

CAPÍTULO I

INTRODUCCION

CAPÍTULO I

INTRODUCCIÓN

1.1 ANTECEDENTES

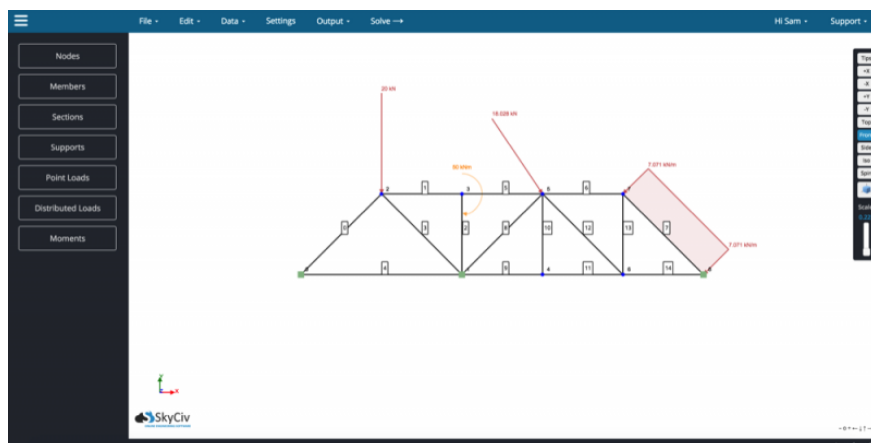
1.1.1 Sky Civ

SkyCiv es una plataforma de software en línea para el diseño y análisis estructural. Es una herramienta utilizada por ingenieros y profesionales de la construcción para llevar a cabo cálculos y análisis estructurales de manera eficiente y precisa.

Sky Civ es un software de ingeniería estructural basado en la nube y un proveedor de API establecido en 2013 de Paul Comino y Sam Carigliano. Como una compañía creada por ingenieros, está dedicada a construir un producto que pueda satisfacer las necesidades cambiantes de la profesión. Hasta el momento, el software ha sido utilizado por ingenieros en más de 160 países para resolver más 8 millones de proyectos. Con oficinas en Sydney y Chicago.

Se destaca por su interfaz de usuario intuitiva y fácil de usar, lo que facilita el modelado y el análisis de estructuras. Además, ofrece resultados en tiempo real, informes profesionales y la posibilidad de colaborar en proyectos en línea. (SkyCiv, 2023)

Figura 1.1 Pantalla principal de SkyCiv



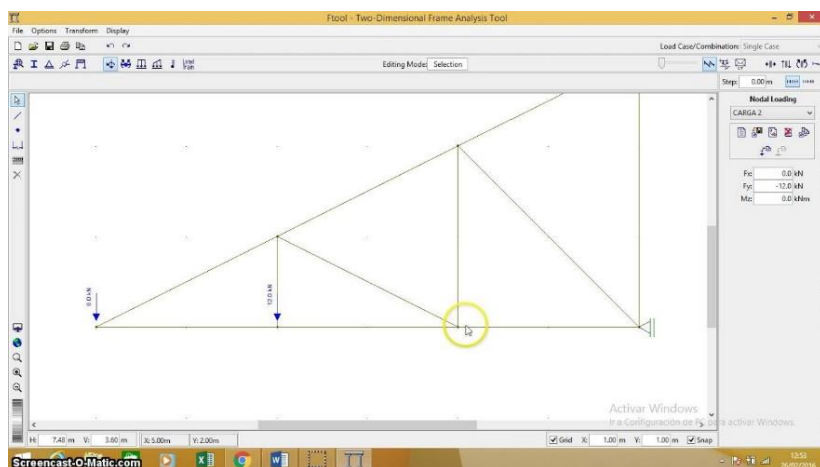
Nota. Captura de imagen de pantalla de cálculo de cerchas del programa (SkyCiv, 2023)

1.1.2 Ftool

Ftool es un programa interactivo, gráfico y de descarga gratuita que fomenta el aprendizaje del análisis estructural. Su creador es Luiz Fernando Campos, doctor en Ingeniería Estructural por la Universidad de Cornell (Estados Unidos), profesor asociado de la Pontificia Universidad Católica de Río de Janeiro (Brasil) y miembro del Instituto Tecgraf de Desarrollo Técnico-Científico de Software de la misma universidad. Él fue el ponente de la charla “Desarrollos recientes con el programa Ftool”, organizado el 21 de abril por la Carrera de Ingeniería Civil de la Universidad de Lima.

Ftool es un programa para el análisis estructural de estructuras planas. Tiene como principal objetivo el prototipado simple y eficiente de estructuras. El programa se desarrolló inicialmente para su uso en el aula, pero se ha convertido en una herramienta de uso frecuente en proyectos estructurales profesionales. Por lo tanto, se lanzó una edición avanzada, con licencia comercial, para satisfacer las necesidades de los diseñadores estructurales, manteniendo la edición básica gratuita. (Ftool, 2023)

Figura 1.2 Pantalla principal de Ftool



Nota. Captura de imagen de pantalla de cálculo de cerchas del software ftool (Ftool, 2023)

1.2 PROBLEMA

1.2.1. Planteamiento del problema

En el ámbito de la ingeniería estructural, los métodos matriciales se dividen en dos enfoques fundamentales: el método de la flexibilidad y el método de la rigidez. Estas dos variantes difieren en cuanto a las incógnitas que se resuelven y la forma en que se abordan los problemas estructurales.

El Método de la Flexibilidad se centra en determinar las fuerzas que actúan en la estructura. Utiliza matrices de flexibilidad para relacionar estas fuerzas aplicadas con los desplazamientos resultantes en la estructura, siendo especialmente útil cuando se conoce la carga aplicada y se busca calcular las fuerzas internas y las reacciones.

Por otro lado, el Método de la Rigidez considera las incógnitas como los desplazamientos de la estructura. Utiliza matrices de rigidez para relacionar estos desplazamientos con las fuerzas internas en la estructura. Este enfoque resulta valioso cuando se tienen los valores de los desplazamientos y se busca calcular las fuerzas internas y las reacciones.

A pesar de la eficacia del Método de Rigidez, el cálculo manual asociado a este método se vuelve complejo y laborioso, especialmente en estructuras con muchas barras. La posibilidad de errores es alta, y corregirlos en un proceso largo y complejo es desafiante. Otro obstáculo es la manipulación de matrices en la construcción de la matriz global y el cálculo de la inversa, lo que es esencial para obtener resultados precisos en el análisis estructural.

Uno de los principales problemas que enfrentan estudiantes y profesionales de la Ingeniería Civil es el procedimiento de cálculo relacionado con el Método de Rigidez aplicado a cerchas. Identificar si una cercha es isostática o hiperestática requiere un profundo conocimiento del criterio de estática en cerchas, añadiendo complejidad al proceso.

A medida que los avances tecnológicos han permitido la implementación de este método en el análisis computacional, la mayoría de los programas de ingeniería actualmente utilizan este enfoque en sus códigos.

Esta evolución tecnológica y la complejidad inherente al cálculo manual de cerchas han generado la necesidad de desarrollar una solución que simplifique y automatice este proceso. Es en este contexto donde surge la idea de crear un software dedicado al cálculo de cerchas en 2D, que permita a los usuarios ver cómo se calcula la Matriz de Rigidez de cada elemento y, al mismo tiempo, reduzca significativamente el tiempo requerido para obtener resultados precisos. Este software se ha desarrollado en lenguaje de programación Java, lo que lo hace accesible y versátil para estudiantes y profesionales de la Ingeniería Civil.

1.2.2. Formulación del problema

Se busca simplificar y automatizar el cálculo de cerchas en el contexto del método matricial de la rigidez sin alterar su procedimiento fundamental, con el propósito de mejorar significativamente la eficiencia en este proceso de análisis estructural.

1.2.3 Sistematización del problema

En la actualidad, el Método de Rigidez se ha convertido en una herramienta ampliamente utilizada en numerosos programas estructurales. Su aplicabilidad y facilidad de aprendizaje lo convierten en una opción óptima para los ingenieros en el análisis y diseño de estructuras. El Método de Rigidez ha demostrado ser eficiente para resolver problemas complejos al dividir un dominio en subdominios o elementos de geometría simple. Esta subdivisión permite tratar analíticamente e individualmente cada elemento, para luego ensamblarlos y obtener una solución para todo el conjunto.

Basándose en esta premisa, surge la idea de desarrollar un software para el cálculo de cerchas. Este software permitirá al usuario visualizar la Matriz de Rigidez de cada elemento, lo que significa que se podrán observar los cálculos y determinaciones realizadas por el programa. La implementación de este software brindará a los ingenieros una herramienta valiosa para analizar y diseñar cerchas de manera eficiente. Al visualizar la Matriz de Rigidez de cada elemento, los usuarios podrán comprender mejor los cálculos y resultados obtenidos, lo que les permitirá tener una mayor confianza en la precisión de los mismos.

1.3 OBJETIVOS

1.3.1 General

- ❖ Elaborar un software para solución de cálculo estructural de cerchas en 2D tanto Isostáticas como Hiperestáticas utilizando el Método de Rigidez, a través del lenguaje de programación Java.

1.3.2 Específicos

- ❖ Analizar los requerimientos de los usuarios.
- ❖ Investigar el Método de Rigidez aplicado en el cálculo de cerchas.
- ❖ Ampliar el conocimiento en el manejo del programa NetBeans que es el IDE (Entorno de Desarrollo Integrado, por sus siglas en inglés Integrated Development Environment) que se utilizara para programar con el lenguaje Java.
- ❖ Crear algoritmos de programación sencillos, de fácil entendimiento en la introducción de datos, así como también para la salida y visualización de resultados.
- ❖ Diseñar una Interfaz gráfica de programación sencilla, de fácil entendimiento para los estudiantes de la facultad de tecnología tanto en la introducción, procesamiento e interpretación de los resultados.
- ❖ Determinar las reacciones, fuerzas internas y desplazamiento en los nudos del sistema estructural, de acuerdo a las solicitaciones de cargas en miembros y nodos.
- ❖ Programar el software computarizado en idioma español utilizando el lenguaje de programación Java.
- ❖ Diseñar una interfaz para mostrar el procedimiento cálculo de manera que sea fácil de entender para el usuario.

1.4 HIPÓTESIS

- ❖ Con este programa, se espera lograr una mejora significativa en la velocidad y precisión de los cálculos en comparación con los métodos

tradicionales. Se espera que, al aprovechar técnicas computacionales y algoritmos eficientes, el programa pueda resolver cerchas de manera rápida y precisa, ofreciendo resultados más confiables y consistentes en comparación con los métodos manuales.

1.5 JUSTIFICACIÓN

Las razones por las cuales se elabora el trabajo de investigación son las siguientes:

1.5.1 Académica

Este proyecto se origina en la necesidad de proporcionar a los estudiantes de ingeniería civil una herramienta efectiva que les permita comprender y aplicar los métodos matriciales en el análisis de cerchas, lo cual es fundamental para su formación académica. Los conceptos de análisis matricial y el método de la rigidez son pilares en la ingeniería estructural moderna. Ofrecer un software que simplifique y automatice estos cálculos no solo facilita el aprendizaje de los estudiantes, sino que también mejora su comprensión de los principios fundamentales de la ingeniería civil.

1.5.2 Técnica

La base técnica de este proyecto radica en la necesidad de aplicar los avances tecnológicos para simplificar y mejorar el cálculo de cerchas en ingeniería civil. Los métodos matriciales, como el método de la rigidez, son fundamentales en el análisis estructural, pero realizar estos cálculos de manera manual puede ser complejo y propenso a errores, especialmente en estructuras con un gran número de barras.

El desarrollo de un software basado en el método de la rigidez, implementado en el lenguaje de programación Java, aprovecha la capacidad de las computadoras modernas para realizar cálculos de manera eficiente y precisa. Esto no solo ahorra tiempo, sino que también reduce la probabilidad de errores humanos en el proceso de cálculo.

Además, la incorporación de una interfaz gráfica de usuario en Java hace que el software sea accesible y fácil de usar tanto para estudiantes como para profesionales de la ingeniería civil. Esta combinación de precisión, eficiencia y accesibilidad técnica hace que el proyecto

sea relevante y valioso en el campo de la ingeniería estructural y contribuye al avance tecnológico en la disciplina.

1.5.3 Social

Este proyecto se basa en la importancia de formar ingenieros civiles altamente competentes que puedan contribuir al desarrollo y la seguridad de las estructuras en la sociedad. Al simplificar y agilizar el cálculo de cerchas, este software no solo beneficia a los estudiantes, sino que también tiene un impacto positivo en la profesión de la ingeniería civil en general. Al permitir que los futuros ingenieros adquieran habilidades sólidas en análisis estructural, se contribuye a la seguridad y eficiencia de las construcciones, lo que a su vez beneficia a la sociedad al garantizar edificios y estructuras más seguros y confiables.

1.6 ALCANCE Y LIMITACIONES

1.6.1 Alcance

El estudio investigativo tendrá los siguientes alcances:

- ❖ El programa trabajará utilizando secciones constantes en cada barra y asumiendo que los materiales serán homogéneos a lo largo de toda su longitud.
- ❖ Solución de cerchas en el plano, resultado de esfuerzos internos desplazamientos de nudos y esfuerzos externos (reacciones).
- ❖ Solución de cerchas sometidas a cargas puntuales ubicadas en los nudos, considerando un enfoque estático. Las cargas deben ser puntuales y estar ubicadas en los nudos de la cercha, lo que implica una idealización de la armadura.
- ❖ Solución de cerchas cuando se presente un desplazamiento impuesto en alguno de los nudos.
- ❖ Pestaña propia donde podrá ver el procedimiento de cálculo.
- ❖ Tendrá la opción para guardar el procedimiento de cálculo en un archivo txt, xls, doc u otro.
- ❖ Visualizar la matriz de rigidez de cada barra y la matriz ensamblada de toda la estructura.
- ❖ El software estará programado para trabajar con tres sistemas de unidades.

1.6.2 Limitaciones

El estudio investigativo tendrá las siguientes limitaciones:

- ❖ Las armaduras que serán resueltas por el programa a crear serán planas, es decir, el programa solamente se limitará al cálculo de estructuras en dos dimensiones.
- ❖ No se permite articulaciones entre las barras.
- ❖ En el caso que ya exista un desplazamiento en un nudo como dato se le debe introducir dicho dato en las mismas unidades.
- ❖ No se permiten apoyos inclinados.
- ❖ No se admiten cargas inclinadas ni distribuidas; por lo tanto, es necesario descomponer estas cargas antes de ingresarlas en el programa.
- ❖ Las barras de cada estructura deberán ser rectas a lo largo de su eje longitudinal.

CAPÍTULO II
FUNDAMENTO TEÓRICO
DE CERCHAS

CAPÍTULO II

FUNDAMENTOS TEÓRICOS DE CERCHAS

2.1 FUNDAMENTO TEÓRICOS

2.1.1 Cerchas

Las cerchas han sido parte sustancial de las cubiertas durante casi dos mil años. Una cercha es una estructura triangulada, de sección variable, autoportante, que aprovecha al máximo el material del que está compuesta, minimizando los esfuerzos de las barras y respondiendo, con su forma, eficientemente a su función.

Las cerchas son estructuras compuestas por una serie de elementos rectos, generalmente barras metálicas o de madera, conectadas en sus extremos mediante articulaciones llamadas nodos. Estas estructuras se utilizan comúnmente en la construcción de cubiertas, puentes, galpones y otras aplicaciones (como se ve en la imagen de abajo). donde se requiere un sistema de soporte resistente y liviano.

Figura 2.1 Fotografía de una cercha en un techo



Nota. Foto de una cubierta de una casa, con una cercha de madera (Techwood-House, 2023)

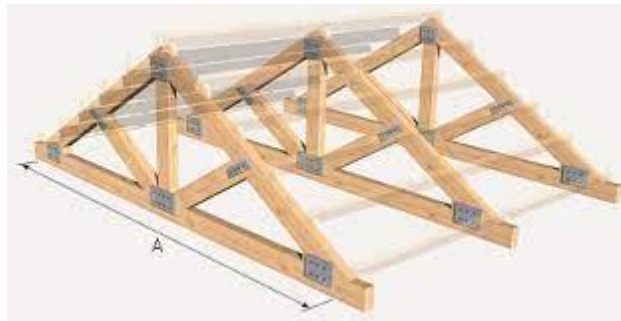
Las cubiertas son múltiples y variadas, tanto las que se utilizan en la actualidad como las ya utilizadas a lo largo del tiempo, aunque hay que tener en cuenta que ha sido con la madera, con la que más largamente se han realizado las estructuras portantes, los entablados, incluso los propios acabados, las tejas. Pero, no obstante, casi todos los materiales que históricamente se han ido incorporando a la construcción han formado y

forman parte de algún tipo de cubierta y de su estructura portante. Cubierta es aquel elemento destinado al cerramiento más o menos horizontal, dependiendo del volumen definido del edificio, situado en la parte más alta y es sin duda la palabra más utilizada para la definición de este conjunto de elementos, que ineludiblemente incluye una estructura portante, un entablado o elemento superficial de soporte y un acabado exterior continuo o de piezas que asegure su funcionamiento como protector de la intemperie. Las cerchas se caracterizan por su forma triangular, lo que les confiere una gran resistencia y estabilidad. Cada barra en una cercha se denomina miembro o elemento, y los nodos son los puntos de conexión entre los elementos. Los elementos de una cercha pueden ser comprimidos o tensionados dependiendo de las cargas aplicadas y la geometría de la estructura.

La forma triangular de las cerchas es fundamental para su funcionamiento. Al ser una estructura en equilibrio, la carga aplicada se distribuye a través de los diferentes elementos de manera eficiente, lo que permite una resistencia óptima. Además, las cerchas trianguladas son rígidas y evitan la deformación excesiva de la estructura bajo cargas, lo que las hace ideales para soportar grandes luces sin necesidad de pilares intermedios.

El diseño y análisis de cerchas se basa en principios de estática y resistencia de materiales. Mediante cálculos y análisis estructural, se determinan las fuerzas internas, los desplazamientos y las reacciones en los nodos de la cercha para garantizar la estabilidad y la seguridad de la estructura. Las cerchas más utilizadas son las de madera y las metálicas.

Figura 2.2 Cercha de Madera



Nota. Imagen de unas cerchas de madera (Redactor de wikiHow, s.f.)

Figura 2.3 Cercha metálica



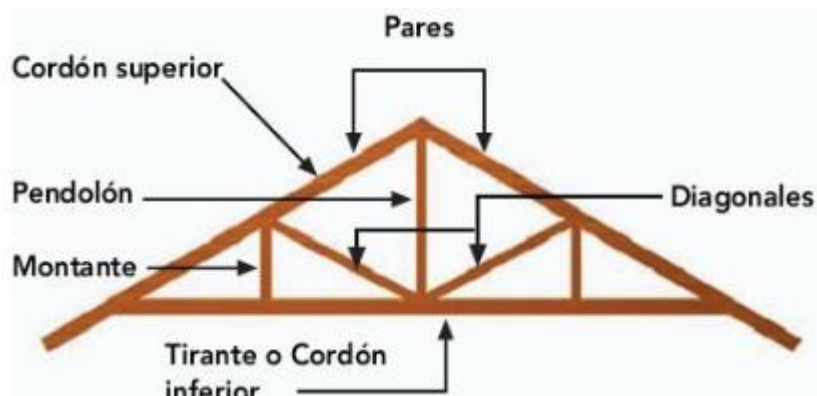
Nota. Imagen de unas cerchas metálicas (Ingeniería Construcción S.A.S, 2023)

En resumen, las cerchas son estructuras trianguladas compuestas por barras y nodos, diseñadas para proporcionar soporte resistente y eficiente en una variedad de aplicaciones de construcción. Su forma triangular y su distribución de carga equilibrada las hacen una opción popular en la ingeniería estructural. (López, 2009)

2.1.2 Elementos que componen una cercha

La fabricación de una cercha es de fácil y rápida confección, estas pueden ser prefabricadas o armada a pie de obra, y gracias a su diseño este permite salvar grandes luces. El tamaño no está limitado por el largo de cada pieza comercial de madera puesto que existen sistemas de unión que permiten conformar elementos que tengan mayores dimensiones, los elementos que conforman una cercha son los siguientes:

Figura 2.4 Partes de una cercha



Nota. Imagen detallada identificando las partes de una cercha (IngeCivil, 2018)

- ❖ **Par o pierna:** Son cada una de las dos piezas que están inclinadas que forman las aguas de una techumbre.
- ❖ **Tirante:** Es la pieza horizontal de una cercha encargada de unir el extremo inferior de los pares con la finalidad de impedir su separación.
- ❖ **Diagonales:** Es la pieza inclinada que tiene por función unir el par con el tirante.
- ❖ **Pendolón:** Es el elemento vertical que une un punto de la cumbrera con otro punto del tirante.
- ❖ **Montante:** Elemento vertical que une un punto del par con otro punto del tirante.

2.1.3 Tipos de cerchas y su clasificación

Existen varios tipos de cerchas utilizadas en la construcción, cada una con características y aplicaciones particulares. Las cuales se pueden clasificar según su función del equilibrio, su forma y distribución de las piezas interiores.

2.1.3.1 Por su función de equilibrio

El grado estático en cerchas se refiere al número de grados de libertad o movimientos independientes permitidos en cada nodo de la cercha. Los grados de libertad pueden ser desplazamientos o rotaciones en diferentes direcciones.

Es importante destacar que el grado estático puede variar según el tipo de cercha y las restricciones específicas en los nodos. En algunos casos, se pueden introducir apoyos adicionales o restricciones para controlar los movimientos en ciertas direcciones y reducir el grado estático en nodos específicos de la cercha. Para definir si una cercha es isostática o hiperestática se presenta la siguiente relación. (Alvarez, 2020)

N° de Ecuaciones = N° de Incógnitas **Isostáticas**

N° de Ecuaciones < N° de Incógnitas **Hiperestáticas**

$$NR+NB = NN*NE$$

NN= N° de Nudos
NB= N° de Barras
NR= N° de Reacciones
NE= Ecuaciones de la Estática por Nudo

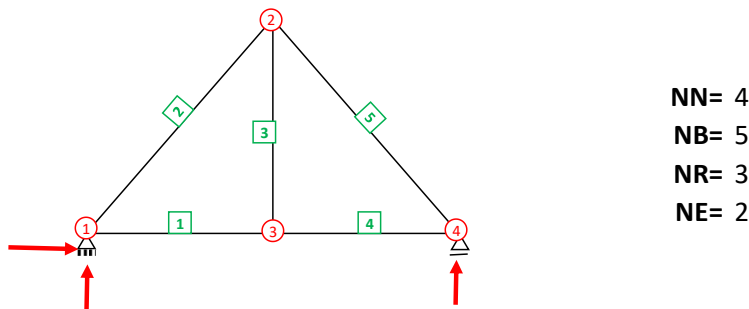
Ecuaciones de la estática en plantearse en el nudo de una cercha:

$$\Sigma F_x = 0$$

$$\Sigma F_y = 0$$

- a) **Cerchas isostáticas.** Tipo de estructura que se puede analizar mediante principios y fórmulas que revelen los valores estáticos. De naturaleza determinada estáticamente, que, si se llegara a eliminar algunos de los componentes que unen al armazón, fallaría catastróficamente todo el sistema.

Figura 2.5 Cercha Isostática



Nota. Gráfico de una cercha Isostática. Elaboración propia

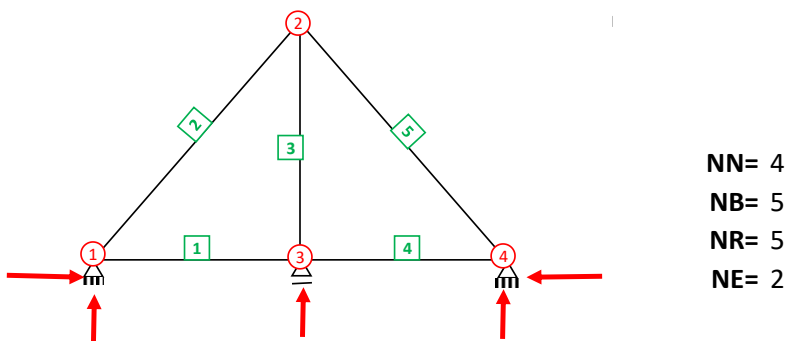
$$NR+NB = NN*NE$$

$$3+5 = 4*2$$

$$8 = 8 \quad \text{Isostáticas}$$

- b) **Cerchas hiperestáticas.** Su configuración se caracteriza por su estado de equilibrio, el momento flector es de valor igual a 0 en cada barra del sistema. A pesar de esto, la cercha puede que presente condiciones de inestabilidad, ya que su diseño de nudos fijos se asemeja a una estructura isostática.

Figura 2.6 Cercha Hiperestática



Nota. Gráfico de una cercha Hiperestática. Elaboración propia

$$NR+NB = NN*NE$$

$$5+5 = 4*2$$

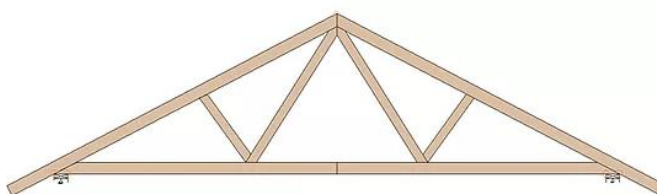
$$10 < 8 \quad \text{Hiperestática}$$

2.1.3.2 Por su forma

Está referido a la figura geométrica que representan los elementos que la confeccionan, en esta clasificación por su diseño y versatilidad, hay una amplia variedad de esta clasificación entre estas encontramos las más conocidas de forma triangular, tijera, rectangular, entre otras.

- a) **Triangular.** Es la más utilizada y permite ocuparla en todos tipos de luces. Su pendiente fluctúa generalmente entre los 12° y 45° Si tiene una pendiente mayor genera mayor altura interior de difícil aprovechamiento.

Figura 2.7 Cercha Triangular

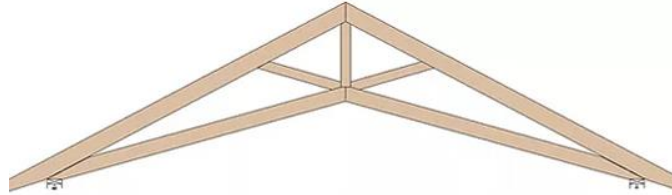


Nota. Gráfico de una cercha Triangular. (Techwood-House, 2023)

- b) **Tijera.** Se caracteriza por tener tanto su tirante como su par, en forma inclinada, fluctuando el Angulo del par superior entre 15° y 35° . La ventaja

de este tipo de estructura es que se logra una mayor altura en la parte central del espacio que cubre.

Figura 2.8 Cercha Tijera



Nota. Gráfico de una cercha Tijera. (Techwood-House, 2023)

- c) **Rectangular.** Generalmente se conoce por el nombre de viga armada o de celosía. Puede salvar luces desde los 7 hasta 30 m. Se emplea como estructura de techumbre, entre pisos, y arriostramiento longitudinal.

Figura 2.9 Cercha Rectangular



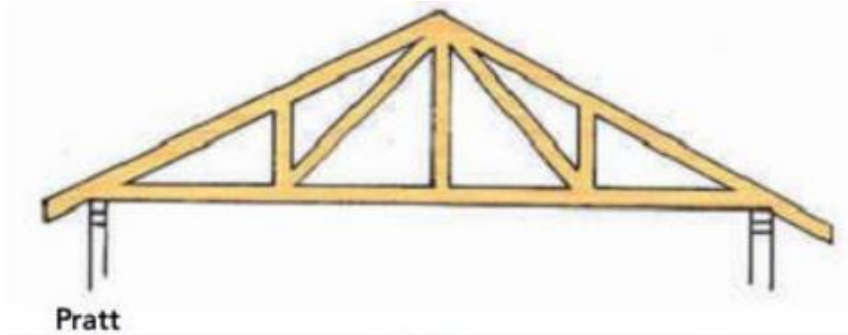
Nota. Gráfico de una cercha Rectangular. (Techwood-House, 2023)

2.1.3.3 Por su distribución de piezas

Entre estas se encuentran las cerchas tipo howe, pratt, fink, entre otras.

- a) **Cercha Pratt.** Es una cercha simple y ampliamente utilizada. Consiste en elementos diagonales que se inclinan hacia adentro y se encuentran en los nodos centrales de la estructura, y elementos verticales o montantes en los nodos extremos. Esta configuración proporciona estabilidad y resistencia eficiente. Se recomienda el uso para pendientes entre 25° y 45° y luces hasta 30 m.

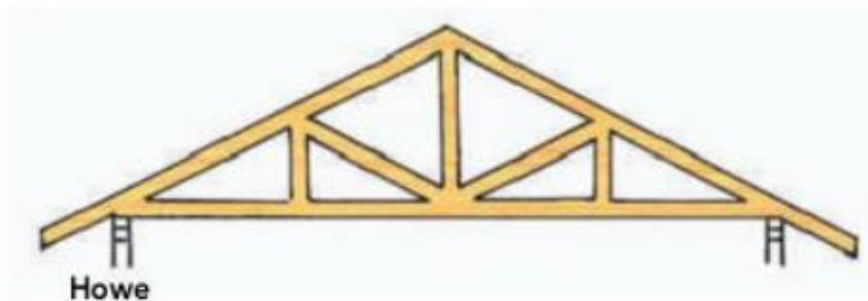
Figura 2.10 Cercha Pratt



Nota. Imagen de una cercha Pratt. (Structuralia, 2021)

- b) **Cercha Howe.** Similar a la cercha Pratt, pero con las diagonales inclinadas hacia afuera en lugar de hacia adentro. Las diagonales están bajo compresión y los montantes bajo tensión. Este tipo de cercha se utiliza cuando se desea reducir la altura de la estructura o cuando se requiere mayor resistencia a la compresión en los elementos. Es apta para ser trabajada en un mismo material.

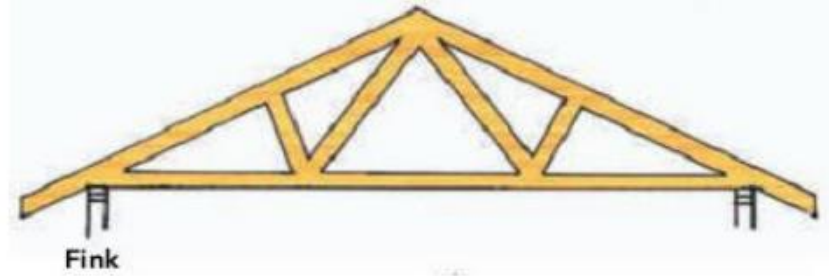
Figura 2.11 Cercha Howe



Nota. Imagen de una cercha Howe. (Structuralia, 2021)

- c) **Cercha Fink.** La cercha Fink ofrece economía en términos de peso de acero para techos de poca altura y altura alta, ya que los miembros se subdividen en elementos más cortos. Hay muchas formas de organizar y subdividir las cuerdas y los miembros internos. Este tipo de cercha se usa comúnmente para construir techos en casas. Permite luces de entre 12 a 18 m, siempre que la pendiente sea superior a 45°.

Figura 2.12 Cercha Fink



Nota. Imagen de una cercha Fink (Structuralia, 2021)

Estos son solo algunos ejemplos de los diferentes tipos de cerchas que se utilizan en la construcción. Cada tipo tiene sus ventajas y se elige según los requisitos específicos del proyecto, como la luz a cubrir, las cargas a soportar, el diseño arquitectónico y las consideraciones económicas. (Ingenieria Construccion S.A.S, 2023)

2.1.4 Desplazamiento en Cerchas

La historia del análisis de desplazamientos en cerchas se remonta a los primeros avances en la teoría de estructuras a fines del siglo XIX y principios del siglo XX.

Uno de los primeros en desarrollar métodos para el análisis de cerchas fue el ingeniero británico Henry Bow, quien en 1848 publicó el libro "Análisis de las Estructuras de Vigas", donde abordó el análisis estático de cerchas mediante la aplicación de ecuaciones de equilibrio. Sin embargo, en ese momento no se consideraban los desplazamientos de los nodos de las cerchas.

Fue a mediados del siglo XX cuando se comenzó a reconocer la importancia de considerar los desplazamientos en el análisis de cerchas. El avance significativo en este sentido se debe al ingeniero estructural Hardy Cross. En 1930, Cross introdujo el método de Cross para el análisis de cerchas, que se basaba en el análisis de fuerzas en los miembros de la cercha y la consideración de los desplazamientos en los nodos.

El método de Cross permitía obtener las fuerzas internas y los desplazamientos de una cercha mediante el uso de ecuaciones de compatibilidad de desplazamientos y ecuaciones de equilibrio. Este método fue un hito importante en el análisis de cerchas, ya que permitía

obtener resultados más precisos y realistas al considerar las deformaciones reales de la estructura.

