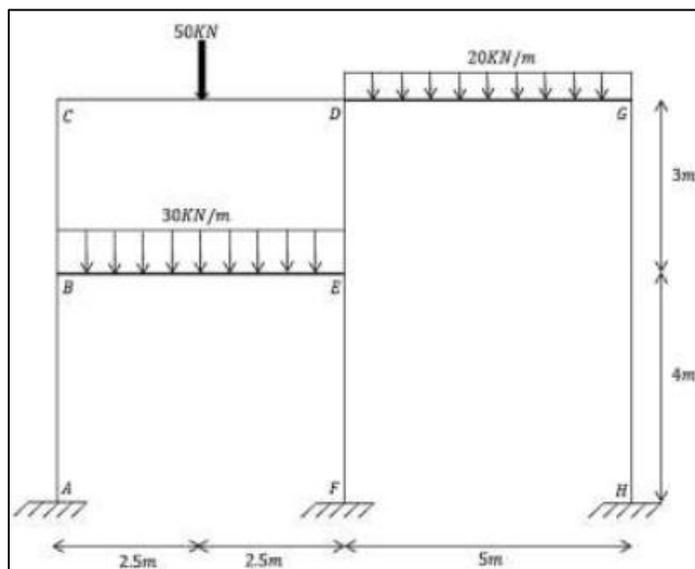


Figura 46: Problema 2

Fuente: Ortiz Soto, D. (2015). Analisis de estructuras Problemas resueltos (Segunda ed.). México (p. 248)

5.2.2.1. Resultados obtenidos de la fuente bibliográfica

$$R_{xA} = 11.1623 \text{ kN} ; \quad R_{yA} = 95.9845 \text{ kN} ; \quad M_A = -16.7131 \text{ kN} \cdot \text{m}$$

² Tomado de Ortiz Soto, D. (2015). Analisis de estructuras Problemas resueltos (Segunda ed.). México (p. 248-259) . Ejercicio 2.17

$$R_{xF} = -6.7375 \text{ kN}; \quad R_{yF} = 158.1651 \text{ kN}; \quad M_F = 7.2135 \text{ kN} \cdot \text{m}$$

$$R_{xF} = -4.4248 \text{ kN}; \quad R_{yF} = 45.8505 \text{ kN}; \quad M_F = 10.1696 \text{ kN} \cdot \text{m}$$

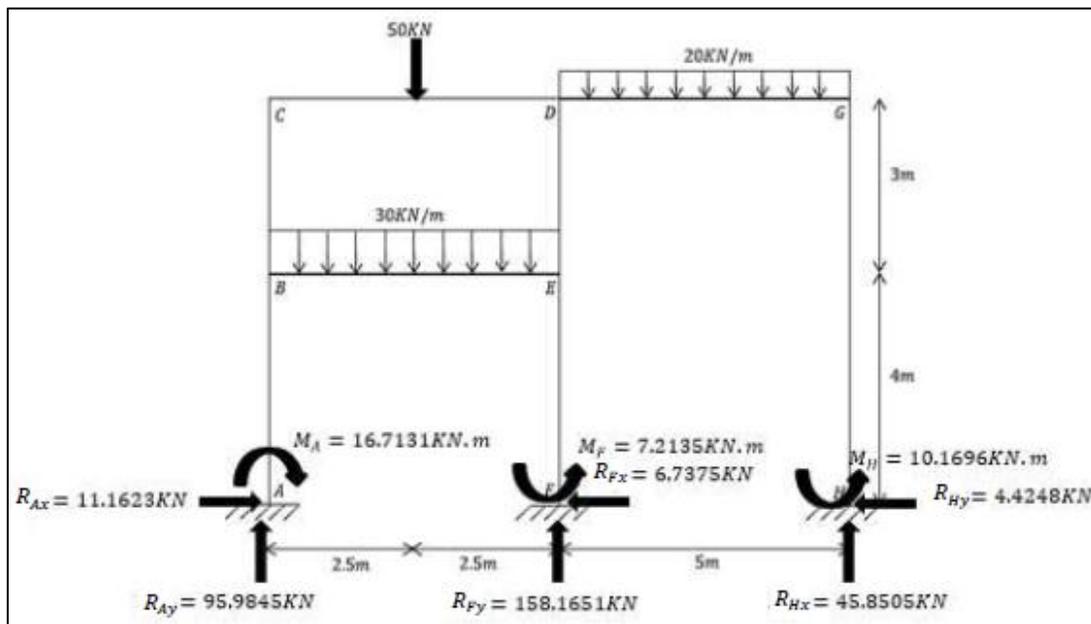


Figura 47: Reacciones problema 2

Fuente: Ortiz Soto, D. (2015). Analisis de estructuras Problemas resueltos (Segunda ed.). México (p. 259)

5.2.2.2. Resultados obtenidos mediante el software

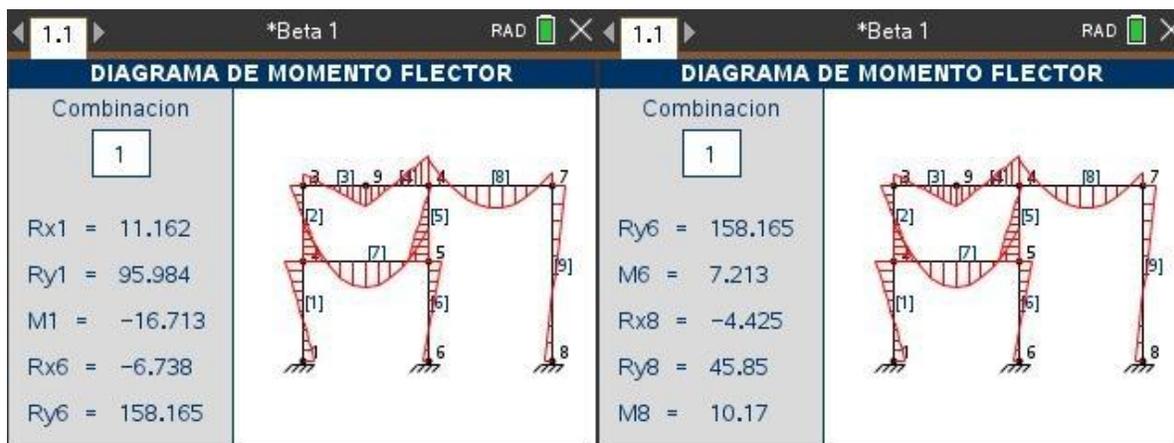


Figura 48: Reacciones problema 2 usando Ac-Struc

5.2.3. Problema 3:

Use el método del análisis matricial por rigidez para calcular las reacciones en los apoyos de la estructura rígida mostrada en la Figura 49. En la ilustración se indican los valores relativos del área y del momento de inercia de la sección transversal de cada elemento de la estructura.

$I=500\text{in}^4$ y $A=20\text{in}^2$. El módulo de elasticidad es igual para los dos elementos.³

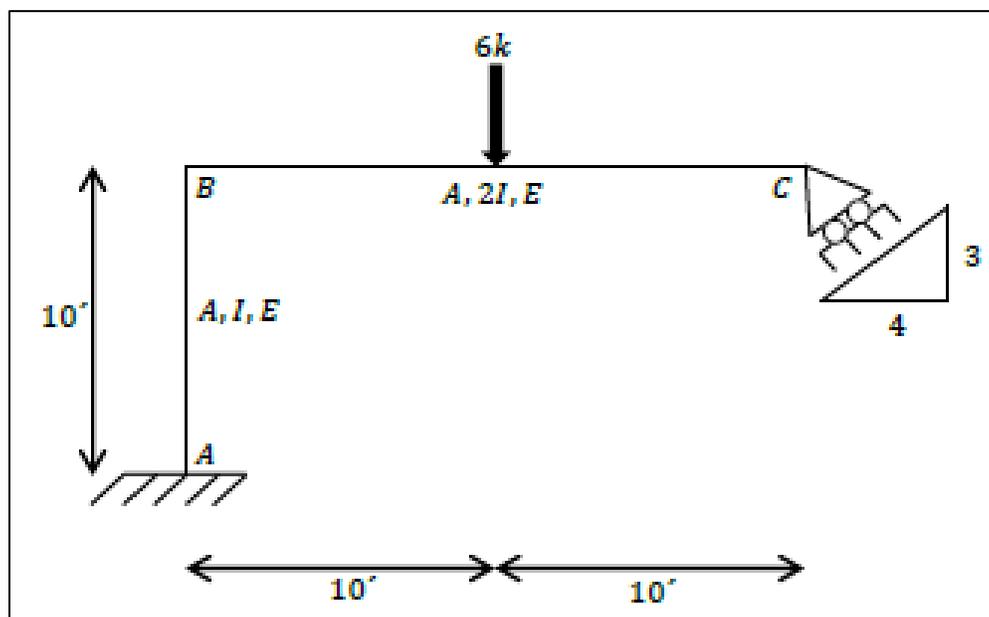


Figura 49: Problema 3

Fuente: Ortiz Soto, D. (2015). *Análisis de estructuras Problemas resueltos* (Segunda ed.). México (p. 231)

³ Tomado de Ortiz Soto, D. (2015). *Análisis de estructuras Problemas resueltos* (Segunda ed.). México (p. 231-237) . Ejercicio 2.15

5.2.3.1. Resultados obtenidos de la fuente bibliográfica

$$R_C - 2.4 = 0.5825 \Rightarrow R_C = 0.5825 + 2.4 = 2.9825 \Rightarrow \therefore R_C = 2.9825k \swarrow$$

$$R_{Ax} = 1.7895 \Rightarrow \therefore R_{Ax} = 1.7895k \rightarrow \quad R_{Ay} = 3.6140 \Rightarrow \therefore R_{Ay} = 3.6140k \uparrow$$

$$M_A = -5.6146 \Rightarrow \therefore M_A = 5.6146k \cdot \text{pie} \curvearrowright$$

Figura 50: Reacciones problema 3

Fuente: Ortiz Soto, D. (2015). Analisis de estructuras Problemas resueltos (Segunda ed.). México (p. 237)

5.2.3.2. Resultados obtenidos mediante el software

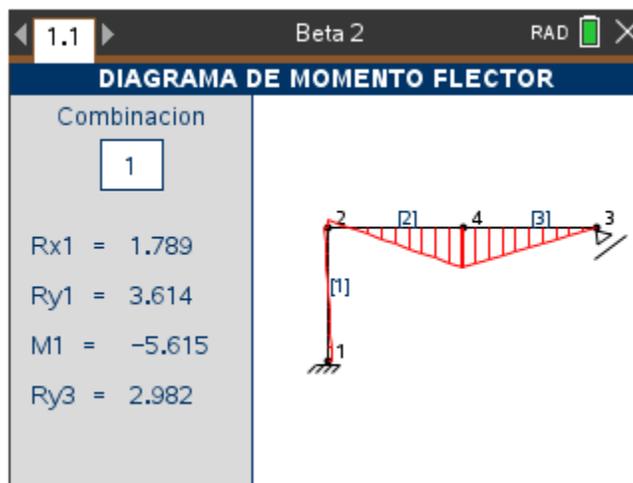


Figura 51: Reacciones problema 3 usando Ac-Struc

5.2.4 Problema 4:

Para la armadura de la Figura 52: Calcular las reacciones en los apoyos, indicar que barras no trabajan, determinar las fuerzas axiales en las barras restantes⁴

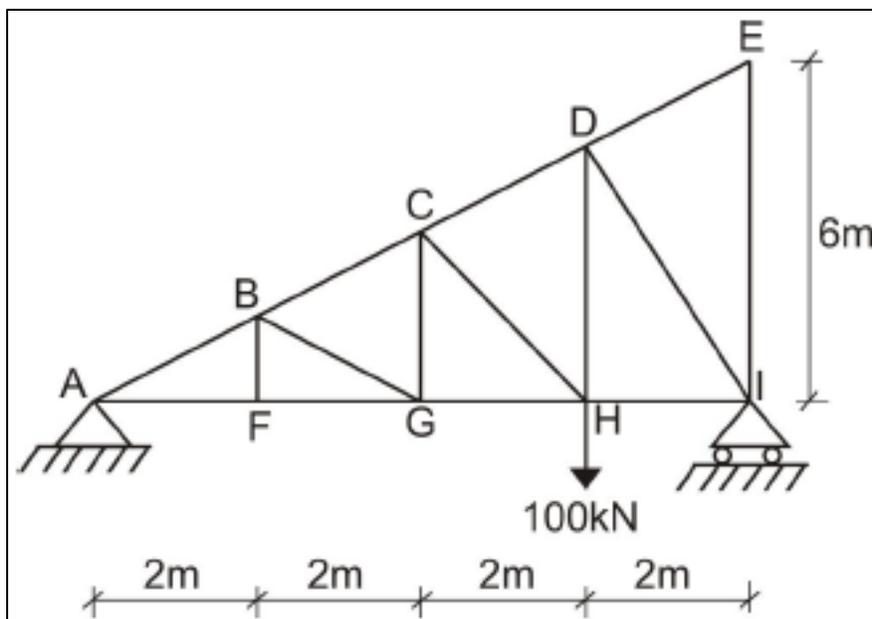


Figura 52: Problema 4

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 113)

5.2.4.1. Resultados obtenidos de la fuente bibliográfica

$$R_{xA} = 0 \text{ kN}; \quad R_{yA} = 25 \text{ kN}; \quad R_{yI} = 75 \text{ kN}$$

⁴ Tomado de Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 113-115) Problema 4.2

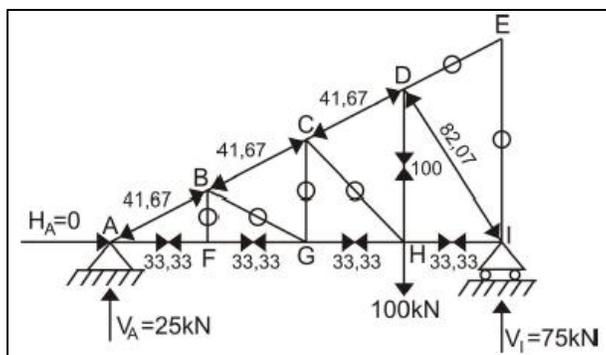


Figura 53: Solución problema 4

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 113)

5.2.4.2. Resultados obtenidos mediante el software

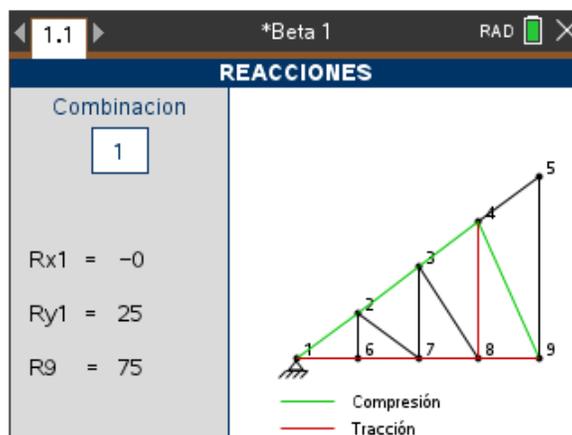


Figura 54: Reacciones problema 4 usando Ac-Struc

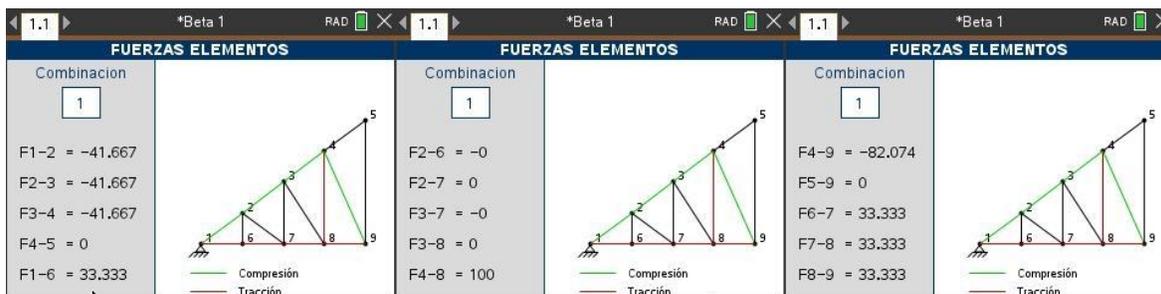


Figura 55: Fuerzas axiales de los elementos problema 4 usando Ac-Struc

5.2.5. Problema 5:

Para la armadura mostrada en la Figura 56, usando el método de los nudos, determinar las fuerzas en las barras CD y DF⁵