Posteriormente, con el desarrollo de la computación y los avances en métodos numéricos, se pudieron realizar análisis más complejos de cerchas considerando desplazamientos. Los métodos de elementos finitos y métodos de rigidez, como se mencionó anteriormente, se convirtieron en herramientas ampliamente utilizadas para el análisis de cerchas y otras estructuras.

Hoy en día, el análisis de desplazamientos en cerchas se realiza utilizando software de modelado y análisis estructural avanzado, que permite obtener resultados precisos y eficientes para el diseño y la evaluación de cerchas en una variedad de aplicaciones de ingeniería civil y arquitectura

En general, un sólido deformable real, como cualquier medio continuo es un sistema físico con un número infinito de grados de libertad. Así sucede que en general para describir la deformación de un sólido necesitándose explicitar un campo vectorial de desplazamientos sobre cada uno de sus puntos. Este campo de desplazamientos en general no es reductible a un número finito de parámetros, y por tanto un sólido deformable de forma totalmente general no tiene un número finito de grados de libertad.

Sin embargo, para barras largas elásticas o prismas mecánicos de longitud grande comparada con el área de su sección transversal, el campo de desplazamientos viene dado por la llamada curva elástica cuya deformación siempre es reductible a un conjunto finito de parámetros. En concreto, fijados los desplazamientos y giros de las secciones extremas de una barra elástica, queda completamente determinada su forma. Así, para una estructura formada por barras largas elásticas, fijados los desplazamientos de los nudos, queda completamente determinada la forma deformada de dicha estructura. Esto hace que las estructuras de barras largas puedan ser tratadas muy aproximadamente mediante un número finito de grados de libertad y que puedan ser calculadas resolviendo un número finito de ecuaciones algebraicas. El método matricial proporciona esas ecuaciones en forma de

sistema matricial que relaciona los desplazamientos de los extremos de las barras con variables dependientes de las fuerzas exteriores. Por esta razón se ha buscado tratar de simplificar los métodos o modelos de análisis. (López, 2009)

2.1.5. Método de Rigidez

El método de rigidez, también conocido como método de los desplazamientos, es una técnica utilizada en ingeniería estructural para analizar y resolver problemas relacionados con las estructuras de armaduras. Este método se basa en la premisa de que las estructuras pueden considerarse como conjuntos de elementos rígidos conectados entre sí mediante elementos flexibles, como barras o vigas.

En el caso específico de las armaduras, que son estructuras compuestas por barras conectadas en sus nodos, el método de rigidez se utiliza para determinar los desplazamientos, las fuerzas internas y las reacciones en los nodos de la armadura bajo la acción de cargas externas. El objetivo es obtener una solución que satisfaga el equilibrio de fuerzas y la compatibilidad de deformaciones.

El método de rigidez directo, está orientado a la formulación de un problema estructural a partir de la matriz de rigidez de cada uno de los elementos que componen el sistema estructural, así como del vector de los desplazamientos y de las fuerzas tanto de acción como de reacción. Todos estos parámetros son manejados de manera vectorial y matricial. El proceso de análisis se puede considerar como una secuencia de cuatro etapas bien definidas:

1. Fuerzas externas o desplazamientos actuando sobre la estructura.
2. Acciones o fuerzas internas actuando sobre los elementos.
3. Reacciones o respuesta de los elementos a los desplazamientos.
4. Reacciones o respuesta de la estructura a la acción de fuerzas.

La relación existente entre acción y respuesta se puede representar matricialmente de la siguiente manera:

$$\{D\} = [f]\{F\} \quad (2.1)$$

$$\{F\} = [K]\{D\} \quad (2.2)$$

Donde:

$\{D\}$ = Vector de desplazamiento

$[f]$ = Matriz de flexibilidad

$\{F\}$ = Vector de fuerzas

$[k]$ = Matriz de Rigidez

En la ecuación 2.1 $[f]$ representa la matriz de flexibilidad y corresponde al método de las fuerzas o de flexibilidad, mientras que en la ecuación 2.2 $[k]$ representa la matriz de rigidez de la estructura en el método de los desplazamientos o de la rigidez. Ambas matrices se relacionan recíprocamente, es decir la matriz de rigidez es la inversa de la matriz de flexibilidad y viceversa.

Para comprender mejor el significado de las matrices de flexibilidad y rigidez respectivamente, es bueno recordar la definición de los coeficientes de flexibilidad y de rigidez y que se pueden resumir así:

2.1.5.1 Coeficiente de flexibilidad

La fuerza necesaria aplicada en el nodo “i” de un elemento estructural, para producir un desplazamiento unitario en el nodo “j” mientras los otros nodos no soportan carga alguna.

2.1.5.2 Coeficiente de rigidez

El desplazamiento necesario generado en el nodo “i” de un elemento estructural, por una fuerza unitaria aplicada en el nodo en el nodo “j” mientras los otros nodos permanecen fijos.

Si se expande la ecuación matricial de rigidez se obtiene:

$$\begin{Bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ F_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \cdot & \cdot & \cdot & k_{1n} \\ k_{21} & k_{22} & \cdot & \cdot & \cdot & k_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ k_{n1} & k_{n2} & \cdot & \cdot & \cdot & k_{nn} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ \cdot \\ \cdot \\ D_n \end{Bmatrix} \quad (2.2.a)$$

Donde:

$k_{11} \dots k_{nn}$ son los coeficientes de rigidez

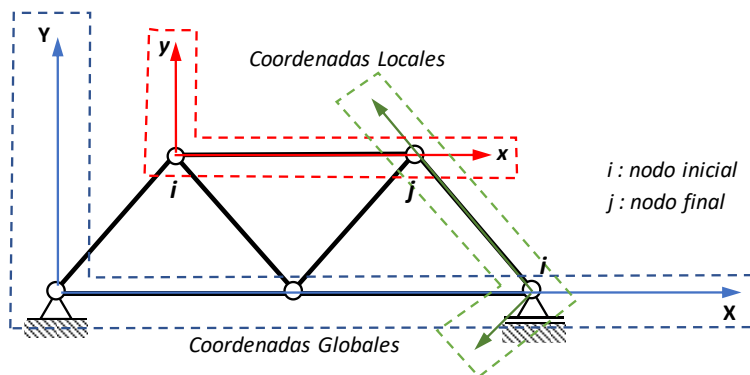
(Alvarez, 2020)

2.1.5.3 Sistemas de Coordenadas

La configuración de un sistema estructural debe estar definida tanto para la estructura sin deformaciones como para cuando se producen los desplazamientos de sus nudos. Todos los elementos de la estructura en general, deben estar referenciados a un sistema de referencia general, el mismo que resulta ser un sistema de coordenadas ortogonales y cartesianas

Es preciso distinguir entre un sistema de coordenadas generales o también llamadas globales de la estructura y un sistema de coordenadas particulares o locales para cada elemento.

Figura 2.13 Cercha con sistema de coordenadas



Nota. Figura de una cercha donde especifica el sistema de coordenadas globales y locales.

(Alvarez, 2020)

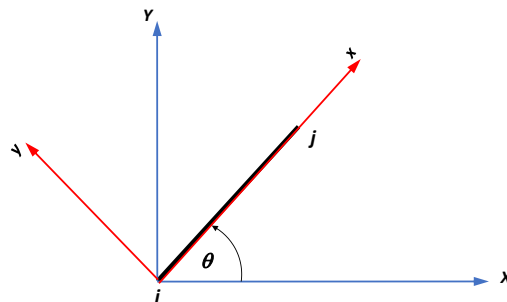
2.1.5.3.1 Sistema de coordenadas locales

La estructura reticular mostrada en la figura 15 presenta un cierto número de elementos; observando la barra $i-j$ se tiene que el eje $x-x$ se encuentra orientado según el eje longitudinal del elemento mientras que el eje $y-y$ tiene su origen en el nudo inicial (i) de la barra ($i-j$) y se supone entonces que el sentido positivo del eje local $x-x$ va del nudo inicial al nudo final. Para definir el eje $y-y$, se gira en el sentido antihorario un ángulo de noventa grados con respecto al eje $x-x$ que define su dirección y sentido positivo. Para el caso en el que se requiera los ejes de rotación estos se definen aplicando la regla de la mano derecha en el nudo inicial del elemento.

2.1.5.3.2 Sistema de coordenadas global

Es un sistema de referencia establecido por el ingeniero estructural para cada tipo de estructura según sea su naturaleza. Para estructuras planas Los ejes globales se denominan $X-Y$ y generalmente se ubican en el primer apoyo externo de la estructura.

Figura 2.14 Eje local y global



Nota. Figura que relación entre ejes global y local para estructuras planas. (Alvarez, 2020)

En resumen, todo lo que se refiere a nudos debe expresarse en coordenadas globales todo lo relacionado con elementos debe referirse a las coordenadas locales correspondientes. (Alvarez, 2020)

2.1.5.3.3. Transformación de ejes locales a ejes globales

Si un elemento reticulado, arbitrariamente orientado y sometido a fuerzas axiales se encuentra referido a un sistema de coordenadas globales, mediante una matriz, llamada

matriz de transformación se puede referir el reticulado a un sistema de coordenadas globales.

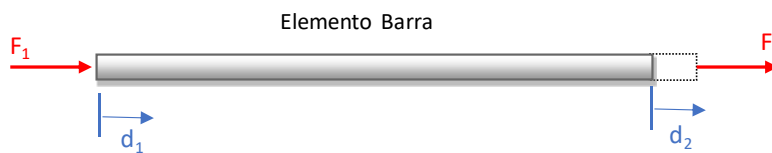
Para transformar los ejes locales a ejes globales vamos a demostrar la relación entre desplazamientos de los nudos y fuerzas externas. (Pardo, 2022)

2.1.5.3.3.1 Relación entre desplazamientos de los nudos y fuerzas externas

El trabajo consiste en encontrar una relación entre los desplazamientos de la barra y las fuerzas externas, mediante las sollicitaciones internas de esfuerzo y deformación.

Imaginaremos entonces que los nudos se desplazan unas distancias d_1 y d_2 . Además de estos desplazamientos, existen fuerzas externas que se aplican en la barra, que por equilibrio serán iguales y opuestas:

Fuerzas aplicadas en los nudos de barra



d_1 = desplazamiento del nudo 1
 d_2 = desplazamiento del nudo 2

Por Equilibrio

$$F_2 = -F_1 \quad (2.3)$$

Ahora bien, podemos plantear una relación entre los esfuerzos internos de la barra y las deformaciones unitarias que estos esfuerzos generan.

Los esfuerzos internos de la barra se definen por:

$$\sigma = E \xi \quad (2.4)$$

Donde:

σ = Esfuerzo axial

E = Modulo de Elasticidad

ξ = Deformacion Unitaria

Pero:

$$\sigma = \frac{F}{A} \xi = \frac{\Delta L}{L} = \frac{d_2 - d_1}{L} \quad (2.4 a)$$

F = fuerza aplicada en la sección

A = área de la sección transversal del elemento

L = longitud del elemento

En estas ecuaciones, se puede poder todo en función de las fuerzas y de los desplazamientos conociendo la definición de esfuerzo y de deformación unitaria. Nos centraremos en este caso en el nudo 2, donde la deformación unitaria es $\xi = (d_2 - d_1)/L$.

Luego se forma una ecuación donde F2 está intrínsecamente relacionada con los desplazamientos en los extremos.

$$\frac{F_2}{A} = E \left(\frac{d_2 - d_1}{L} \right) \quad (2.4.b)$$

$$F_2 = \frac{EA}{L} \left(\frac{d_2 - d_1}{L} \right) = \frac{EA}{L} \left(\frac{-d_1 + d_2}{L} \right) \quad (2.4.c)$$

Debido a que $F_2 = -F_1$, entonces se puede escribir la ecuación anterior en función de F1:

$$F_1 = \frac{EA}{L} (d_1 - d_2) \quad (2.5)$$

Se forma entonces un sistema de ecuaciones en función a 4 variables. F1, F2, u1, u2, Se forma el sistema:

$$F_1 = \frac{EA}{L} (d_1 - d_2) \quad (2.5.a)$$

$$F_2 = \frac{EA}{L} (-d_1 + d_2) \quad (2.5.b)$$

Escrito el sistema matricialmente, se tiene lo siguiente:

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} \quad (2.6)$$

Siguiendo el formato la ecuación queda:

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \end{Bmatrix} \quad (2.6 a)$$

Donde:

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \text{Vector de cargas}$$

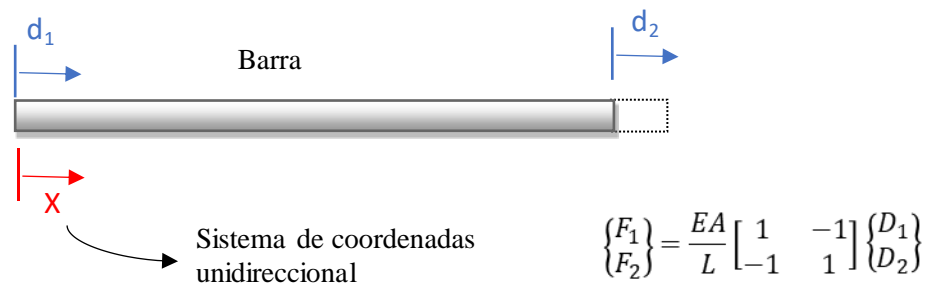
$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \text{Matriz de Rigidez}$$

$$\begin{Bmatrix} D_1 \\ D_2 \end{Bmatrix} = \text{Vector de Desplazamiento}$$

2.1.5.3.3.2 Demostración

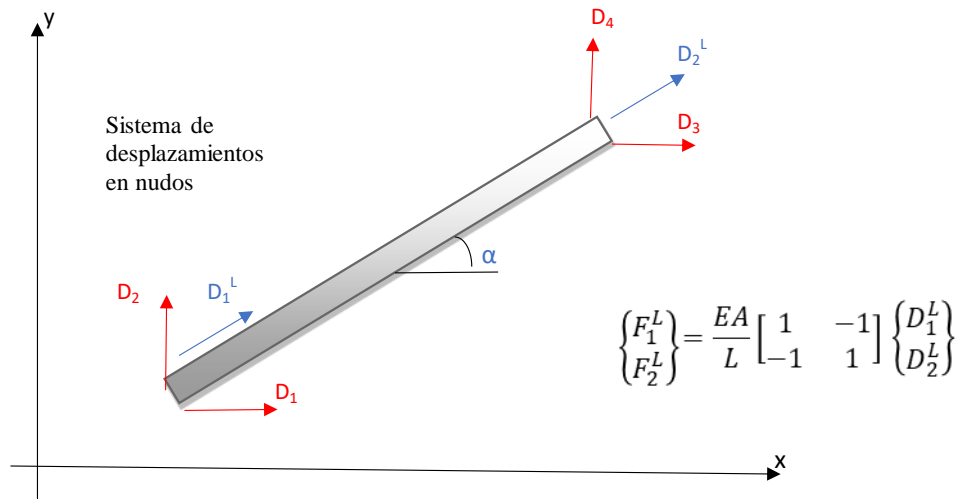
Esta transformación de coordenadas de 1D a 2D genera en cada nudo, un grado de libertad extra (en total dos grados de libertad por nudo). En el caso de transformación de coordenadas de 1D a 3D se generarán 2 grados de libertad adicionales para cada nudo (con un total de 3 grados de libertad). (Pardo, 2022)

Arrancamos con el elemento de cercha mencionado en el apartado anterior donde se definió la ecuación 2.6. El elemento cuenta con un grado de libertad por nudo, como se muestra a continuación:



Para que este elemento expuesto a deformaciones axiales pueda trabajar en un espacio bidireccional, lo rotaremos un ángulo alfa. De tal manera, se generan en cada nudo dos grados de libertad en un sistema global de coordenadas.

Figura 2.15 Desplazamiento en Nudos

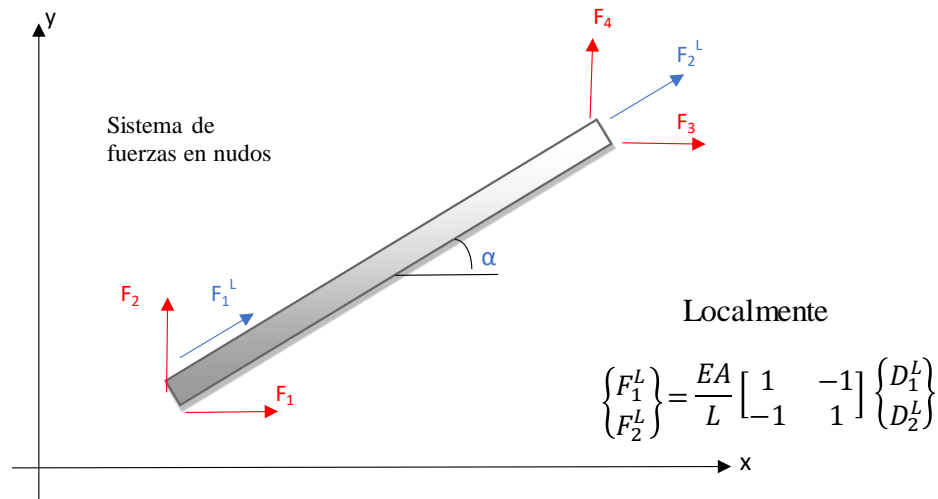


Nota. Gráfico de desplazamiento en nudos. Elaboración propia.

Nota que los desplazamientos en coordenadas globales no llevan superíndice mientras que los desplazamientos en coordenadas locales llevan el superíndice “L”.

En lo que consiste el trabajo a continuación es en buscar una relación entre el desplazamiento local D_1^L y los desplazamientos globales D_1 y D_2 . De la misma manera con el otro nudo. Así como se busca realizar la transformación de los vectores desplazamiento de locales a globales, lo mismo pasa con las fuerzas, de locales a globales:

Figura 2.16 Fuerzas en Nudos



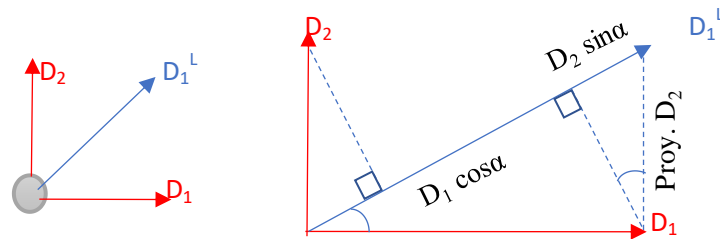
Nota. Gráfico de fuerzas en nudos. Elaboración propia

2.1.5.3.3 Transformación de coordenadas de desplazamientos

Centrándonos en el nudo izquierdo del elemento rotado. Si descomponemos los desplazamientos globales D_1 y D_2 , se tiene lo siguiente.

Relación entre desplazamientos locales y globales en un nudo.

Figura 2.17 Transformación de Desplazamiento



Nota. Gráfico de transformación de desplazamiento. Elaboración propia

$$D_1^L = D_1 \cos \alpha + D_2 \sin \alpha \quad (2.7)$$

D_1 Local puede componerse por las proyecciones de los desplazamientos D_1 y D_2 sobre $U1$ Local. Repitiendo el procedimiento para los desplazamientos del otro nudo, se obtienen las dos siguientes ecuaciones.

Analógicamente para el otro nudo:

$$D_1^L = D_1 \cos \alpha + D_2 \sin \alpha \quad (2.7.a)$$

$$D_2^L = D_3 \cos \alpha + D_4 \sin \alpha \quad (2.7.b)$$

Escribiendo lo anterior en forma matricial, se tiene:

$$\begin{Bmatrix} D_1^L \\ D_2^L \end{Bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{Bmatrix} \quad (2.8)$$

Finalmente toca reemplazar los desplazamientos en coordenadas locales, por los desplazamientos en coordenadas globales. Este reemplazo debe hacerse en el sistema de ecuaciones que relaciona desplazamientos con fuerzas externas, de la siguiente manera:

$$\begin{Bmatrix} D_1^L \\ D_2^L \end{Bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{Bmatrix}$$

Reemplazando en el sistema local, ecuación (2.6 a)

$$\begin{Bmatrix} F_1^L \\ F_2^L \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} D_1^L \\ D_2^L \end{Bmatrix}$$

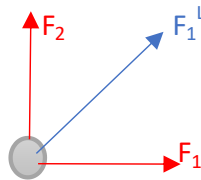
El sistema de ecuaciones anterior entonces queda:

$$\begin{Bmatrix} F_1^L \\ F_2^L \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{Bmatrix} \quad (2.9)$$

2.1.5.3.4 Transformación de coordenadas de fuerzas externas

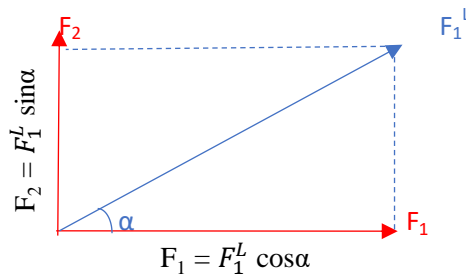
De manera similar a los desplazamientos, ahora nos enfocamos en las fuerzas en los nudos.

Las fuerzas locales llevan el superíndice L mientras que las fuerzas en coordenadas globales no llevan estos superíndices. Si nos concentramos en las fuerzas del nudo izquierdo se tiene lo siguiente:



Relación entre fuerzas locales y globales en nudo

Figura 2.18 Transformación de Fuerza



Nota. Gráfico de transformación de fuerzas. Elaboración propia

Para nudos 1 y 2 entonces:

$$F_1 = F_1^L \cos\alpha \quad (2.10)$$

$$F_2 = F_1^L \sin\alpha \quad (2.10 \text{ a})$$

$$F_3 = F_2^L \cos\alpha \quad (2.10 \text{ b})$$

$$F_4 = F_2^L \sin\alpha \quad (2.10 \text{ c})$$

Esta vez descompondremos las fuerzas locales en su respectiva proyección sobre los ejes X e Y globales. De esta manera, por ejemplo, la proyección de F_1L sobre el eje X es $F_1L \cdot \cos(\alpha)$ y este resultado es la fuerza X en coordenadas globales para este nudo. Repitiendo el proceso para X y Y para ambos nudos, se tienen las fórmulas en la imagen anterior.

Escribiendo estas fórmulas en coordenadas en forma matricial se tiene:

Matricialmente

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ 0 & \cos \alpha \\ 0 & \sin \alpha \end{bmatrix} \begin{Bmatrix} F_1^L \\ F_2^L \end{Bmatrix} \quad (2.11)$$

A diferencia del procedimiento realizado a estas alturas con los desplazamientos, ahora reemplazaremos las fuerzas locales de la ecuación Desplazamientos-Fuerzas, dentro del último sistema de ecuaciones obtenido, de la siguiente manera:

$$\begin{Bmatrix} F_1^L \\ F_2^L \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{Bmatrix}$$

Reemplazando en la ecuación (2.11)

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ 0 & \cos \alpha \\ 0 & \sin \alpha \end{bmatrix} \begin{Bmatrix} F_1^L \\ F_2^L \end{Bmatrix}$$

Así, tenemos el sistema que relaciona desplazamientos con fuerzas completo.

Reescribiendo el sistema final se tiene:

Sistema Local

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ 0 & \cos \alpha \\ 0 & \sin \alpha \end{bmatrix} \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{Bmatrix} \quad (2.12)$$

De la anterior ecuación, la matriz de rigidez para un elemento de cercha es la siguiente:

Matriz de Rigidez del Elemento

$$[k] = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ 0 & \cos \alpha \\ 0 & \sin \alpha \end{bmatrix} \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \quad (2.11)$$

Entonces realizando la multiplicación matricial con las ecuaciones 2.11 y 2.9 la matriz de rigidez $[k]$ en coordenadas globales para un elemento de reticulado arbitrariamente orientado queda de la siguiente manera:

$$K_{ij} = \frac{AE}{L} \begin{bmatrix} \cos^2 \alpha & \cos \alpha \sin \alpha & -\cos^2 \alpha & -\cos \alpha \sin \alpha \\ \cos \alpha \sin \alpha & \sin^2 \alpha & -\cos \alpha \sin \alpha & -\sin^2 \alpha \\ -\cos^2 \alpha & -\cos \alpha \sin \alpha & \cos^2 \alpha & \cos \alpha \sin \alpha \\ -\cos \alpha \sin \alpha & -\sin^2 \alpha & \cos \alpha \sin \alpha & \sin^2 \alpha \end{bmatrix} \quad (2.11.a)$$

Para escribirlo de una manera más académica y resumida:

$$[k] = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \quad (2.11.b)$$

$$\text{Donde:} \quad c^2 = \cos^2 \theta \quad s^2 = \sin^2 \theta \quad cs = \cos \theta \sin \theta$$

Este es el principio del método de rigidez directo, es decir que la matriz de rigidez de cada elemento se prepara en coordenadas locales y se transforma a coordenadas globales. (Pardo, 2022)

2.1.5.4 Grados De Libertad

Los grados de libertad para una estructura se definan como el número suficiente de coordenadas globales independientes, ubicadas en cada nodo para describir la configuración inicial y/o final de sus desplazamientos o fuerzas actuando sobre el nodo respectivo. En general los grados de libertad se clasifican en:

- ❖ Grados de libertad potenciales, GLP
- ❖ Grados de libertad restringidos, GLR
- ❖ Grados de libertad efectivos, GLE

Los grados de libertad potenciales son todos los grados de libertad identificados en la estructura, tomando en cuenta los de los apoyos y todos aquellos que se definan de acuerdo a los fines que en el análisis se persiguen. Se los determina a partir del número total de nodos en la estructura multiplicado por las coordenadas independientes y necesarias requeridas para la estructura.

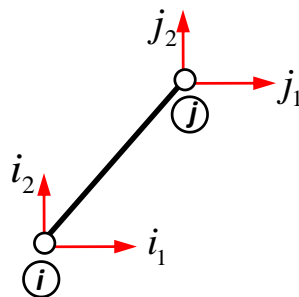
Los grados de libertad restringidos son aquellos que por alguna razón están impedidos de generar movimiento sobre alguna de las coordenadas globales definidas en los nodos, como es el caso de las restricciones en los apoyos.

Los grados de libertad efectivos son los que resultan de la diferencia entre los grados de libertad potenciales y los grados de libertad restringidos, vale decir:

$$GLE = GLP - GLR \quad (2.12)$$

En el caso de las estructuras planas, los casos se definen según el tipo de estructura que se esté tratando.

Figura 2.19 Grados de Libertad



Nota. Imagen detallando los Grados de Libertad en una Barras. Elaboración propia

Elemento de reticulado en el plano

$$i_1 = 2 \cdot i - 1 \qquad j_1 = 2 \cdot j - 1$$

$$i_2 = 2 \cdot i \qquad j_2 = 2 \cdot j$$

Para el caso de reticulados y vigas en el plano, se tiene dos grados de libertad, asociados con los desplazamientos nodales. Dos desplazamientos lineales (vertical y horizontal) para una barra de reticulado y dos desplazamientos, uno lineal y otro de rotación para un elemento viga. (Alvarez, 2020).

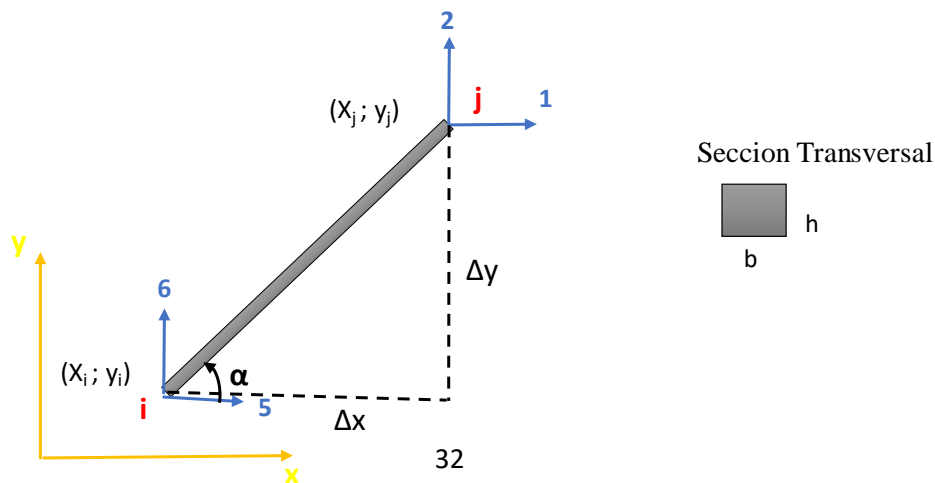
2.1.5.5 Cosenos directores y Matriz de Rigidez global para cada elemento

Como ya se mencionó, para el sistema de coordenadas globales se considera la x positiva hacia la derecha y la y positiva hacia arriba. Por otra parte, cada elemento tiene un sistema coordenado propio o local, el cual se usa para especificar el sentido de sus desplazamientos y sus cargas internas. Este sistema es definido a través de los ejes x' , y' con el origen en el nodo cercano N y el eje x' coincidiendo con el eje longitudinal del elemento y señalando hacia el extremo alejado F .

Para hacer compatibles las matrices de rigidez de los elementos, estas deben transformarse del sistema local al global. Los ángulos más pequeños entre los ejes y , x globales positivos y el eje local y' positivo se designan como θ_y y θ_z .

Los cosenos de esos ángulos se denominan cosenos directores y se evalúan con las siguientes ecuaciones:

Figura 2.20 Cosenos directores



Nota. Gráfico para identificar los cosenos directores. Elaboración propia

$$\Delta x = X_j - X_i \quad (2.13)$$

$$\Delta y = Y_j - Y_i \quad (2.13.a)$$

$$L = \sqrt{\Delta x^2 + \Delta y^2} \quad (2.14)$$

$$C_x = \Delta x / L \quad (2.14.a)$$

$$S_y = \Delta y / L \quad (2.14.b)$$

$$C = \frac{X_j - X_i}{\sqrt{\Delta x^2 + \Delta y^2}} = \text{Cos} \alpha \quad (2.15)$$

$$S = \frac{Y_j - Y_i}{\sqrt{\Delta x^2 + \Delta y^2}} = \text{Sen} \alpha \quad (2.16)$$

Estos son los coeficientes que conforman la matriz de rigidez del elemento

$$[k] = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}$$

Donde:

A= Area de la seccion transversal

i= Nodo inicial (X_i ; Y_i)

j= Nodo final (X_j ; Y_j)

E= Modulo de Elasticidad

S,C= Coeficiente de rigidez

L= Longitud de la Barra

2.1.5.6 Planteamiento General del Método De Rigidez en la Solución de Cerchas

Cuando una estructura se va a analizar empleando el método matricial de la rigidez, se requiere de la subdivisión de esta en una serie de elementos finitos discretos e identificar sus puntos extremos como nodos. Si se trata de una armadura, las barras o elementos que la componen representan los elementos finitos y los nodos son equivalentes a las juntas.

Cada elemento y cada nodo se identifican arbitrariamente mediante un número encerrado en un cuadrado y un número dentro de un círculo respectivamente, y los extremos lejanos G (del inglés Far) y cercano N (del inglés Near) de cada elemento se simbolizan en forma arbitraria con una flecha a lo largo del mismo cuya punta o cabeza se dirige hacia el extremo alejado. Se establece un sistema de coordenadas globales con un origen que puede situarse en cualquier punto.

Hay dos grados de libertad o dos posibles desplazamientos por nodo de armadura, los cuales individualmente se especificarán a través de un número de código y serán referenciados mediante una flecha orientada en su dirección coordenada global positiva.

Los desplazamientos incógnita deben codificarse numéricamente primero, puesto que, como se verá más adelante, cuando se formule la matriz de rigidez de la estructura y se seccione, esto nos proporcionará la manera más directa para hallar sus valores.

En general, en un nodo donde no haya algún soporte, los desplazamientos horizontal y vertical son desconocidos. Por su parte, en un apoyo simple o rodillo sólo se genera una reacción que es una fuerza que actúa perpendicularmente a la superficie en el punto de contacto, así que tal tipo de soporte es capaz de no permitir el desplazamiento en la dirección de su reacción, pero incapaz de impedir el desplazamiento perpendicular a la dirección de su fuerza reactiva en su nodo de aplicación. En tanto, un apoyo articulado, pasador o bisagra no permite los desplazamientos horizontal y vertical en su nodo de ubicación debido a las fuerzas reactivas surgidas en tales direcciones. Cabe mencionar que

los argumentos para el apoyo en turno son válidos si a este no se le impone un desplazamiento inicial como lo puede ser un asentamiento.

(Soto, 2014)

Una vez que se forma la matriz de rigidez de la estructura, los componentes de fuerzas global F que actúan sobre la armadura pueden relacionar con sus desplazamientos globales D utilizando

$$\{F\} = [K]\{D\}$$

Esta ecuación ya esta definida en apartado 2.2 se conoce como la ecuación de rigidez de la estructura.

Como siempre se han asignado los números más bajo de código para identificar los grados de libertad no restringidos, esto permitirá ahora hacer una partición de la ecuación en la forma siguiente: Esta ecuación matricial siempre puede ser reordenada considerando la definición de grados de libertad efectivos y restringidos de la siguiente forma:

$$\begin{Bmatrix} F_e \\ F_r \end{Bmatrix} = \begin{bmatrix} K_{ee} & K_{er} \\ K_{re} & K_{rr} \end{bmatrix} \begin{Bmatrix} D_e \\ D_r \end{Bmatrix} \quad (2.17)$$

Donde:

- $\{F_e\}$ Vector de las fuerzas según los grados de libertad efectivos.
- $\{F_r\}$ Vector de las fuerzas según los grados de libertad restringidos
- $\{D_e\}$ Vector de los desplazamientos según los grados de libertad efectivos.
- $\{D_r\}$ Vector de los desplazamientos según los grados de libertad restringidos (generalmente son iguales a cero).

Expandiendo la ecuación 2.17 se obtiene:

$$\{F_e\} = [K_{ee}]\{D_e\} + [K_{er}]\{D_r\} \quad (2.17.a)$$

$$\{F_r\} = [K_{re}]\{D_e\} + [K_{rr}]\{D_r\} \quad (2.17.b)$$

Despejando de la ecuación 2.17.a el vector $\{D_e\}$

$$\{D_e\} = [K_{ee}]^{-1} (\{F_e\} - [K_{er}]\{D_r\}) \quad (2.18)$$

Reemplazando en la ecuación 2.17.b y factorizando el vector $\{D_r\}$ se tiene:

$$\{F_r\} = [K_{re}][K_{ee}]^{-1}\{F_e\} + ([K_{rr}] - [K_{re}][K_{ee}]^{-1}[K_{er}])\{D_r\} \quad (2.19)$$

Las ecuaciones matriciales 2.18 y 2.19 constituyen la base de la solución del análisis matricial de estructuras. En el caso de que los desplazamientos restringidos sean nulos (caso muy común para los apoyos de la estructura) el vector $\{D_r\}$ es cero y las ecuaciones 2.18 y 2.19 se reducen a:

$$\{D_e\} = [K_{ee}]^{-1}\{F_e\} \quad (2.18.a)$$

$$\{F_r\} = [K_{re}]\{D_e\} + [K_{rr}]\{D_r\} \quad (2.19.a)$$

De esta ecuación puede obtenerse una solución directa para todos los desplazamientos de junta desconocidos entonces, a partir de la ecuación (2.19.a), con $\mathbf{D}_r = \mathbf{0}$ se obtiene la ecuación para obtener las reacciones desconocidas.

$$\{F_r\} = [K_{re}]\{D_e\} \quad (2.20)$$

Las fuerzas internas del elemento se pueden determinar mediante la ecuación 2.7 ya definidas en apartado de transformación de coordenadas de desplazamiento:

$$D_1^L = D_1 \cos \alpha + D_2 \sin \alpha$$

Figura 2.21 Fuerzas Internas 1

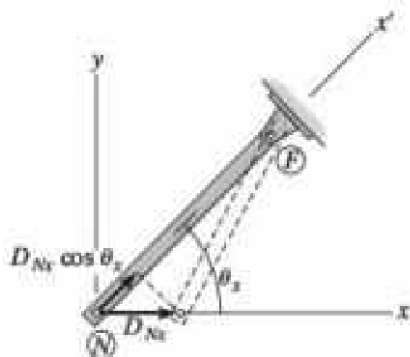
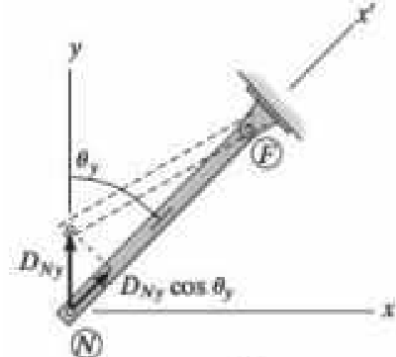


Figura 2.22 Fuerzas Internas 2



Nota. Gráficos para demostrar cómo se comportan las ecuaciones de las fuerzas internas de una barra. (R.C.HIBBELER, 2012)

Figura 2.23 Fuerzas Internas 3

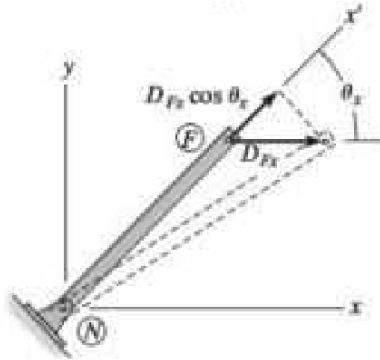
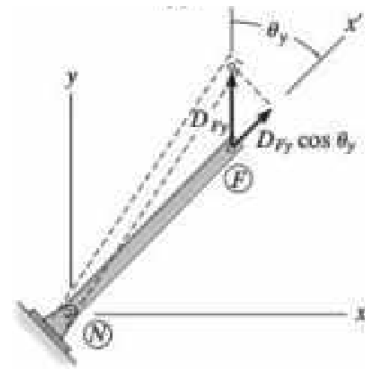


Figura 2.24 Fuerzas Internas 4



Nota. Gráficos para demostrar cómo se comportan las ecuaciones de las fuerzas internas de una barra. (R.C.HIBBELER, 2012)

De la ecuación 2.11b podemos remplazar

$$c = \cos \alpha \quad s = \sin \alpha$$

$$D_N = D_{NX} \cos \alpha + D_{NY} \sin \alpha$$

$$D_F = D_{FX} \cos \alpha + D_{FY} \sin \alpha$$

Podemos remplazar

$$D_N = D_{NX} c + D_{NY} s \quad (2.20.a)$$

$$D_F = D_{FX} c + D_{FY} s \quad (2.20.b)$$

Que puede escribirse matricialmente como:

$$\begin{Bmatrix} D_N \\ D_F \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \begin{Bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{Bmatrix} \quad (2.21)$$

O bien se puede escribir:

$$\{d\}=[T]\{D\} \quad (2.21.a)$$

Donde

$$T = \begin{bmatrix} C & S & 0 & 0 \\ 0 & 0 & C & S \end{bmatrix} \quad (2.21.b)$$

De la deducción anterior, T transformar los cuatro desplazamientos D globales x, y, en los dos desplazamientos d, locales. Por lo tanto, T se conoce como la matriz de transformación del desplazamiento.

$$q=[k][T]\{D\} \quad (2.22)$$

Al expandir esta ecuación se obtiene una ecuación la cual determina tensión en el elemento.

Escrita matricialmente y reemplazando la ecuación 2.22.

$$\begin{Bmatrix} D_N \\ D_F \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} C & S & 0 & 0 \\ 0 & 0 & C & S \end{bmatrix} \begin{Bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{Bmatrix} \quad (2.22.a)$$

Como $q_N = -q_F$ para el equilibrio, solo debe encontrarse una de las fuerzas. Aquí se determina q_F , la cual ejerce tensión en el elemento.

$$q = \frac{AE}{L} \begin{bmatrix} -C & -S & C & S \end{bmatrix} \begin{bmatrix} D_{NX} \\ D_{NY} \\ D_{FX} \\ D_{FY} \end{bmatrix} \quad (2.22.b)$$

En particular, si el resultado que se calcula mediante esta ecuación es negativo, entonces el elemento está en compresión. (R.C.HIBBELER, 2012)

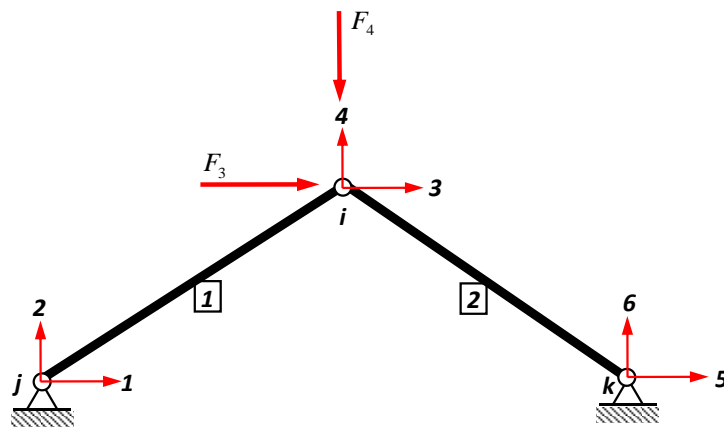
2.1.5.7 Ensamblaje de la Matriz de Rigidez General

Ya formada todas las matrices de rigidez de los elementos en coordenadas globales es necesario ensamblarlas en el orden correcto para que se pueda determinar la matriz de rigidez \mathbf{k} de toda la armadura.

Este proceso de combinación de matrices de los elementos depende de una cuidadosa identificación de los miembros de cada matriz del elemento. Como se analizó en la sección anterior, esto se hace por la designación de filas y columnas de la matriz mediante los cuatro números de código N_X, N_Y, F_X, F_Y que se utilizan para la identificación de grados de libertad globales que puedan ocurrir en cada extremo del elemento. La matriz de rigidez de la estructura tendrá entonces un orden que será igual al número de código mayor asignado a la armadura, ya que se representa la cantidad total de grados de libertad de la estructura. Cuando se ensamble las matrices \mathbf{k} , cada elemento en \mathbf{k} se podrá entonces en su misma designación de fila y columna en la matriz de rigidez de la estructura \mathbf{k} .

En particular cuando dos o más elementos están conectados a la misma junta o nodo, entonces algunos de los elementos de cada matriz del elemento se asignarán a la misma posición en la matriz \mathbf{k} . Cuando esto ocurre, los elementos asignados a la ubicación común deben sumarse algebraicamente. (R.C.HIBBELER, 2012)

Figura 2.25 Grados de Libertad



Nota. Gráficos para demostrar cómo se comportan los grados de libertad en la estructura. (Alvarez, 2020)

Cuando todas las matrices de rigidez de cada elemento de la estructura hayan sido determinadas, se procede al ensamblaje de la matriz de la estructura según los grados de libertad comunes a todas las barras que concurren al mismo nodo Tal como se muestra en la figura 2.25. Las ecuaciones de equilibrio, exigen que las cargas o fuerzas externas aplicadas en los nodos sean iguales a la suma de las sollicitaciones de extremo de los elementos que concurren al nodo. Es decir que:

$$\{F_i\} = \begin{Bmatrix} F_{xi} \\ F_{yi} \end{Bmatrix} \quad (2.23)$$

Donde, $\{F_i\}$ es el vector de las cargas o fuerzas externas aplicada en el nodo “i”

De esta manera, las ecuaciones matriciales de equilibrio para las barras $\boxed{1}$ y $\boxed{2}$ de la estructura mostrada en la figura 2.26, se pueden escribir de la siguiente forma:

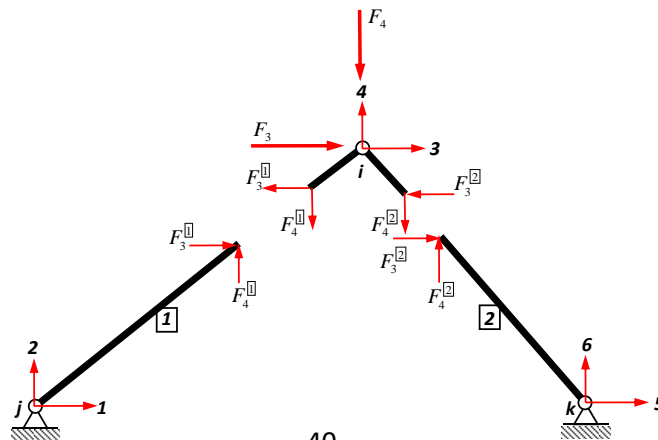
$$\begin{aligned} F_i^{\boxed{1}} &= k_i^{\boxed{1}} \cdot D_i^{\boxed{1}} \\ F_i^{\boxed{2}} &= k_i^{\boxed{2}} \cdot D_i^{\boxed{2}} \end{aligned} \quad (2.24)$$

Donde, en general se tiene:

- $F_i^{\boxed{n}}$ es el vector de las sollicitaciones en el extremo “i” de la barra “n”
- $D_i^{\boxed{n}}$ es el vector de deformación en el extremo “i” de la barra “n”
- $k_{ij}^{\boxed{n}}$ es uno de los coeficientes de rigidez del nodo “i” que proviene de un desplazamiento unitario en el nodo “j” de la barra “n”

En función de la notación establecida para para el nodo “i” del reticulado mostrado en la figura 26, y las condiciones de equilibrio en el nodo “i” resulta ser:

Figura 2.26 Equilibrio de fuerzas



Nota. Equilibrio de fuerzas en el nodo “i” de las barras $\boxed{1}$ y $\boxed{2}$. (Alvarez, 2020)

$$\begin{aligned} F_3 &= F_3^{\boxed{1}} + F_3^{\boxed{2}} \\ F_4 &= F_4^{\boxed{1}} + F_4^{\boxed{2}} \end{aligned} \quad (2.25)$$

Las ecuaciones de equilibrio 9.25 pueden representarse en forma matricial así:

Para la barra $\boxed{1}$

$$\begin{Bmatrix} F_3^{\boxed{1}} \\ F_4^{\boxed{1}} \end{Bmatrix} = \begin{bmatrix} k_{33}^{\boxed{1}} & k_{34}^{\boxed{1}} \\ k_{43}^{\boxed{1}} & k_{44}^{\boxed{1}} \end{bmatrix} \begin{Bmatrix} D_3^{\boxed{1}} \\ D_4^{\boxed{1}} \end{Bmatrix}$$

Para la barra $\boxed{2}$

$$\begin{Bmatrix} F_3^{\boxed{2}} \\ F_4^{\boxed{2}} \end{Bmatrix} = \begin{bmatrix} k_{33}^{\boxed{2}} & k_{34}^{\boxed{2}} \\ k_{43}^{\boxed{2}} & k_{44}^{\boxed{2}} \end{bmatrix} \begin{Bmatrix} D_3^{\boxed{2}} \\ D_4^{\boxed{2}} \end{Bmatrix} \quad (2.26)$$

Luego desarrollando el producto matricial en cada una de estas ecuaciones se tiene:

$$\begin{aligned} F_3 &= k_{33}^{\boxed{1}} \cdot D_3^{\boxed{1}} + k_{34}^{\boxed{1}} \cdot D_4^{\boxed{1}} + k_{33}^{\boxed{2}} \cdot D_3^{\boxed{2}} + k_{34}^{\boxed{2}} \cdot D_4^{\boxed{2}} \\ F_4 &= k_{43}^{\boxed{1}} \cdot D_3^{\boxed{1}} + k_{44}^{\boxed{1}} \cdot D_4^{\boxed{1}} + k_{43}^{\boxed{2}} \cdot D_3^{\boxed{2}} + k_{44}^{\boxed{2}} \cdot D_4^{\boxed{2}} \end{aligned} \quad (2.27)$$

Planteando ahora la condición de compatibilidad de las deformaciones se tiene que:

$$\begin{aligned} D_3^{\boxed{1}} &= D_3^{\boxed{2}} = D_3 \\ D_4^{\boxed{1}} &= D_4^{\boxed{2}} = D_4 \end{aligned} \quad (2.28)$$

Reemplazando la ecuación 2.28 en las ecuaciones 2.26 y luego en la ecuación 2.24 se obtiene:

$$\begin{aligned} F_3 &= \left(k_{33}^{\boxed{1}} + k_{33}^{\boxed{2}} \right) \cdot D_3 + \left(k_{34}^{\boxed{1}} + k_{34}^{\boxed{2}} \right) \cdot D_4 \\ F_4 &= \left(k_{43}^{\boxed{1}} + k_{43}^{\boxed{2}} \right) \cdot D_3 + \left(k_{44}^{\boxed{1}} + k_{44}^{\boxed{2}} \right) \cdot D_4 \end{aligned} \quad (2.29)$$

El significado físico de los resultados obtenidos en la ecuación 2.29, indica que cuando dos o más elementos de una estructura concurren a un mismo nodo, la rigidez global en la

dirección del grado de libertad común, es igual a la suma de las rigideces de todas las barras que llegan al mismo nodo. Expresado esto en términos matriciales se tiene:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} = \begin{bmatrix} k_{11}^{[1]} & k_{12}^{[1]} & k_{13}^{[1]} & k_{14}^{[1]} & 0 & 0 \\ \dots & k_{22}^{[1]} & k_{23}^{[1]} & k_{24}^{[1]} & 0 & 0 \\ \dots & \dots & (k_{33}^{[1]} + k_{33}^{[2]}) & (k_{34}^{[1]} + k_{34}^{[2]}) & k_{35}^{[2]} & k_{36}^{[2]} \\ \dots & \dots & \dots & (k_{44}^{[1]} + k_{44}^{[2]}) & k_{45}^{[2]} & k_{46}^{[2]} \\ \dots & \dots & \dots & \dots & k_{55}^{[2]} & k_{56}^{[2]} \\ \text{Sim} & \dots & \dots & \dots & \dots & k_{66}^{[2]} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{Bmatrix} \quad (2.30)$$

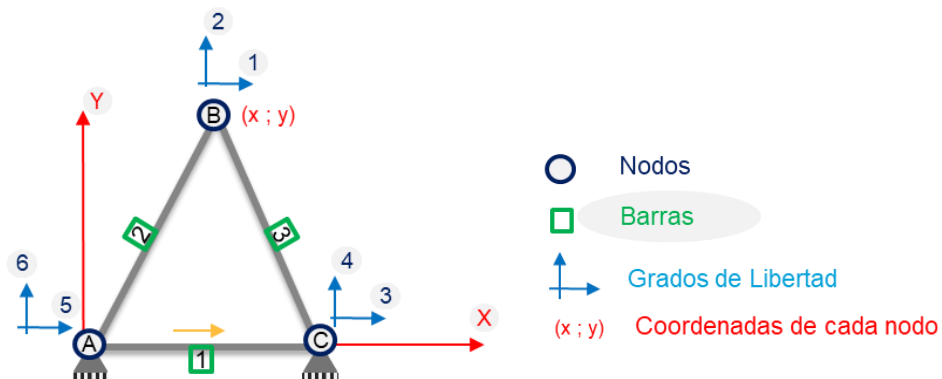
Los dígitos en la parte superior de la matriz de rigidez corresponden a los grados de libertad con los que se define la estructura. Se observa que las matrices de rigidez de cada elemento que tiene grados de libertad comunes con otra barra, se superponen en la matriz de rigidez general. A este procedimiento de superponer los coeficientes de rigidez se denomina ensamblaje de la matriz de rigidez general. (Alvarez, 2020)

2.1.6 Procedimiento de cálculo

1) Numeración de nodos, barras y grados de libertad

- Comenzar a numerar los grados de libertad, desde el nodo no restringido ver figura 2.27.
- Marcar el sentido por donde se va comenzar a calcular ver figura 2.27.

Figura 2.27 Numeración de nodos



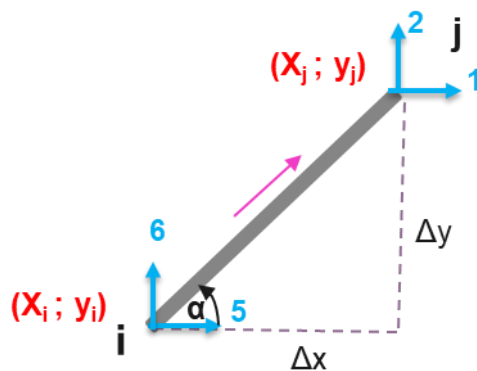
Nota. Gráficos para ilustrar la numeración de nudos, barras y grados de libertad.
Elaboración Propia

2) Definir el Sistema de Coordenadas

- Es aconsejable colocar el sistema de coordenadas en el primer nudo de la izquierda, así la estructura comensura con la coordenada (0;0).

3) Cálculo de Longitud, Área y Modulo de Elasticidad

- El módulo de elasticidad depende del material de la armadura
- El área es el área transversal de la barra



$$\Delta x = X_j - X_i \quad \Delta y = y_j - y_i$$

$$L = \sqrt{\Delta x^2 + \Delta y^2}$$

$$C = \Delta x / L \quad S = \Delta y / L$$

$$C = \frac{X_j - X_i}{\sqrt{\Delta x^2 + \Delta y^2}} = \cos \alpha$$

$$S = \frac{y_j - y_i}{\sqrt{\Delta x^2 + \Delta y^2}} = \sin \alpha$$

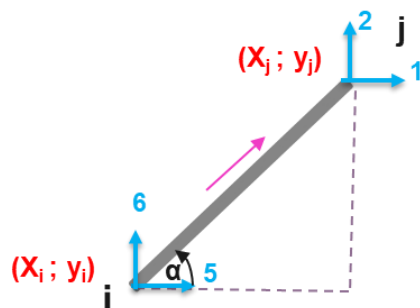
4) Matriz de Rigidez de Cada Elemento

$$K_{ij} = \frac{AE}{L} \begin{bmatrix} N_x & N_y & F_x & F_y \\ C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \end{bmatrix} \begin{matrix} N_x \\ N_y \\ F_x \\ F_y \end{matrix}$$

N = Grado de libertad del nudo inicial

F = Grado de libertad del nudo final

5) Ensamblaje de matriz de rigidez de toda la estructura



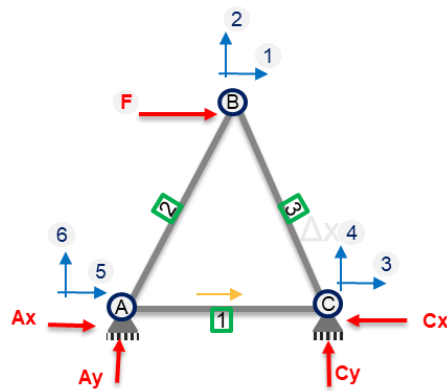
$$K_{ij} = \frac{AE}{L} \begin{bmatrix} X & X & X & X \\ X & X & X & X \\ X & X & X & X \\ X & X & X & X \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 1 \\ 2 \end{matrix}$$

Ejemplo ilustrativo de ensamblado global

$$K = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \end{bmatrix} \end{matrix}$$

6) Vectores de Fuerzas y Desplazamientos

Figura 2.28 Vectores de desplazamiento



✓ Vector de Desplazamientos

$$\begin{bmatrix} d1 \\ d2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

✓ Vector de Fuerzas

$$\begin{bmatrix} F \\ 0 \\ -Cx \\ Cy \\ Ax \\ Ay \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Nota. Gráficos para ilustrar el vector de fuerzas y desplazamientos. Elaboración Propia.

7) Establecimiento de Sistema de Ecuaciones

$$\{ F \} = [k] \{ D \}$$

$$\begin{Bmatrix} F_e \\ F_r \end{Bmatrix} = \begin{bmatrix} K_{ee} & K_{er} \\ K_{re} & K_{rr} \end{bmatrix} \begin{Bmatrix} D_e \\ D_r \end{Bmatrix}$$

Donde:

F_e = vector de fuerzas según GLE

F_r = Vector de fuerzas según GLR

D_e = Vector de desplazamiento según GLE

D_r = Vector de desplazamiento según GLR

$$\begin{bmatrix} F \\ 0 \\ -Cx \\ Cy \\ Ax \\ Ay \end{bmatrix} \begin{matrix} F_e \\ \\ F_r \end{matrix} = \begin{bmatrix} x & K_{ee} & x & x & x & x \\ x & x & x & K_{er} & x & x \\ \hline x & x & x & x & x & x \\ x & K_{re} & x & x & x & x \\ x & x & x & K_{rr} & x & x \\ x & x & x & x & x & x \end{bmatrix} \begin{bmatrix} D_e \\ d1 \\ d2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} D_r \\ \\ \\ D_r \end{matrix}$$

1 2 3 4 5 6

8) Resolución del sistema reducido (obtención de desplazamientos y las reacciones)

- Aplicamos la siguiente ecuación que ya están definidas en el anterior apartado

Para cálculo de desplazamiento ecuaciones 2.18:

$$F_e = K_{ee} D_e + K_{er} D_r$$

$$D_e = [K_{ee}]^{-1} F_e \quad \text{Cuando no hay un desplazamiento en los nudos.}$$

$$D_e = [K_{ee}]^{-1} (F_e - K_{er} D_r) \quad \text{Cuando hay desplazamiento en los nudos}$$

Para cálculo de reacciones ecuaciones 2.19:

$$F_r = K_{er} D_e \quad \text{Cuando no hay un desplazamiento en los nudos.}$$

$$F_r = K_{er} D_e + K_{rr} D_r \quad \text{Cuando hay desplazamiento en los nudos}$$

9) Fuerzas internas de cada barra

- Utilizamos la siguiente ecuación (2.22b) que ya fue deducida en el anterior párrafo.

$$q = \frac{AE}{L} \begin{bmatrix} -C & -S & C & S \end{bmatrix} \begin{bmatrix} D_{NX} \\ D_{NY} \\ D_{FX} \\ D_{FY} \end{bmatrix}$$

Donde:

q = Fuerza interna de la barra

D_N = Desplazamiento del nodo inicial de la barra

D_F = Desplazamiento del nodo final de la barra

2.2 CONCEPTOS FUNDAMENTALES

- ❖ **Coefficiente de flexibilidad.** Es La fuerza necesaria aplicada en el nodo “i” de un elemento estructural, para producir un desplazamiento unitario en el nodo “j” mientras los otros nodos no soportan carga alguna.
- ❖ **Coefficiente de rigidez.** Es el desplazamiento necesario generado en el nodo “i” de un elemento estructural, por una fuerza unitaria aplicada en el nodo en el nodo “j” mientras los otros nodos permanecen fijos.
- ❖ **coordenadas locales.** Las coordenadas locales son un sistema de referencia utilizado para describir la posición o ubicación de un punto, objeto o entidad dentro de un sistema de coordenadas específico. Este sistema de coordenadas se define en relación con un punto de origen y los ejes de referencia asociados.
- ❖ **coordenadas globales.** En el contexto de la ingeniería estructural, las coordenadas globales se refieren a un sistema de coordenadas utilizado para describir y ubicar los elementos de una estructura en relación con un punto de referencia fijo en el espacio. Estas coordenadas se utilizan para definir la posición y la orientación de los componentes estructurales en el entorno tridimensional.
- ❖ **Definición de nodo.** Generalmente la ubicación de un sistema de coordenadas locales, se asume que pueda estar ubicado en los nudos iniciales de cada elemento, pero también a veces es necesario definir la configuración de algún otro punto que no sea necesariamente un nudo o apoyo, en estos casos corresponde definir un nuevo nudo en el punto donde se requiera realizar el análisis, por lo tanto para generalizar la ubicación de estos nudos particulares y los de barras es conveniente denominarlos nodos antes que nudos.
Por lo tanto, un nodo resulta ser un punto cualquiera que se encuentra contenido en el elemento entre los extremos del mismo.

CAPÍTULO III
FUNDAMENTO TEÓRICO
DE SOFTWARE

CAPÍTULO III

FUNDAMENTOS TEÓRICOS DE SOFTWARE

3.1 QUÉ ES UN SOFTWARE

El software es un conjunto de reglas o programas que dan instrucciones a un ordenador para que realice tareas específicas. También se conoce como aplicaciones de software, paquetes de software, herramientas de software y programas de software. El software puede utilizarse para gestionar datos, automatizar procesos y crear aplicaciones o productos informáticos. Su complejidad puede variar desde un simple programa de tratamiento de textos hasta complejos sistemas informáticos que controlan infraestructuras críticas en sectores como la sanidad y el transporte. (Armetrics, 2022)

3.2 ORIGEN E HISTORIA DEL SOFTWARE

El origen e historia del software están estrechamente relacionados con el desarrollo de la computación y la evolución de las tecnologías de la información a lo largo del tiempo. Aquí tiene una breve descripción de los hitos más importantes en la historia del software:

- ❖ **Década de 1940:** Los primeros ordenadores electrónicos, como la ENIAC (Electronic Numerical Integrator and Computer), no tenían software en el sentido moderno. En cambio, se programaban manualmente mediante el cableado de conexiones y el ajuste de interruptores físicos para realizar cálculos específicos.
- ❖ **Década de 1950:** Con la invención de los lenguajes de programación de nivel más alto, como el Fortran y el COBOL, los programadores podían escribir instrucciones en un formato más comprensible en lugar de utilizar el cableado directo. Esto marcó el comienzo del desarrollo de software como lo conocemos hoy.
- ❖ **Década de 1960:** Se desarrollaron sistemas operativos para gestionar y coordinar las actividades de los ordenadores. IBM lanzó el sistema operativo OS/360, uno de los primeros sistemas operativos comerciales ampliamente utilizados.

- ❖ **Década de 1970:** El concepto de software de código abierto comenzó a tomar forma con proyectos como el sistema operativo UNIX. Además, se introdujeron las microcomputadoras, lo que hizo que la programación fuera más accesible para las personas fuera del ámbito académico o empresarial.
- ❖ **Década de 1980:** Se popularizaron las interfaces gráficas de usuario (GUI) con la introducción de sistemas como el Apple Macintosh y Microsoft Windows. Esto hizo que las computadoras fueran más amigables para los usuarios finales.
- ❖ **Década de 1990:** La World Wide Web se desarrolló y se popularizó, lo que llevó a la creación de navegadores web y aplicaciones en línea. Además, esta década vio el auge de las aplicaciones de software empresarial y de consumo.
- ❖ **Década de 2000 en adelante:** La proliferación de dispositivos móviles, la computación en la nube y la expansión de la conectividad a Internet continuaron impulsando la evolución del software. Además, surgieron enfoques de desarrollo ágil y metodologías de desarrollo de software más colaborativas.

Hoy en día, el software desempeña un papel crítico en prácticamente todos los aspectos de la vida moderna, desde la comunicación y el entretenimiento hasta la gestión empresarial y la investigación científica. Ha evolucionado enormemente desde sus humildes comienzos y seguirá siendo una parte central de la tecnología y la sociedad en el futuro. (Armetrics, 2022)

3.3 TIPOS DE SOFTWARE

Los tipos de software pueden clasificarse a grandes rasgos en dos categorías principales:

- ❖ **Software de sistema**, o sistemas operativos, proporcionan la capa base sobre la que se ejecutan todas las demás aplicaciones de software. El software de sistema incluye el propio sistema operativo, los controladores de dispositivos, el middleware, los navegadores web, los reproductores multimedia y muchos más.
- ❖ **Software de aplicación**, o aplicaciones de software, son programas diseñados para realizar tareas específicas. Puede incluir software como procesadores de texto y herramientas de hojas de cálculo, o productos de software como sistemas

de planificación de recursos empresariales que ayudan a las empresas a gestionar sus operaciones. (Arimerics, 2022)

3.4 INTRODUCCIÓN A JAVA

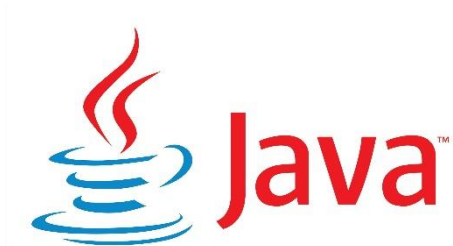
Java es un lenguaje de programación desarrollado por Sun Microsystems. Java fue presentado en la segunda mitad del año 1995 y desde entonces se ha convertido en un lenguaje de programación muy popular.

Java se destaca por ser un lenguaje orientado a objetos, lo que significa que se basa en la creación de objetos que interactúan entre sí para realizar tareas. Esto permite una estructura modular y flexible en la programación, facilitando el desarrollo de aplicaciones grandes y complejas.

Una de las características clave de Java es su portabilidad. Las aplicaciones escritas en Java pueden ejecutarse en diferentes plataformas, como Windows, macOS y Linux, sin necesidad de reescribir el código fuente. Esto se debe a que Java utiliza una máquina virtual Java (JVM) que interpreta el código compilado, lo que brinda una capa de abstracción entre el lenguaje de programación y el sistema operativo subyacente.