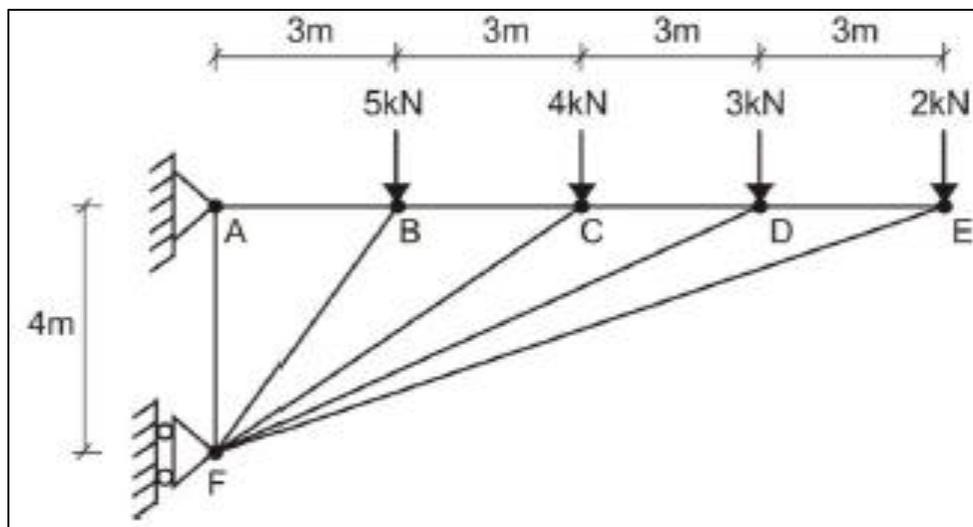


Figura 56: Problema 5

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 117)

5.2.5.1. Resultados obtenidos de la fuente bibliográfica

$$F_{CD} = 12.75 \text{ kN (Tracción)}; \quad F_{DF} = 7.387 \text{ kN (Compresión)}$$

⁵ Tomado de Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 117-118) Problema 4.4

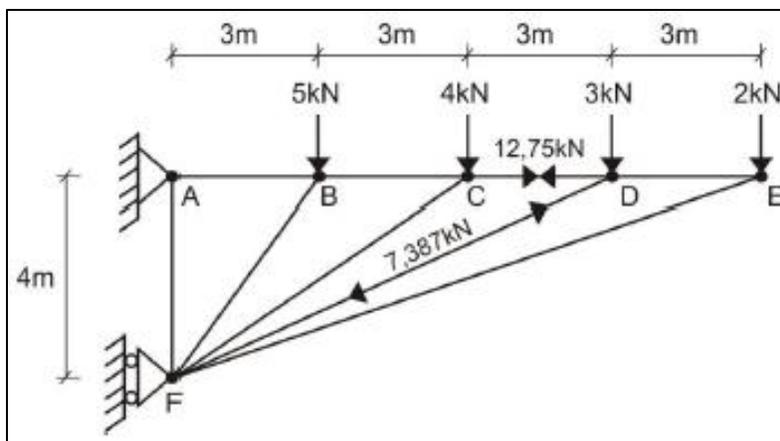


Figura 57: Solución problema 5

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 118)

5.2.5.2. Resultados obtenidos mediante el software

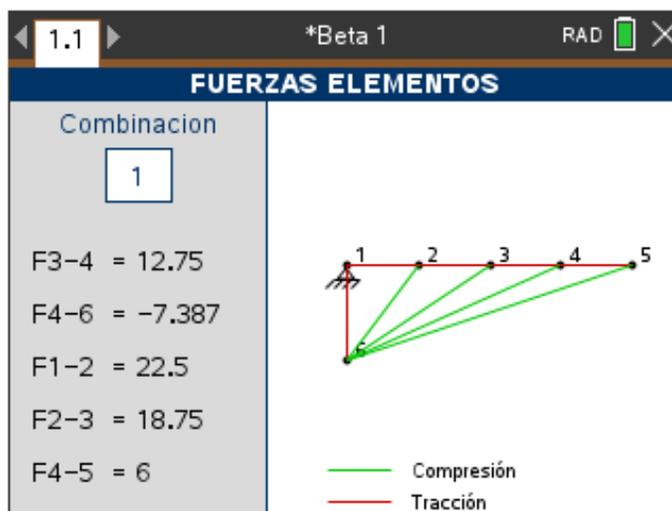


Figura 58: Solución problema 5 usando Ac-Struc

5.2.6. Problema 6:

Para la armadura de la Figura 59, calcular las reacciones en los apoyos. determinar las fuerzas axiales en cada una de las barras.⁶

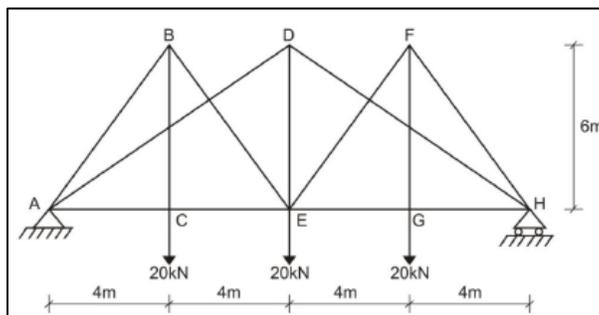


Figura 59: Problema 6

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 122)

5.2.6.1. Resultados obtenidos de la fuente bibliográfica

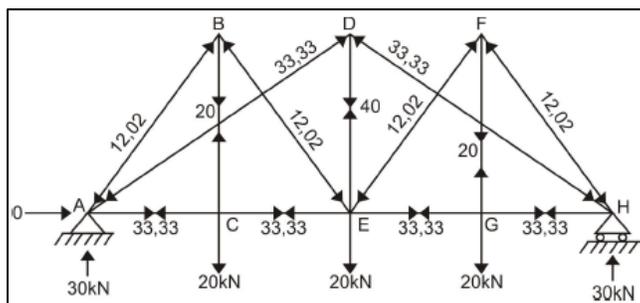


Figura 60: Solución problema 6

Fuente: Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 124)

⁶ Tomado de Villarreal Castro, G. (2011). Estática Problemas Resueltos. Lima, Perú. (p. 122-124) Problema 4.6

5.2.6.2. Resultados obtenidos mediante el software

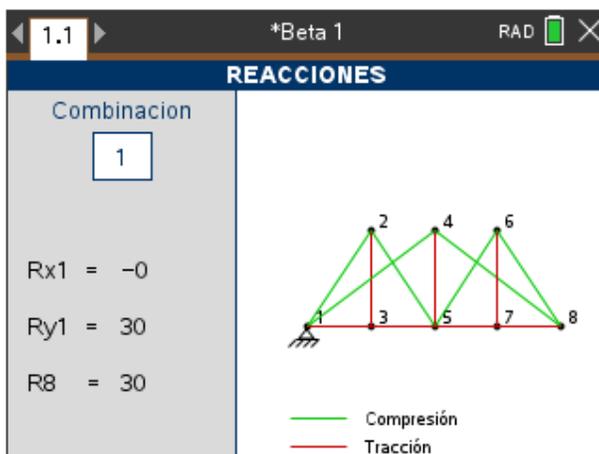


Figura 61: Reacciones problema 6 usando Ac-Struc

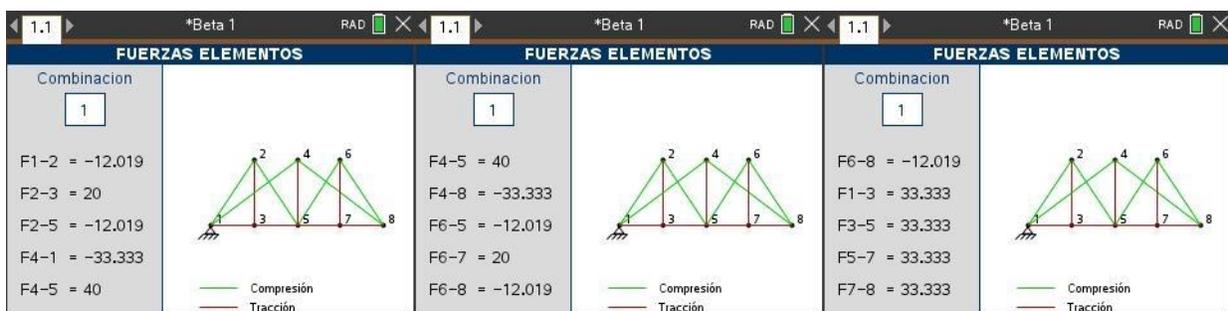


Figura 62: Fuerzas axiales de los elementos problema 6 usando Ac-Struc

5.2.6. Problema 7:

Resolver la viga continua con extremos empotrados que se muestra en la Figura 63⁷

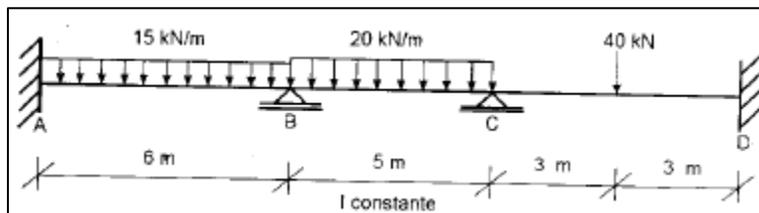


Figura 63: Problema 7

Fuente: Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (p. 171)

5.2.7.1. Resultados obtenidos de la fuente bibliográfica

$$M_A = -45.1 \text{ kN} \cdot \text{m}; \quad M_B = -44.9 \text{ kN} \cdot \text{m}$$

$$M_C = -35.3 \text{ kN} \cdot \text{m}; \quad M_D = -27.4 \text{ kN} \cdot \text{m}$$

5.2.7.2. Resultados obtenidos mediante el software

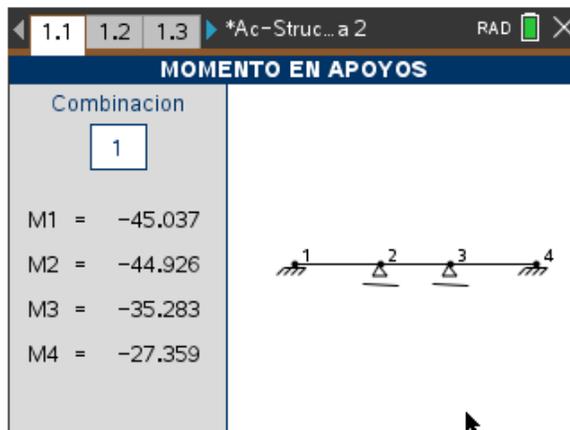


Figura 64: Momentos problema 7 usando el software Ac-Struc

⁷ Tomado de Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (pp. 170-173)

5.2.8. Problema 8:

Resuelva la viga mostrada en la Figura 65⁸

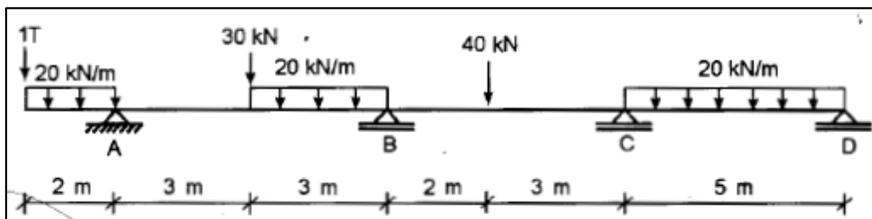


Figura 65: Problema 8

Fuente: Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (p. 168)

5.2.8.1. Resultados obtenidos de la fuente bibliográfica

$$M_A = -60 \text{ kN} \cdot \text{m} ; \quad M_B = -38.4 \text{ kN} \cdot \text{m}$$

$$M_C = -38.5 \text{ kN} \cdot \text{m} ; \quad M_D = 0 \text{ kN} \cdot \text{m}$$

5.2.8.2. Resultados obtenidos mediante el software

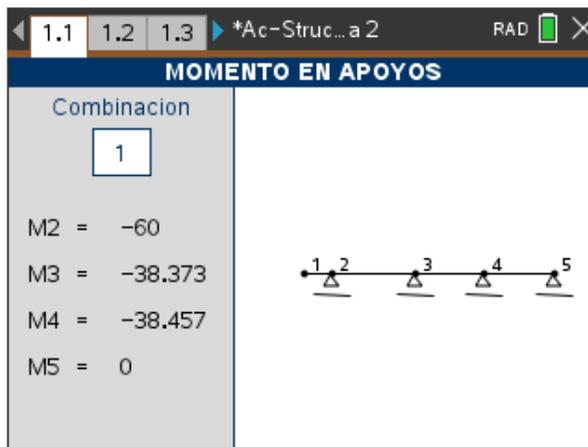


Figura 66: Momentos problema 8 usando el software Ac-Struc

⁸ Tomado de Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (pp. 168-170)

5.2.9. Problema 9:

La viga continua de la Figura 67 se hace de hormigón reforzado ($f'_c=21 \text{ N/mm}^2$) y tiene las secciones indicadas. Estúdiense el efecto de un asentamiento de 15 mm en el apoyo B.⁹

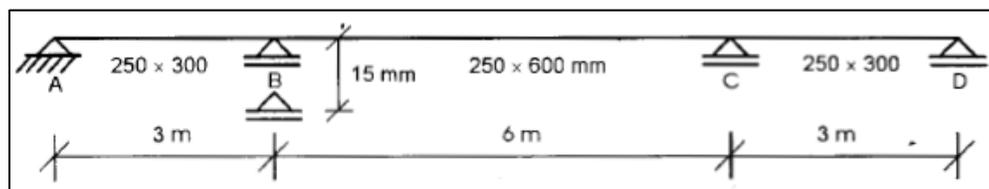


Figura 67: Problema 9

Fuente: Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (p. 174)

5.2.9.1. Resultados obtenidos de la fuente bibliográfica

$$M_A = 0 \text{ kN} \cdot \text{m}; \quad M_B = 31.5 \text{ kN} \cdot \text{m}$$

$$M_C = -13.2 \text{ kN} \cdot \text{m}; \quad M_D = 0 \text{ kN} \cdot \text{m}$$

5.2.9.2. Resultados obtenidos mediante el software

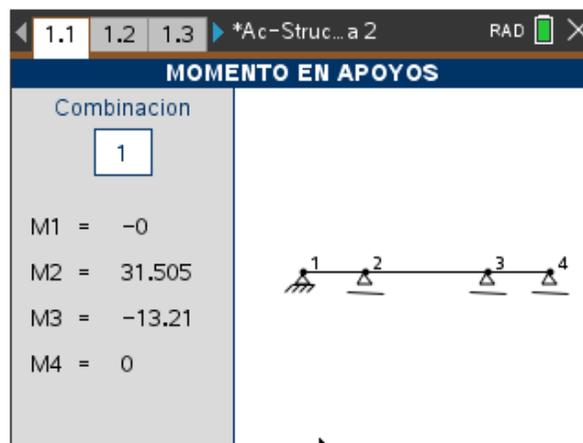


Figura 68: Momentos problema 9 usando el software Ac-Struc

⁹ Tomado de Escamilla, J. U. (s.f.). Análisis de Estructuras (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería (pp. 173-177)

Capítulo 6: Conclusiones y recomendaciones

6.1. Conclusiones

El software desarrollado permite realizar análisis estructural a pórticos, armaduras isostáticas y vigas hiperestáticas de manera fácil y sencilla.