Dada la facilidad de contar con estos marcos de trabajo (framework) gratuitos, son muchos los desarrollos que existen a nivel mundial y que se han convertido en aplicaciones importantes para las empresas. Lo que inició como un compilador para electrodomésticos, poco a poco se convirtió en una herramienta de uso general para el desarrollo de aplicaciones web o para dispositivos móviles. (Luis Joyanes Aguilar, 2011)

Figura 3.1 Logo del lenguaje Java



Nota. Captura del logo del lenguaje Java. (Gomez Jimenez Enrique, 2019)

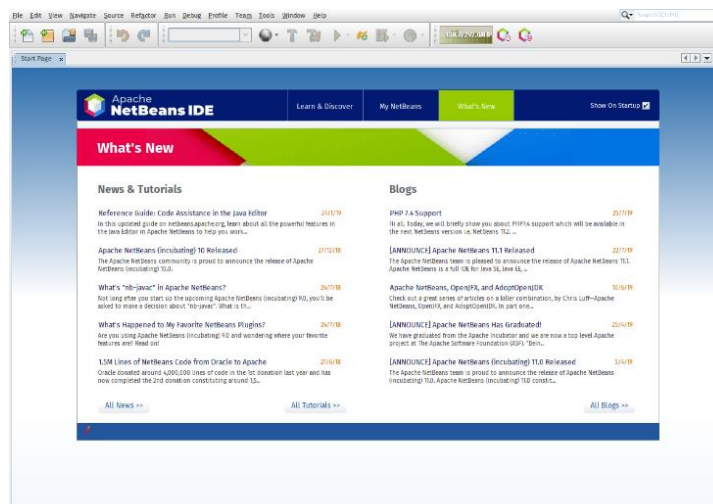
3.5 INTRODUCCIÓN A NETBEANS

NetBeans es un entorno de desarrollo de uso libre que fue creado para utilizar lenguaje Java. Existe una variedad de módulos diseñados especialmente para hacerlo extensible. Esta cualidad hace que NetBeans sea lo suficientemente poderoso y escriba aplicaciones Java para escritorio, así como para dispositivos móviles. Cuenta con una interfaz gráfica para aplicaciones de escritorio rica en componentes y librerías de reutilización.

NetBeans proporciona un entorno completo para escribir, compilar, depurar y ejecutar código. Ofrece características como resaltado de sintaxis, finalización automática de código, navegación rápida por el código, depuración paso a paso, refactorización de código, integración con sistemas de control de versiones y soporte para pruebas y perfiles de rendimiento. Una de las ventajas de NetBeans es su estrecha integración con el lenguaje de programación Java. Ofrece un amplio soporte para el desarrollo de aplicaciones Java, incluyendo la creación de interfaces gráficas de usuario (GUI) con la biblioteca Swing, la gestión de proyectos, la administración de dependencias y la generación automática de código.

Además, NetBeans cuenta con una comunidad activa que desarrolla y mantiene una amplia gama de complementos y extensiones que agregan funcionalidades adicionales al IDE. Estos complementos pueden incluir soporte para otros lenguajes de programación, frameworks específicos, herramientas de desarrollo y características personalizadas según las necesidades del desarrollador. (Gomez Jimenez Enrique, 2019)

Figura 3.2 Software NetBeans



Nota. Captura de imagen del entorno de NetBeans. (Gomez Jimenez Enrique, 2019)

3.6 REVISIÓN BIBLIOGRÁFICA

Se realizará una revisión bibliográfica para esta investigación, la cual abarcará las siguientes fuentes:

- ❖ Apuntes de la materia de Elementos Finitos: Estos apuntes proporcionarán información tanto teórica como práctica sobre el método de Elementos Finitos aplicado a la solución de cerchas en 2D. Se espera que estos apuntes cubran los conceptos fundamentales, las técnicas de análisis y los procedimientos de cálculo necesarios para abordar el problema de las cerchas.
- ❖ Libros especializados en la solución de cerchas utilizando el método de la matriz de rigidez: La consulta de libros dedicados específicamente a la solución de cerchas mediante el método de la matriz de rigidez permitirá obtener un mayor nivel de detalle y conocimiento sobre esta técnica en particular. Estos libros pueden proporcionar ejemplos prácticos, análisis detallados y casos de estudio que serán de utilidad para comprender y aplicar el método en el contexto de la investigación.
- ❖ Apuntes del Instituto relacionados con el lenguaje de programación Java: Estos apuntes, que se enfocan en la explicación del lenguaje de programación Java mediante ejemplos, serán una fuente valiosa para adquirir los conocimientos necesarios en el desarrollo del software. Se espera que los apuntes abarquen desde los conceptos básicos hasta aspectos más avanzados del lenguaje, con el objetivo de proporcionar una comprensión completa y práctica.
- ❖ Investigación en libros y recursos en línea sobre programación: Se llevará a cabo una investigación exhaustiva tanto en libros como en recursos en línea para ampliar el conocimiento sobre programación en general. Se buscarán fuentes confiables que cubran diversos aspectos de la programación, lo que permitirá adquirir una base sólida de conocimientos y habilidades necesarias para alcanzar los objetivos de la investigación.

En resumen, la revisión bibliográfica comprenderá la consulta de apuntes de Elementos Finitos, libros especializados en la solución de cerchas por el método de la matriz de rigidez, apuntes del Instituto sobre programación en Java y una investigación en libros y recursos en línea sobre programación. Estas fuentes proporcionarán el conocimiento necesario para el desarrollo de la investigación y el cumplimiento de los objetivos planteados.

Para obtener datos relevantes, se utilizó una combinación de métodos de investigación, como la observación directa, entrevistas y cuestionarios. El cuestionario está detallado en el [Anexo 1](#). También se respetaron los principios éticos, garantizando la confidencialidad y el consentimiento informado de los participantes.

Los resultados de la investigación proporcionaron una visión más clara del comportamiento de los estudiantes durante los exámenes y los trabajos prácticos. Se identificaron patrones comunes y se analizaron los factores que influyen en el desempeño de los estudiantes en cada contexto.

3.7 METODOLOGÍA DE DESARROLLO DE SOFTWARE

3.7.1. Metodología Ágil

La metodología ágil es un enfoque de desarrollo de software que se basa en principios y valores colaborativos, flexibles y adaptativos. Surgió como alternativa a los enfoques tradicionales de desarrollo de software, como el modelo en cascada, que se caracterizaban por procesos rígidos y secuenciales.

El concepto de metodología ágil se basa en el Manifiesto Ágil, un documento que fue redactado en 2001 por un grupo de expertos en desarrollo de software. Los principios fundamentales del Manifiesto Ágil son:

Individuos e interacciones sobre procesos y herramientas: Se enfatiza la importancia de la comunicación y la colaboración entre los miembros del equipo de desarrollo, ya que esto facilita la comprensión de los requisitos y la toma de decisiones efectivas.

Software funcionando sobre documentación extensiva: Se valora más el software funcional y en ejecución que la documentación detallada. La prioridad es entregar valor rápidamente al cliente y obtener retroalimentación temprana.

Colaboración con el cliente sobre negociación de contratos: La interacción continua con el cliente es esencial para entender sus necesidades cambiantes y adaptar el desarrollo en consecuencia.

Respuesta ante el cambio sobre seguir un plan: En lugar de seguir un plan rígido, la metodología ágil reconoce la incertidumbre inherente en el desarrollo de software y fomenta la adaptabilidad ante cambios en los requisitos.

El objetivo principal de la metodología ágil es aumentar la satisfacción del cliente, entregar productos de alta calidad y responder de manera efectiva a los cambios en el entorno y las necesidades del usuario, lo que la convierte en una opción popular para el desarrollo de software en la actualidad. (Maida & Pacienza, 2015)

3.7.2. Tipos de Metodología Ágiles

Existen varios tipos de metodologías ágiles, cada una con sus características específicas y enfoques particulares para el desarrollo de proyectos. Algunos de los tipos más conocidos son:

- ❖ **Scrum:** Es una de las metodologías ágiles más populares y ampliamente utilizadas. Se basa en ciclos cortos de trabajo llamados "sprints", que generalmente duran de 1 a 4 semanas. Durante cada sprint, se desarrolla una parte funcional del producto y al final del sprint se presenta el resultado al cliente.
- ❖ **Kanban:** Se centra en la visualización del flujo de trabajo y la gestión de tareas a través de tableros Kanban. Las tareas se mueven a través de diferentes etapas del proceso, lo que permite un control más efectivo del trabajo en progreso.
- ❖ **Extreme Programming (XP):** Se enfoca en la calidad del software mediante prácticas como la programación en pareja, las pruebas automatizadas y la

integración continua. XP también promueve una comunicación cercana entre los desarrolladores y los usuarios del software.

- ❖ **Crystal:** Es una familia de metodologías ágiles que se adapta a diferentes tamaños y complejidades de proyectos. Crystal busca optimizar la colaboración y la comunicación en equipos de desarrollo.
- ❖ **Dynamic Systems Development Method (DSDM):** Esta metodología ágil se enfoca en el desarrollo rápido y flexible de sistemas de información. DSDM se centra en la entrega temprana de funcionalidades de alto valor y la incorporación constante de comentarios del cliente.
- ❖ **Feature Driven Development (FDD):** Se basa en la creación de modelos de diseño y desarrollo basados en características o funcionalidades. FDD promueve la colaboración y la comunicación efectiva entre los miembros del equipo.
- ❖ **Lean Software Development:** Inspirada en el sistema de producción Lean, esta metodología ágil busca eliminar el desperdicio y maximizar el valor entregado al cliente. Se enfoca en la eficiencia y la entrega continua.
- ❖ **Agile Unified Process (AUP):** Es una versión ágil de RUP (Rational Unified Process) que combina conceptos ágiles con buenas prácticas de desarrollo. AUP se centra en la simplicidad y la adaptabilidad. (Maida & Pacienza, 2015)

3.7.3. Metodología utilizada para esta investigación AUP (Proceso Unificado Ágil)

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas

tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP. (Ingeniería de Software, 2023)

3.7.3.1 Fases del desarrollo de la metodología AUP

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- ❖ **Incepción:** El objetivo de esta fase es obtener una comprensión común cliente – equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- ❖ **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- ❖ **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- ❖ **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción (Ingeniería de Software, 2023)

3.8 MEDIOS Y RECURSOS

Los medios que se emplearán para realizar el prototipo y la documentación incluyen los siguientes:

- ❖ Computadora: Se utilizará una computadora para llevar a cabo el desarrollo del programa. En esta computadora se instalará un entorno de desarrollo integrado (IDE) que permitirá escribir, compilar y depurar el código del programa.
- ❖ IDE: Se utilizará el IDE NetBeans, el cual es un entorno de desarrollo integrado libre y gratuito, diseñado principalmente para el lenguaje de programación Java. NetBeans ofrece una amplia gama de herramientas y funcionalidades que facilitan el proceso de desarrollo de software.
- ❖ Bibliografía: Se consultará bibliografía relacionada con el método teórico de la Matriz de Rigidez en la resolución de cerchas. Esta bibliografía proporcionará los fundamentos teóricos necesarios para comprender y aplicar correctamente el método en el desarrollo del programa.
- ❖ Lenguaje de programación Java: El programa se desarrollará utilizando el lenguaje de programación Java. Java es un lenguaje ampliamente utilizado en el desarrollo de aplicaciones y cuenta con una amplia comunidad de soporte y recursos disponibles.
- ❖ Recursos en línea: Además de la bibliografía, se utilizarán recursos en línea como tutoriales, documentación oficial de Java y foros de programación. Estos recursos proporcionarán información adicional, ejemplos prácticos y soluciones a posibles problemas durante el desarrollo del programa.

En conclusión, se utilizará una computadora con el IDE NetBeans instalado para desarrollar el programa. Se consultará bibliografía relacionada con el método teórico de la Matriz de Rigidez y se utilizará el lenguaje de programación Java. Además, se hará uso de recursos en línea para obtener información adicional y solucionar problemas durante el desarrollo del programa.

CAPÍTULO IV
PROGRAMACIÓN GENERAL
DEL SOFTWARE

CAPÍTULO IV

PROGRAMACIÓN GENERAL DEL SOFTWARE

4.1. DESARROLLO DE LA METODOLOGIA AUP

A continuación, te describo cómo se aplicaría cada fase de la metodología AUP para el desarrollo del software de análisis de cerchas:

FASE 1 Inicio:

En esta fase, se identificaría el propósito y el alcance del proyecto, definiendo claramente el objetivo del programa de análisis de estructuras y los resultados esperados. Se determinarían los requisitos iniciales, como los parámetros iniciales del software y las funcionalidades clave a desarrollar. Se realizaría un análisis de viabilidad para asegurar que el proyecto sea viable técnicamente y en términos de recursos. Se comenzaría a planificar el proyecto, estableciendo un cronograma inicial.

FASE 2 Elaboración:

Durante esta fase, se enfocaría en el análisis y diseño detallado del programa. Se definirían los casos de uso clave y se elaborarían los diagramas de clases y diagramas de secuencia para modelar la arquitectura del software. Se detallarían los requerimientos específicos, como la interacción con la interfaz gráfica, el cálculo de fuerzas y desplazamientos, y la visualización de resultados. Se realizaría un análisis de riesgos para identificar posibles desafíos y mitigarlos. Además, se llevaría a cabo una revisión de la arquitectura para garantizar que el diseño sea sólido y adecuado para cumplir con los objetivos del programa.

FASE 3 Construcción:

Durante esta fase, se llevaría a cabo la implementación del programa en iteraciones. Se comenzaría por desarrollar las funcionalidades clave

identificadas en la fase de Elaboración, como el ingreso de coordenadas, tipos de apoyo, tipos de carga, cálculos de fuerzas y desplazamientos, entre otros. Se realizarían pruebas continuas para validar y asegurar la calidad del código. El equipo de desarrollo se enfocaría en agregar funcionalidades y construir el núcleo del programa, iterando en ciclos cortos y entregando incrementos funcionales en cada iteración.

FASE 4 Transición:

En esta fase, se prepararía el programa para su entrega y puesta en producción. Se llevarían a cabo pruebas de aceptación y se corregirían los errores y problemas identificados. Se realizaría la documentación completa del programa, incluyendo manuales de usuario y un video tutorial para instalación y el uso del programa, ver [Anexo 8](#).

Es importante destacar que la metodología AUP permite adaptar y ajustar las fases y actividades según las necesidades del proyecto, por lo que el equipo de desarrollo puede modificar el enfoque y la planificación en función de los requisitos específicos del programa de análisis de estructuras. La iteratividad e incrementalidad de AUP también permitirá obtener retroalimentación temprana del cliente y realizar ajustes a medida que se avanza en el desarrollo del programa.

4.1.1 FASE 1 Inicio

4.1.1.1 Presupuesto

Tabla 4.1 Presupuesto para desarrollo del Software

| PRESUPUESTO | | | |
|---|--------------------|-----------------------|------------------------------|
| DESCRIPCION | COSTO (Bs) | FINANCIAMIENTO | RECURSOS PROPIOS (Bs) |
| Investigación del método de matriz de rigidez | 40 | ---- | 40 |
| Investigación del lenguaje de programación Java | 50 | ---- | 50 |
| Curso de lenguaje de programación java | 250 | ---- | 250 |
| Diseño de Front ends | 10 | ---- | 10 |
| Diseño de Back ends | 10 | ---- | 10 |
| Diseño de Logo para el Software | 30 | ---- | 30 |
| Capacitación de Photoshop | 180 | ---- | 180 |
| Prueba para detectar fallos y corregirlos | 40 | ---- | 40 |
| Mantenimiento de la computadora | 120 | ---- | 120 |
| Manual de Usuario | 120 | ---- | 120 |
| Poner el software en funcionamiento | 20 | ---- | 20 |
| 1 Resmas de Papel Bond | 38 | ---- | 38 |
| Impresiones | 150 | ---- | 150 |
| TOTAL | 1.058,00 Bs | | 1.058,00 Bs |
| | | | 1.058,00 Bs |

Elaboración propia.

4.1.1.2 Cronograma de Actividades

4.1.1.2.1 Cronograma

Un cronograma es una representación gráfica y detallada de las actividades o tareas de un proyecto dispuestas en una secuencia temporal. Proporciona una visión clara de cuándo se llevarán a cabo las actividades y cuánto tiempo se asigna a cada una. Los cronogramas son herramientas esenciales en la gestión de proyectos, ya que ayudan a planificar, organizar y dar seguimiento al progreso del proyecto.

4.1.1.2.2 Diagrama de GANTT

Un diagrama de Gantt es una representación gráfica de un cronograma de proyectos que muestra las tareas del proyecto en una línea de tiempo. Las tareas se representan como barras horizontales que indican su inicio, duración y finalización previstos. El diagrama de Gantt es una herramienta popular en la gestión de proyectos, ya que permite a los equipos visualizar fácilmente las fechas de inicio y finalización de las tareas, las dependencias entre tareas y la asignación de recursos. Facilita la comunicación y el seguimiento del progreso del proyecto. El diagrama de Gantt esta detallado en el [Anexo 7](#)

4.1.1.3 Requerimiento Funcionales y no Funcionales

Los requerimientos para este programa son las especificaciones detalladas que definen las funcionalidades y características esenciales que el software debe tener. Estos requerimientos guían el proceso de desarrollo y aseguran que el programa cumpla con los objetivos establecidos.

4.1.1.3.1 Requerimiento Funcional

Tabla 4.2 Requerimiento Funcionales

| ID | Nombre del Requerimiento | Descripción |
|------|--|--|
| RF01 | ESPECIFICACIÓN DEL SISTEMA DE UNIDADES | El programa debe permitir al usuario seleccionar el sistema de unidades con el que trabajará durante el análisis de estructuras. Las opciones disponibles deben incluir el Sistema |

| | | |
|------|--|---|
| | | Internacional (SI), el Sistema Imperial y el Sistema Métrico. |
| RF02 | DETALLAR EL NÚMERO DE BARRAS | El programa debe solicitar al usuario el ingreso del número total de barras que conforman la estructura a analizar. Este valor será utilizado para definir la cantidad de elementos estructurales individuales considerados en el análisis. |
| RF03 | CARACTERÍSTICAS DEL MATERIAL COMO EL ÁREA Y EL MÓDULO DE ELASTICIDAD | El usuario debe proporcionar el valor del módulo de elasticidad del material y el área transversal de cada barra. Esta información será utilizada para el cálculo de las fuerzas y desplazamientos en las barras. El programa debe aceptar este valor en unidades apropiadas, según el sistema de unidades seleccionado. |
| RF04 | COORDENADAS DEL NODO INICIAL DE LA BARRA | El programa debe solicitar al usuario ingresar las coordenadas (X, Y) del nodo inicial de cada barra de la estructura. Estas coordenadas representan la posición del extremo inicial de la barra en el plano 2D. |
| RF05 | COORDENADAS DEL NODO FINAL DE LA BARRA | El usuario debe ingresar las coordenadas (X, Y) del nodo final de cada barra de la estructura. Estas coordenadas indican la posición del extremo final de la barra en el mismo plano 2D. |
| RF06 | GRADOS DE LIBERTAD | El programa debe permitir al usuario definir y asignar los grados de libertad para cada nodo de la estructura, lo que incluye traslaciones en los ejes X, Y, así como rotaciones alrededor de los ejes X, Y. El usuario debe poder especificar las restricciones de grados de libertad según sea necesario para simular apoyos fijos o móviles. |
| RF07 | INTRODUCIR EL TIPO DE APOYO | El programa debe permitir al usuario introducir el tipo de apoyo para cada nodo de la estructura. Al ingresar las coordenadas del nodo, se presentará |

| | | |
|------|---|--|
| | | <p>un menú desplegable (Combo Box) que contendrá las opciones de tipos de apoyo disponibles, tales como apoyo fijo, apoyo móvil. El usuario deberá seleccionar el tipo de apoyo adecuado para cada nodo según las condiciones reales de la estructura.</p> |
| RF08 | <p>INTRODUCIR EL TIPO DE CARGA</p> | <p>El programa debe permitir al usuario introducir el tipo de carga que se aplicará a la estructura. Al ingresar las coordenadas del punto donde se ubicará la carga, se presentará un menú desplegable (Combo Box) que contendrá las opciones de tipos de carga disponibles. El usuario deberá seleccionar el tipo de carga adecuado para cada punto de aplicación.</p> <p>Además, el programa debe ofrecer la opción de especificar la magnitud y dirección de la carga, en términos de fuerza, según el tipo de carga seleccionado.</p> |
| RF09 | <p>INTRODUCIR LAS FUERZAS QUE ESTA SOMETIDA LA ESTRUCTURA</p> | <p>El programa debe permitir al usuario introducir las fuerzas externas que están sometidas a la estructura. Para ello, se proporcionará una tabla o formulario donde el usuario podrá ingresar las magnitudes y direcciones de las fuerzas que actúan</p> |
| RF10 | <p>INTODUCIR LOS DESPLZAMIENTOS DE LOS NUDOS</p> | <p>El programa debe permitir al usuario introducir los desplazamientos de los nodos de la estructura después de realizar el análisis. Se proporcionará una tabla o formulario donde el usuario podrá ingresar los valores numéricos de los desplazamientos en las direcciones X, Y, para cada nodo de la estructura.</p> |
| RF11 | <p>BOTON DIBUJAR</p> | <p>El programa debe contar con un botón denominado "Dibujar" que permita visualizar la</p> |

| | | |
|------|----------------|--|
| | | representación gráfica de la estructura analizada. Al hacer clic en este botón, se mostrará una representación gráfica de las barras en un plano 2D, y también se ira calculando la matriz de rigidez de cada barra que se mostrará en la pestaña de procedimiento. |
| RF12 | BOTON CALCULAR | <p>El programa debe contar con un botón denominado "Calcular" que permita iniciar el proceso de análisis estructural una vez que se hayan ingresado todos los datos necesarios, como las coordenadas de los nodos, las propiedades de las barras, los tipos de apoyo y las fuerzas externas.</p> <p>Al hacer clic en el botón "Calcular", el programa debe realizar el análisis estructural utilizando el Método de Rigidez, según lo establecido previamente. Se calcularán los desplazamientos, las reacciones de apoyo, las fuerzas internas.</p> |

Elaboración Propia

4.1.1.3.2 Requerimiento No Funcional

Tabla 4.3 Requerimiento No Funcionales

| Tipo | ID | Nombre del Requerimiento | Descripción |
|------------|------|----------------------------|--|
| Usabilidad | RF01 | FUNCIONALIDAD DEL PROGRAMA | La interfaz gráfica debe ser intuitiva y fácil de usar, permitiendo que los usuarios ingresen datos, realicen análisis y visualicen resultados sin dificultad. |
| | RF02 | PORTABILIDAD | El programa debe ser compatible con diferentes sistemas operativos y |

| | | | |
|-----------------|------|---------------------------|--|
| | | | dispositivos, permitiendo su ejecución en diversas plataformas |
| Seguridad | RF03 | SEGURIDAD DEL PROGRAMA | Se deben implementar medidas de seguridad para proteger los datos y resultados del análisis, evitando accesos no autorizados o pérdida de información |
| | RF04 | SOPORTE TÉCNICO | Debe existir un canal de soporte técnico para los usuarios, brindando asistencia en caso de problemas o dudas relacionadas con el programa |
| Multiplataforma | RF05 | ESCALABILIDAD | El programa debe ser capaz de manejar estructuras de diferentes tamaños y complejidades, sin afectar significativamente el rendimiento y la precisión. |
| | RF06 | INTEROPERABILIDAD | El programa debe permitir la importación y exportación de datos en formatos estándar, lo que facilita la integración con otros programas y herramientas. |
| Rendimiento | RF07 | RENDIMIENTO DEL PROGRAMA | El programa debe ser eficiente y responder rápidamente a las acciones del usuario, incluso al analizar estructuras complejas con un gran número de barras y nodos. |
| | RF08 | PRECISIÓN Y CONFIABILIDAD | Los cálculos y resultados del análisis deben ser precisos y confiables, asegurando que el programa brinde resultados consistentes y exactos en todas las situaciones. |
| Desempeño | RF09 | EFICIENCIA ENERGÉTICA | El programa debe ser diseñado para minimizar el consumo de recursos de hardware, como memoria y capacidad de procesamiento, para reducir el impacto ambiental y mejorar la vida útil de los dispositivos |

| | | | |
|--|------|----------------|---|
| | RF10 | MANTENIBILIDAD | El código del programa debe ser estructurado, bien documentado y fácil de mantener, facilitando futuras actualizaciones y mejoras |
|--|------|----------------|---|

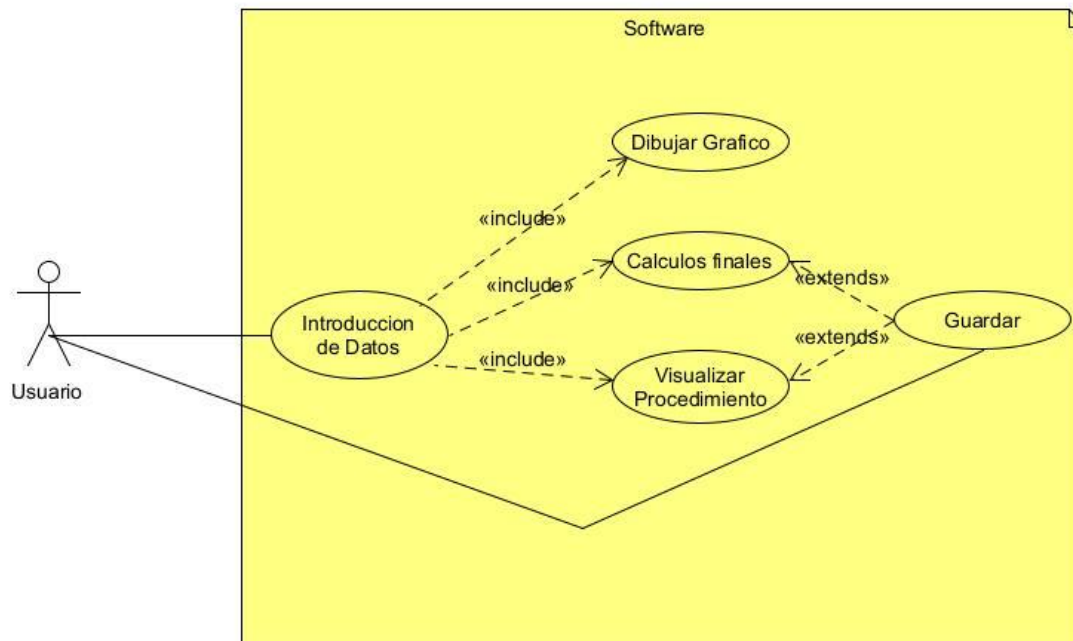
Elaboración Propia

4.1.2. FASE 2 Elaboración

Durante esta fase, se crea la aplicación para el software y el diseño incluye la metáfora que dará vida al software. El resultado de la fase de elaboración es la línea de base de la arquitectura, que representa esencialmente un esqueleto del sistema.

4.1.2.1 Modelado de casos de uso y descripción

Figura 4.1 Caso de Uso – Usuario



Nota. Diagrama realizado en el programa Umlet. Elaboración Propia

4.1.2.1.1 Descripción del caso de Uso-Usuario

Tabla 4.4 Descripción del Caso de Uso-Introducción de Datos

| Nombre de Caso de Uso | Introducción de datos |
|--------------------------|--|
| Objetivo | Permite la introducción de datos |
| Actor | Usuario |
| Precondiciones | Usuario de software |
| Flujo Principal | *El usuario ingresa los datos *Si los datos están bien cargados el sistema comenzará el calculo |
| Flujo Alternativo | *Si los datos están mal introducidos el software no realizara los cálculos |
| Post condiciones | Si los datos están bien ingresado al software, procederá a realizar el calculo |

Elaboración Propia

Tabla 4.5 Descripción del Caso de Uso- Dibujar Gráfico

| Nombre de Caso de Uso | Dibujar Grafico |
|--------------------------|--|
| Objetivo | Recibe las coordenadas y lo grafica |
| Actor | Usuario |
| Precondiciones | Debe existir un usuario de software |
| Flujo Principal | *El software recibe las coordenadas *Si las coordenadas están bien se procederá a graficar *La librería Plot recibe las coordenadas y lo grafica en su panel |
| Flujo Alternativo | *Si los datos están mal introducidos el software no realizara la grafica |
| Post condiciones | Si los datos están bien ingresado al software, procederá a realizar la gráfica en la pestaña de gráficos |

Elaboración Propia

Tabla 4.6 Descripción del Caso de Uso- Cálculo final

| Nombre de Caso de Uso | Calculo final |
|------------------------|--|
| Objetivo | Recibe los datos |
| Actor | Usuario |
| Precondiciones | Debe existir un usuario de software |
| Flujo Principal | *El software recibe los datos *Si los datos están bien se procederá a realizar los cálculos *Al presionar el botón calcular, saldrá una venta con los resultados finales |

| | |
|--------------------------|--|
| Flujo Alternativo | *Si los datos están mal introducidos el software no realizara el calculo |
| Post condiciones | Si los datos están bien ingresado al software, procederá a realizar los cálculos |

Elaboración Propia

Tabla 4.7 Descripción del Caso de Uso- Visualizar el Procedimiento

| | |
|------------------------------|---|
| Nombre de Caso de Uso | Visualizar el Procedimiento |
| Objetivo | Recibe los datos |
| Actor | Usuario |
| Precondiciones | Debe existir un usuario que cargue los datos al software |
| Flujo Principal | *El software recibe los datos *Si los datos están bien se procederá a realizar los cálculos y podrá visualizar el procedimiento *Al presionar el botón calcular, se podrá visualizar el procedimiento en la pestaña de procedimientos |
| Flujo Alternativo | *Si los datos están mal introducidos el software no realizara el procedimiento del calculo |
| Post condiciones | Si los datos están bien ingresado al software, procederá a realizar los cálculos y se podrá visualizar el procedimiento |

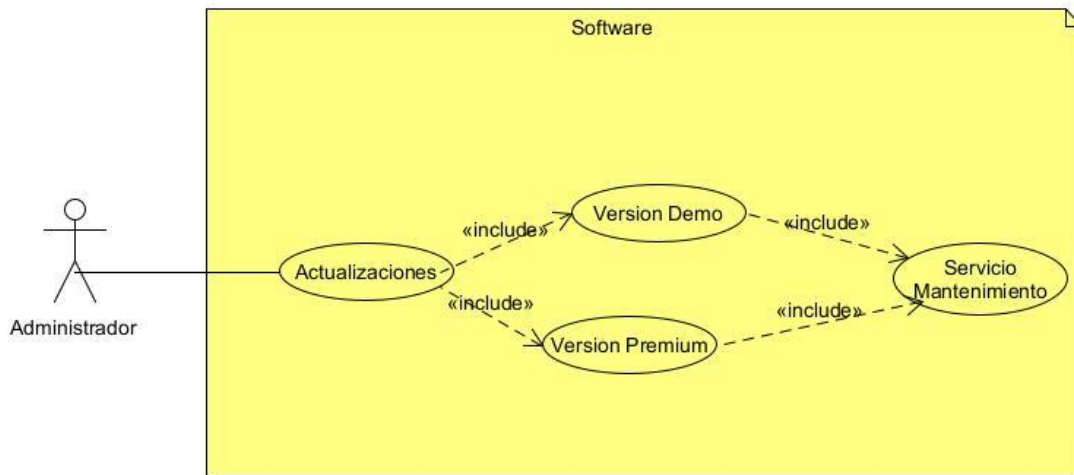
Elaboración Propia

Tabla 4.8 Descripción del Caso de Uso- Guardar

| | |
|------------------------------|---|
| Nombre de Caso de Uso | Guardar |
| Objetivo | Guardar los cálculos |
| Actor | Usuario |
| Precondiciones | Debe existir un usuario que guarde los datos |
| Flujo Principal | *El software guarda el procedimiento *Si los datos están bien se procederá a realizar los cálculos, entonces los el usuario podrá guardar los datos *Al presionar el botón calcular, se podrá visualizar el procedimiento en la pestaña de procedimientos |
| Flujo Alternativo | *Si los datos están mal introducidos el software no podrá cálculos |
| Post condiciones | Si los datos están bien ingresado al software, procederá a realizar los cálculos y se podrá guardar el procedimiento |

Elaboración Propia

Figura 4.2 Caso de Uso – Administrador



Nota. Diagrama realizado en el programa Umlet. Elaboración Propia

4.1.2.1.2 Descripción del caso de Uso-Administrador

Tabla 4.9 Descripción del Caso de Uso- Actualizaciones

| Nombre de Caso de Uso | Actualizaciones |
|--------------------------|--|
| Objetivo | Realiza las actualizaciones |
| Actor | Administrador |
| Precondiciones | Debe existir un Usuario |
| Flujo Principal | *El software se tiene que actualizar *Si el software se actualiza tendrá más funciones para el calcular |
| Flujo Alternativo | *Si el cálculo tiene errores frecuentemente, el software debe actualizarse muy seguido |
| Post condiciones | Si no existe error en los cálculos el software no se actualizar frecuentemente |

Elaboración Propia

Tabla 4.10 Descripción del Caso de Uso- Versión Demo

| Nombre de Caso de Uso | Versión Demo |
|------------------------|---|
| Objetivo | Es una versión de prueba del software |
| Actor | Administrador |
| Precondiciones | Debe existir un Administrador |
| Flujo Principal | *El software solo tiene algunas funciones |

| | |
|--------------------------|---|
| | *Si usuario quiere utilizar más funciones debe comunicarse con el administrador |
| Flujo Alternativo | *Verificar el calculo |
| Post condiciones | La versión Demo está limitada funciones, solo es una versión de prueba del software |

Elaboración Propia

Tabla 4.11 Descripción del Caso de Uso- Versión Premium

| Nombre de Caso de Uso | Versión Premium |
|------------------------------|---|
| Objetivo | Utilizar todas las funciones |
| Actor | Administrador |
| Precondiciones | Debe existir un Usuario |
| Flujo Principal | *El software utiliza todas sus funciones *El usuario podrá utilizar todas sus funciones del software |
| Flujo Alternativo | *Verificar los resultados |
| Post condiciones | En la versión premium se puede utilizar todas las funciones |

Elaboración Propia

Tabla 4.12 Descripción del Caso de Uso- Servicio de Mantenimiento

| Nombre de Caso de Uso | Servicio de Mantenimiento |
|------------------------------|---|
| Objetivo | Servicio de mantenimiento a todas versiones de software |
| Actor | Administrador |
| Precondiciones | Debe existir un Usuario |
| Flujo Principal | *El servicio se brinda a todas las versiones *El usuario podrá utilizar el servicio de mantenimiento personalizado |
| Flujo Alternativo | *Verificar el software que funciones correctamente |
| Post condiciones | Si las versiones de software funcionan correctamente y el servicio de mantenimiento no se dará frecuentemente. |

Elaboración Propia

4.1.2.2 Definición de riesgos del proyecto

Algunos posibles riesgos podrían incluir:

- ❖ **Complejidad de la Implementación:** La programación de algoritmos y modelos matemáticos para el análisis de cerchas puede ser compleja y propensa a errores, lo que podría afectar la precisión de los resultados.
- ❖ **Errores en los Cálculos:** Si los algoritmos de cálculo no se implementan correctamente, los resultados podrían ser inexactos, lo que podría llevar a decisiones erróneas en el diseño de estructuras.
- ❖ **Interfaz Ininteligible:** Una interfaz de usuario poco clara o difícil de usar podría afectar la usabilidad del software, lo que a su vez podría llevar a errores de entrada por parte del usuario.
- ❖ **Incompatibilidad de Plataformas:** Problemas de compatibilidad con diferentes sistemas operativos o versiones de software podrían limitar la accesibilidad y la adopción del programa.
- ❖ **Dificultad para Verificar Resultados:** Verificar la precisión de los resultados del software puede ser complicado, lo que podría llevar a la confianza limitada en su utilidad.
- ❖ **Rendimiento Insatisfactorio:** El software podría no cumplir con las expectativas de rendimiento en términos de velocidad de cálculo y capacidad para manejar cerchas de diferentes tamaños.