El uso de software reduce considerablemente el tiempo requerido para realizar análisis estructural, no obstante, mantiene un nivel de precisión elevado.

El dispositivo portátil TI-Nspire CX CAS es un instrumento ideal para ingenieros civiles en formación debido a su facilidad de uso, su portabilidad, su capacidad de procesamiento y su principal característica la cual permite el desarrollo de softwares independientes que permite la creación de aplicaciones para las diferentes áreas de la ingeniería civil como lo son: Vías, Suelos, Aguas, Estructuras.

El software desarrollado resulta ser una herramienta de mucha utilidad para estudiantes, sin embargo, este no debe reemplazar el correcto estudio teórico de los temas.

Lua es un lenguaje de programación versátil, moderno y fácil de aprender, lo cual lo convierte en el lenguaje ideal para iniciar en el área del desarrollo de software.

Es de gran importancia que el departamento de ingeniería civil de la Universidad de Pamplona imparta cursos y seminarios sobre el desarrollo de software y el lenguaje de programación Lua, así como fomentar la adquisición de dispositivos portátiles TI-Nspire CX CAS por parte de los estudiantes.

El software requiere un nivel de conocimiento intermedio acerca de los temas de análisis estructural, ya que así podrá ser utilizado de manera correcta aprovechando la totalidad de las funcionalidades incluidas, así como también la correcta interpretación de resultados.

Los softwares deben incluir un manual de usuario ya que este permite al usuario comprender el uso correcto y completo, evitando de esta manera se cometan errores debido al desconocimiento del funcionamiento del software.

El manual técnico permite a futuros desarrolladores la expansión del software, con lo cual es posible aumentar las funcionalidades del software sin la necesidad de la presencia del desarrollador inicial.

6.2. Recomendaciones

Los estudiantes deben usar el software únicamente para comprobar los resultados, ya que este solo es una herramienta la cual no sustituye el estudio teórico de los métodos de análisis estructural.

Se recomienda a docentes de ingeniería civil de la Universidad de Pamplona que soliciten a sus estudiantes que en sus proyectos académicos relacionados con el análisis estructural se incluyan capturas de pantalla de los resultados obtenidos mediante el software.

Se debe continuar con el desarrollo de softwares para el área de ingeniería civil, en especial con los del área de diseño y análisis estructural.

Se recomienda a futuros desarrolladores completar el software incluyendo nuevas funcionalidades como lo es el análisis de estructuras en tres dimensiones (3D).

Se recomienda al departamento de ingeniería civil la creación de clases centradas en la programación del dispositivo portátil TI Nspire CX CAS, así como fomentar la adquisición del producto, ya que este podría aumentar la competitividad del programa en comparación con otras instituciones de educación superior.

Bibliografía

- Albornoz, M. C., Berón, M., & Montejano, G. (s.f.). *Interfaz Gráfica de Usuario: el Usuario como Protagonista del Diseño*. Universidad Nacional de San Luis, San Luis.
- Cervera Ruiz, M., & Blanco Diaz, E. (2001). *Mecánica de estructuras Libro 1 Resistencia de materiales* (Segunda ed.). Barcelona, España: Edicions UPC.
- Cervera Ruiz, M., & Blanco Diaz, E. (2004). *Mecánica de estructuras Libro 2 Métodos de análisis* (Segunda ed.). Barcelona, España: Edicions UPC.
- Escamilla, J. U. (s.f.). *Análisis de Estructuras* (Segunda ed.). Colombia: Escuela Colombiana de Ingeniería.
- Espinoza, A. J. (1995). *MARPLA, Programa para el análisis estructural matricial de marcos planos*. Tesis para optar el título profesional de Ingeniero Civil, Universidad Nacional Autónoma de México, Facultad de Ingeniería Civil, México.
- Hinojosa Gabriel, O. (2009). *Software para la enseñanza de la dinámica estructural*. Tesis para optar por el grado de Maestro en Ingeniería, Universidad Nacional Autónoma de México, Programa de maestría y doctorado en ingeniería, México.
- Kinney, J. S. (1962). *Indeterminate Structural Analysis*. Addison-Wesley.
- Lerusalimschy, R., de Figueiredo, L. H., & Celes, W. (29 de Agosto de 2019). *Manual de Referencia de Lua 5.1*. Recuperado el 29 de Noviembre de 2019, de LUA:
<https://www.lua.org/manual/5.1/es/manual.html>
- M. Gómez , L. S. (2000). *Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación*.
- McCormac, J. C., & Brown, R. H. (2011). *Diseño de Concreto reforzado* (Octava ed.). México D.F.: Alfaomega.

- Meza Huaynate, M. J. (2019). *Desarrollo del software de aplicación ESDID para el cálculo de espectros elásticos de diseños determinísticos para edificaciones*. Tesis para optar el título profesional de Ingeniero Civil, Universidad Nacional del Centro del Perú, Facultad de Ingeniería Civil, Huancayo.
- Miller, G. A. (1955). *The Magical Number Seven, Plus or Minus Two*. Psychological Review, Philadelphia.
- Nieto García, E. (05 de Marzo de 2009). *Transformación de la matriz de rigidez*. Recuperado el 29 de 11 de 2019, de OpenCourseWare Universidad de Sevilla:
http://ocwus.us.es/mecanica-de-medios-continuos-y-teoria-de-estructuras/calculo-de-estructuras-1/apartados/apartado6_6.html
- Ortiz Berrocal, L. (2007). *Resistencia de Materiales* (Tercera ed.). España: McGraw-Hill.
- Ortiz Soto, D. (2015). *Análisis de estructuras Problemas resueltos* (Segunda ed.). México.
- Plata Arrieta, C. M. (2015). *Aplicación del software de la Ti-Nspire CX CAS como herramienta de apoyo en el análisis geotécnico para el diseño de cimentaciones superficiales, con base en el reglamento colombiano de construcción sismo resistentes NSR-10*. Tesis para optar el título profesional de Ingeniero Civil, Universidad de Pamplona, Departamento de ingeniería civil, Pamplona.
- Regalado Ramírez, J., & Romero Garrido, A. (2004.). *Software Interactivo para el Diseño de Trabes Armadas con el método LFRD*. Tesis Licenciatura, Universidad de las Américas Puebla, Departamento de Ingeniería Civil, Cholula.
- Samperio Fuentes, L. E. (2004). *Software para el Diseño de Vigas I de Concreto Presforzado*. Tesis profesional, Universidad de las Américas Puebla, Departamento de Ingeniería Civil, Cholula.

Silva V., G. S. (2016). Método Matricial para el Análisis de Armaduras Planas y Espaciales.

L'esprit Ingénieur.

Sommerville, I. (2005). *Ingeniería de Software* (Septima ed.). Madrid, España: Pearson Educación.

Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). México: PEARSON EDUCACIÓN.

Texas Instruments. (s.f.). *TI-Nspire Writing Lua Scripts Guide*.

Universidad de Pamplona. (s.f.). Recuperado el 19 de Noviembre de 2019, de Imagen

Corporativa Universidad de Pamplona:

http://www.unipamplona.edu.co/unipamplona/portaIG/home_1/recursos/corporativo/15022011/descargas_unipamplona.jsp

Villarreal Castro, G. (2009). *Análisis estructural*. Lima, Perú.

Villarreal Castro, G. (2011). *Estática Problemas Resueltos*. Lima, Peru.

White, R. N., Gergely, P., & Sexsmith, R. G. (1972). *Structural engineering [Ingeniería estructural]* (Vol. 1 a 2). John Wiley & Sons. Inc.

Anexo A: Manual de usuario de Ac-Struc

A.1. Inicio de Ac-Struc

Al iniciar el software se muestra la ventana principal (Figura 69), desde esta se puede tener acceso a todas las funcionalidades únicamente a través de la tecla “MENU” del dispositivo portátil.



Figura 69: Ventana de Inicio del Software Ac-Struc

A.2. Nuevo archivo

Para seleccionar la opción de Archivo Nuevo se deben realizar lo siguiente Menú>Archivo>Nuevo, el software automáticamente trasladará al usuario a la ventana de archivo nuevo (Figura 70) en donde podrá seleccionar el tipo de estructura que desea analizar y el método con el cual realizará el análisis.

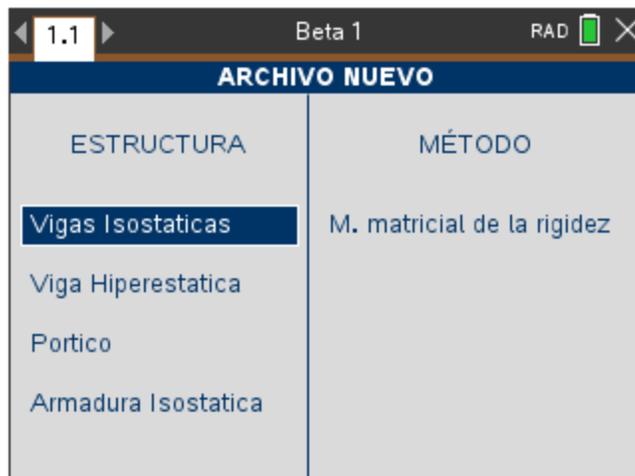


Figura 70: Ventana Archivo Nuevo

A.3. Abrir Archivo

Para seleccionar la opción de Abrir Archivo se deben realizar lo siguiente Menú>Archivo>Abrir, el software automáticamente trasladara al usuario a la ventana de Abrir Archivo (Figura 71) en donde el usuario podrá seleccionar archivo que desea abrir

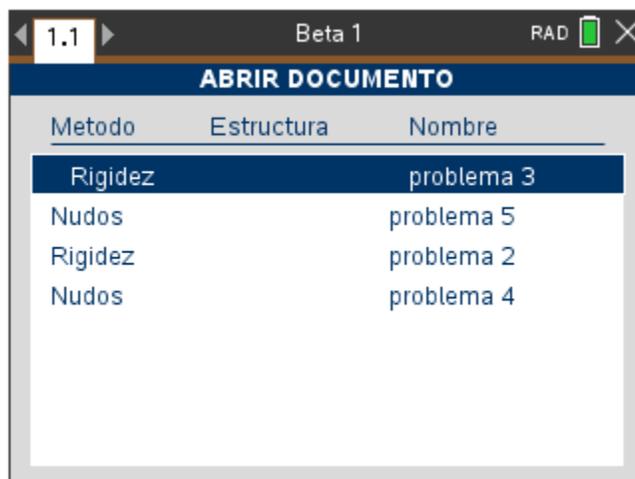


Figura 71: Ventana Abrir Archivo

A.4. Método Matricial de Rigidez

A.4.1. Procedimiento para ingresar datos

A.4.1.1. Agregar Materiales

Para acceder a la opción de agregar materiales es necesario realizar los siguientes pasos
Menú > Insertar > Materiales

Para agregar un nuevo material es necesario presionar el botón “Agregar” que se muestra en pantalla , el software automáticamente trasladara al usuario a la ventana para ingresar los datos correspondientes al material (Figura 72),acto seguido se deberá seleccionar la forma de la sección transversal del elemento y completar los cuadros de texto con los valores correspondientes. Una vez completado el formulario es necesario presionar la tecla “Aceptar” que se muestra en pantalla

The image shows a software dialog box titled "AGREGAR MATERIALES". At the top, there is a navigation bar with "1.1" on the left, "Beta 1" in the center, and "RAD" with a green indicator and a close button on the right. The main area of the dialog contains the following elements:

- A dropdown menu labeled "Sección:" with "Rectangular" selected.
- Two input fields labeled "B=" and "H=" for defining the dimensions of the rectangular section.
- A diagram of a rectangle with "B" labeled for width and "H" for height.
- Two input fields for material properties: "Mod. Elast.=" and "α=" followed by "x10⁻⁶".
- Two buttons at the bottom: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 72: Ventana Agregar Materiales

En caso de no conocer la forma del elemento ni sus dimensiones, o que la forma del elemento no corresponda con las mostradas en el software, el usuario deberá seleccionar la opción de “Personalizado” en las opciones de sección y deberá completar los valores de I, A y E correspondientes a al material a agregar (Figura 73)

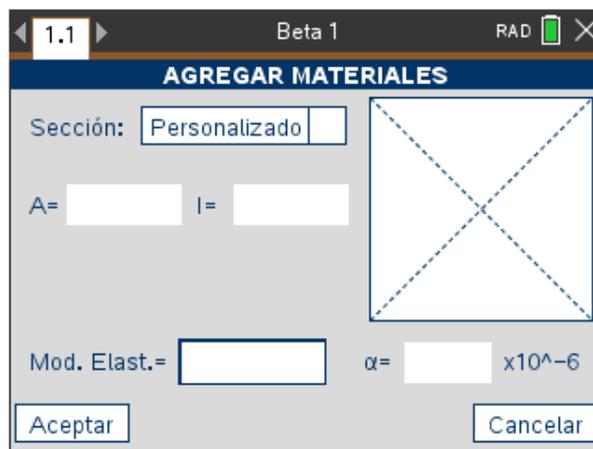


Figura 73: Agregar Sección Personalizada

A.4.1.2. Agregar Nudos

Para acceder a la opción de Agregar Nudos es necesario realizar los siguientes pasos
Menú > Insertar > Nudos.

Para agregar un nuevo nudo, el usuario deberá ingresar en los cuadros de diálogo de X y Y los valores correspondientes a la posición del nudo en el plano cartesiano, acto seguido deberá presionar el botón “Agregar”

Para modificar o eliminar un nudo es necesario que el usuario seleccione el nudo con el cursor, modifique los valores de la posición si es necesario y acto seguido presione la opción que desea realizar, ya sea “Modificar” o “Eliminar”

En la Figura 74 se muestra la apariencia de la ventana Insertar Nudos, así como la apariencia que adquiere un nudo cuando es seleccionado (Nudo 3)

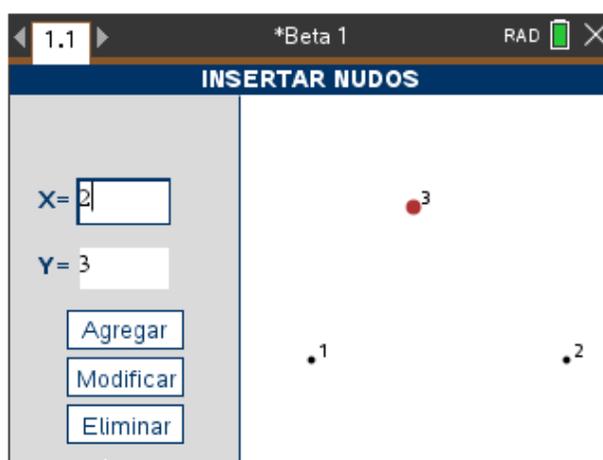


Figura 74: Ventana Insertar Nudos

A.4.1.3. Agregar Elementos

Para acceder a la opción de agregar elementos es necesario realizar los siguientes pasos
Menú > Insertar > Elementos

Para agregar un nuevo elemento, el usuario deberá ingresar en los cuadros de diálogo los valores correspondientes al material del cual está fabricado y los nudos de los cuales inicia y termina el elemento, acto seguido deberá presionar el botón “Agregar”

Para modificar o eliminar un elemento es necesario que el usuario seleccione el elemento con el cursor, modifique los valores del material, el nudo inicial o el nudo final si es necesario y acto seguido presione la opción que desea realizar, ya sea “Modificar” o “Eliminar”

En la Figura 75 se muestra la apariencia de la ventana Insertar Elementos, así como la apariencia que adquiere un elemento cuando es seleccionado (Elemento 2)

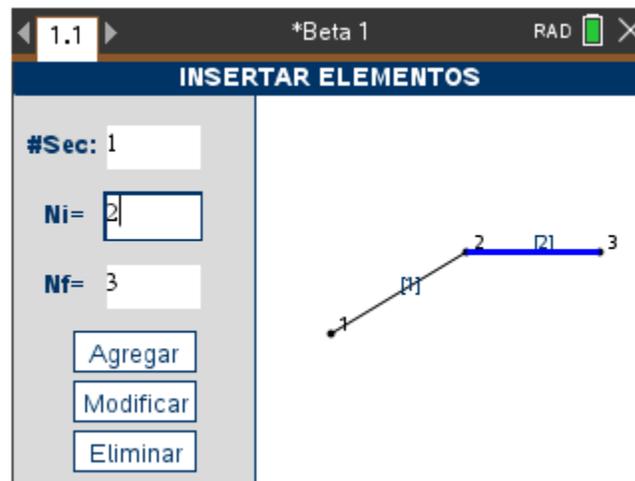


Figura 75: Ventana Insertar Elementos

A.4.1.4. Agregar Apoyos y Enlaces

Para acceder a la opción de agregar Apoyos y Enlaces es necesario realizar los siguientes pasos Menú > Insertar > Apoyos y Enlaces.

Para ingresar o modificar apoyo o enlace el usuario deberá seleccionar la opción que desea y completar en el recuadro el nudo que se desea modificar, acto seguido es necesario presionar la tecla de “Modificar” para cambiar el tipo de apoyo o enlace. El sistema

automáticamente completa el cuadro colocando en el recuadro de datos el valor del nudo siguiente para ayudarle al usuario a ahorrar tiempo ingresando datos.

En la Figura 76 se muestra la apariencia de la ventana de apoyos, así como la apariencia de los nudos con un apoyo tipo: “Empotramiento” (Nudo 1), un enlace de tipo: “Nudo Rígido” (Nudo 2) y un “Apoyo Fijo” (Nudo 3)

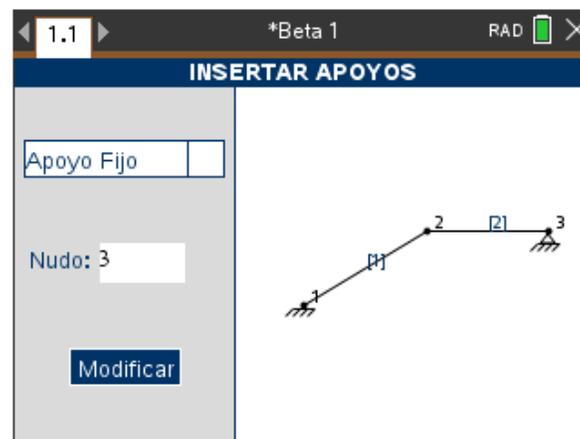


Figura 76: Ventana Insertar Apoyos y Enlaces

A.4.1.5. Insertar Cargas Puntuales

Para acceder a la opción de agregar Cargas Puntuales es necesario realizar los siguientes pasos Menú > Insertar > Cargas Puntuales.

Por motivos de practicidad las cargas puntuales y momentos solo se podrán insertar en los nudos, por lo tanto, si se desea agregar una carga puntual o un momento en la mitad de un

elemento primero será necesario agregar un nudo en la mitad de este tal y como se muestra en la Figura 77.

Para insertar cargas puntuales se debe seleccionar el tipo de carga y en los cuadros de datos se debe ingresar los valores correspondientes, también es posible seleccionar el nudo con el cursor y el sistema automáticamente asignara en los cuadros de datos los valores de las fuerzas y el momento aplicado en el nudo seleccionado.

Para evitar una saturación de información debido a una gran cantidad de datos mostrados en pantalla, las fuerzas y momentos que se muestran en pantalla solo corresponden a aquellos cuyo tipo concuerdan con el tipo que ha sido seleccionado en “Tipo de Carga”, sin embargo, aunque se cambie el tipo de carga los datos permanecerán, solo que no serán visibles por el usuario. Así mismo, en la ventana no se mostrarán las cargas aplicadas sobre los elementos.

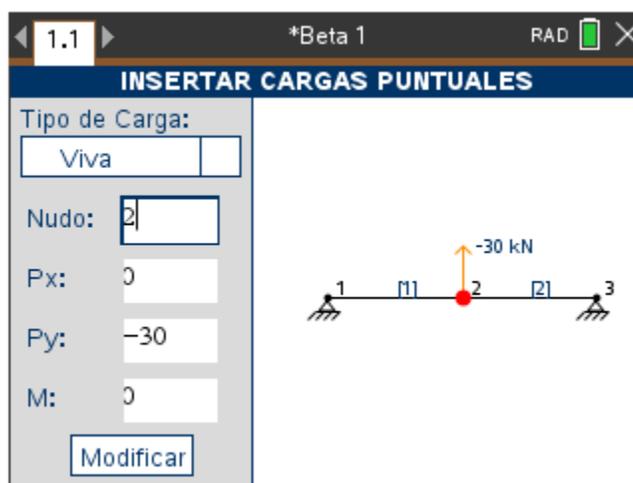


Figura 77: Ventana Insertar Cargas Puntuales

A.4.1.6. Insertar Cargas Distribuidas

Para acceder a la opción de agregar Cargas Distribuidas es necesario realizar los siguientes pasos Menú > Insertar > Cargas Distribuidas.

Para insertar cargas distribuidas o fuerzas axiales se debe seleccionar el tipo de carga y en los cuadros de datos se debe ingresar los valores correspondientes, también es posible seleccionar el elemento con el cursor y el sistema automáticamente asignara en los cuadros de datos los valores de las cargas aplicadas en el elemento, tal y como se muestra en la Figura 78.

Para evitar una saturación de información debido a una gran cantidad de datos mostrados en pantalla, los datos que se muestran en pantalla solo corresponden a aquellos cuyo tipo concuerdan con el tipo que ha sido seleccionado en “Tipo de Carga”, sin embargo, aunque se cambie el tipo de carga los datos permanecerán, solo que no serán visibles por el usuario. Así mismo, en la ventana no se mostrarán las fuerzas y momentos aplicados en los nudos.



Figura 78: Ventana Insertar Cargas distribuidas

A.4.1.7. Definir Combinaciones de Carga

Para acceder a la opción de definir Combinaciones de Carga es necesario realizar los siguientes pasos Menú > Definir > Comb. Cargas.

En pantalla se mostrarán las combinaciones de carga agregadas (Figura 79),



Figura 79: Ventana Definir Combinaciones de Carga

Para agregar una nueva combinación es necesario presionar con el cursor el botón “Agregar” que se encuentra en la parte inferior izquierda. El software automáticamente moverá al usuario a la ventana para definir los valores de la combinación (Figura 80), en esta el usuario deberá ingresar el coeficiente correspondiente a la carga.

El coeficiente del tipo de carga “Unitaria” no puede ser modificado, es decir siempre tendrá el valor de uno (1) por lo tanto el usuario deberá tener especial cuidado cuando ingrese cargas con este tipo de carga. El tipo de carga “Auxiliar” es como su nombre un tipo auxiliar que

le permite al usuario agregar un tipo de carga adicional que no se encuentre predefinido en el sistema.

La deformación por temperatura le permite al usuario decidir si la combinación tendrá en consideración las fuerzas axiales producidas por la deformación termina, es decir, si el usuario no desea que el sistema evalúe las ya mencionadas fuerzas axiales deberá seleccionar la opción de “No”, en el caso contrario si desea que sean evaluadas estas fuerzas deberá seleccionar la opción “Si”.

En el caso de la deformación por temperatura se debe tener una consideración adicional: para que el sistema evalúe la deformación termina en la estructura los materiales deben tener un coeficiente de dilatación termina (α) diferente de cero (0), el cual es un dato solicitado por el sistema en la sección de Agregar Materiales

The image shows a software dialog box titled "AGREGAR COMBINACION". The window title bar includes "1.1", "*Beta 1", "RAD", and a close button. The dialog contains the following elements:

- Unitaria:
- Sismica:
- Viva:
- Viento:
- Muerta:
- Auxiliar:
- Deformación por temperatura: Si No
- Aceptar
- Cancelar

Figura 80: ventana para definir los valores de la combinación

A.4.1.8. Definir Variación de temperatura

Para acceder a la opción de definir Combinaciones de Carga es necesario realizar los siguientes pasos Menú > Definir > Variación Temperatura

En la ventana se le solicitara al usuario que ingrese los valores correspondientes a la Temperatura inicial y Temperatura final del sistema.

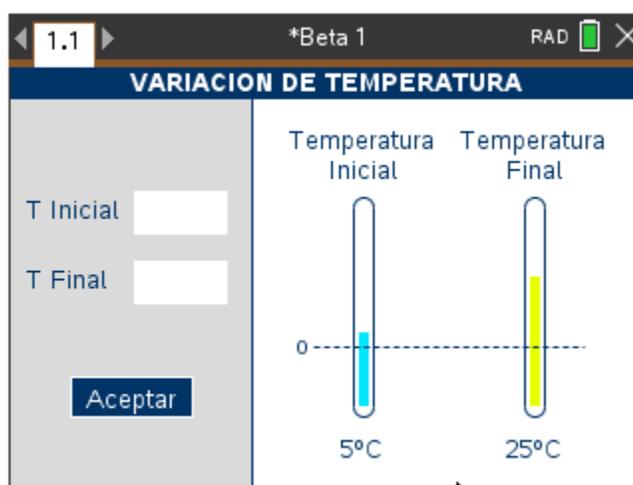


Figura 81: Ventana Variación de Temperatura

A.4.2. Procedimiento para ver los resultados

La opción de guardar y las opciones para visualizar los diferentes tipos de resultados solo se habilitarán una vez que se la estructura haya sido analizada

A.4.2.1. Analizar estructura

Para analizar la estructura es necesario realizar los siguientes pasos, tras ingresar los datos requeridos por el software de debe presionar Menú > Analizar > Analizar Estructura.

A.4.2.2. Deformaciones

Para ver las deformaciones de la estructura es necesario realizar los siguientes pasos Menú > Ver > Deformaciones.

Para ver las deformaciones de los nudos solo es necesario desplazar el cursor sobre el nudo a visualizar. También se pueden observar la totalidad de los valores en la parte izquierda, en caso de que el usuario no pueda ver el dato que desea consultar solo deberá hacer uso de los botones de desplazamiento del teclado (Arriba y Abajo) para ver los datos restantes. Para cambiar la combinación de carga visualizada, el usuario deberá usar los botones de desplazamiento (Izquierda y Derecha)



Figura 82: Ventana Desplazamientos

A.4.2.3. Reacciones, diagrama de fuerza cortante y momento flector

Para ver las reacciones y el diagrama de fuerza cortante o momento flector de la estructura es necesario realizar los siguientes pasos Menú > Ver > Diagrama Cortante o Menú > Ver > Diagrama Momento según corresponda.

En la Figura 83 se muestra la ventana del diagrama de fuerza cortante y las reacciones de la estructura para la combinación seleccionada, la cual se encuentra en la parte superior izquierda.

En caso de que el usuario no pueda ver las reacciones que desea consultar solo deberá hacer uso de los botones de desplazamiento del teclado (Arriba y Abajo) para ver los datos restantes. Para cambiar la combinación seleccionada solo es necesario oprimir los botones de desplazamiento del dispositivo (Izquierda y Derecha).

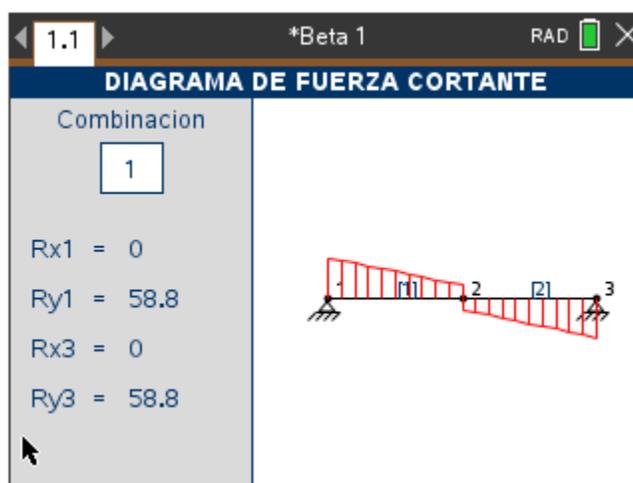


Figura 83: Ventana Diagrama de fuerza cortante

A.4.2.4. Ecuaciones de fuerza cortante y momento flector

Para ver las ecuaciones de la fuerza cortante o el momento flector de la estructura es necesario realizar los siguientes pasos Menú > Ver > Diagrama Cortante Elemento o Menú > Ver > Diagrama Momento Elemento según corresponda.

En estas ventanas el usuario puede ver los diagramas de fuerza cortante o momento flector por elemento, además en la parte inferior se muestra la ecuación del cortante o del momento como se muestra en la Figura 84. Si el usuario desea cambiar la combinación o el elemento mostrado solo deberá usar las teclas de desplazamiento del dispositivo: Con las teclas Izquierda y Derecha puede aumentar o disminuir el número que corresponde a la combinación o el elemento según este seleccionado. Con las teclas Arriba y Abajo el usuario puede alternar entre la combinación, el elemento o la celda X, la cual puede ser completada con el valor de X en el cual se desea conocer un valor específico del cortante o del momento, según corresponda.

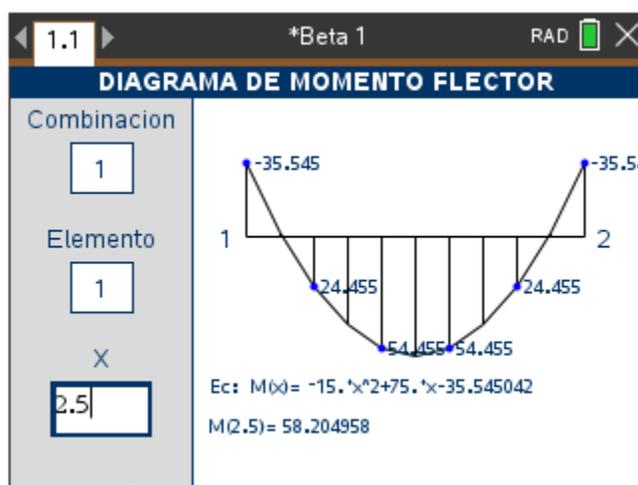


Figura 84: Ventana Diagrama de momento flector por elemento

A.4.2.5. Matriz de rigidez K de los elementos

Para ver la matriz de rigidez de los elementos es necesario realizar los siguientes pasos Menú > Ver > Matriz K Coordenadas Locales o Menú > Ver > Matriz K Coordenadas Globales según corresponda.

Para cambiar el elemento mostrado es necesario que el recuadro de elemento este seleccionado, acto seguido es necesario usar las teclas Izquierda y Derecha del dispositivo para alternar entre elementos. En caso de no estar seleccionado el recuadro de elemento solo se debe presionar el botón direccional “Arriba” para seleccionarlo.