Tabla 4.13 Escala de Probabilidades, Impacto y Severidad

| Probabilidad | | Impacto | | Severidad | |
|--------------|-------------|-------------|-------------|-------------|-------------|
| Rango | Descripción | Rango | Descripción | Rango | Descripción |
| 0.00 a 0.30 | Baja | 0.00 a 0.30 | Baja | 0.00 a 0.30 | Baja |
| 0.30 a 0.60 | Media | 0.30 a 0.60 | Media | 0.30 a 0.60 | Media |
| 0.60 a 1.00 | Alta | 0.60 a 1.00 | Alta | 0.60 a 1.00 | Alta |

Elaboración propia

Es importante identificar y abordar estos riesgos de manera proactiva mediante estrategias de mitigación y planes de contingencia adecuados. Un enfoque planificado en la gestión de riesgos puede ayudar a minimizar los posibles impactos negativos en el desarrollo y uso del software de análisis de cerchas.

Tabla 4.14 Identificación de Riesgo

| Riesgo | Probabilidad | Impacto | Severidad |
|--------------------------------------|---------------------|----------------|------------------|
| Complejidad de la Implementación | 0.50 | 0.30 | 0.30 |
| Errores en los Cálculos | 0.30 | 0.70 | 0.20 |
| Interfaz Ininteligible | 0.30 | 0.20 | 0.30 |
| Incompatibilidad de Plataformas | 0.40 | 0.60 | 0.20 |
| Dificultad para Verificar Resultados | 0.20 | 0.10 | 0.10 |
| Rendimiento Insatisfactorio | 0.50 | 0.70 | 0.50 |

Elaboración propia

4.1.2.2.1 Plan de respuesta ante riesgos

Riesgo 1: Complejidad de la Implementación

- ❖ **Respuesta:** Implementar revisiones de código y pruebas exhaustivas para asegurar que los algoritmos estén correctamente implementados.
- ❖ **Plan de Contingencia:** En caso de identificar errores, contar con una asistencia de equipo de desarrollo capacitado para abordar rápidamente y corregir las fallas.

Riesgo 2: Errores en los Cálculos

- ❖ **Respuesta:** Realizar pruebas de validación comparando resultados con soluciones analíticas.
- ❖ **Plan de Contingencia:** Si se encuentran discrepancias, se debe verificar y ajustar los algoritmos de cálculo.

Riesgo 3: Interfaz Ininteligible

- ❖ **Respuesta:** Invertir en un diseño de interfaz intuitiva y realizar pruebas de usabilidad con usuarios reales.
- ❖ **Plan de Contingencia:** Si los usuarios reportan dificultades, se deben realizar ajustes en la interfaz y proporcionar documentación clara.

Riesgo 4: Incompatibilidad de Plataformas

- ❖ **Respuesta:** Probar el software en diferentes sistemas operativos y versiones antes del lanzamiento.
- ❖ **Plan de Contingencia:** Si se encuentran problemas de compatibilidad, trabajar en parches o actualizaciones para abordarlos.

Riesgo 5: Dificultad para Verificar Resultados

- ❖ **Respuesta:** Realizar validaciones cruzadas y comparaciones con soluciones conocidas para verificar los resultados.
- ❖ **Plan de Contingencia:** Si persisten dudas, buscar asesoramiento de expertos en análisis estructural.

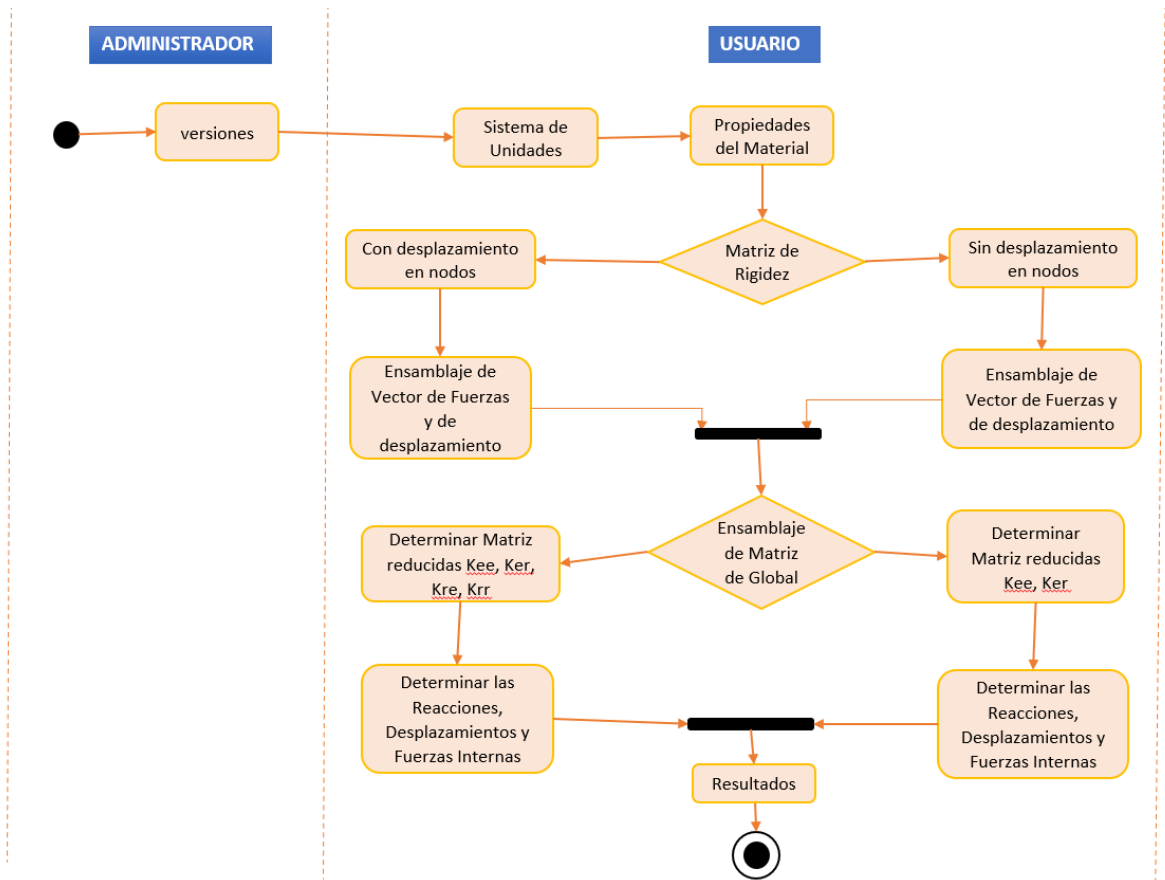
Riesgo 6: Rendimiento Insatisfactorio

- ❖ **Respuesta:** Optimizar los algoritmos y realizar pruebas de carga para asegurar el rendimiento adecuado.
- ❖ **Plan de Contingencia:** Si el rendimiento es insatisfactorio, considerar la posibilidad de dividir el análisis en etapas o escalabilidad.

4.1.2.3 Diagrama de actividades

El diagrama de actividades en el diseño de un software es una herramienta gráfica que se utiliza para modelar y visualizar el flujo de actividades y acciones dentro de un proceso o función en el sistema. Este diagrama se basa en la notación de la UML (Lenguaje de Modelado Unificado) y es especialmente útil para representar procesos empresariales, flujos de trabajo y comportamientos en un nivel más alto de abstracción.

Figura 4.3 Diagrama de Actividades

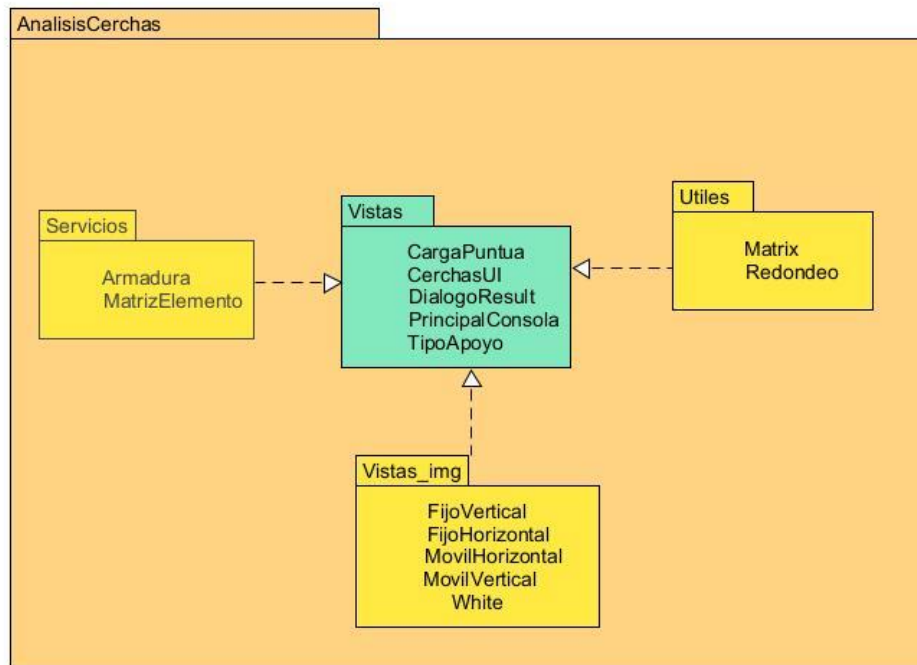


Nota. Diagrama realizado en el programa Visio. Elaboración Propia

4.1.2.4 Diagrama de paquetes

El diagrama de paquetes sirve para organizar los elementos de un software en un grupo, es decir mostrar que ciertas clases o componentes son parte de un subsistema en particular, como lo muestra la Figura

Figura 4.4 Diagrama de Paquetes



Nota. Diagrama realizado en el programa UMLET. Elaboración Propia

Paquete Servicios: Paquete que tiene la funcionalidad de calcular de calcular la matriz de rigidez y la matriz global de toda la estructura, va instanciado al paquete vistas.

Paquete Útiles: Paquete que tiene las clases para calcular la matriz inversa y controlará los redondeos de los cálculos.

Paquete Vista img: Paquete donde se carga los logos para instanciarlos al paquete principal.

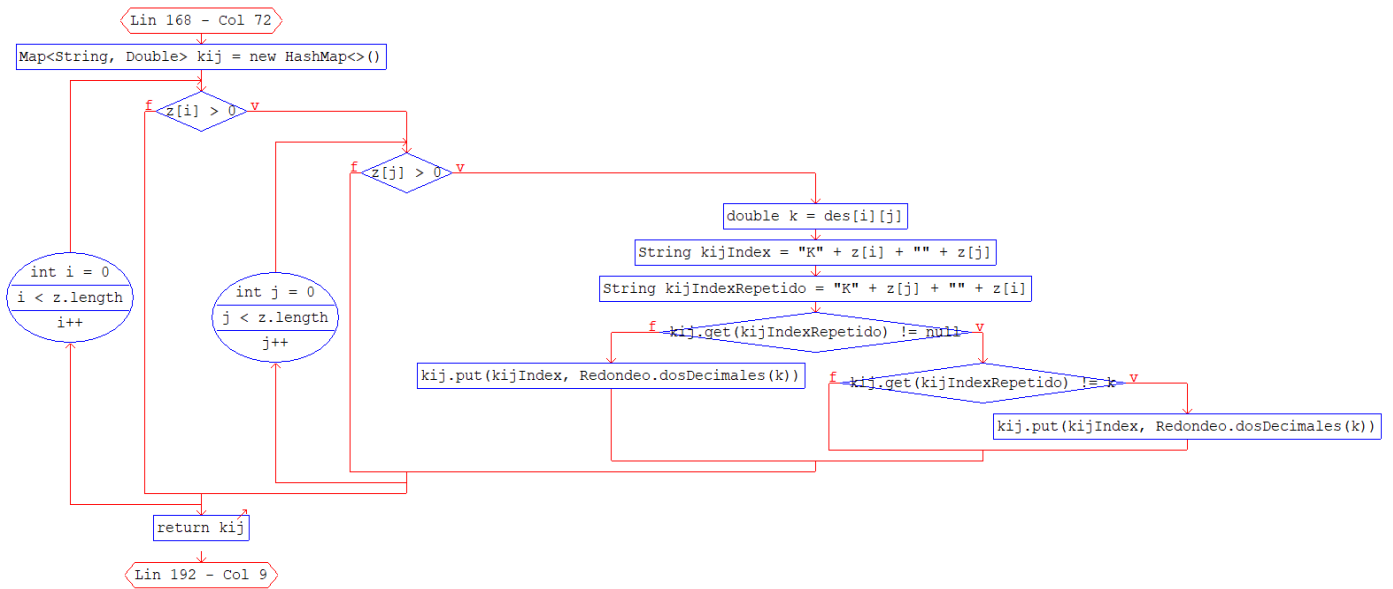
Paquete Vistas: Este es el paquete principal de la programación donde están todas las clases y donde heredan todos los demás paquetes.

4.1.2.5 Diagramas de Flujos

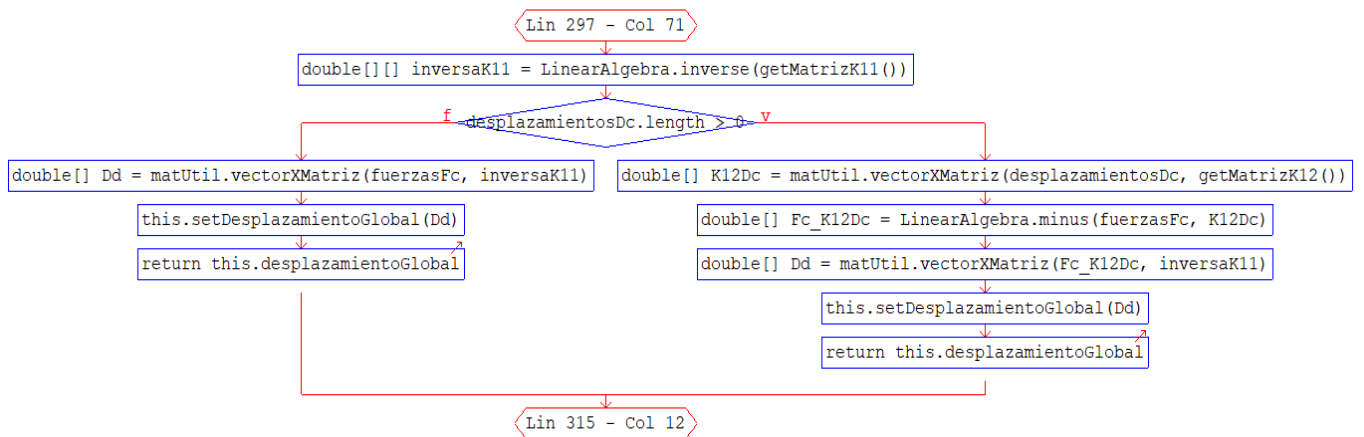
4.1.2.5.1 Diagrama de flujo para construcción de Matriz Rigidez de cada barra



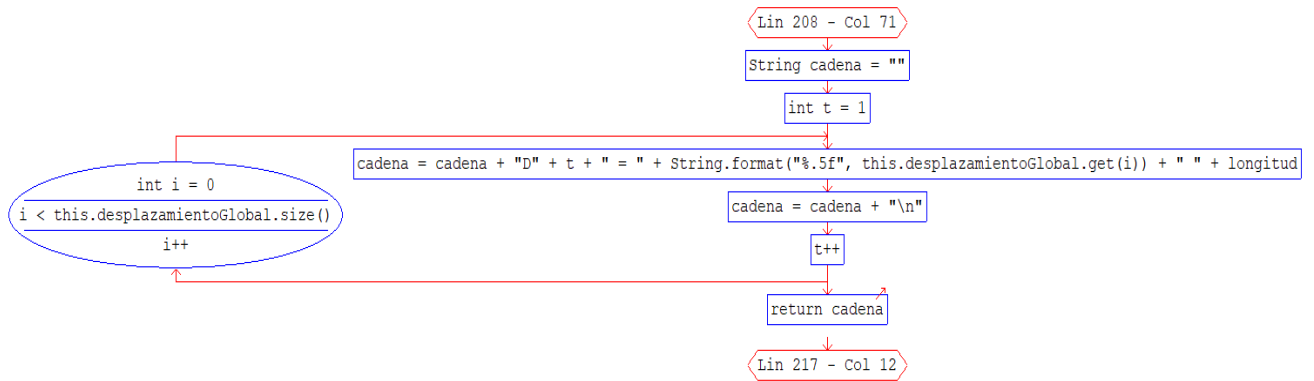
4.1.2.5.2 Diagrama de flujo del código para el ensamblado de matriz global



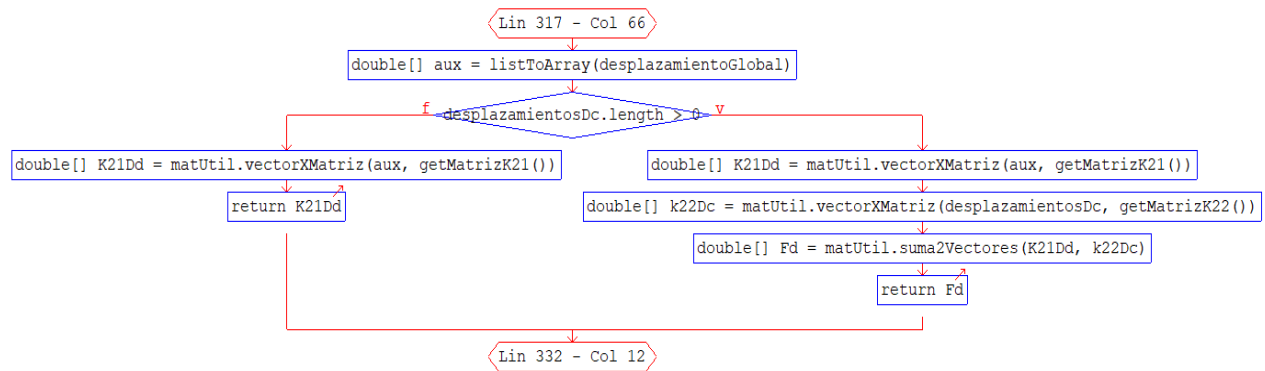
4.1.2.5.3 Diagrama de flujo del código del cálculo de desplazamientos de una cercha



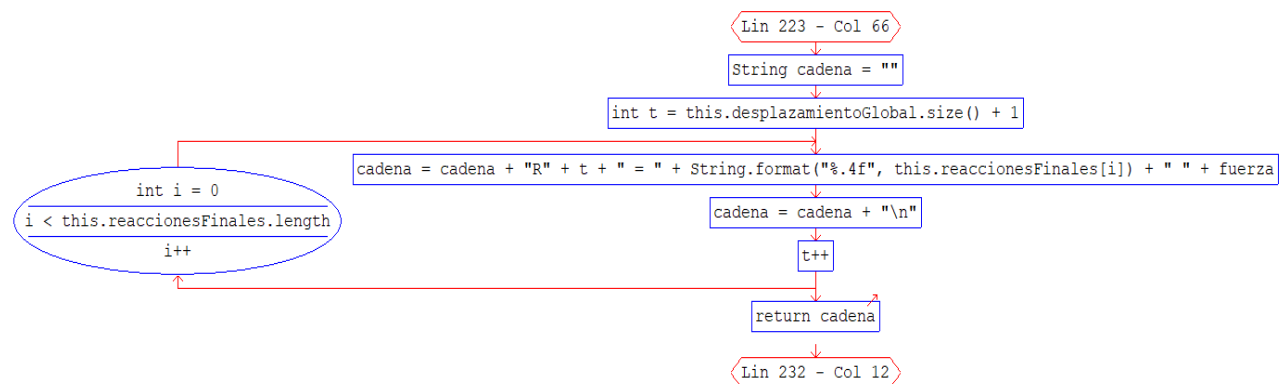
4.1.2.5.4 Diagrama de flujo para imprimir los desplazamientos



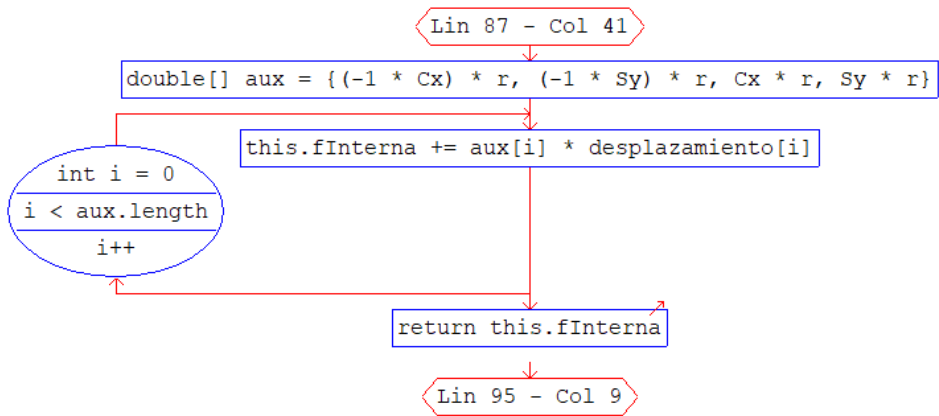
4.1.2.5.5 Diagrama de flujo del código para el cálculo de Reacciones



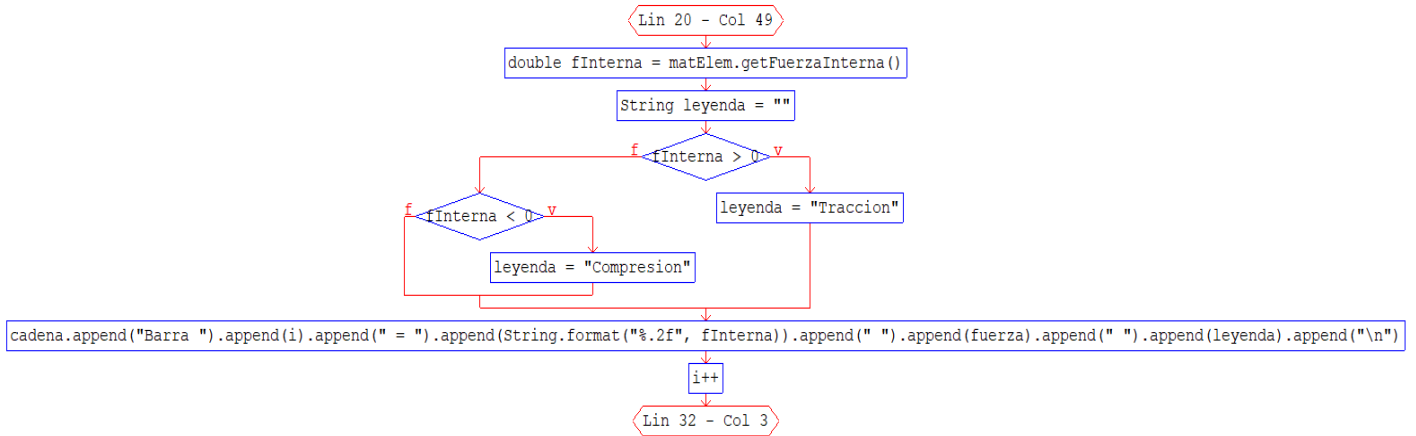
4.1.2.5.6 Diagrama de flujo para imprimir las Reacciones



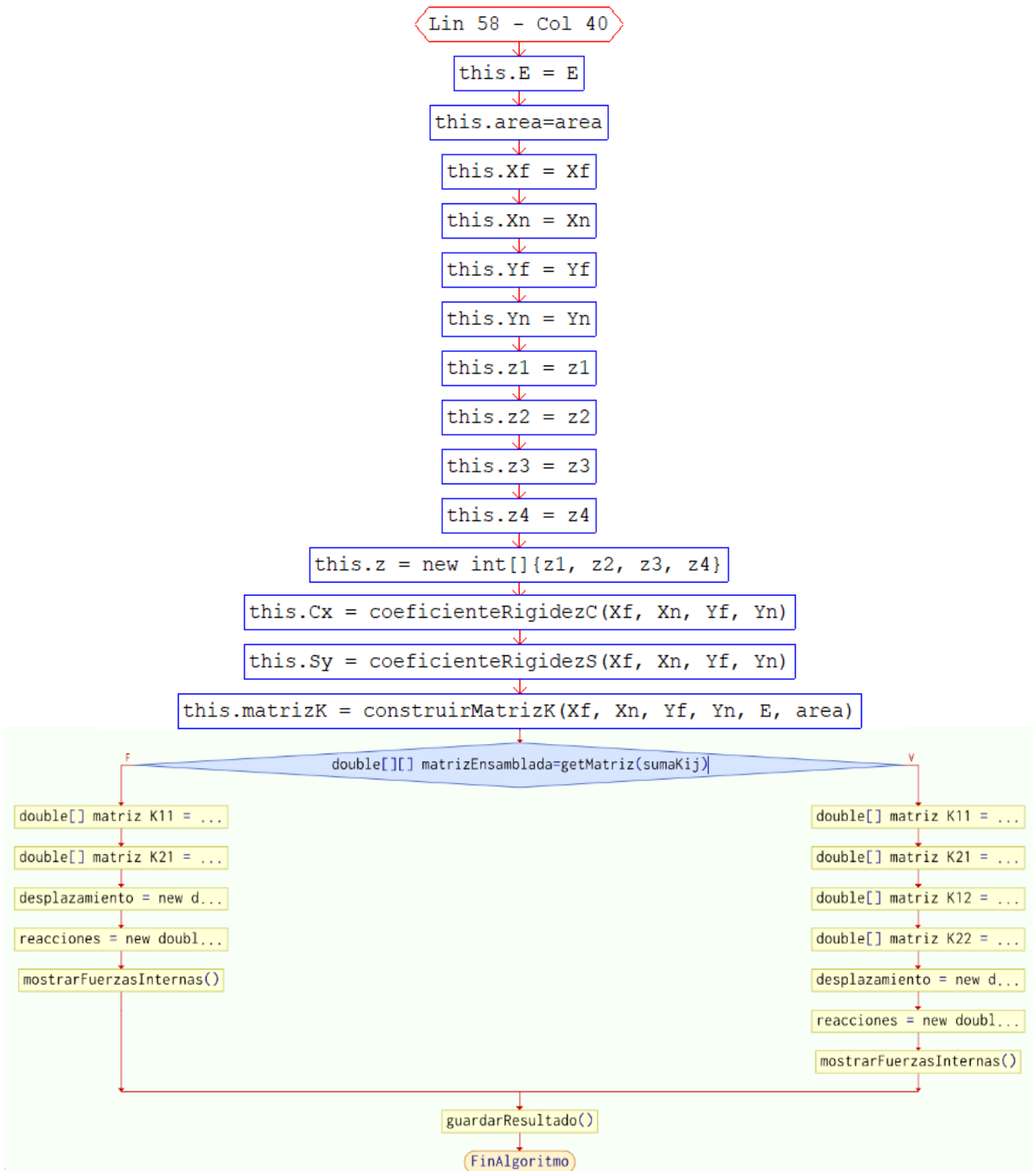
4.1.2.5.7 Diagrama de flujo del código para el cálculo de Fuerzas Internas



4.1.2.5.8 Diagrama de flujo para Imprimir Fuerzas Internas



4.1.2.5.9 Diagrama general para el cálculo de cerchas



4.1.3. FASE 3 Construcción

4.1.3.1 Características del Software

El programa ha sido desarrollado utilizando el lenguaje de programación Java, lo que le otorga la ventaja de ser multiplataforma y compatible con diversos sistemas operativos. Una de las características sobresalientes del programa es su capacidad para abordar una amplia variedad de factores en el análisis, incluyendo cargas, apoyos y geometría. Esto proporciona resultados más exhaustivos y permite un enfoque integral en el estudio de las cerchas.

No obstante, es relevante tener en cuenta ciertas limitaciones, ya que el programa no admite apoyos inclinados ni cargas inclinadas. Aunque se destaca por su versatilidad en analizar cerchas de distintas configuraciones, estas limitaciones pueden requerir aproximaciones adicionales para resolver problemas con condiciones específicas.