Las filas y columnas de la matriz se encuentran numeradas para evitar confusiones, tal y como se muestra en la Figura 85, si se desea desplazar entre las diferentes columnas de la matriz solo es necesario que este seleccionado la matriz y acto seguido oprimir los botones direccionales “Izquierda” o “Derecha” del dispositivo. Si la matriz no se encuentra seleccionada solo es necesario oprimir el botón direccional “Abajo” del dispositivo

	1	2	3
1	942477.796	0	0
2	0	360642.014	504898.819
3	0	504898.819	942477.796
4	-942477.796	0	0
5	0	-360642.014	-504898.819
6	0	504898.819	471238.898

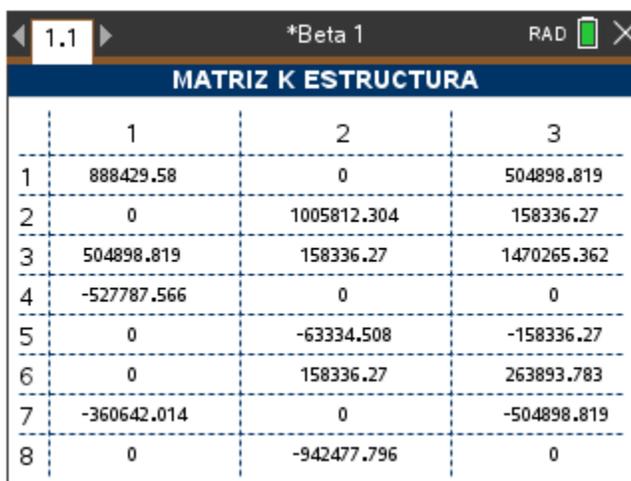
Figura 85: Ventana de matriz K en coordenadas locales

A.4.2.6. Matriz de rigidez K de la estructura

Para ver la matriz de rigidez de la estructura es necesario realizar los siguientes pasos

Menú > Ver > Matriz K Estructura

Las filas y columnas de la matriz de rigidez de la estructura se encuentran numeradas para evitar confusiones, tal y como se muestra en la Figura 86. Si el usuario desea desplazarse entre las diferentes filas y columnas de la matriz, solo es necesario oprimir los botones direccionales.



	1	2	3
1	888429.58	0	504898.819
2	0	1005812.304	158336.27
3	504898.819	158336.27	1470265.362
4	-527787.566	0	0
5	0	-63334.508	-158336.27
6	0	158336.27	263893.783
7	-360642.014	0	-504898.819
8	0	-942477.796	0

Figura 86: Ventana Matriz de rigidez K de la estructura

A.5. Método de Nudos

A.5.1. Procedimiento para ingresar datos

A.5.1.1. Agregar Nudos

Para acceder a la opción de agregar Apoyos es necesario realizar los siguientes pasos
Menú > Insertar > Apoyos.

En esta ventana (Figura 87) el software solicitará que el usuario introduzca los nudos que corresponden a el apoyo fijo (Apoyo 1) y el apoyo móvil (Apoyo 2), además, el ángulo de inclinación(en grados) de la superficie en donde se encuentra el apoyo móvil.

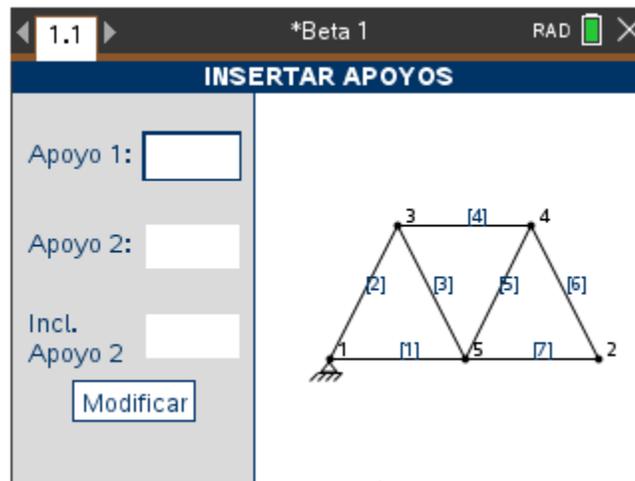


Figura 87: Ventana Insertar apoyos método de nudos

A.5.2. Visualización de resultados

A.5.2.1. Reacciones en los apoyos

Para ver las reacciones en los apoyos es necesario realizar los siguientes pasos Menú > Ver > Reacciones Apoyos. Para realizar el cambio entre la combinación mostrada se debe presionar la tecla direccional “derecha” o la tecla direccional “izquierda” del dispositivo. La reacción del apoyo secundario (apoyo móvil) es perpendicular a la superficie sobre la cual se encuentra apoyado el mismo.



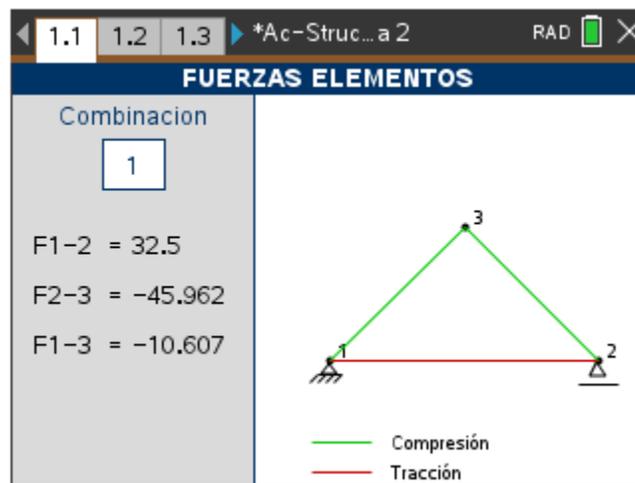
Figura 88: Ventana de reacciones de las armaduras isostáticas

A.5.2.2. Fuerzas internas en las barras

Para ver las fuerzas internas en las barras es necesario realizar los siguientes pasos Menú > Ver > Fuerzas Barras. Si un elemento se encuentra sometido a tracción, este se mostrará de color rojo, de la misma forma el valor de la reacción será positivo, por el contrario, si un

elemento se encuentra sometido a compresión, este se mostrará de color verde en la pantalla, el valor correspondiente será negativo. esto se debe a la consideración inicial la cual indica que todos los elementos se encuentran sometidos a tracción, es decir, las fuerzas salen de los nudos.

El cambio de la combinación mostrada se realiza a través de las teclas direccionales “izquierda” y “derecha”. En caso de que no se pueda ver la totalidad de las fuerzas internas en las barras solo es necesario usar las teclas direccionales “Arriba” y “Abajo” para visualizar las fuerzas faltantes.



A.6. Tres momentos

A.6.1. Procedimiento para ingresar datos

A.6.1.1. Agregar nudos

El software bloquea automáticamente la casilla para ingresar datos de la posición del nudo en el eje Y, además, para evitar errores es necesario que el usuario ingrese los nudos de izquierda a derecha en orden, también se deben incluir como nudos ambos extremos de un elemento en voladizo. En la Figura 89 se muestra la manera correcta de ingresar las coordenadas de los nudos para una viga hiperestática la cual tiene un voladizo en su parte izquierda

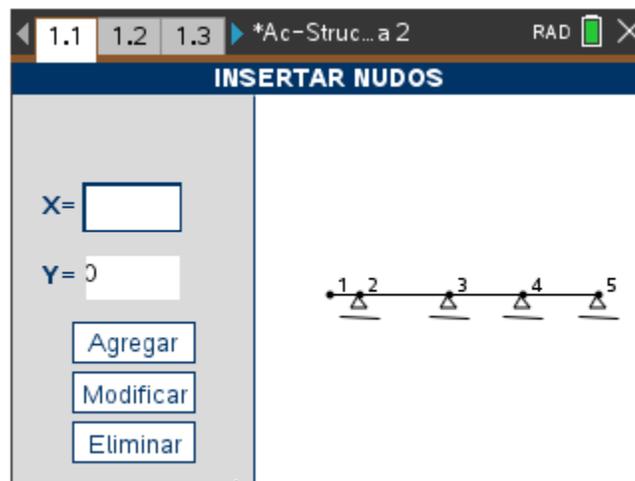


Figura 89: Ventana de ingresar nudos para el cálculo de vigas hiperestáticas

A.6.1.2. Agregar elementos

Similar a los nudos, los elementos deben ingresarse en orden de izquierda a derecha, además, estos no deben saltarse nudos, por ejemplo, un elemento no puede tener como nudo inicial (3) y como nudo final (5) ya que en este caso no se está teniendo en cuenta el nudo 4. La manera correcta es ingresar dos elementos uno del nudo (3) al nudo (4) y otro del nudo (4) al nudo (5)

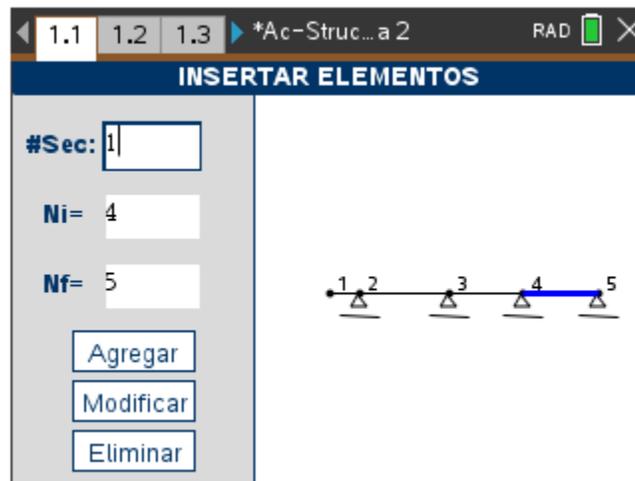


Figura 90: Ventana de ingresar elementos para el cálculo de vigas hiperestáticas

Anexo B: Guía para el análisis de estructuras

B.1. Guía para el análisis de estructuras usando el método matricial de rigidez

Los elementos de la estructura de la Figura 91 tienen un módulo de elasticidad $E=21\text{kN/mm}^2$.

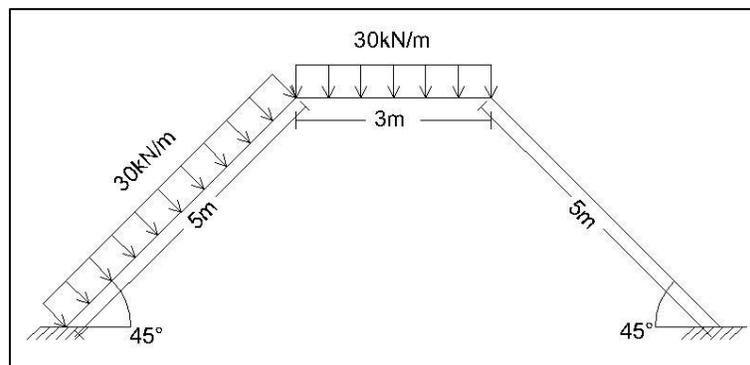


Figura 91: Estructura hiperestática

B.1.1. Análisis detallado

Para analizar la estructura se numerarán a continuación una serie de pasos los cuales permitirán el cálculo de las deformaciones y reacciones de la estructura.

B.1.1.1. Paso 1: Numerar los nudos y los elementos de la estructura.

La numeración de los nudos y elementos de la estructura es arbitraria, sin embargo, para este ejemplo práctico se numeran de la siguiente manera (Figura 92)

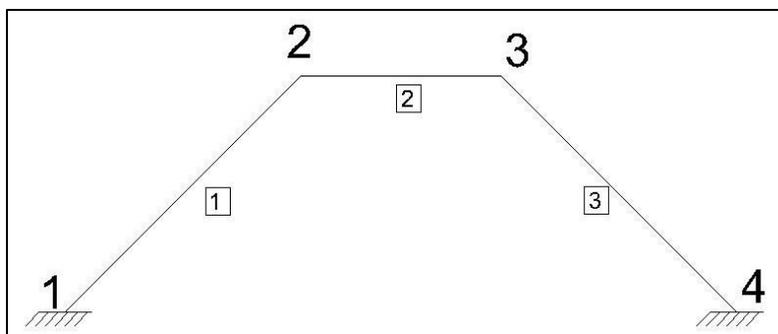


Figura 92: Numeración de nudos y elementos

B.1.1.2. Paso 2: Completar la tabla de datos

A partir de los datos obtenidos a partir del planteamiento del problema y el la ilustración de la Figura 92 se debe completar la Tabla 8 y la Tabla 9, las cuales se muestran a continuación.

Tabla 8: Tabla de datos metodo matricial de rigidez

Elemento	Nudo inicial	Nudo final	Longitud (m)	B (m)	H (m)	Área (m ²)	Inercia (m ²)	Modulo Elasticidad (kN/m ²)	θ (Grados)
1	1	2	5	0.4	0.4	0.16	0.0021	21000000	45
2	2	3	3	0.3	0.3	0.09	0.0007	21000000	0
3	3	4	5	0.4	0.4	0.16	0.0021	21000000	-45

Tabla 9: Tabla de rigideces método matricial de rigidez

Elemento	AE	EI	AE/L	12EI/L ³	6EI/L ²	4EI/L	2EI/L	Sen(θ)	Cos(θ)
1	3360000	44800	672000	4300.8	10752	35840	17920	0.707	0.707
2	1890000	14175	630000	6300	9450	18900	9450	0	1
3	3360000	44800	672000	4300.8	10752	35840	17920	-0.707	0.707

Donde:

B y H= son las dimensiones de la sección transversal del elemento.

Área (A) = Área de la sección transversal del elemento.

Inercia (I)= Para una sección rectangular, $I = \frac{B*H^3}{12}$

Θ = Angulo de inclinación del elemento

B.1.1.3. Paso 3: Ensamblar la matriz de rigidez de cada elemento.

Para ensamblar la matriz de rigidez de cada elemento en coordenadas locales es necesario tener en cuenta los datos de la Tabla 9 y en la Ecuación 10.

Ecuación 10: Matriz de rigidez K coordenadas locales

$$[K_L] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

Debido a que elemento 1 y el elemento 3 tienen las mismas dimensiones y están compuestas por el mismo material podemos considerar que $K_{1L} = K_{3L}$

$$[K_{1L}] = [K_{3L}] = \begin{bmatrix} 672000 & 0 & 0 & -672000 & 0 & 0 \\ 0 & 4300.8 & 10752 & 0 & -4300.8 & 10752 \\ 0 & 10752 & 35840 & 0 & -10752 & 17920 \\ -672000 & 0 & 0 & 672000 & 0 & 0 \\ 0 & -4300.8 & -10752 & 0 & 4300.8 & -10752 \\ 0 & 10752 & 17920 & 0 & -10752 & 35840 \end{bmatrix}$$

$$[K_{2L}] = \begin{bmatrix} 630000 & 0 & 0 & -630000 & 0 & 0 \\ 0 & 6300 & 9450 & 0 & -6300 & 9450 \\ 0 & 9450 & 18900 & 0 & -9450 & 9450 \\ -630000 & 0 & 0 & 630000 & 0 & 0 \\ 0 & -6300 & -9450 & 0 & 6300 & -9450 \\ 0 & 9450 & 9450 & 0 & -9450 & 18900 \end{bmatrix}$$

B.1.1.4. Paso 4: Reacciones de los elementos

Completar la Tabla 10 la cual muestra las reacciones de los apoyos de cada elemento, para esto es necesario asumir que el elemento no se encuentra inclinado, es decir, es paralelo al eje X. Además, se debe considerar que el elemento se encuentra empotrado en ambos extremos.

Tabla 10: Tabla de reacciones de los elementos

Elemento	W	RxI	RxJ	RyI	RyJ	MI	MJ
1	30	0	0	75	75	62.5	-62.5
2	30	0	0	45	45	22.5	-22.5
3	0	0	0	0	0	0	0

B.1.1.5. Paso 5: Ensamblar el vector de empotramiento

Para ensamblar el vector de empotramiento en coordenadas locales de cada elemento es necesario tener cuenta los datos de la Tabla 10 y la Ecuación 11

Ecuación 11: Vector de empotramiento en coordenadas locales

$$Emp_L = \begin{bmatrix} RxI \\ RyI \\ MI \\ RxJ \\ RyJ \\ MJ \end{bmatrix}$$

$$Emp_{1L} = \begin{bmatrix} 0 \\ 75 \\ 62.5 \\ 0 \\ 75 \\ -62.5 \end{bmatrix}; \quad Emp_{2L} = \begin{bmatrix} 0 \\ 45 \\ 22.5 \\ 0 \\ 45 \\ -22.5 \end{bmatrix}; \quad Emp_{3L} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.1.1.6. Paso 6: Ensamblar la matriz de transformación de cada elemento

Para ensamblar la matriz de transformación de cada elemento basándose en los datos de la Tabla 9 y en la Ecuación 12

Ecuación 12: Matriz de transformación

$$[T] = \begin{bmatrix} \text{Cos}(\theta) & -\text{Sen}(\theta) & 0 & 0 & 0 & 0 \\ \text{Sen}(\theta) & \text{Cos}(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{Cos}(\theta) & -\text{Sen}(\theta) & 0 \\ 0 & 0 & 0 & \text{Sen}(\theta) & \text{Cos}(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 0.707 & -0.707 & 0 & 0 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.707 & -0.707 & 0 \\ 0 & 0 & 0 & 0.707 & 0.707 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$; T_3 = \begin{bmatrix} 0.707 & 0.707 & 0 & 0 & 0 & 0 \\ -0.707 & 0.707 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.707 & 0.707 & 0 \\ 0 & 0 & 0 & -0.707 & 0.707 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

B.1.1.7. Paso 7 Matriz K de coordenadas locales a coordenadas globales

Convertir la matriz de rigidez de coordenadas locales a coordenadas globales utilizando

la Ecuación 13

Ecuación 13: Matriz de rigidez en coordenadas globales

$$[K_g] = [T] * [K] * [T^T]$$

Dónde: $[T^T]$ es la matriz de transformación transpuesta

Ya que el elemento 2 tiene una inclinación de $\theta = 0^\circ$ entonces:

$$[K_{2G}] = [K_{2L}]$$

Para los elementos 1 y 3:

$$[K_{1G}] = \begin{bmatrix} 338150.4 & 333849.6 & -7602.8 & -338150.4 & -333849.6 & -7602.8 \\ 333849.6 & 338150.4 & 7602.8 & -333849.6 & -338150.4 & 7602.8 \\ -7602.8 & 7602.8 & 35840 & 7602.8 & -7602.8 & 17920.8 \\ -338150.4 & -333849.6 & 7602.8 & 338150.4 & 333849.6 & 7602.8 \\ -333849.6 & -338150.4 & -7602.8 & 333849.6 & 338150.4 & -7602.8 \\ -7602.8 & 7602.8 & 17920 & 7602.8 & -7602.8 & 35840 \end{bmatrix}$$

$$[K_{3G}] = \begin{bmatrix} 338150.4 & -333849.6 & 7602.8 & -338150.4 & 333849.6 & 7602.8 \\ -333849.6 & 338150.4 & 7602.8 & 333849.6 & -338150.4 & 7602.8 \\ 7602.8 & 7602.8 & 35840 & -7602.8 & -7602.8 & 17920.8 \\ -338150.4 & 333849.6 & -7602.8 & 338150.4 & -333849.6 & -7602.8 \\ 333849.6 & -338150.4 & -7602.8 & -333849.6 & 338150.4 & -7602.8 \\ 7602.8 & 7602.8 & 17920 & -7602.8 & -7602.8 & 35840 \end{bmatrix}$$

B.1.1.8. Paso 8

Convertir del vector de empotramiento de coordenadas locales a coordenadas globales usando la Ecuación 14

Ecuación 14: Vector de empotramiento en coordenadas globales

$$[Emp_g] = [Emp_L] * [T]$$

$$Emp_{1g} = \begin{bmatrix} -53.033 \\ 53.033 \\ 62.5 \\ -53.033 \\ 53.033 \\ -62.5 \end{bmatrix}; \quad Emp_{2g} = \begin{bmatrix} 0 \\ 45 \\ 22.5 \\ 0 \\ 45 \\ -22.5 \end{bmatrix}; \quad Emp_{3g} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.1.1.9. Paso 9

Ensamblar la matriz de rigidez K de la estructura usando las matrices de rigidez K de cada elemento en coordenadas globales, para esto se tendrá en cuenta que el tamaño de la matriz K de la estructura depende de la cantidad de nudos que esta tenga multiplicada por tres, en este caso, la matriz de la estructura tendrá un tamaño de 12x12 debido a que tiene 4 nudos.

Para ensamblar de manera sencilla la estructura la matriz de rigidez K de la estructura se dividirá en grupos de 3 columnas y 3 filas, cada grupo de filas y cada grupo de columnas debe ser numerado en orden como se muestra en la Figura 93

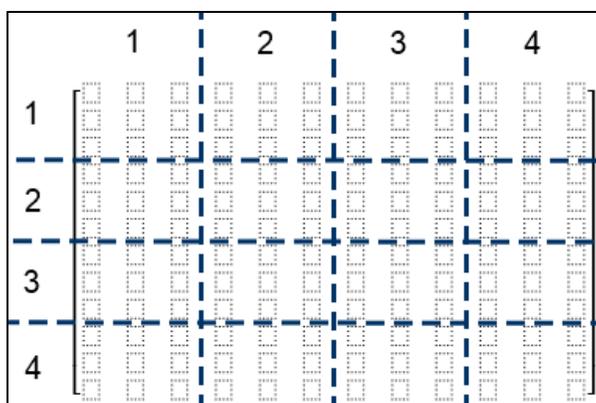


Figura 93: Agrupación de filas y columnas en la matriz de rigidez K de la estructura

De la misma forma se dividirá la matriz de rigidez de cada elemento en coordenadas globales, sin embargo, la numeración corresponde a al nudo inicial y al nudo final, en ese orden. Finalizada la numeración, se procede a agregar los valores de la matriz K de los elementos en la matriz K de la estructura, para esto se tendrá en cuenta la numeración, cada grupo de valores de la matriz K de los elementos deberá ser colocado en las mismas numeraciones de la matriz K de la estructura como se muestra en la Figura 94

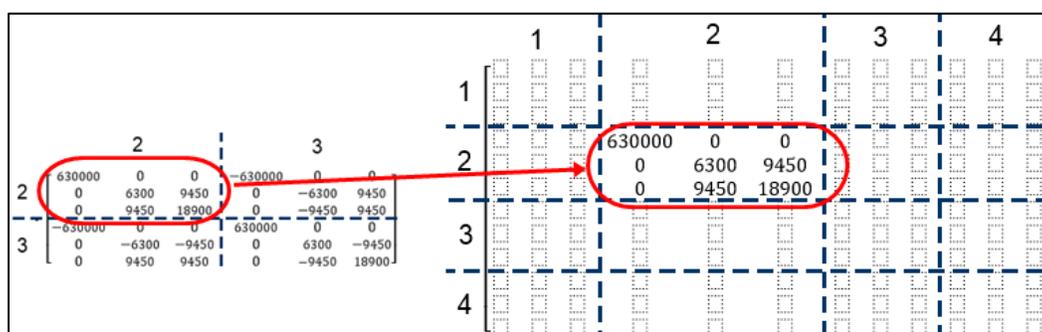


Figura 94: Método para ensamblar la matriz K de la estructura

De esta forma se debe realizar para todos los elementos, en caso de que el grupo de datos se sobreponga a otro, suceso que es muy común, se deben sumar ambos grupos de datos, en la Figura 95 se muestra la matriz de rigidez K de la estructura completamente ensamblada

$$[K] = \begin{bmatrix} 338150.4 & 333849.6 & -7602.8 & -338150.4 & -333849.6 & -7602.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 333849.6 & 338150.4 & 7602.8 & -333849.6 & -338150.4 & 7602.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7602.8 & 7602.8 & 35840 & 7602.8 & -7602.8 & 17920 & 0 & 0 & 0 & 0 & 0 & 0 \\ -338150.4 & -333849.6 & 7602.8 & 968150.4 & 333849.6 & 7602.8 & -630000 & 0 & 0 & 0 & 0 & 0 \\ -333849.6 & -338150.4 & -7602.8 & 333849.6 & 344450.4 & 1847.2 & 0 & -6300 & 9450 & 0 & 0 & 0 \\ -7602.8 & 7602.8 & 17920 & 7602.8 & 1847.2 & 54740 & 0 & -9450 & 9450 & 0 & 0 & 0 \\ 0 & 0 & 0 & -630000 & 0 & 0 & 968450.4 & -333849.6 & 7602.8 & -338150.4 & 333849.6 & 7602.8 \\ 0 & 0 & 0 & 0 & -6300 & -9450 & -333849.6 & 344450.4 & -1847.2 & 333849.6 & -338150.4 & 7602.8 \\ 0 & 0 & 0 & 0 & 9450 & 9450 & 7602.8 & -1847.2 & 54740 & -7602.8 & -7602.8 & 7602.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & -338150.4 & 333849.6 & -7602.8 & 338150.4 & -333849.6 & -7602.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 333849.6 & -338150.4 & -7602.8 & -333849.6 & 338150.4 & -7602.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7602.8 & 7602.8 & 17920 & -7602.8 & -7602.8 & 35840.8 \end{bmatrix}$$

Figura 95: Matriz de rigidez K de la estructura

B.1.1.10. Paso 10

Ensamblar el vector de deformaciones, para esto se tendrá en cuenta cada uno de los tipos de apoyos o enlaces de la estructura, el número de filas del vector de deformaciones es igual al número de columnas de la matriz de rigidez K de la estructura. Las filas del vector de deformaciones se deben agrupar y numerar de manera similar a la matriz de rigidez K de la estructura, tal y como se muestra en la Figura 96.

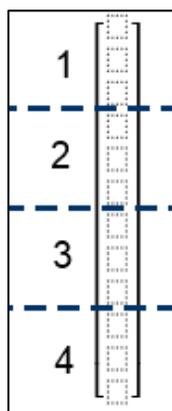


Figura 96: Agrupación de filas en el vector de deformaciones

El primer recuadro de cada agrupación corresponde al desplazamiento en X del nudo equivalente al número de dicha agrupación, el segundo recuadro corresponde al desplazamiento en Y del nudo equivalente al número de dicha agrupación y el tercer recuadro corresponde al giro del nudo equivalente al número de dicha agrupación. Si el nudo analizado posee deformación en el eje X entonces el recuadro se debe completar con la simbología: “dx-n”, si el nudo no posee deformación en el eje X se debe completar con Cero (0). Si el nudo analizado posee deformación en el eje Y el recuadro se debe completar con:” dy-n” en caso contrario se completa con Cero (0). Si el nudo analizado posee giro se debe completar con: “g-n”, en el caso contrario se completa con Cero (0). En donde “n” representa el nudo analizado.

Para la estructura propuesta los nudos 1 y 4 corresponden a empotramientos, por lo cual, carece de giros y deformación de ambos sentidos. Los nudos 2 y 3 son nudos rígidos lo que implica que posee giro y deformación en ambos sentidos. A continuación, se muestra el vector de empotramiento de la estructura planteada.

$$\delta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ dx2 \\ dy2 \\ g2 \\ dx3 \\ dy3 \\ g3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.1.1.11. Paso 11

Ensamblar el vector de reacciones de la estructura, para esto es necesario partir del vector de deformaciones calculado en la sección B.1.1.10. Paso 10 la cual corresponde al Paso 10. Si el valor de la deformación en el eje X es cero (0) entonces el valor en el vector de reacciones toma el valor de “Rx-n”; si el valor de la deformación en el eje Y es cero, el valor en el vector de reacciones toma el valor de “Ry-n”; si el valor del giro es cero, el valor en el vector de reacciones toma el valor de “M-n”. En caso de que la casilla del vector deformaciones tenga un valor diferente a cero (0) entonces inmediatamente el valor de la casilla de reacciones toma el valor de cero (0). El vector de reacciones de la estructura se muestra a continuación.

$$R = \begin{bmatrix} Rx1 \\ Ry1 \\ M1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ Rx4 \\ Ry4 \\ M4 \end{bmatrix}$$

B.1.1.12. Paso 12

Ensamblar el vector de empotramiento de la estructura, para este paso se realizará un procedimiento similar al realizado en la sección B.1.1.9. Paso 9 Sin embargo, se utilizaran los vectores de empotramiento en coordenadas globales.

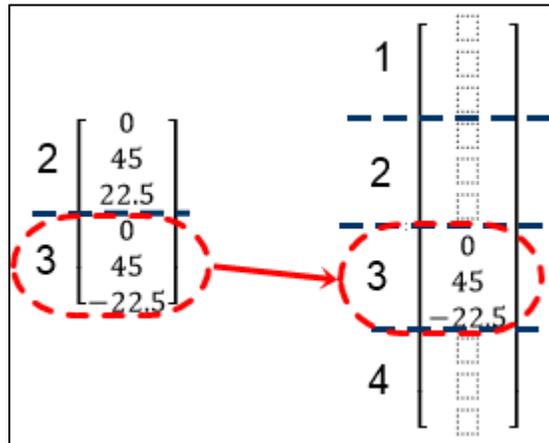


Figura 97: Método para ensamblar el vector de empotramiento de la estructura

$$Emp = \begin{bmatrix} -53.03 \\ 53.03 \\ 62.5 \\ 53.03 \\ 98.03 \\ -40 \\ 0 \\ 45 \\ -22.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.1.1.13. Paso 13

Simplificar la matriz de rigidez K y el vector de deformaciones de la estructura. La simplificación debe realizarse de la siguiente manera, si la fila del vector de deformaciones tiene el valor de cero (0) entonces se elimina junto con la columna equivalente en la matriz de rigidez K de la estructura, tal y como se muestra en la Figura 98

por el contrario no tiene el valor de cero (0) entonces la fila de la matriz de rigidez K simplifica, la fila del vector de empotramientos y la fila del vector de reacciones harán parte de la Parte B. En la Figura 99 se muestra el proceso mencionado anteriormente resaltando de color Azul las filas de la Parte A y de color Amarillo las filas de la Parte B.

R_{x1}	-338150.4	-333849.6	-7602.8	0	0	0	-53.03
R_{y1}	-333849.6	-338150.4	7602.8	0	0	0	53.03
$M1$	7602.8	-7602.8	17920	0	0	0	62.5
0	968150.4	333849.6	7602.8	-630000	0	0	53.03
0	333849.6	344450.4	1847.2	0	-6300	9450	98.03
0	7602.8	1847.2	54740	0	-9450	9450	-40
0	-630000	0	0	968450.4	-333849.6	7602.8	0
0	0	-6300	-9450	-333849.6	344450.4	-1847.2	45
0	0	9450	9450	7602.8	-1847.2	54740	-22.5
R_{x4}	0	0	0	-338150.4	333849.6	-7602.8	0
R_{y4}	0	0	0	333849.6	-338150.4	-7602.8	0
$M4$	0	0	0	7602.8	7602.8	17920	0

Figura 99: Parte A y Parte B de la matriz K simplificada, el vector reacción y el vector empotramiento

La matriz K simplificada, el vector de empotramiento y el vector de reacciones de la parte A se llamarán " K_A ", " Emp_A " y " R_A ", respectivamente. La matriz K simplificada, el vector de empotramiento y el vector de reacciones de la parte B se llamarán " K_B ", " Emp_B " y " R_B ", respectivamente.

B.1.1.15. Paso 15 Calculo de las deformaciones de la estructura

Para calcular las deformaciones de la estructura usara la Ecuación 15, donde $[K_A]^{-1}$ es la matriz inversa de la matriz K_A

Ecuación 15: Calculo de deformaciones de la estructura

$$[\delta] = [K_A]^{-1} * ([R_A] - [Emp_A])$$

$$\begin{bmatrix} dx2 \\ dy2 \\ g2 \\ dx3 \\ dy3 \\ g3 \end{bmatrix} = \begin{bmatrix} 968150.4 & -333849.6 & -7602.8 & -630000 & 0 & 0 \\ -333849.6 & 344450.4 & 1847.2 & 0 & -6300 & 9450 \\ -7602.8 & 1847.2 & 54740 & 0 & -9450 & 9450 \\ -630000 & 0 & 0 & 968450.4 & 333849.6 & -7602.8 \\ 0 & -6300 & -9450 & 333849.6 & 344450.4 & -1847.2 \\ 0 & 9450 & 9450 & -7602.8 & -1847.2 & 54740 \end{bmatrix}^{-1} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 53.03 \\ 98.03 \\ -40 \\ 0 \\ 45 \\ -22.5 \end{pmatrix}$$

$$\begin{bmatrix} dx2 \\ dy2 \\ g2 \\ dx3 \\ dy3 \\ g3 \end{bmatrix} = \begin{bmatrix} 2.7 \times 10^{-3} \\ -2.9 \times 10^{-3} \\ 7.7 \times 10^{-4} \\ 2.6 \times 10^{-3} \\ 2.3 \times 10^{-3} \\ 5.0 \times 10^{-4} \end{bmatrix}$$

B.1.1.16. Paso 16 Calculo de las reacciones de la estructura

Para calcular las reacciones de la estructura usara la Ecuación 16 y los valores de las deformaciones calculados en la sección B.1.1.16. Paso 16 Calculo de las reacciones de la estructura.