4.1.3.2 Pantallas

Pantalla 1 Ventana Principal

ANÁLISIS DE CERCHAS

Formulario Gráfico Procedimientos

PARAMETROS INICIALES:

--Unidades-- Nro. de barras: Area: m² E: KN/m²

DIMENSIONES DE LA BARRA 0

Nodo Inicial X: m Y: m

Nodo Final Xj: m Yj: m

GRADOS DE LIBERTAD

ix iy jx jy **DIBUJAR**

TIPO DE APOYO

X= m Y= m Tipo de Apoyo

TIPO DE CARGA

X= m Y= m Valor= KN Carga Puntual

FUERZAS

F1

F2

F3

F4

DESPLAZAMIENTOS

Δ1

Δ2

Δ3

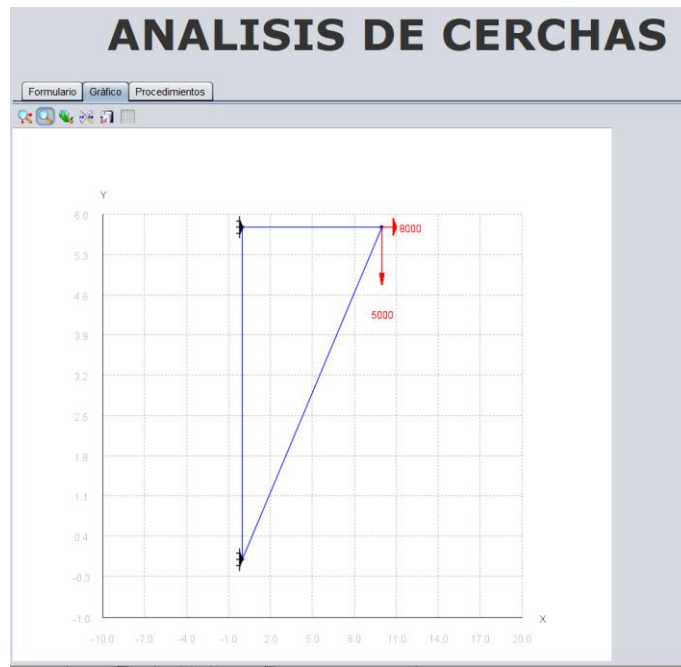
Δ4

CALCULAR

NUEVO

Nota. Ventana principal del programa Análisis de Cerchas

Pantalla 2 Pestaña del gráfico



Nota. Pestaña donde se podrá ver las graficas

Pantalla 3 Pestaña del Procedimiento

PROCEDIMIENTO DE CALCULO:

PROCEDIMIENTO DE CALCULO:

MATRIZ K1
Longitud=2.83 m

| | | | |
|----------|----------|----------|----------|
| 0,18000 | 0,18000 | -0,18000 | -0,18000 |
| 0,18000 | 0,18000 | -0,18000 | -0,18000 |
| -0,18000 | -0,18000 | 0,18000 | 0,18000 |
| -0,18000 | -0,18000 | 0,18000 | 0,18000 |

K11 = 0,18
K22 = 0,18
K33 = 0,18
K44 = 0,18
K13 = -0,18
K24 = -0,18
K12 = 0,18
K23 = -0,18
K34 = 0,18
K14 = -0,18

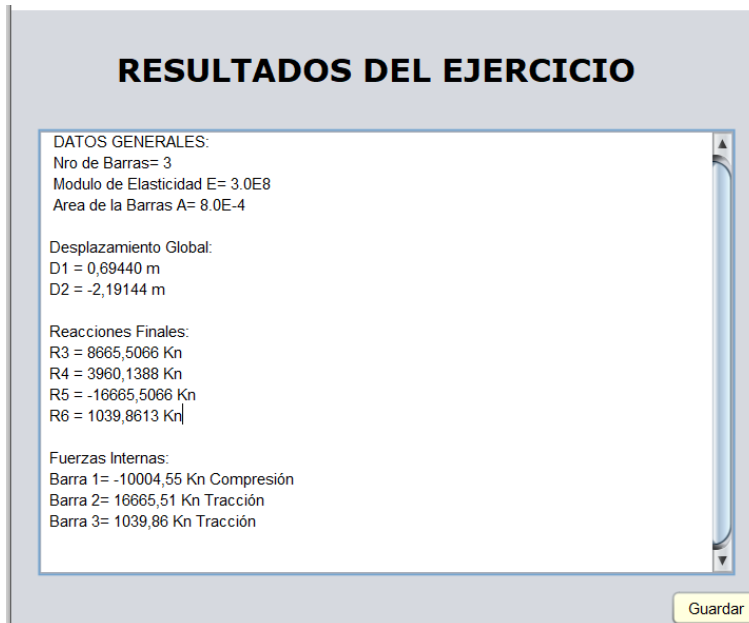
Resultados sumas Kij:

| |
|-------------|
| K31 = -0,18 |
| K42 = -0,18 |
| K41 = -0,18 |
| K11 = 0,18 |
| K22 = 0,18 |
| K33 = 0,18 |
| K44 = 0,18 |
| K21 = 0,18 |
| K32 = -0,18 |
| K43 = 0,18 |
| K13 = -0,18 |
| K24 = -0,18 |
| K12 = 0,18 |

Guardar

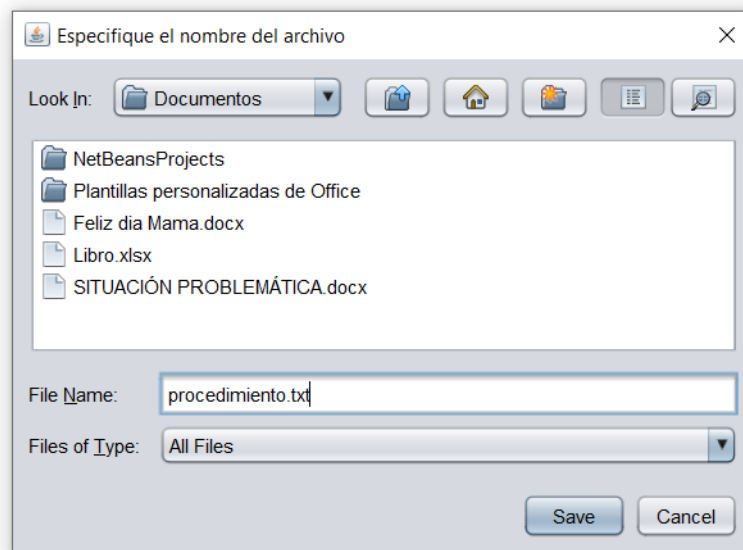
Nota. Pestaña donde se podrá ver el procedimiento de cálculo.

Pantalla 4 Ventana de resultado



Nota: Ventana donde se verá los resultados del software

Pantalla 5 Ventana para guardar el procedimiento



Nota: Ventana donde tendrá la opción de guardar el procedimiento

4.1.3.3 Inicio de la Programación

La primera ventana que se visualiza al iniciar el programa es la ventana principal (Pantalla 1), en la cual se presenta el programa y se coloca el número de barras que tendrá las cerchas, el área, el módulo de elasticidad, las coordenadas para barra, como también las coordenadas para las cargas y los apoyos. Otros datos que se tiene que cargar son los grados de libertad de cada barra.

El código de programación de la ventana principal se brinda a continuación.

```
public class CerchasUI extends javax.swing.JFrame {  
  
    int matrixCont = 1;  
    Armadura armadura = new Armadura();  
    Matrix matUtil = new Matrix();  
    String usrPath;  
    Hashtable<Object, ImageIcon> itemElements;  
    int width = 20;  
    int height = 20;  
  
    /**  
     * Creates new form CerchasUI  
     */  
    public CerchasUI() {  
        initComponents();  
        usrPath = System.getProperty( key:"user.dir") + "\\src\\Vistas\\img\\";  
        itemElements = new Hashtable<>();  
        initCombo();  
        // plot.setBounds(0, 100, 430, 450);  
        plot.setBounds( x:0, y:0, width:700, height:700);  
        plot.setAutoscrolls( autoscrolls:true);  
        plot.setAutoBounds();  
        this.panelDibujo.add( comp:plot);  
    }  
}
```

```
private javax.swing.JButton btnCalcular;  
private javax.swing.JButton btnDibujar;  
private javax.swing.JButton btnGuardar;  
private javax.swing.JComboBox<String> cboApoyo;  
private javax.swing.JComboBox<String> cboCarga;  
private javax.swing.JPanel formPanel2;
```

```
private javax.swing.JPanel panelDibujo;
private javax.swing.JTable tablaFuerzas;
private javax.swing.JTextField tbxBase;
private javax.swing.JTextField tbxE;
private javax.swing.JTextField tbxNroMatrixs;
private javax.swing.JTextField tbxXi;
private javax.swing.JTextField tbxXj;
private javax.swing.JTextField tbxYi;
private javax.swing.JTextField tbxYj;
private javax.swing.JTextField tbxZ1;
private javax.swing.JTextField tbxZ2;
private javax.swing.JTextField tbxZ3;
private javax.swing.JTextField tbxZ4;
private javax.swing.JTextArea textAResultados;
private javax.swing.JTextField txtCarga;
private javax.swing.JTextField txtX;
private javax.swing.JTextField txtXc;
private javax.swing.JTextField txtY;
private javax.swing.JTextField txtYc;
```

4.1.3.4 Ingreso de Datos

Una vez que se ha definido el número de barras de la cercha, el siguiente paso es cargar los datos de cada barra en la interfaz gráfica para representarlas visualmente.

Para ello, se requerirá que el usuario ingrese la información relevante de cada barra, como su longitud, tipo de material, sección transversal, conexiones, y condiciones de apoyo, entre otros datos. Estos datos se utilizarán para calcular las propiedades y características de cada barra, como la rigidez, las fuerzas internas y las deformaciones.

Una vez que se hayan cargado todos los datos de las barras, el programa procederá a dibujar cada una de ellas en la interfaz gráfica. Esto permitirá al usuario visualizar la configuración de la cercha y cómo están conectadas todas las barras. La representación visual de las barras y sus propiedades ayudará al usuario a comprender mejor la estructura de la cercha y a verificar que los datos ingresados sean correctos.

4.1.3.5 Sistema de unidades

Al iniciar el programa se debe elegir el sistema de unidades para ingresar los datos, es una buena práctica permitir que el usuario elija el sistema de unidades al iniciar el programa. Al hacerlo, el usuario podrá trabajar con las unidades con las que se sienta más cómodo o que sean más apropiadas para el proyecto en particular. Esto mejora la experiencia del usuario y facilita el ingreso de datos coherentes con el sistema de unidades deseado.

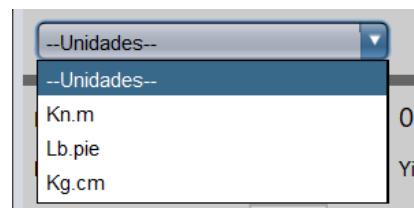
Las opciones para el sistema de unidades que se pueden ofrecer al usuario son:

Sistema Internacional: Este sistema utiliza el metro (m) para longitudes, el kilogramo (kg) para masas y el segundo (s) para tiempos, entre otras unidades básicas. Es ampliamente utilizado en todo el mundo y en el ámbito de la ingeniería civil.

Sistema Ingles: En este sistema, se utilizan unidades como pies (ft) para longitudes y libras (lb) para fuerzas, entre otras. Es más comúnmente utilizado en algunos países como Estados Unidos.

Sistema Métrico Decimal: Es similar al sistema SI, pero con algunas diferencias en la nomenclatura de unidades. Por ejemplo, se pueden usar centímetro (cm) para longitudes y kilo newtons (kN) para fuerzas.

Al ofrecer la opción de elegir el sistema de unidades al inicio del programa, se brinda mayor flexibilidad y adaptabilidad, lo que es especialmente importante cuando se trabaja con usuarios de diferentes regiones o proyectos que requieren diferentes unidades.



El código para programar el tipo de sistema de unidades que se va utilizar es el siguiente.

```

private void cbxSistemaUnidadesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String su = cbxSistemaUnidades.getSelectedItem().toString();

    switch (su) {
        case "Kn.m":
            System.out.println(x:su);
            setLabelText( longitud:"m", fuerza:"Kn");
            break;
        case "Kg.cm":
            System.out.println(x:su);
            setLabelText( longitud:"cm", fuerza:"Kg");
            break;
        case "Lb.pie":
            System.out.println(x:su);
            setLabelText( longitud:"pie", fuerza:"Lb");
            break;
        default:
            System.out.println(x:su);
            break;
    }
}

```

```

private void setLabelText(String longitud, String fuerza) {
    lblAreaU.setText(longitud+"2");
    lblEU.setText(fuerza + "/" + longitud+"2");
    lblBarraXiU.setText( text:longitud);
    lblBarraXjU.setText( text:longitud);
    lblBarraYiU.setText( text:longitud);
    lblBarraYjU.setText( text:longitud);

    lblApoyoXU.setText( text:longitud);
    lblApoyoYU.setText( text:longitud);

    lblCargaXU.setText( text:longitud);
    lblCargaYU.setText( text:longitud);
    lblCargaValorU.setText( text:fuerza);
}

```

4.1.3.6 Propiedades del material

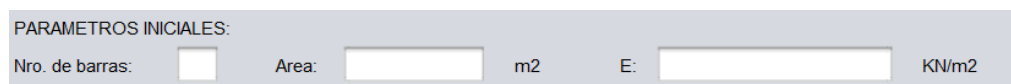
Los primeros datos que se cargarán en el programa para el análisis de las cerchas son:

Número de barras: El usuario deberá ingresar la cantidad total de barras que componen la cercha. Este dato es esencial para definir la configuración y la conectividad de la estructura.

Área: Se debe proporcionar el área de la sección transversal de las barras en las unidades que se especifique en el programa. Esta área es relevante para calcular la rigidez de cada barra, que a su vez influye en las fuerzas internas y las deformaciones resultantes.

Módulo de elasticidad: El módulo de elasticidad es una propiedad del material utilizado en las barras de la cercha. representa la relación entre el esfuerzo aplicado a la barra y la deformación resultante. Este dato es fundamental para realizar el análisis de la estructura y determinar su comportamiento ante las cargas.

Es importante que el usuario proporcione estos datos con precisión, ya que afectarán directamente los resultados obtenidos y garantizarán un análisis estructural confiable y preciso.



Código que restringe el número de barras

```
this.lblCount.setText(text: Integer.toString(i:matrixCont));
int max = Integer.parseInt(s:this.tbxNroMatrixs.getText());
if (matrixCont <= max) {
    MatrizElemento matrizElemento = armadura.calcular(E:e, area, xf:xf, Xn:xn, Yf:yf, Yn:yn, z1, z2, z3, z4);
    Map<String, Double> resultMap = matrizElemento.getKij();

    guardarResultado("MATRIZ K" + matrixCont + "\n");
    guardarResultado(matrixToString:matUtil.matrixToString(m:matrizElemento.getMatrizK()));
    guardarResultado(matrixToString: "\n");
    guardarResultado(matrixToString:armadura.mapToString(kij:resultMap));
    guardarResultado(matrixToString: "\n");
    guardarResultado(matrixToString: "\n");
}
```

```

armadura.resultdosKij.add( e:resultMap);
if (matrixCont == max) {
    this.btnDibujar.setEnabled( b:false);
    this.btnCalcular.setEnabled( b:true);
}
}
matrixCont++;

```

Código de entrada del Área y el Módulo de Elasticidad

```

double e = Double.parseDouble( s:this.tbxE.getText().trim());
double area = Double.parseDouble( s:this.tbxBase.getText().trim());

```

4.1.3.7 Coordenadas de la barra

En el proceso de ingreso de coordenadas X e Y, se divide en dos partes. La primera parte corresponde a las coordenadas del nodo inicial de la barra, mientras que la segunda parte corresponde a las coordenadas del nodo final de la barra. Estas coordenadas se ingresarán según las unidades especificadas en el programa.

Por ejemplo, si el programa está configurado para trabajar con unidades en metros, se ingresarán las coordenadas del nodo inicial y final en metros. Si el programa está configurado para trabajar con unidades en centímetros, entonces se ingresarán las coordenadas en centímetros.

Al dividir las coordenadas en dos partes, se permite al usuario ingresar la información de manera más organizada y comprensible. De esta manera, el programa podrá identificar de forma clara la posición de cada nodo y calcular correctamente las longitudes y orientaciones de las barras de la cercha.

| DIMENSIONES DE LA BARRA | | 0 | | |
|-------------------------|--------------------------|---|--------------------------|---|
| Nodo Inicial | Xi: <input type="text"/> | m | Yi: <input type="text"/> | m |
| Nodo Final | Xj: <input type="text"/> | m | Yj: <input type="text"/> | m |

```

double xf = Double.parseDouble( s:this.tbXj.getText().trim());
double xn = Double.parseDouble( s:this.tbXi.getText().trim());
double yf = Double.parseDouble( s:this.tbYj.getText().trim());
double yn = Double.parseDouble( s:this.tbYi.getText().trim());

```

4.1.3.8 Grados de Libertad de cada barra

Exactamente, los grados de libertad en el análisis de estructuras son los diferentes sentidos de desplazamiento que se permiten en cada barra o elemento de la estructura. Estos grados de libertad se definen para cada nodo de la estructura y se utilizan para describir la capacidad de movimiento de la barra en sus diferentes direcciones.

Para realizar el análisis estructural, es necesario discretizar toda la estructura, es decir, dividirla en elementos más pequeños, como barras o nodos, para simplificar los cálculos y obtener resultados más precisos. En este proceso de discretización, se definen los ejes locales y globales de la estructura. Los ejes locales están asociados a cada barra individual y se utilizan para describir el movimiento de la barra en su propio sistema de coordenadas. Por otro lado, los ejes globales representan el sistema de coordenadas global de la estructura en su conjunto.

Cada barra tendrá ciertos grados de libertad, que pueden ser desplazamientos en los ejes X, Y y Z y posiblemente rotaciones alrededor de los mismos ejes. Por ejemplo, un grado de libertad típico para una barra en 2D sería el desplazamiento en el eje X y el desplazamiento en el eje Y.

El calculista o ingeniero estructural debe definir cuidadosamente estos grados de libertad para cada barra, teniendo en cuenta las restricciones y condiciones de apoyo en los nodos y la interacción entre las diferentes barras de la estructura.

GRADOS DE LIBERTAD

| | | | |
|--|--|--|--|
| ix <input style="width: 30px;" type="text"/> | iy <input style="width: 30px;" type="text"/> | jx <input style="width: 30px;" type="text"/> | jy <input style="width: 30px;" type="text"/> |
|--|--|--|--|

```

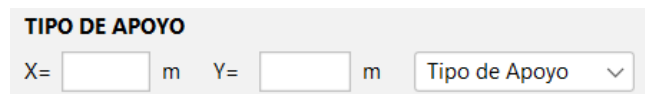
int z1 = Integer.parseInt( s:this.tbz1.getText().trim());
int z2 = Integer.parseInt( s:this.tbz2.getText().trim());
int z3 = Integer.parseInt( s:this.tbz3.getText().trim());
int z4 = Integer.parseInt( s:this.tbz4.getText().trim());

```

4.1.3.9 Tipo de Apoyos

Correcto, para introducir el tipo de apoyo en el análisis de la cercha, se sigue un proceso que involucra ingresar las coordenadas del nodo donde se ubicará el apoyo y luego seleccionar el tipo de apoyo mediante un combo box.

El procedimiento sería el siguiente. Ingresar las coordenadas, El usuario deberá proporcionar las coordenadas X e Y del nodo donde se ubicará el apoyo. Estas coordenadas representan la posición exacta del nodo en el sistema de coordenadas global de la estructura. Selección del tipo de apoyo, Una vez ingresadas las coordenadas del nodo, el usuario utilizará un combo box o menú desplegable para seleccionar el tipo de apoyo que se necesita en ese nodo. Los tipos de apoyo pueden incluir apoyo fijo, apoyo móvil, apoyo con desplazamiento restringido en ciertas direcciones.



Código para introducir el apoyo

```
private void cboApoyoItemStateChanged(java.awt.event.ItemEvent evt) { // GEN-FIRST
    int apoyo = cboApoyo.getSelectedIndex();
    double primerX = Double.parseDouble(s.txtX.getText().toString());
    double primerY = Double.parseDouble(s.txtY.getText().toString());
    double[] arrayz = {primerX, primerY};
    String a = txtCarga.getText();

    switch (apoyo) {
        case 1:

            for (double i = -0.12; i <= 0.12; i += 0.001) {
                double[] inif = {primerX - i, primerY - 0.12};
                double[] finf = {primerX, primerY};
                plot.addLinePlot(name:"Linea", color:Color.BLACK, x:inif, y:finf);
            }

            double[] ini2 = {primerX - 0.20, primerY - 0.12};
            double[] fin2 = {primerX + 0.20, primerY - 0.12};
            plot.addLinePlot(name:"Linea", color:Color.BLACK, x:ini2, y:fin2);

            for (double i = 0.10; i < 0.40; i += 0.10) {
                double[] ini3 = {primerX - 0.20 + i, primerY - 0.12};
                double[] fin3 = {primerX - 0.20 + i, primerY - 0.20};
                plot.addLinePlot(name:"Linea", color:Color.BLACK, x:ini3, y:fin3);
            }

            break;
    }
}
```

```

case 2:

    for (double i = -0.12; i <= 0.12; i += 0.001) {
        double[] inif = {primerX - 0.20, primerY - i};
        double[] finf = {primerX, primerY};
        plot.addLinePlot( name:"Linea", color:Color.BLACK, x:inif, y:finf);
    }

    double[] iniv2 = {primerX - 0.20, primerY + 0.20};
    double[] finv2 = {primerX - 0.20, primerY - 0.20};
    plot.addLinePlot( name:"Linea", color:Color.BLACK, x:iniv2, y:finv2);

    }
break;
case 3:

    for (double i = -0.12; i <= 0.12; i += 0.001) {
        double[] inif = {primerX - i, primerY - 0.12};
        double[] finf = {primerX, primerY};
        plot.addLinePlot( name:"Linea", color:Color.BLACK, x:inif, y:finf);
    }

    double[] iniMh2 = {primerX - 0.20, primerY - 0.12};
    double[] finMh2 = {primerX + 0.20, primerY - 0.12};
    plot.addLinePlot( name:"Linea", color:Color.BLACK, x:iniMh2, y:finMh2);

    double[] iniMh3 = {primerX - 0.20, primerY - 0.15};
    double[] finMh3 = {primerX + 0.20, primerY - 0.15};
    plot.addLinePlot( name:"Linea", color:Color.BLACK, x:iniMh3, y:finMh3);
break;
case 4:

    for (double i = -0.12; i <= 0.12; i += 0.001) {
        double[] inif = {primerX - 0.20, primerY - i};
        double[] finf = {primerX, primerY};
        plot.addLinePlot( name:"Linea", color:Color.BLACK, x:inif, y:finf);
    }

    double[] iniMv2 = {primerX - 0.20, primerY + 0.20};
    double[] finMv2 = {primerX - 0.20, primerY - 0.20};
    plot.addLinePlot( name:"Linea", color:Color.BLACK, x:iniMv2, y:finMv2);

    double[] iniMv3 = {primerX - 0.25, primerY + 0.20};
    double[] finMv3 = {primerX - 0.25, primerY - 0.20};
    plot.addLinePlot( name:"Linea", color:Color.BLACK, x:iniMv3, y:finMv3);

    break;
}
}
} // GEN-LAST:event_cboApoyoItemStateChanged

```

4.1.3.10 Tipo de Cargas

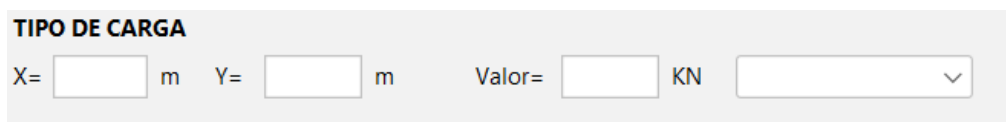
Exactamente, el proceso para introducir el tipo de carga en el análisis de la cercha consiste en los siguientes pasos:

Ingresar las coordenadas del nodo: El usuario debe proporcionar las coordenadas X e Y del nodo donde se aplicará la carga. Estas coordenadas representan la posición exacta del nodo en el sistema de coordenadas global de la estructura.

Selección del sentido de la carga: Una vez ingresadas las coordenadas del nodo, el usuario utilizará un combo box o menú desplegable para elegir el sentido en el cual se ubicará la carga en ese nodo. Los sentidos de carga pueden ser vertical y horizontal.

Definición del valor de la carga: Después de seleccionar el sentido de la carga, se ingresará el valor numérico de la magnitud de la carga que se aplicará en el nodo. Este valor se expresará en unidades adecuadas para la magnitud de la carga, como kilo newtons (kN).

El proceso de introducir la carga en cada nodo es fundamental para modelar adecuadamente las condiciones de carga de la cercha y obtener resultados precisos en el análisis. Las cargas externas aplicadas a la cercha afectarán su comportamiento y respuesta estructural, por lo que es esencial considerar cuidadosamente estas condiciones durante el análisis mediante el Método de Rigidez u otros métodos de análisis estructural.



TIPO DE CARGA

X= m Y= m Valor= KN

Código para la programación, para el tipo de carga.

```
private void cboCargaItemStateChanged(java.awt.event.ItemEvent evt) {  
    int carga = cboCarga.getSelectedIndex();  
    double primerX = Double.parseDouble(s:txtXc.getText().toString());  
    double primerY = Double.parseDouble(s:txtYc.getText().toString());  
}
```



```

switch (carga) {
  case 1: // VERTICAL POSITIVO
    // Para la linea
    double[] inivp = {primerX, primerY};
    double[] finvp = {primerX, primerY + 0.8};
    plot.addLinePlot( name: "Linea", color: Color.RED, x: inivp, y: finvp);

    for (double i = -0.15; i <= 0.15; i += 0.001) {
      double[] inif = {primerX - i, primerY + 0.8};
      double[] finf = {primerX, primerY + 1};
      plot.addLinePlot( name: "Linea", color: Color.RED, x: inif, y: finf);
    }

    // PARA LA etiqueta
    double[] arraya = {primerX, primerY + 1.5};
    String a = txtCarga.getText();
    plot.addLabel( text: a, c: Color.red, where: arraya);

    break;
  case 2: // Vertical negativo
    // Para la linea
    double[] inivn = {primerX, primerY};
    double[] finvn = {primerX, primerY - 0.8};
    plot.addLinePlot( name: "Linea", color: Color.RED, x: inivn, y: finvn);

    for (double i = -0.15; i <= 0.15; i += 0.001) {
      double[] inif = {primerX - i, primerY - 0.8};
      double[] finf = {primerX, primerY - 1};
      plot.addLinePlot( name: "Linea", color: Color.RED, x: inif, y: finf);
    }

    double[] arrab = {primerX, primerY - 1.5};
    String b = txtCarga.getText();
    plot.addLabel( text: b, c: Color.red, where: arrab);
    break;
  case 3: // HORIZONTAL POSITIVO
    // PAR ALA LINEA
    double[] arrays1 = {primerX + 0.8, primerY};
    double[] arrays2 = {primerX, primerY};
    plot.addLinePlot( name: "Linea", color: Color.RED, x: arrays1, y: arrays2);

    for (double i = -0.15; i <= 0.15; i += 0.001) {
      double[] inif = {primerX + 0.8, primerY - i};
      double[] finf = {primerX + 1, primerY};
      plot.addLinePlot( name: "Linea", color: Color.RED, x: inif, y: finf);
    }
}

```

```

        // para la etiqueta
        double[] arrayc = {primerX + 2, primerY};
        String c = txtCarga.getText();
        plot.addLabel( text:c, c:Color.red, where:arrayc);
        break;
    case 4: // HORIZONTAL NEGATIVO
        double[] arra1 = {primerX, primerY};
        double[] arra2 = {primerX - 0.8, primerY};
        plot.addLinePlot( name:"Linea", color:Color.RED, x:arra1, y:arra2);

        for (double i = -0.15; i <= 0.15; i += 0.001) {
            double[] inif = {primerX - 0.8, primerY - i};
            double[] finf = {primerX - 1, primerY};
            plot.addLinePlot( name:"Linea", color:Color.RED, x:inif, y:finf);
        }

        // para la etiqueta
        double[] arrayd = {primerX - 2, primerY};
        String d = txtCarga.getText();
        plot.addLabel( text:d, c:Color.red, where:arrayd);
        break;
    }
} // GEN-LAST:event_cboCargaItemStateChanged

```

4.1.3.11 Fuerzas Externas

Para ingresar las fuerzas externas que actúan en la estructura, se utilizará una tabla en la cual se introducirán los valores de las fuerzas, siguiendo el orden de los grados de libertad de cada nodo.

El orden de los grados de libertad se refiere a los diferentes sentidos de desplazamiento que se permiten en cada nodo de la estructura.

Es importante asegurarse de utilizar las unidades adecuadas que se eligió al principio para las fuerzas y mantener el orden de los grados de libertad al ingresar los valores en la tabla. De esta manera, el programa podrá considerar correctamente las fuerzas externas aplicadas en cada nodo durante el análisis de la estructura mediante el Método de Rigidez u otros métodos de análisis estructural.



El código de programación para las fuerzas es el siguiente:

```
private double[] getFuerzasTabla() {
    List<Double> aux = new ArrayList<>();
    int rows = tablaFuerzas.getRowCount();
    for (int i = 0; i < rows; i++) {
        try {
            String f = tablaFuerzas.getModel().getValueAt( rowIndex:i, columnIndex:0).toString();
            if (!f.isEmpty()) {
                double fe = Double.parseDouble( s:f);
                aux.add( e:fe);
            }
        } catch (NullPointerException e) {
            break;
        }
    }
    double[] fuerzas = new double[aux.size()];

    for (int i = 0; i < fuerzas.length; i++) {
        fuerzas[i] = aux.get( index:i);
    }
    return fuerzas;
}
```

4.1.3.12 Desplazamiento Impuestos

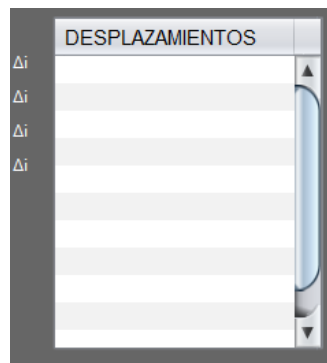
Al ingresar los desplazamientos en el análisis de la cercha, se debe tener en cuenta lo siguiente:

Grados de libertad: Los desplazamientos deben ingresarse siguiendo el orden de los grados de libertad de cada nodo. Por ejemplo, si se tiene un nodo con los dos grados de libertad (desplazamientos en X e Y), entonces se deberán ingresar los desplazamientos en ese orden.

Unidades especificadas en el programa: Los desplazamientos deben ser ingresados en las unidades que se especifiquen en el programa. Si el programa trabaja con el sistema

internacional (SI) y utiliza metros (m) como unidad de longitud, los desplazamientos se ingresarán en metros. Si se utiliza otro sistema de unidades, se respetarán las unidades correspondientes.

Sentido de los desplazamientos: Es importante tener en cuenta el sentido de los desplazamientos al ingresar los valores. Esto implica considerar si el desplazamiento es positivo o negativo según la dirección del movimiento. Por ejemplo, si un desplazamiento en el eje X es hacia la derecha, se considerará positivo, mientras que, si es hacia la izquierda, será negativo.

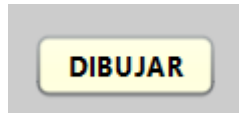


El código de programación para el desplazamiento es el siguiente:

```
private double[] getDesplazamientosDcTabla () {
    List<Double> aux = new ArrayList<>();
    int rows = tablaDesplazamientosDc.getRowCount();
    for (int i = 0; i < rows; i++) {
        try {
            String f = tablaDesplazamientosDc.getModel().getValueAt( rowIndex: i, columnIndex: 0).toString();
            if (!f.isEmpty()) {
                double fe = Double.parseDouble( s:f);
                aux.add( e:fe);
            }
        } catch (NullPointerException e) {
            break;
        }
    }
    double[] desplazamientos = new double[aux.size()];

    for (int i = 0; i < desplazamientos.length; i++) {
        desplazamientos[i] = aux.get( index:i);
    }
    return desplazamientos;
}
```

4.1.3.13 Programación del botón dibujar



El código de programación sigue flujo lógico y secuencial para realizar el cálculo de las cerchas. A continuación, te proporciono una descripción paso a paso del proceso:

Introducción de datos: El usuario ingresa los datos especificados anteriormente, como el número de barras, el área en metros cuadrados y el módulo de elasticidad en kN/m^2 . Estos datos son esenciales para definir las características y propiedades de las barras de la cercha.

Presionar el botón "Dibujar": Una vez que se han ingresado los datos, el usuario presiona el botón "Dibujar". Este botón activa la acción para calcular y dibujar la matriz de rigidez de cada barra.

Cálculo de la matriz de rigidez: En la pestaña de procedimiento, el programa procede a calcular la matriz de rigidez de cada barra de la cercha.

Graficar la barra: Mientras se calcula la matriz de rigidez, el programa va graficando cada barra en la pestaña de gráficos. Esto permite al usuario visualizar la configuración de la cercha y cómo están conectadas las barras.