Ecuación 16: Calculo de reacciones de la estructura

$$[R_B] = [K_B] * [\delta] + [Emp_B]$$

$$\begin{bmatrix} Rx1 \\ Ry1 \\ M1 \\ Rx4 \\ Ry4 \\ M4 \end{bmatrix} = \begin{bmatrix} -338150.4 & 333849.6 & 7602.8 & 0 & 0 & 0 \\ 333849.6 & -338150.4 & 7602.8 & 0 & 0 & 0 \\ -7602.8 & -7602.8 & 17920 & 0 & 0 & 0 \\ 0 & 0 & 0 & -338150.4 & -333849.6 & 7602.8 \\ 0 & 0 & 0 & -333849.6 & -338150.4 & -7602.8 \\ 0 & 0 & 0 & -7602.8 & 7602.8 & 17920 \end{bmatrix} * \begin{bmatrix} dx2 \\ dy2 \\ g2 \\ dx3 \\ dy3 \\ g3 \end{bmatrix} + \begin{bmatrix} -53.03 \\ 53.03 \\ 62.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Rx1 \\ Ry1 \\ M1 \\ Rx4 \\ Ry4 \\ M4 \end{bmatrix} = \begin{bmatrix} -11.6 \\ 130.2 \\ 118.8 \\ -94.74 \\ 65.86 \\ 4609 \end{bmatrix}$$

B.1.1.17. Paso 17 Deformaciones y reacciones de cada elemento en coordenadas

locales

Para convertir las deformaciones y las reacciones de coordenadas globales a coordenadas locales es necesario hacer uso de la Ecuación 17 y de la Ecuación 18, respectivamente. Donde n corresponde al número del elemento que se desea analizar

Ecuación 17: Vector de deformaciones en coordenadas locales

$$[\delta_{nL}] = [T_n]^T * [\delta_{nG}]$$

Ecuación 18: Vector de reacciones en coordenadas locales

$$[R_{nL}] = [T_n]^T * [R_{nG}]$$

$$\begin{bmatrix} dx_{1L} \\ dy_{1L} \\ g_{1L} \\ dx_{2L} \\ dy_{2L} \\ g_{2L} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.25x10^{-4} \\ -3.96x10^{-3} \\ 7.66x10^{-4} \end{bmatrix}; \quad \begin{bmatrix} dx_{2L} \\ dy_{2L} \\ g_{2L} \\ dx_{3L} \\ dy_{3L} \\ g_{3L} \end{bmatrix} = \begin{bmatrix} 2.71x10^{-3} \\ -2.89x10^{-3} \\ 7.66x10^{-4} \\ 2.56x10^{-3} \\ 2.32x10^{-3} \\ 5.0x10^{-4} \end{bmatrix}; \quad \begin{bmatrix} dx_{3L} \\ dy_{3L} \\ g_{3L} \\ dx_{4L} \\ dy_{4L} \\ g_{4L} \end{bmatrix} = \begin{bmatrix} 1.69x10^{-4} \\ 3.45x10^{-3} \\ 5.0x10^{-4} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Rx_{1L} \\ Ry_{1L} \\ M_{1L} \\ Rx_{2L} \\ Ry_{2L} \\ M_{2L} \end{bmatrix} = \begin{bmatrix} 83.87 \\ 100.27 \\ 118.8 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad \begin{bmatrix} Rx_{2L} \\ Ry_{2L} \\ M_{2L} \\ Rx_{3L} \\ Ry_{3L} \\ M_{3L} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad \begin{bmatrix} Rx_{3L} \\ Ry_{3L} \\ M_{3L} \\ Rx_{4L} \\ Ry_{4L} \\ M_{4L} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -113.37 \\ -20.23 \\ 46.09 \end{bmatrix}$$

B.1.1.18. Paso 18 Fuerzas internas de cada elemento

Para el cálculo de las fuerzas internas de cada elemento se aplica la Ecuación 19.

Ecuación 19: Fuerzas en los extremos de cada elemento

$$[F_n] = [K_{nL}] * [\delta_{nL}]$$

$$\begin{bmatrix} Fx_1 \\ Fy_1 \\ M_1 \\ Fx_2 \\ Fy_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} 672000 & 0 & 0 & -672000 & 0 & 0 \\ 0 & 4300.8 & 10752 & 0 & -4300.8 & 10752 \\ 0 & 10752 & 35840 & 0 & -10752 & 17920 \\ -672000 & 0 & 0 & 672000 & 0 & 0 \\ 0 & -4300.8 & -10752 & 0 & 4300.8 & -10752 \\ 0 & 10752 & 17920 & 0 & -10752 & 35840 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.25x10^{-4} \\ -3.96x10^{-3} \\ 7.66x10^{-4} \end{bmatrix} = \begin{bmatrix} 83.87 \\ 25.27 \\ 56.30 \\ -83.87 \\ -25.27 \\ 70.03 \end{bmatrix}$$

$$\begin{bmatrix} Fx_2 \\ Fy_2 \\ M_2 \\ Fx_3 \\ Fy_3 \\ M_3 \end{bmatrix} = \begin{bmatrix} 672000 & 0 & 0 & -672000 & 0 & 0 \\ 0 & 4300.8 & 10752 & 0 & -4300.8 & 10752 \\ 0 & 10752 & 35840 & 0 & -10752 & 17920 \\ -672000 & 0 & 0 & 672000 & 0 & 0 \\ 0 & -4300.8 & -10752 & 0 & 4300.8 & -10752 \\ 0 & 10752 & 17920 & 0 & -10752 & 35840 \end{bmatrix} * \begin{bmatrix} 2.71x10^{-3} \\ -2.89x10^{-3} \\ 7.66x10^{-4} \\ 2.56x10^{-3} \\ 2.32x10^{-3} \\ 5.0x10^{-4} \end{bmatrix} = \begin{bmatrix} 94.47 \\ -20.86 \\ -30.03 \\ -94.47 \\ 20.86 \\ -32.55 \end{bmatrix}$$

$$\begin{bmatrix} Fx_3 \\ Fy_3 \\ M_3 \\ Fx_4 \\ Fy_4 \\ M_4 \end{bmatrix} = \begin{bmatrix} 672000 & 0 & 0 & -672000 & 0 & 0 \\ 0 & 4300.8 & 10752 & 0 & -4300.8 & 10752 \\ 0 & 10752 & 35840 & 0 & -10752 & 17920 \\ -672000 & 0 & 0 & 672000 & 0 & 0 \\ 0 & -4300.8 & -10752 & 0 & 4300.8 & -10752 \\ 0 & 10752 & 17920 & 0 & -10752 & 35840 \end{bmatrix} * \begin{bmatrix} 1.69x10^{-4} \\ 3.45x10^{-3} \\ 5.0x10^{-4} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 113.37 \\ 20.23 \\ 55.05 \\ -113.37 \\ -20.23 \\ 46.09 \end{bmatrix}$$

B.1.2. Análisis usando el software Ac-struct

B.1.2.1. Paso 1: Nuevo archivo

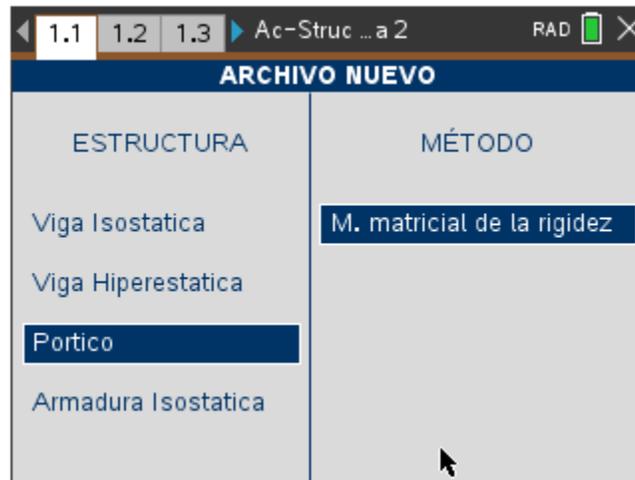


Figura 100: Nuevo pórtico, metodo matricial de rigidez

B.1.2.2. Paso 2: Agregar materiales

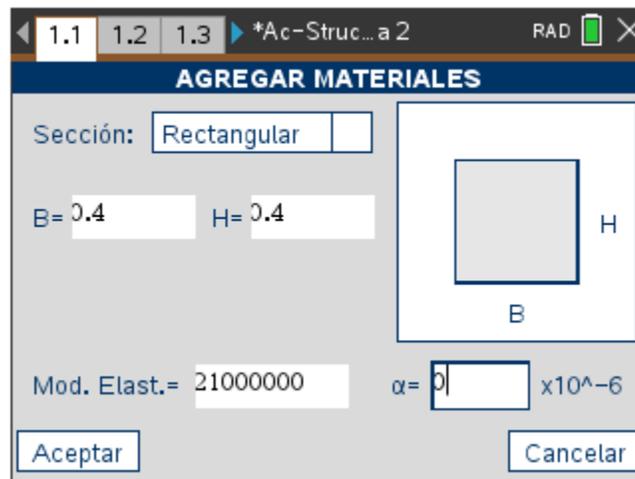


Figura 101: Sección rectangular 0.4m x 0.4m



Figura 102: Materiales agregados

B.1.2.3. Paso 3: Agregar nudos

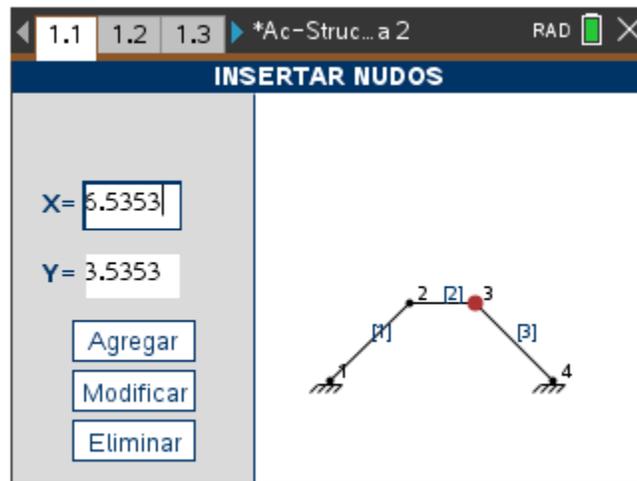


Figura 103: Nudos insertados

B.1.2.4. Paso 4: Agregar elementos

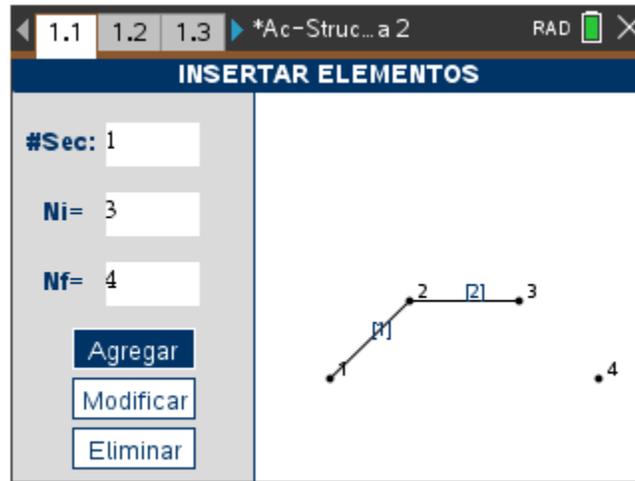


Figura 104: Insertar elementos:

B.1.2.5. Paso 5: Agregar apoyos

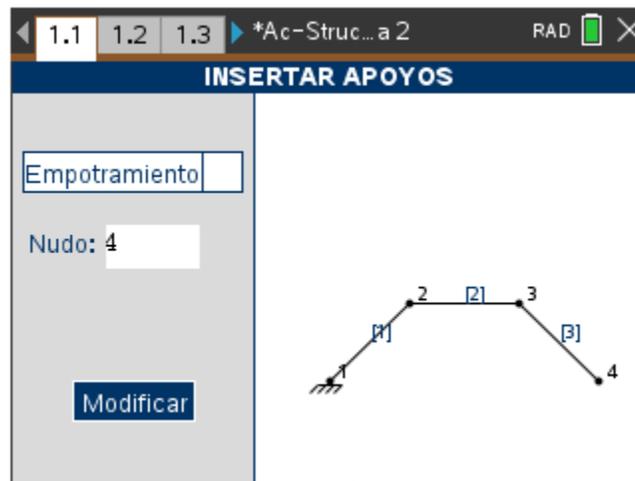


Figura 105: Insertar apoyos

B.1.2.6. Paso 6: Agregar cargas puntuales

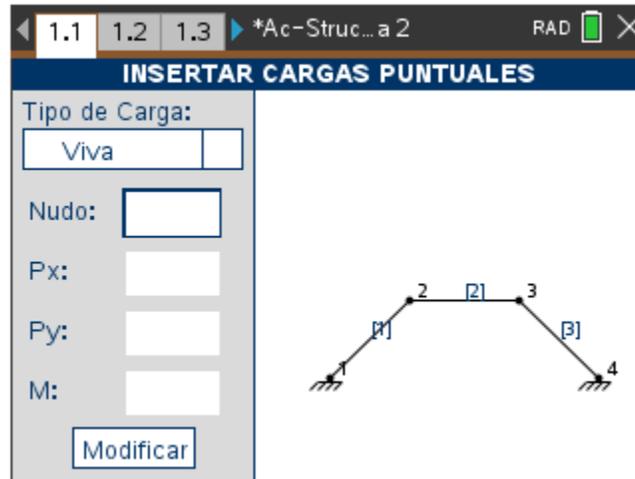


Figura 106: Insertar cargas puntuales

B.1.2.7. Paso 7: Agregar cargas distribuidas

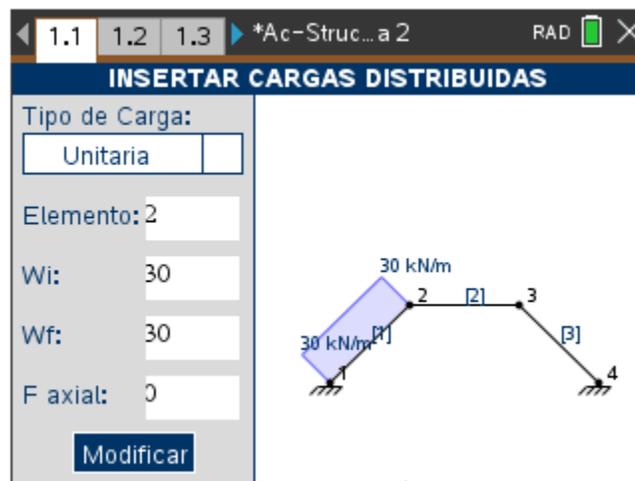


Figura 107: Insertar cargas distribuidas

B.1.2.8. Paso 8: Definir combinaciones de carga

1.1 1.2 1.3 *Ac-Struc... a 2 RAD X

AGREGAR COMBINACION

Unitaria: 1 Sismica: 0

Viva: 0 Viento: 0

Muerta: 0 Auxiliar: 0

Deformacion por temperatura: Si No

Aceptar Cancelar

Figura 108: Agregar combinación

1.1 1.2 1.3 *Ac-Struc... a 2 RAD X

DEFINIR COMBINACIONES DE CARGA

# Comb.	Caracteristicas
1	1U+ 0L+ 0D+ 0S+ 0W+ 0A+ 0T

Agregar Eliminar

Figura 109: Combinaciones agregadas

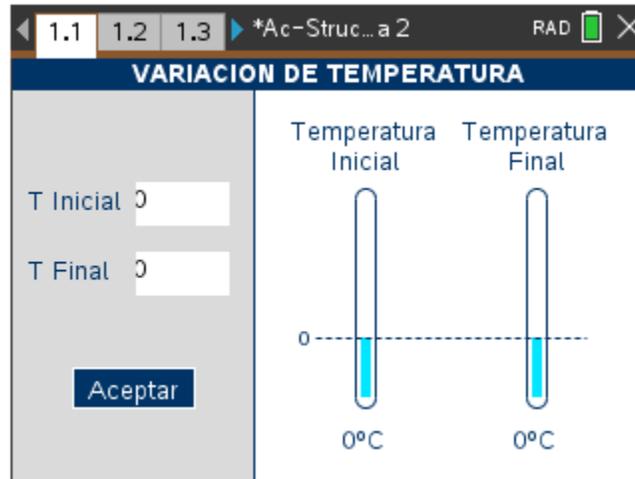
B.1.2.9. Paso 9: Definir variación de temperatura

Figura 110: Definir variación de temperatura

B.1.2.10. Paso 10: Analizar estructura

Figura 111: Analizar estructura

B.1.2.11. Paso 11: Ver resultados

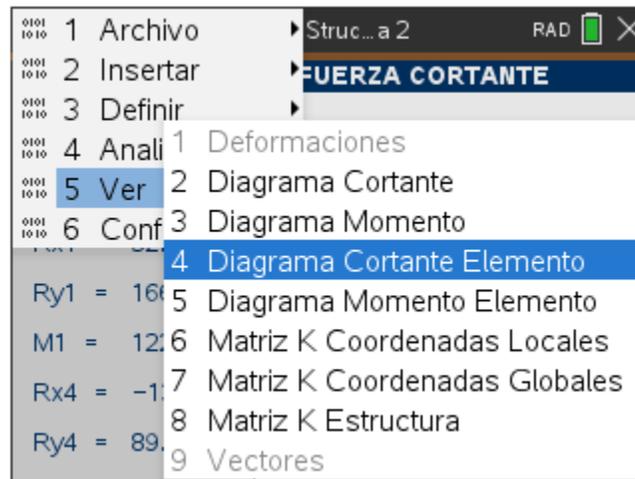


Figura 112: Resultados

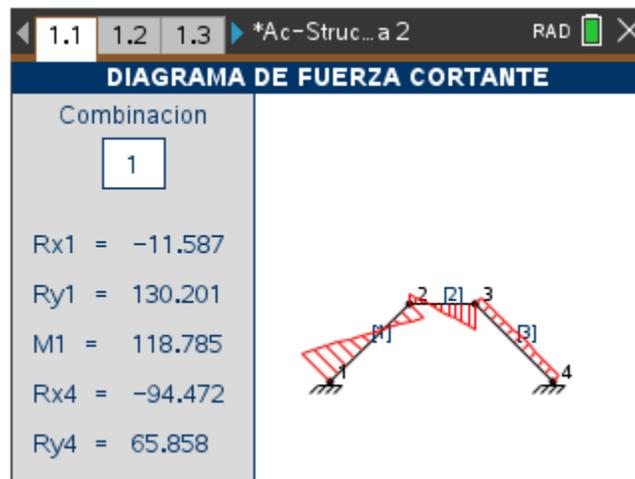


Figura 113: Reacciones de la estructura

B.2. Guía para el análisis de armaduras isostáticas usando el método de nudos

B.2.1. Análisis detallado

B.2.1.1. Paso 1: Grado de indeterminación estática de la estructura

Para comprobar que es posible resolver la armadura isostática usando el método propuesto, es necesario que la estructura cumpla con la siguiente igualdad (Ecuación 20).

Ecuación 20: Grado de indeterminación estática de la estructura

$$N_E + N_R = 2 * N_N$$

Dónde: N_E es el número de elementos de la estructura, N_R es el número de reacciones y N_N es el número de nudos de la estructura

Para la armadura propuesta, $N_E = 3$, $N_R = 3$ y $N_N = 3$, por lo tanto, es posible continuar con el cálculo de la estructura utilizando el método propuesto.

B.2.1.2. Paso 2: Equilibrio de fuerzas en cada nudo

Se realiza el equilibrio de fuerzas en cada nudo, se asume que los elementos se encuentran sometidos a tracción, es decir, las fuerzas salen de los nudos.

B.2.1.2.1. Nudo 1.

$$\sum F_X = F_{1-2} * \cos(0) + F_{1-3} * \cos(45) + R_{X1} = 0$$

$$\sum F_Y = F_{1-2} * \sin(0) + F_{1-3} * \sin(45) + R_{Y1} = 0$$

B.2.1.2.2. Nudo 2.

$$\sum F_X = F_{1-2} * \cos(180) + F_{2-3} * \cos(135) = 0$$

$$\sum F_Y = F_{1-2} * \sin(180) + F_{2-3} * \sin(135) + R_2 = 0$$

B.2.1.2.3. Nudo 3.

$$\sum F_X = F_{1-3} * \cos(-135) + F_{2-3} * \cos(-45) + 25 = 0$$

$$\sum F_Y = F_{1-3} * \sin(-135) + F_{2-3} * \sin(-45) - 40 = 0$$

B.2.1.3. Paso 3: Plantear las ecuaciones de manera matricial

Se plantea cada una de las ecuaciones de manera matricial, la cual tendrá la siguiente

forma:

$$[M] * [F] + [F_{ext}] = 0$$

Dónde, $[M]$ es la matriz de coordenadas, $[F]$ es el vector de fuerzas desconocidas y $[F_{ext}]$ son las fuerzas externas a las cuales se encuentra sometida la estructura. A continuación, se muestran las ecuaciones de equilibrio de la estructura planteada expresadas en forma matricial.

$$\begin{bmatrix} \cos(0) & \cos(45) & 0 & 1 & 0 & 0 \\ \sin(0) & \sin(45) & 0 & 0 & 1 & 0 \\ \cos(180) & 0 & \cos(135) & 0 & 0 & 0 \\ \sin(180) & 0 & \sin(135) & 0 & 0 & 1 \\ 0 & \cos(-135) & \cos(-45) & 0 & 0 & 0 \\ 0 & \sin(-135) & \sin(-45) & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} F_{1-2} \\ F_{1-3} \\ F_{2-3} \\ R_{x1} \\ R_{y1} \\ R_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 25 \\ -40 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.2.1.4. Paso 4: Resolver las incógnitas

Para resolver las incógnitas se utiliza la ecuación $[F] = -([M]^{-1} * [F_{ext}])$, que se obtiene al despejar el vector $[F]$ de la ecuación. A continuación, se muestra el resultado de la operación

$$\begin{bmatrix} F_{1-2} \\ F_{1-3} \\ F_{2-3} \\ R_{x1} \\ R_{y1} \\ R_2 \end{bmatrix} = \begin{bmatrix} 32.5 \\ -10.6066 \\ -45.9619 \\ -25 \\ 7.5 \\ 32.5 \end{bmatrix}$$

B.2.1.5. Paso 5: Interpretación de los resultados

Si el valor de la fuerza del elemento es positivo significa que el elemento se encuentra sometido a tracción, si por el contrario es negativo significa que el elemento en cuestión se encuentra sometido a compresión. Si uno de los valores de las reacciones es negativo significa que la dirección asumida inicialmente es incorrecta y que la fuerza realmente está siendo aplicada en el sentido contrario.

B.2.2. Análisis usando el software Ac-Struc

B.2.2.1. Paso 1: Nueva armadura



Figura 114: Nuevo método de los nudos

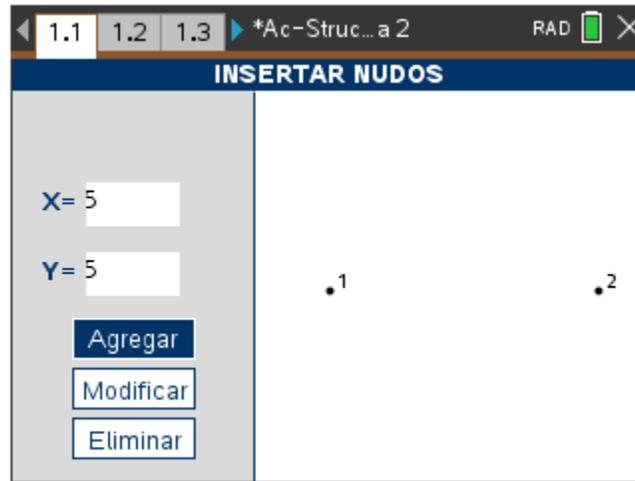
B.2.2.2. Paso 2: Insertar nudos

Figura 115: Insertar nudos armadura isostática

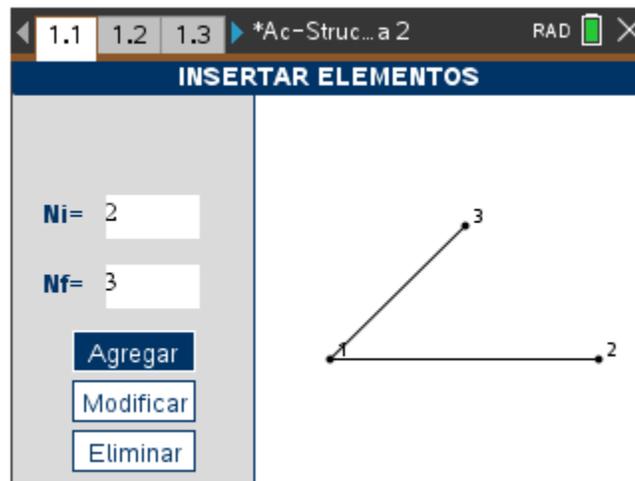
B.2.2.3. Paso 3: Insertar elementos

Figura 116: Insertar elemento armadura isostática

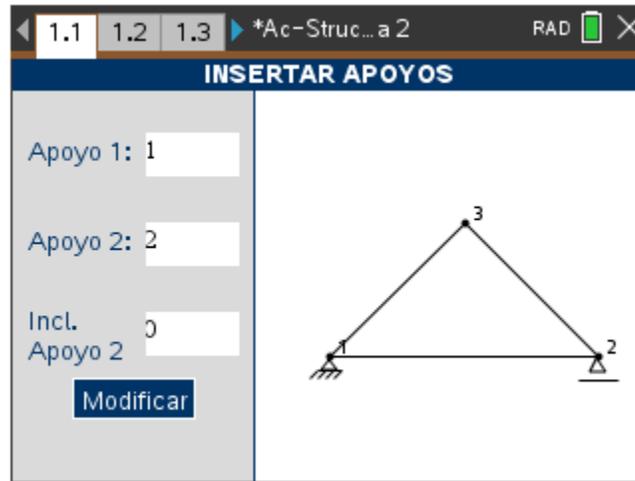
B.2.2.4. Paso 4: Insertar apoyos

Figura 117: Insertar apoyos armadura isostática

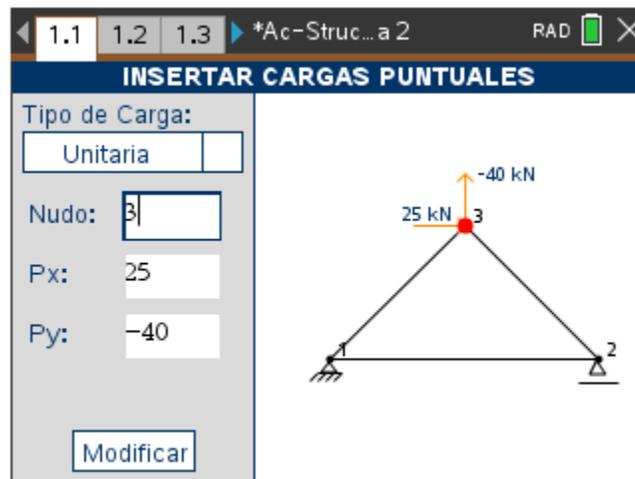
B.2.2.5. Paso 5: Insertar cargas puntuales

Figura 118: Insertar cargas puntuales armadura isostática

B.2.2.6. Paso 6: Definir combinaciones de carga.

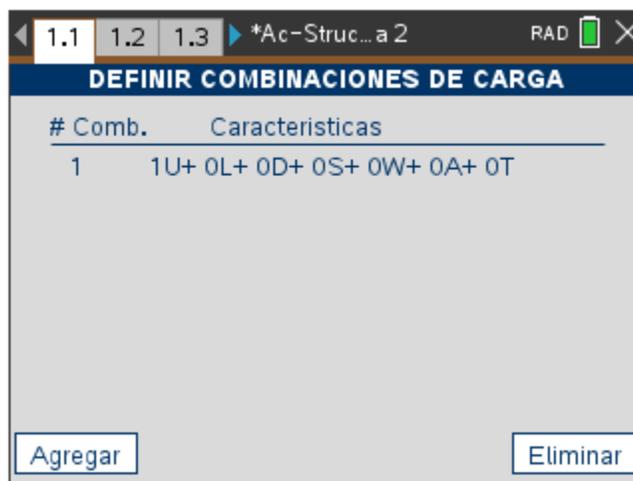


Figura 119: Combinaciones agregadas

B.2.2.7. Paso 7: Analizar armadura isostática

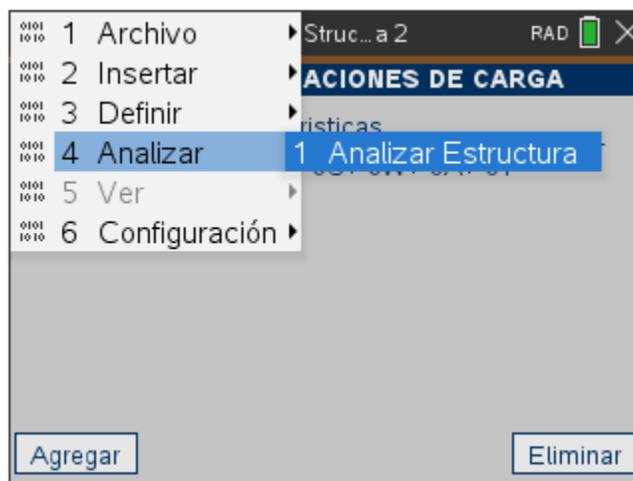


Figura 120: Analizar armadura isostática

B.2.2.8. Paso 8: Ver resultados



Figura 121: Resultados armadura isostática

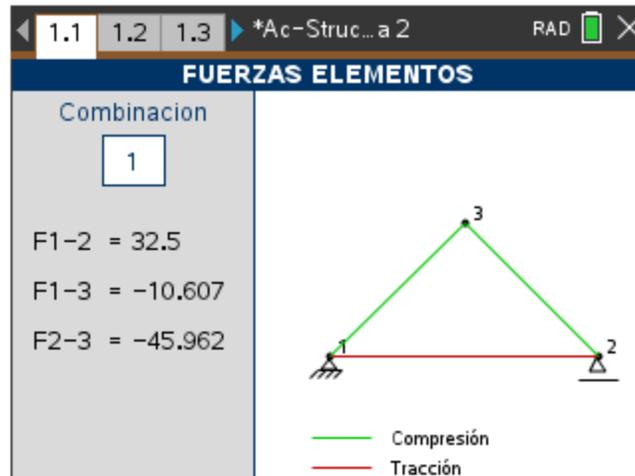


Figura 122: Fuerzas de los elementos de las armaduras isostáticas