El proceso de cálculo y dibujo progresivo permite que el usuario tenga una idea clara del análisis y diseño de la cercha en tiempo real. La representación gráfica de la cercha en la pestaña de gráficos también ayuda al usuario a visualizar cómo se están calculando las propiedades y comportamiento estructural de la cercha.

```

private void btnDibujarActionPerformed(java.awt.event.ActionEvent evt) { // GEN-FIRST:event_btnDibujarActionPerform

    dibujar();

    double e = Double.parseDouble( s:this.tbxE.getText().trim());
    double area = Double.parseDouble( s:this.tbxArea.getText().trim());
    // double h = Double.parseDouble(this.tbxAltura.getText().trim());
    double xf = Double.parseDouble( s:this.tbXXj.getText().trim());
    double xn = Double.parseDouble( s:this.tbXxi.getText().trim());
    double yf = Double.parseDouble( s:this.tbXyj.getText().trim());
    double yn = Double.parseDouble( s:this.tbXyi.getText().trim());

    int z1 = Integer.parseInt( s:this.tbz1.getText().trim());
    int z2 = Integer.parseInt( s:this.tbz2.getText().trim());
    int z3 = Integer.parseInt( s:this.tbz3.getText().trim());
    int z4 = Integer.parseInt( s:this.tbz4.getText().trim());

    this.lblCount.setText( text:Integer.toString( i:matrixCont));
    int max = Integer.parseInt( s:this.tbNroMatrixs.getText());
    if(titulo<1){
        guardarResultado( matrixToString: "PROCEDIMIENTO DE CALCULO: \n");
        guardarResultado( matrixToString: " \n");
        titulo++;
    }

    if (matrixCont <= max) {
        MatrizElemento matrizElemento = armadura.calcular( E:e, area, Xf:xf, Xn:xn, Yf:yf, Yn:yn, z1, z2, z3, z4);
        Map<String, Double> resultMap = matrizElemento.getKij();

        guardarResultado("MATRIZ K" + matrixCont + "\n");

        double longitud = matrizElemento.getLongitud();
        guardarResultado("Longitud=" + longitud + " " +lblBarraXiU.getText().trim() + "\n");
        guardarResultado( matrixToString:matUtil.matrixToString( m:matrizElemento.getMatrizK()));
        guardarResultado( matrixToString: "\n");
        guardarResultado( matrixToString:armadura.mapToString( kij:resultMap));
        guardarResultado( matrixToString: "\n");
        guardarResultado( matrixToString: "\n");

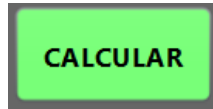
        armadura.resultdosKij.add( e:resultMap);
        if (matrixCont == max) {
            this.btnDibujar.setEnabled( b:false);
            this.btnCalcular.setEnabled( b:true);
            this.btnNuevo.setEnabled( b:true);
        }
    }

    matrixCont++;
    this.limpiarCampos();
} // GEN-LAST:event_btnDibujarActionPerformed

```

Las funciones del botón Dibujar están detallados en el [Anexo 6](#)

4.1.3.14 Programación del botón calcular



El proceso descrito después de presionar el botón "Calcular" sigue una secuencia lógica de cálculos y operaciones necesarias para analizar la cercha y obtener los resultados requeridos. A continuación, se proporciona una descripción paso a paso de este proceso:

Presionar el botón "Calcular": Después de guardar los datos ingresados previamente, el usuario presiona el botón "Calcular". Este botón inicia el proceso de cálculo de la cercha y activa las operaciones necesarias para obtener los resultados.

Visualización del progreso: Durante el cálculo, el progreso se muestra en la pestaña de procedimiento en la vista principal. Esta visualización permite al usuario seguir el avance del cálculo y asegurarse de que se esté realizando correctamente.

Suma de matrices para formar la matriz de rigidez global: Como parte del cálculo, el programa suma las matrices de rigidez de cada barra para formar la matriz de rigidez global de la cercha. Esta matriz representa la rigidez y comportamiento de toda la estructura y es fundamental para analizarla adecuadamente.

Ordenamiento de desplazamientos y formación de la matriz global: Se lleva a cabo un proceso de ordenamiento de los desplazamientos de cada barra mediante un código apropiado. Luego, estos desplazamientos se utilizan para formar la matriz global que contiene tanto los desplazamientos como las fuerzas externas aplicadas en la cercha.

Obtención de las matrices K_{ee} y K_{er} : A partir de la matriz global, se obtienen las sub matrices K_{ee} y K_{er} que se utilizan para calcular los desplazamientos de la cercha.

Obtención de las matrices K_{re} y K_{rr} : También a partir de la matriz global, se obtienen las sub matrices K_{re} y K_{rr} , que se utilizan para calcular las reacciones de la estructura.

Cálculo de desplazamientos y reacciones: Utilizando las sub matrices obtenidas, se realiza el cálculo de los desplazamientos de cada nodo y las reacciones en los apoyos de la cercha.

Al seguir este proceso, el programa podrá determinar las fuerzas internas, desplazamientos y reacciones en la cercha mediante el Método de Rigidez, lo que proporcionará una visión completa del comportamiento estructural de la cercha bajo las condiciones de carga y apoyo especificadas. Es esencial que todos los cálculos y operaciones se realicen con precisión para obtener resultados confiables y precisos.

```
Map<String, Double> sumasKij = armadura.getSumasKij();

String resultado = "";
double e = Double.parseDouble(s:this.tbxE.getText().trim());
double area = Double.parseDouble(s:this.tbxAarea.getText().trim());
int max = Integer.parseInt(s:this.tbxNroMatrixs.getText());
resultado += " DATOS GENERALES:\n";
resultado += " Nro de Barras= "+max+"\n";
resultado += " Modulo de Elasticidad E= "+e+"\n";
resultado += " Area de la Barras A= "+area+"\n";
resultado += "\n";

guardarResultado(matrixToString: "Resultados sumas Kij:\n");
guardarResultado(matrixToString: armadura.mapToString(kij:sumasKij));
double[][] matrizEnsamblada = armadura.getMatrizEnsamblada(sumasKij);

guardarResultado(matrixToString: "\n");
guardarResultado(matrixToString: "Matriz Ensamblada :\n");
guardarResultado(matrixToString: matUtil.matrixToString(m:matrizEnsamblada));

double[] fuerzasFc = getFuezasFcTabla();
armadura.setFuerzasFc(fuerzas:fuerzasFc);

double[] desplazamientosDc = getDesplazamientosDcTabla();
armadura.setDesplazamientosDc(fuerzas:desplazamientosDc);

double[][] matrizK11 = armadura.getMatrizK11();
guardarResultado(matrixToString: "\n");
guardarResultado(matrixToString: "Matriz K11:\n");
guardarResultado(matrixToString: matUtil.matrixToString(m:matrizK11));

double[][] matrizK21 = armadura.getMatrizK21();

guardarResultado(matrixToString: "\n");
guardarResultado(matrixToString: "Matriz K21:\n");
guardarResultado(matrixToString: matUtil.matrixToString(m:matrizK21));
```



```

double[][] matrizK12 = armadura.getMatrizK12();

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Matriz K12:\n");
guardarResultado( matrixToString: matUtil.matrixToString( m:matrizK12));

double[][] matrizK22 = armadura.getMatrizK22();

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Matriz K22:\n");
guardarResultado( matrixToString: matUtil.matrixToString( m:matrizK22));

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Calculo de Matriz Inversa:\n");
guardarResultado( matrixToString: "\n");

double determinate = matUtil.determinante( matriz:matrizK11);
guardarResultado("Determinate: " + Redondeo.dosDecimales( num: determinate) + "\n");

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Matriz Adjunta:\n");
double[][] matAdjunta = matUtil.matrizAdjunta( matriz:matrizK11);
guardarResultado( matrixToString: matUtil.matrixToString( m:matAdjunta));

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Traspuesta de la Matriz Adjunta:\n");
double[][] traspuestMatAdjunta = matUtil.matrizTraspuesta( matriz:matAdjunta);
guardarResultado( matrixToString: matUtil.matrixToString( m:traspuestMatAdjunta));

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Matriz Inversa:\n");
double[][] inversaK11 = matUtil.matrizInversa( matriz:matrizK11);
guardarResultado( matrixToString: matUtil.matrixToString( m:inversaK11));
//-----calculo de Dd-----*****

List<Double> desplazamientosDd = armadura.getDesplazamientoGlobal(matUtil);

//double[] desplazamientoG = matUtil.vectorXMatriz(fuerzasFc, inversaK11);
armadura.setDesplazamientoGlobal( despl:desplazamientosDd);

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Desplazamiento Global:\n");
String dg = armadura.imprimirDesplazamientoGlobal( longitud:lblBarraXiU.getText().trim());
resultado += "Desplazamiento Global:\n" + dg + "\n";
guardarResultado( matrixToString: dg);

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Reacciones Finales:\n");

//double[] reacFinales = matUtil.vectorXMatriz(desplazamientosDd, matrizK21);
double[] reacFinales = armadura.getReaccionesFinalesFd(matUtil); // matUtil.vectorXMatriz(despl

```

```

//double[] reacFinales = matUtil.vectorXMatriz(desplazamientosDd, matrizK21);
double[] reacFinales = armadura.getReaccionesFinalesFd(matUtil); // matUtil.vectorXMatriz(des

armadura.setReaccionesFinales( rfinales:reacFinales);
String rf = armadura.imprimirReaccionesFinales( fuerza:lblCargaValorU.getText().trim());
resultado += "Reacciones Finales:\n" + rf + "\n";
guardarResultado( matrixToString: rf);

armadura.calculaDesplazamientos();

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Desplazamiento en Cada Barra:\n");
guardarResultado( matrixToString: armadura.desplazamientosToString());

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Fuerzas Internas:\n");
String fi = armadura.mostrarFuerzasInternas(matUtil, fuerza:lblCargaValorU.getText().trim());
resultado += "Fuerzas Internas:\n" + fi + "\n";

guardarResultado( matrixToString: fi);
guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "\n =====");
guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "\n");

mostrarResumenResultado(resultado);
this.btnNuevo.setEnabled( b:true);
} // GEN-LAST:event_btnCalcular

```

Los Métodos y funciones que se utilizaron para el botón calcular están descritos en el [Anexo 2](#)

4.1.3.15 Resultados

Al presionar el botón "Calcular", se mostrará los resultados de desplazamientos, reacciones y fuerzas internas en una ventana separada es una forma conveniente y clara de presentar la información al usuario. Esta ventana de resultados permitirá al usuario analizar y comprender rápidamente cómo se comporta la cercha bajo las condiciones de carga y apoyo especificadas.

La ventana de resultados podría incluir lo siguiente:

Desplazamientos: Mostrar los desplazamientos resultantes en cada nodo de la cercha. Estos desplazamientos pueden estar en metros (u otra unidad especificada) y ser positivos o negativos según el sentido del movimiento.

Reacciones: Presentar las reacciones en los apoyos de la cercha. Estas reacciones son las fuerzas que ejercen los apoyos para mantener la estructura en equilibrio y pueden ser fuerzas verticales, horizontales.

Fuerzas internas: Mostrar las fuerzas internas en cada barra de la cercha, como fuerzas axiales, cortantes y momentos. Estas fuerzas representan las fuerzas que actúan dentro de cada barra debido a las cargas externas y los desplazamientos de la estructura.

Al mostrar los resultados en una ventana aparte es una práctica recomendada para presentar los datos de manera ordenada y legible, lo que facilita la interpretación de los resultados y proporciona una experiencia de usuario más efectiva y satisfactoria.

```
private void mostrarResumenResultado (String resultado) {
    DialogoResult dialog = new DialogoResult(new javax.swing.JFrame(), modal:true, resultado);
    /*dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });*/
    dialog.setVisible( b:true);
}
```

4.1.3.15.1 Reacciones en los apoyos

El código para mostrar las reacciones es el siguiente

```
guardarResultado ( matrixToString: "\n");
guardarResultado ( matrixToString: "Reacciones finales:\n");
double[] reacFinales = matUtil.vectorXMatriz( v:desplazamientoG, mtx:matrizK21);
armadura.setReaccionesFinales( rfinales:reacFinales);
String rf = armadura.imprimirReaccionesFinales();
resultado += "Reacciones finales:\n" + rf + "\n";
guardarResultado ( matrixToString: rf);
```

Los métodos para el cálculo de las reacciones se muestran en el [Anexo 4](#)

4.1.3.15.2 Desplazamientos

El código para mostrar el desplazamiento es el siguiente

```

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Desplazamiento global:\n");
String dg = armadura.imprimirDesplazamientoGlobal();
resultado += "Desplazamiento global:\n" + dg + "\n";
guardarResultado( matrixToString: dg);

```

Los métodos para el cálculo de los desplazamientos se muestran en el [Anexo 3](#)

4.1.3.15.3 Fuerzas Internas

El código para mostrar las fuerzas internas el siguiente

```

guardarResultado( matrixToString: "\n");
guardarResultado( matrixToString: "Fuerzas internas:\n");

String fi = armadura.mostrarFuerzasInternas(matUtil);

resultado += "Fuerzas internas:\n" + fi + "\n";

        guardarResultado( matrixToString: fi);
        guardarResultado( matrixToString: "\n");
        guardarResultado( matrixToString: "\n fin.");
        guardarResultado( matrixToString: "\n");
        guardarResultado( matrixToString: "\n");

        mostrarResumenResultado(resultado);
} // GEN-LAST:event_btnCalcular

```

Los métodos para el cálculo de las fuerzas internas se muestran en el [Anexo 5](#)

4.1.3.15.4 Mostrar el Procedimiento y Resumen de Resultados.

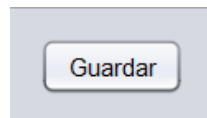
En la pestaña donde dice procedimiento, es donde se mostrar todo el procedimiento de cálculo y tendrá la opción de guardar en un archivo de texto. El código del procedimiento es el mismo código para el botón calcular es decir que procedimiento se va ir haciendo conforme se vaya calculando la cercha.

El código para mostrar el resumen de resultados es el siguiente:

```
private void mostrarResumenResultado(String resultado) {
    DialogoResult dialog = new DialogoResult(new javax.swing.JFrame(), modal:true, resultado);
    /*dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });*/
    dialog.setVisible( b:true);
}
```

4.1.3.16 Botón Guardar el procedimiento

Este botón tendrá la opción de guardar el procedimiento ya sea en un archivo txt, pdf y Excel



```
private void guardarResultado(String matrixToString) {
    String text = this.textAResultados.getText();
    text += matrixToString;
    this.textAResultados.setText( t:text);
}
```

4.1.4. FASE 4 Transición

4.1.4.1 Pruebas Funcionales del software

Las pruebas funcionales del software son un conjunto de actividades diseñadas para verificar que cada función y característica del programa cumple con los requisitos especificados y produce los resultados esperados. En el contexto del programa de análisis de cerchas en 2D, las pruebas funcionales verificarían si las funcionalidades relacionadas con el análisis de cerchas están operando correctamente. Aquí tienes un enfoque general para realizar pruebas funcionales:

Identificación de Escenarios de Prueba: Define escenarios de prueba que cubran todas las funcionalidades clave del programa, como la introducción de coordenadas, tipos de apoyo, tipos de carga, cálculo de resultados y presentación de resultados.

Diseño de Casos de Prueba: Crea casos de prueba específicos para cada escenario. Cada caso de prueba debe incluir la entrada esperada (datos de entrada) y el resultado esperado (resultados calculados y presentados).

Ejecución de Pruebas: Ingresas los datos en el programa de acuerdo con los casos de prueba diseñados y observa los resultados generados. Asegúrate de seguir exactamente los pasos descritos en los casos de prueba.

Comparación de Resultados: Compara los resultados generados por el programa con los resultados esperados según los casos de prueba. Verifica si los cálculos y las presentaciones de resultados son coherentes y precisos.

Pruebas de Integración: Además de las pruebas individuales de funciones, realiza pruebas de integración para asegurarte de que todas las partes del programa trabajan juntas de manera adecuada.

Pruebas de Usabilidad: No solo se trata de si el programa funciona correctamente, sino también de cómo lo hace desde la perspectiva del usuario. Realiza pruebas de usabilidad para garantizar que la interfaz de usuario sea intuitiva y fácil de usar.

Pruebas de Estrés y Rendimiento: Evalúa cómo se comporta el programa bajo condiciones extremas, como cargas altas o situaciones de uso intensivo, para asegurarte de que el rendimiento es satisfactorio.

Realizar pruebas funcionales sólidas es esencial para garantizar que tu software de análisis de cerchas en 2D funcione correctamente, produzca resultados precisos y sea confiable para los usuarios.

A continuación, se presentarán imágenes que ilustran el proceso de realización de las pruebas, junto con la identificación de algunos errores detectados durante dicho proceso.

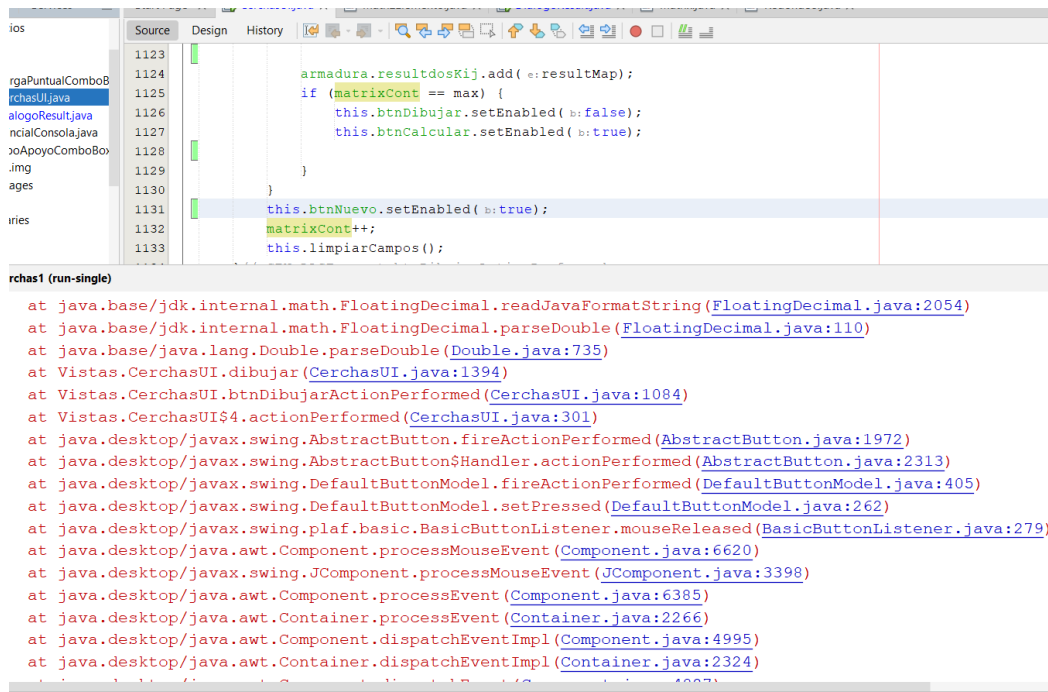
Figura 4.5 Errores de try catch



```
Output - AnalisisCerchas1 (run-single)
ant -f C:\Users\Marcelo\OneDrive\Escritorio\Respaldo\Modificado2\AnalisisCerchas -Dnb.internal.action.name=run.single -Djavac.includes
init:
Deleting: C:\Users\Marcelo\OneDrive\Escritorio\Respaldo\Modificado2\AnalisisCerchas\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\Marcelo\OneDrive\Escritorio\Respaldo\Modificado2\AnalisisCerchas\build\build-jar.properties
Compiling 1 source file to C:\Users\Marcelo\OneDrive\Escritorio\Respaldo\Modificado2\AnalisisCerchas\build\classes
Note: C:\Users\Marcelo\OneDrive\Escritorio\Respaldo\Modificado2\AnalisisCerchas\src\Vistas\CerchasUI.java uses unchecked or unsafe operations
Note: Recompile with -Xlint:unchecked for details.
compile-single:
run-single:
Kn.m
Lb.pie
Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException: For input string: "a"
    at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)
    at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
    at java.base/java.lang.Double.parseDouble(Double.java:735)
    at Vistas.CerchasUI.dibujar(CerchasUI.java:1390)
    at Vistas.CerchasUI.btnDibujarActionPerformed(CerchasUI.java:1084)
    at Vistas.CerchasUI$actionPerformed(CerchasUI.java:301)
    at java.desktop/javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1972)
    at java.desktop/javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2313)
    at java.desktop/javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:405)
    at java.desktop/javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:262)
    at java.desktop/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:279)
```

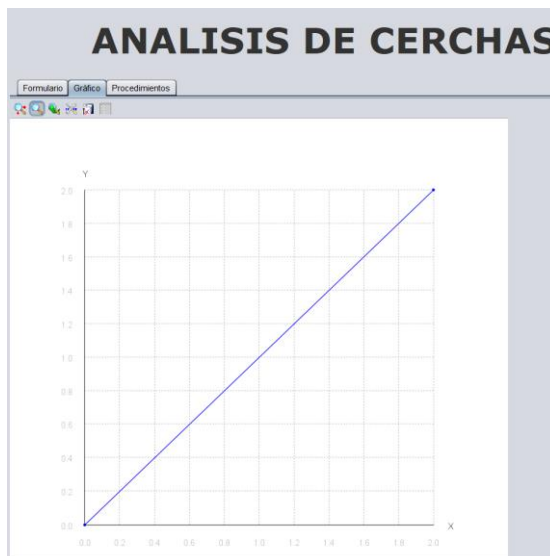
Nota. Error al convertir la variable de string a int. Captura de pantalla de NetBeans

Figura 4.6 El contador no reinicia



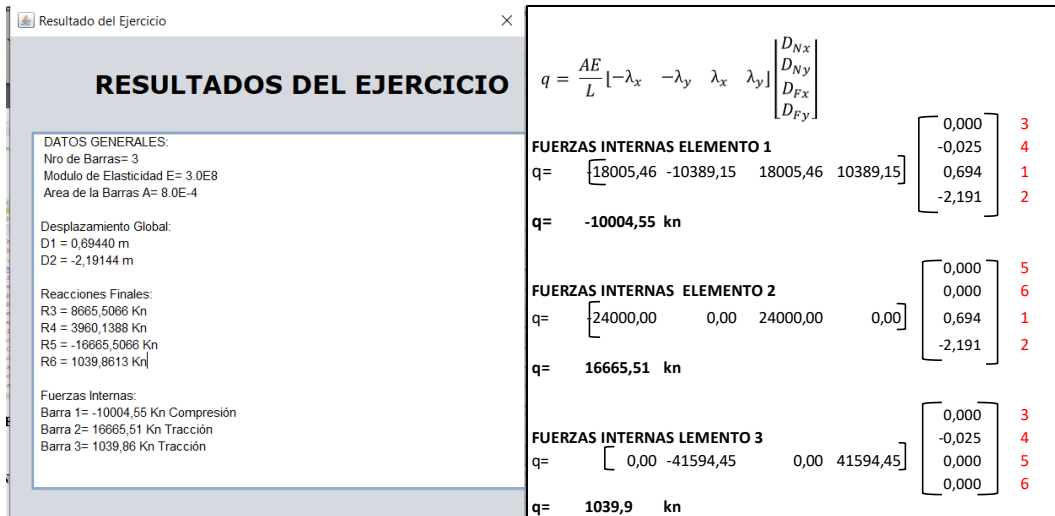
Nota. Error la variable contadora no reinicia para el evento del botón de un nuevo proyecto. Captura de pantalla de NetBeans

Figura 4.7 Pestaña de Gráficos



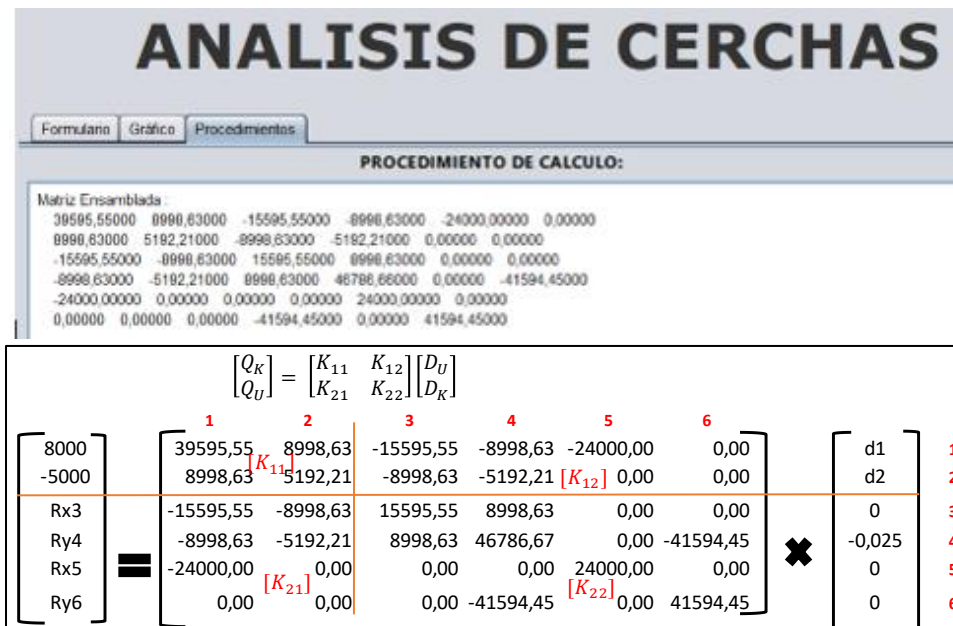
Nota. Pruebas en la pestaña de gráficos. Captura de pantalla de NetBeans

Figura 4.8 Comparación de resultados finales



Nota. Comparando los resultados calculados manualmente de las fuerzas internas.
 Captura de pantalla

Figura 4.9 Comparación de resultados de la matriz ensamblada



Nota. Comparando los resultados calculados manualmente en la matriz ensamblada de toda la estructura. Captura de pantalla

CAPÍTULO V
ANÁLISIS DE
RESULTADOS

CAPÍTULO V

ANÁLISIS DE RESULTADOS

Los resultados obtenidos tras completar el desarrollo del software se presentarán de manera exhaustiva. Para validar la precisión de estos resultados, se realizará una comparación directa con un software comercial ampliamente reconocido: el Sap2000 v14.0.0. El propósito subyacente es evaluar la concordancia entre los datos analizados a través de la nueva aplicación y los resultados generados por el software comercial establecido. Esta comparación será clave para determinar el nivel de coherencia y correspondencia entre ambas fuentes de información, lo que a su vez respaldará la confiabilidad y la exactitud de los cálculos obtenidos mediante el nuevo software desarrollado.

5.1. EJERCICIOS DE CERCHA EN DOS DIMENSIONES

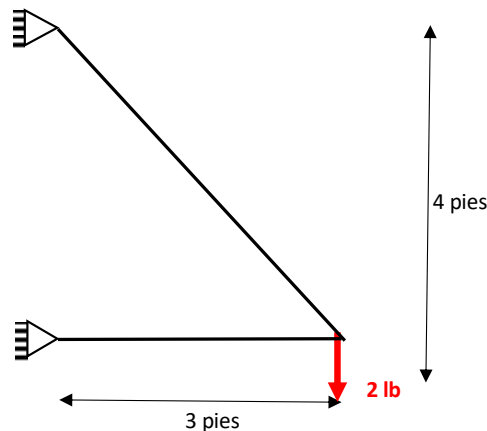
En el análisis, se considerará las siguientes cerchas, cuyos datos son los siguientes:

5.1.1 EJERCICIO 1

Datos:

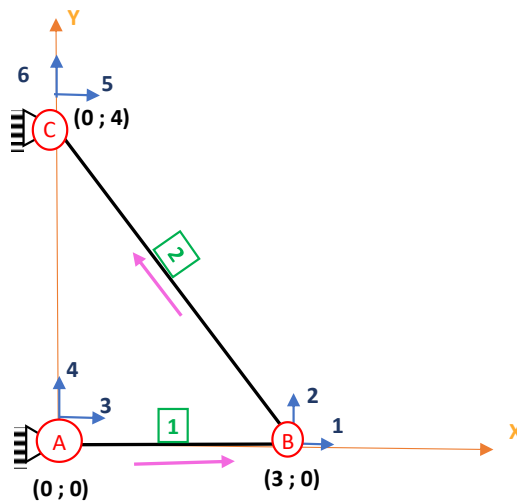
- Área de las barras $A=0.0006 \text{ pie}^2$
- Módulo de elasticidad del material $E=8 \times 10^6 \text{ lb/pie}^2$

Figura 5.1. Ejercicio 1 de cercha en 2D



Nota. Ejercicio de armaduras con dos barras. (R.C.HIBBELER, 2012)

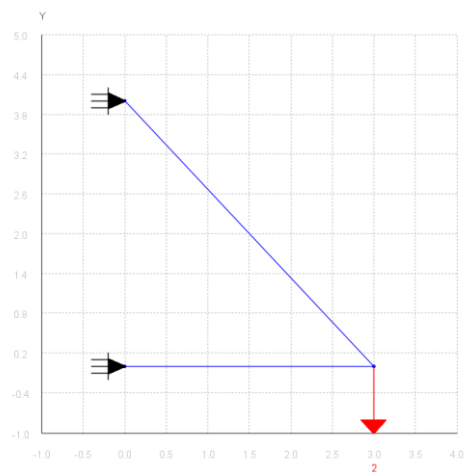
Figura 5.2 Discretización de la estructura Ejercicio 1



Nota. Discretización de desplazamientos de nudos y de barras. Elaboración de propia

5.2. RESULTADOS POR EL SOFTWARE ANÁLISIS DE CERCHAS

Figura 5.3 Gráfico del ejercicio en el software



Nota. Datos cargados en software generados en la pestaña de gráficos. Elaboración de propia

5.2.1 Procedimiento de cálculo

Una de las ventajas sobresalientes de este software reside en su capacidad para brindar una visualización detallada y secuencial de todo el proceso de cálculo. Esto significa que los usuarios tienen la posibilidad de seguir paso a paso el procedimiento completo de análisis. Esta funcionalidad no solo enriquece la comprensión del proceso, sino que también contribuye a la transparencia y claridad en la generación de resultados. Al permitir que los usuarios visualicen cada etapa, desde la entrada de datos hasta la presentación de resultados, el software fomenta una comprensión más profunda y facilita el análisis crítico de los cálculos, lo que es esencial tanto para fines educativos como para proyectos profesionales.

PROCEDIMIENTO DE CALCULO:

MATRIZ K1

Longitud= 3.0 pie

| | | | |
|-------------|----------|-------------|----------|
| 1600,00000 | 0,00000 | -1600,00000 | -0,00000 |
| 0,00000 | 0,00000 | -0,00000 | -0,00000 |
| -1600,00000 | -0,00000 | 1600,00000 | 0,00000 |
| -0,00000 | -0,00000 | 0,00000 | 0,00000 |

MATRIZ K2

Longitud= 5.0 pie

| | | | |
|------------|------------|------------|------------|
| 345,60000 | -460,80000 | -345,60000 | 460,80000 |
| -460,80000 | 614,40000 | 460,80000 | -614,40000 |
| -345,60000 | 460,80000 | 345,60000 | -460,80000 |
| 460,80000 | -614,40000 | -460,80000 | 614,40000 |

Matriz Ensamblada :

| | | | | | |
|-------------|------------|-------------|---------|------------|------------|
| 1945,60000 | -460,80000 | -1600,00000 | 0,00000 | -345,60000 | 460,80000 |
| -460,80000 | 614,40000 | 0,00000 | 0,00000 | 460,80000 | -614,40000 |
| -1600,00000 | 0,00000 | 1600,00000 | 0,00000 | 0,00000 | 0,00000 |
| 0,00000 | 0,00000 | 0,00000 | 0,00000 | 0,00000 | 0,00000 |
| -345,60000 | 460,80000 | 0,00000 | 0,00000 | 345,60000 | -460,80000 |
| 460,80000 | -614,40000 | 0,00000 | 0,00000 | -460,80000 | 614,40000 |

Matriz K11:

| | |
|------------|------------|
| 1945,60000 | -460,80000 |
| -460,80000 | 614,40000 |

Matriz K21:

| | |
|-------------|------------|
| -1600,00000 | 0,00000 |
| 0,00000 | 0,00000 |
| -345,60000 | 460,80000 |
| 460,80000 | -614,40000 |

Matriz K12:

| | | | |
|-------------|---------|------------|------------|
| -1600,00000 | 0,00000 | -345,60000 | 460,80000 |
| 0,00000 | 0,00000 | 460,80000 | -614,40000 |

Matriz K22:

| | | | |
|------------|---------|------------|------------|
| 1600,00000 | 0,00000 | 0,00000 | 0,00000 |
| 0,00000 | 0,00000 | 0,00000 | 0,00000 |
| 0,00000 | 0,00000 | 345,60000 | -460,80000 |
| 0,00000 | 0,00000 | -460,80000 | 614,40000 |

Calculo de la Matriz Inversa de K11:

Determinante= 983040.0

Matriz Adjunta:

| | |
|------------|------------|
| 1945,60000 | -460,80000 |
| -460,80000 | 614,40000 |

Traspuesta de la Matriz Adjunta:

| | |
|------------|------------|
| 1945,60000 | -460,80000 |
| -460,80000 | 614,40000 |

Matriz Inversa:

| | |
|---------|---------|
| 0,00063 | 0,00047 |
| 0,00047 | 0,00198 |

Desplazamiento Global:

D1 = -0,00094 pie

D2 = -0,00396 pie

Reacciones Finales:

R3 = 1,50 Lb

R4 = 0,00 Lb

R5 = -1,50 Lb

R6 = 2,00 Lb

Desplazamiento en Cada Barra:

Barra 1

0,0000

0,0000

-0,0009

-0,0040

Barra 2

-0,0009

-0,0040

0,0000

0,0000

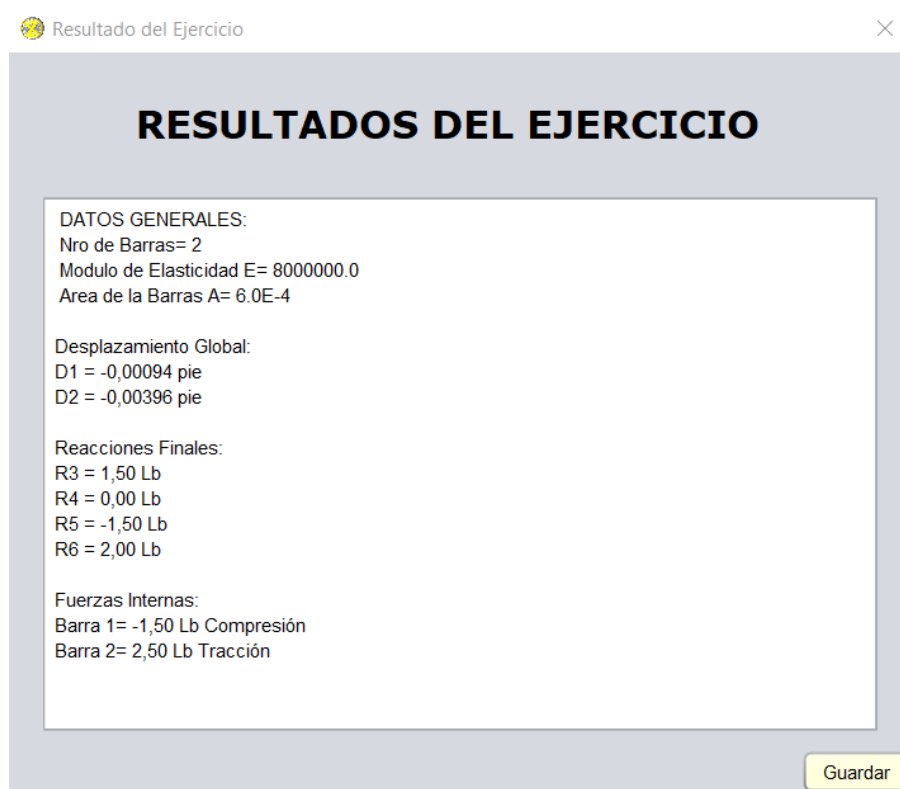
Fuerzas Internas:

Barra 1= -1,50 Lb Compresión

Barra 2= 2,50 Lb Tracción

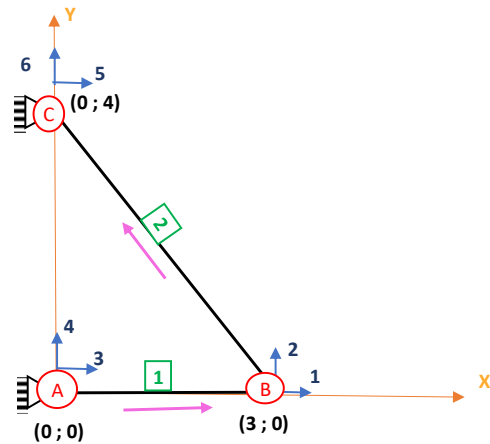
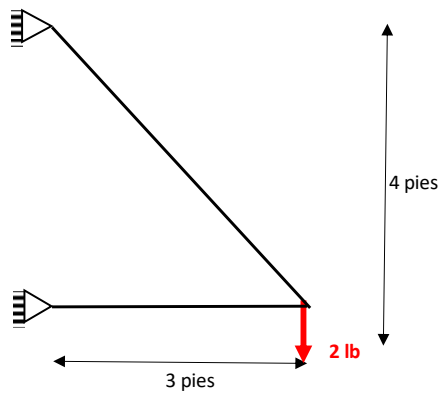
5.2.2 Resumen de los resultados finales

Figura 5.4 Resultados finales del ejercicio



Nota. Resultados finales del ejercicio calculados en el software. Captura de pantalla del software.

5.2.3 Resultados calculados manualmente



BARRA 1

$$E = 8000000,0 \text{ lb/pie}^2$$

$$A = 0,0006 \text{ pie}^2$$

$$L = 3,00 \text{ pie}$$

$$AE/L = 1600,00 \text{ lb/pie}$$

$$X_j = 3,00 \text{ pies}$$

$$X_i = 0,00 \text{ pies}$$

$$\lambda_x = 1,00$$

$$Y_j = 0,00 \text{ pies}$$

$$Y_i = 0,00 \text{ pies}$$

$$\lambda_y = 0,00$$

$$\lambda_x^2 = 1,000$$

$$\lambda_y^2 = 0,000$$

$$\lambda_x \lambda_y = 0,000$$

$$\lambda_x = \frac{X_j - X_i}{L}$$

$$\lambda_y = \frac{Y_j - Y_i}{L}$$

MATRIZ 1

| | | | | | |
|---------|-----|---------|-----|-----|---|
| | 3 | 4 | 1 | 2 | |
| 1600,0 | 0,0 | -1600,0 | 0,0 | 0,0 | 3 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 4 |
| -1600,0 | 0,0 | 1600,0 | 0,0 | 0,0 | 1 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 2 |

BARRA 2

$$E = 8000000,0 \text{ lb/pie}^2$$

$$A = 0,0006 \text{ pie}^2$$

$$L = 5,00 \text{ pie}$$

$$AE/L = 960,00 \text{ lb/pie}$$

$$X_j = 0,00 \text{ pies}$$

$$X_i = 3,00 \text{ pies}$$

$$\lambda_x = -0,60$$

$$Y_j = 4,00 \text{ pies}$$

$$Y_i = 0,00 \text{ pies}$$

$$\lambda_y = 0,80$$

$$\lambda_x^2 = 0,360$$

$$\lambda_y^2 = 0,640$$

$$\lambda_x \lambda_y = -0,480$$

MATRIZ 2

| | | | | | |
|--------|--------|--------|--------|-----|---|
| | 1 | 2 | 5 | 6 | |
| 345,6 | -460,8 | -345,6 | 460,8 | 0,0 | 1 |
| -460,8 | 614,4 | 460,8 | -614,4 | 0,0 | 2 |
| -345,6 | 460,8 | 345,6 | -460,8 | 0,0 | 5 |
| 460,8 | -614,4 | -460,8 | 614,4 | 0,0 | 6 |

APLICACIÓN DE LA TEORIA DEL METODO DE LA RIGIDEZ

$$\begin{bmatrix} F_e \\ F_r \end{bmatrix} = \begin{bmatrix} K_{ee} & K_{er} \\ K_{re} & K_{rr} \end{bmatrix} \begin{bmatrix} D_e \\ D_r \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ -2 \\ \text{Rx3} \\ \text{Ry4} \\ \text{Rx5} \\ \text{Ry6} \end{bmatrix} = \begin{bmatrix} 1945,60 & -460,80 & -1600,00 & 0,00 & -345,60 & 460,80 \\ -460,80 & 614,40 & 0,00 & 0,00 & 460,80 & -614,40 \\ -1600,00 & 0,00 & 1600,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ -345,60 & 460,80 & 0,00 & 0,00 & 345,60 & -460,80 \\ 460,80 & -614,40 & 0,00 & 0,00 & -460,80 & 614,40 \end{bmatrix} \begin{bmatrix} d1 \\ d2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

CALCULO DE DESPLAZAMIENTOS EN LOS NUDOS

$$D_e = [K_{ee}]^{-1} F_e$$

$$\begin{bmatrix} d1 \\ d2 \end{bmatrix} = \begin{bmatrix} 0,00063 & 0,00047 \\ 0,00047 & 0,00198 \end{bmatrix} \times \begin{bmatrix} 0,00 \\ -2,00 \end{bmatrix} = \begin{bmatrix} -0,0009 \text{ pie} \\ -0,0040 \text{ pie} \end{bmatrix}$$

CALCULO DE REACCIONES

$$F_r = K_{er} D_e$$

$$\begin{bmatrix} \text{Rx3} \\ \text{Ry4} \\ \text{Rx5} \\ \text{Ry6} \end{bmatrix} = \begin{bmatrix} -1600,00 & 0,00 \\ 0,00 & 0,00 \\ -345,60 & 460,80 \\ 460,80 & -614,40 \end{bmatrix} \times \begin{bmatrix} -0,0009 \\ -0,0040 \end{bmatrix} = \begin{bmatrix} 1,5 \text{ lb} \\ 0 \text{ lb} \\ -1,50 \text{ lb} \\ 2,00 \text{ lb} \end{bmatrix}$$

ESFUERZOS EN LOS MIEMBROS

$$q = \frac{AE}{L} \begin{bmatrix} -\lambda_x & -\lambda_y & \lambda_x & \lambda_y \end{bmatrix} \begin{pmatrix} D_{ix} \\ D_{iy} \\ D_{jx} \\ D_{jy} \end{pmatrix}$$

FUERZAS INTERNAS BARRA 1

$$q = \begin{bmatrix} -1600,00 & 0,00 & 1600,00 & 0,00 \end{bmatrix} \begin{bmatrix} 0,0 \\ 0,0 \\ -0,0009 \\ -0,0040 \end{bmatrix}$$

$$q = -1,50 \text{ lb}$$

FUERZAS INTERNAS BARRA 2

$$q = \begin{bmatrix} 576,00 & -768,00 & -576,00 & 768,00 \end{bmatrix} \begin{bmatrix} -0,0009 \\ -0,0040 \\ 0,0 \\ 0,0 \end{bmatrix}$$

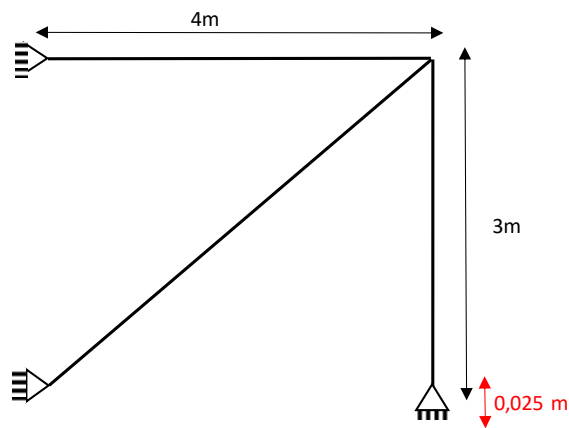
$$q = 2,50 \text{ lb}$$

5.4 EJERCICIO 2

Datos:

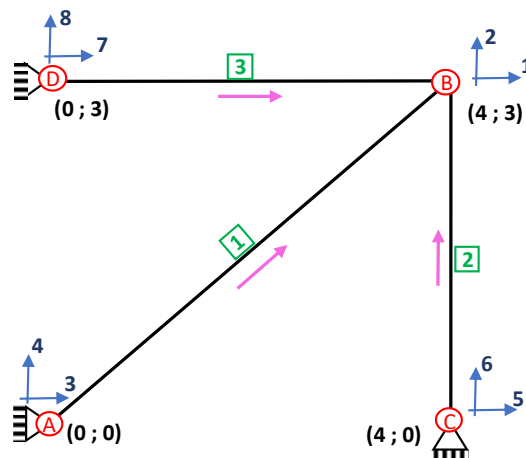
- Área de las barras $A = 0.0004 \text{ m}^2$
- Módulo de elasticidad del material $E = 200 \times 10^5 \text{ Kn/m}^2$
- Desplazamiento impuesto en el apoyo C de 0.025 metros

Figura 5.5 Ejercicio 2 de cercha en 2D



Nota. Ejercicio de armaduras con tres barras. Elaboración Propia

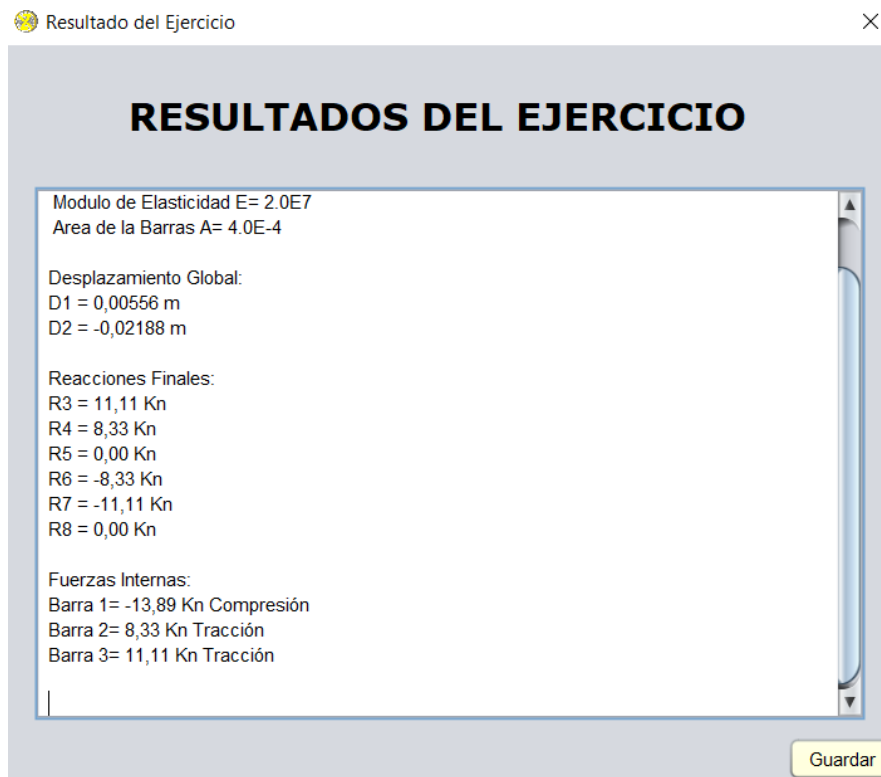
Figura 5.6 Discretización del ejercicio 2



Nota. Discretización de las armaduras con tres barras. Elaboración Propia

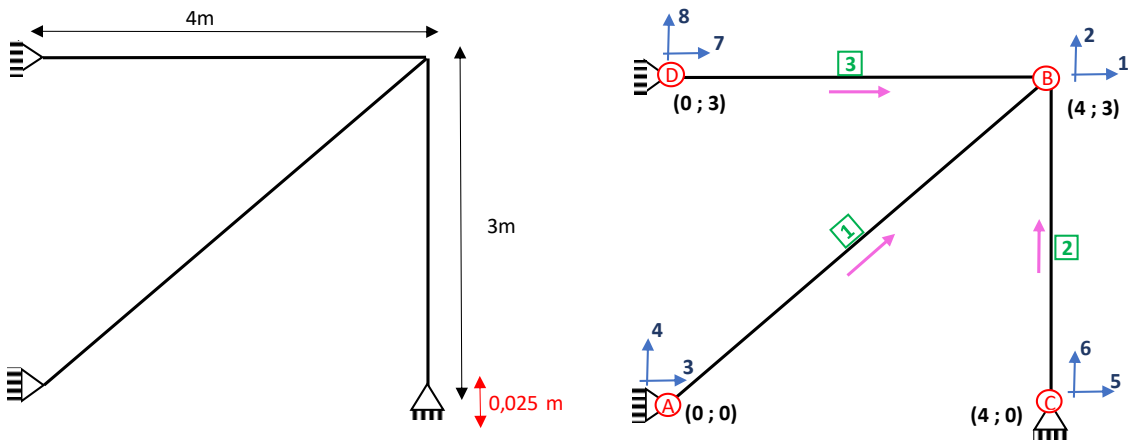
5.4.1 Resultados en el software.

Figura 5.7 Resultados finales del ejercicio 2



Nota. Resultados finales del ejercicio 2 calculados en el software. Captura de pantalla del software

5.4.2 Resultados calculados manualmente.



BARRA 1

$E = 20000000,0 \text{ KN/m}^2$
 $A = 0,0004 \text{ m}^2$
 $L = 5,00 \text{ m}$
 $AE/L = 1600,00 \text{ KN/m}$
 $XF = 4,00 \text{ m}$
 $XN = 0,00 \text{ m}$
 $\lambda_x = 0,80$
 $YF = 3,00 \text{ m}$
 $YN = 0,00 \text{ m}$
 $\lambda_y = 0,60$
 $\lambda_x^2 = 0,640$
 $\lambda_y^2 = 0,360$
 $\lambda_x \lambda_y = 0,480$

MATRIZ 1

$$\begin{bmatrix}
 3 & 4 & 1 & 2 \\
 1024,00 & 768,00 & -1024,00 & -768,00 \\
 768,00 & 576,00 & -768,00 & -576,00 \\
 -1024,00 & -768,00 & 1024,00 & 768,00 \\
 -768,00 & -576,00 & 768,00 & 576,00
 \end{bmatrix}
 \begin{matrix}
 3 \\
 4 \\
 1 \\
 2
 \end{matrix}$$

BARRA 2

$E = 20000000,0 \text{ KN/m}^2$
 $A = 0,0004 \text{ m}^2$
 $L = 3,00 \text{ m}$
 $AE/L = 2666,67 \text{ KN/m}$
 $XF = 4,00 \text{ m}$
 $XN = 4,00 \text{ m}$
 $\lambda_x = 0,00$
 $YF = 3,00 \text{ m}$
 $YN = 0,00 \text{ m}$
 $\lambda_y = 1,00$
 $\lambda_x^2 = 0,000$
 $\lambda_y^2 = 1,000$
 $\lambda_x \lambda_y = 0,000$

MATRIZ 2

$$\begin{bmatrix}
 5 & 6 & 1 & 2 \\
 0,00 & 0,00 & 0,00 & 0,00 \\
 0,00 & 2666,67 & 0,00 & -2666,67 \\
 0,00 & 0,00 & 0,00 & 0,00 \\
 0,00 & -2666,67 & 0,00 & 2666,67
 \end{bmatrix}
 \begin{matrix}
 5 \\
 6 \\
 1 \\
 2
 \end{matrix}$$

BARRA 3

E= 20000000,00 KN/m²
 A= 0,0004 m²
 L= 4,00 m
 AE/L= 2000,00 KN/m
 XF= 4,00 m
 XN= 0,00 m
 $\lambda_X = 1,00$
 YF= 3,00 m
 YN= 3,00 m
 $\lambda_Y = 0,00$
 $\lambda_x^2 = 1,000$
 $\lambda_y^2 = 0,000$
 $\lambda_x \lambda_y = 0,000$

MATRIZ 3

$$\begin{bmatrix} 7 & 8 & 1 & 2 \\ 2000,00 & 0,00 & -2000,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 \\ -2000,00 & 0,00 & 2000,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix}
 \begin{matrix} 7 \\ 8 \\ 1 \\ 2 \end{matrix}$$

APLICACIÓN DE LA TEORIA DEL METODO DE LA RIGIDEZ

$$\begin{bmatrix} F_e \\ F_r \end{bmatrix} = \begin{bmatrix} K_{ee} & K_{er} \\ K_{re} & K_{rr} \end{bmatrix} \begin{bmatrix} D_e \\ D_r \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ Rx3 \\ Ry4 \\ Rx5 \\ Ry6 \\ Rx7 \\ Ry8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3024,00 & 768,00 & -1024,00 & -768,00 & 0,00 & 0,00 & -2000,00 & 0,00 \\ 768,00 & 3242,67 & -768,00 & -576,00 & 0,00 & -2666,67 & 0,00 & 0,00 \\ -1024,00 & -768,00 & 1024,00 & 768,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ -768,00 & -576,00 & 768,00 & 576,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & -2666,67 & 0,00 & 0,00 & 0,00 & 2666,67 & 0,00 & 0,00 \\ -2000,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 2000,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix} \begin{matrix} d1 \\ d2 \\ 0 \\ 0 \\ 0 \\ -0,025 \\ 0 \\ 0 \end{matrix}$$

CALCULO DE DESPLAZAMIENTOS EN LOS NUDOS

$$D_e = [K_{ee}]^{-1} (F_e - K_{er} D_r)$$

$$[K_{ee}]^{-1} = \begin{bmatrix} 0,0004 & -0,0001 \\ -0,0001 & 0,0003 \end{bmatrix}$$

$$K_{er} D_r = \begin{bmatrix} 0,00 \\ 66,67 \end{bmatrix}$$

$$\begin{matrix} d1 \\ d2 \end{matrix} = \begin{bmatrix} 3024,00 & 768,00 \\ 768,00 & 3242,67 \end{bmatrix} \begin{bmatrix} 0,00 \\ -66,67 \end{bmatrix} = \begin{matrix} 0,0056 \text{ m} \\ -0,0219 \text{ m} \end{matrix}$$

CALCULO DE REACCIONES

$$F_r = K_{er} D_e + K_{rr} D_r$$

| | | | | | | |
|-----|---|---|---|---|---|------------|
| Rx3 | = | $\begin{bmatrix} 11,11111 \\ 8,33 \\ 0,00 \\ 58,33 \\ -11,11 \\ 0,00 \end{bmatrix}$ | + | $\begin{bmatrix} 0 \\ 0,00 \\ 0,00 \\ -66,67 \\ 0,00 \\ 0,00 \end{bmatrix}$ | = | 11,111 Kn |
| Ry4 | | | | | | 8,3333 Kn |
| Rx5 | | | | | | 0 Kn |
| Ry6 | | | | | | -8,3333 Kn |
| Rx7 | | | | | | -11,111 Kn |
| Ry8 | | | | | | 0 Kn |

CÁLCULO DE FUERZAS INTERNAS

$$q = \frac{AE}{L} \begin{bmatrix} -\lambda_x & -\lambda_y & \lambda_x & \lambda_y \end{bmatrix} \begin{pmatrix} Dix \\ Diy \\ Djx \\ Djy \end{pmatrix}$$

FUERZAS INTERNAS BARRA 1

$$q = \begin{bmatrix} -1280,00 & -960,00 & 1280,00 & 960,00 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ 0,006 \\ -0,022 \end{bmatrix} \begin{matrix} 3 \\ 4 \\ 1 \\ 2 \end{matrix}$$

q= -13,89 kn

FUERZAS INTERNAS BARRA 2

$$q = \begin{bmatrix} 0,00 & -2666,67 & 0,00 & 2666,67 \end{bmatrix} \begin{bmatrix} 0 \\ -0,025 \\ 0,006 \\ -0,022 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 1 \\ 2 \end{matrix}$$

q= 8,33 kn

FUERZAS INTERNAS BARRA 3

$$q = \begin{bmatrix} -2000,00 & 0,00 & 2000,00 & 0,00 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ 0,006 \\ -0,022 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 1 \\ 2 \end{matrix}$$

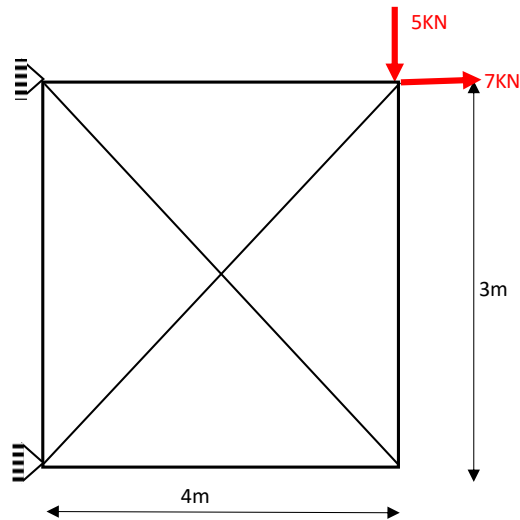
q= 11,1 kn

5.5 EJERCICIO 3

Datos:

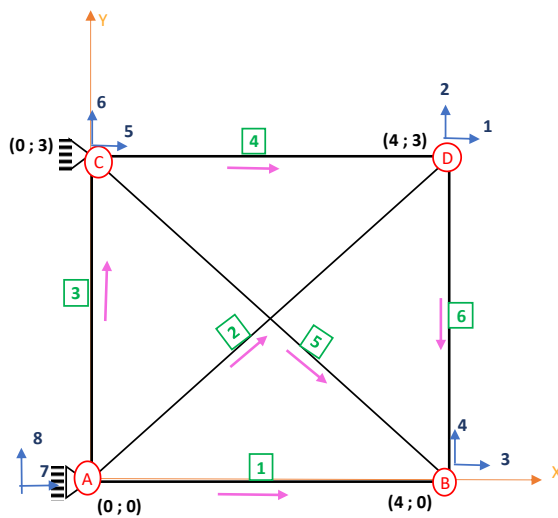
- Área de las barras (A) constante
- Módulo de elasticidad del material (E) también se mantendrá constante.

Figura 5.8. Ejercicio 2 de cercha en 2D



Nota. Ejercicio de armaduras con seis barras. Elaboración Propia

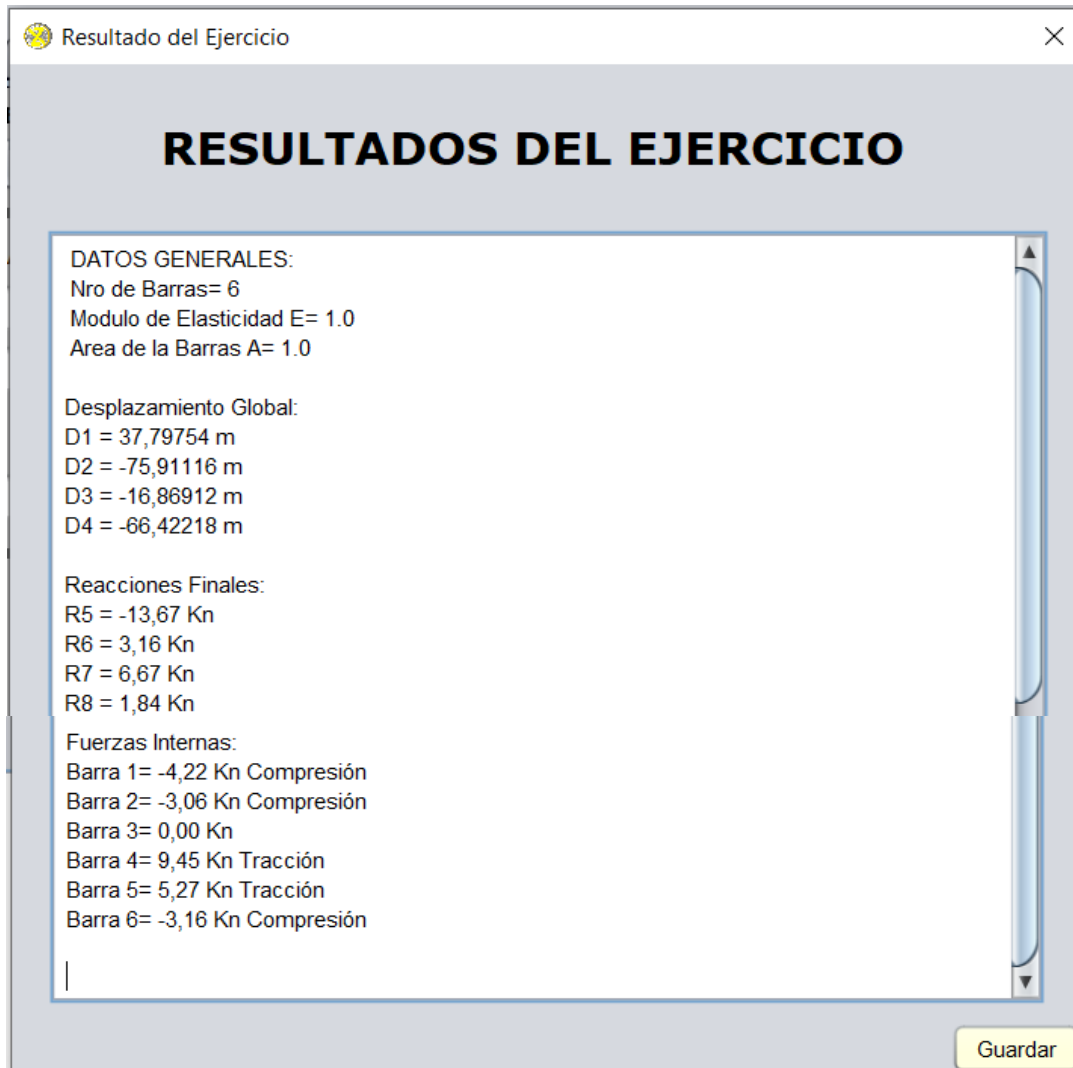
Figura 5.9 Discretización del ejercicio 3



Nota. Discretización de las armaduras con tres barras. Elaboración Propia.

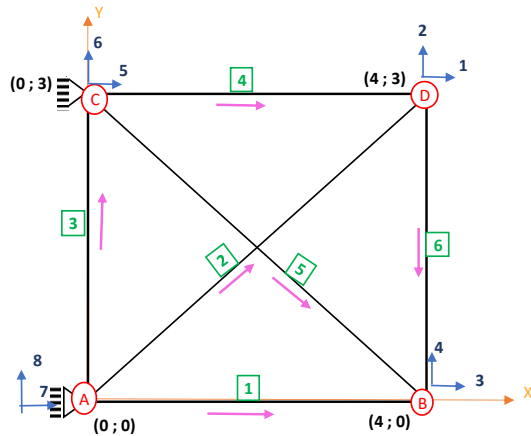
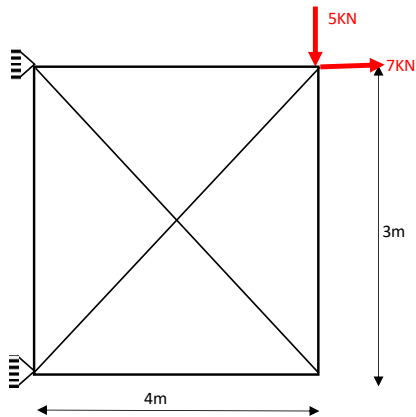
5.5.1 Resultados en el software.

Figura 5.10 Resultados finales del ejercicio 3



Nota. Resultados finales del ejercicio 3 calculados en el software. Captura de pantalla del software

5.5.2 Resultados calculados manualmente.



BARRA 1

$E = 1 \text{ KN/m}^2$
 $A = 1 \text{ m}^2$
 $L = 4,00 \text{ m}$
 $AE/L = 0,25 \text{ KN/m}$
 $XF = 4,00 \text{ m}$
 $XN = 0,00 \text{ m}$
 $\lambda_x = 1,00$
 $YF = 0,00 \text{ m}$
 $YN = 0,00 \text{ m}$
 $\lambda_y = 0,00$
 $\lambda_x^2 = 1,0$
 $\lambda_y^2 = 0,0$
 $\lambda_x \lambda_y = 0,0$

MATRIZ 1

$$\begin{bmatrix}
 7 & 8 & 3 & 4 \\
 0,25 & 0,00 & -0,25 & 0,00 \\
 0,00 & 0,00 & 0,00 & 0,00 \\
 -0,25 & 0,00 & 0,25 & 0,00 \\
 0,00 & 0,00 & 0,00 & 0,00
 \end{bmatrix}
 \begin{matrix}
 7 \\
 8 \\
 3 \\
 4
 \end{matrix}$$

BARRA 2

$E = 1 \text{ KN/m}^2$
 $A = 1 \text{ m}^2$
 $L = 5,00 \text{ m}$
 $AE/L = 0,20 \text{ KN/m}$
 $XF = 4,00 \text{ m}$
 $XN = 0,00 \text{ m}$
 $\lambda_x = 0,80$
 $YF = 3,00 \text{ m}$
 $YN = 0,00 \text{ m}$
 $\lambda_y = 0,60$
 $\lambda_x^2 = 0,64$
 $\lambda_y^2 = 0,36$
 $\lambda_x \lambda_y = 0,48$

MATRIZ 2

$$\begin{bmatrix}
 7 & 8 & 1 & 2 \\
 0,13 & 0,10 & -0,13 & -0,10 \\
 0,10 & 0,07 & -0,10 & -0,07 \\
 -0,13 & -0,10 & 0,13 & 0,10 \\
 -0,10 & -0,07 & 0,10 & 0,07
 \end{bmatrix}
 \begin{matrix}
 7 \\
 8 \\
 1 \\
 2
 \end{matrix}$$

BARRA 3

E= 1 KN/m²
 A= 1 m²
 L= 3,00 m
 AE/L= 0,33 KN/m
 XF= 0,00 m
 XN= 0,00 m

$\lambda_X = 0,00$
 YF= 3,00 m
 YN= 0,00 m
 $\lambda_Y = 1,00$
 $\lambda_x^2 = 0,0$
 $\lambda_y^2 = 1,0$
 $\lambda_x \lambda_y = 0,0$

MATRIZ 3

$$\begin{bmatrix} 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,33 & 0,00 & -0,33 \\ 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & -0,33 & 0,00 & 0,33 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 5 \\ 6 \end{matrix}$$

BARRA 4

E= 1 KN/m²
 A= 1 m²
 L= 4,00 m
 AE/L= 0,25 KN/m
 XF= 4,00 m
 XN= 0,00 m

$\lambda_X = 1,00$
 YF= 3,00 m
 YN= 3,00 m
 $\lambda_Y = 0,00$
 $\lambda_x^2 = 1,0$
 $\lambda_y^2 = 0,0$
 $\lambda_x \lambda_y = 0,0$

MATRIZ 4

$$\begin{bmatrix} 0,25 & 0,00 & -0,25 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 \\ -0,25 & 0,00 & 0,25 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 1 \\ 2 \end{matrix}$$

BARRA 5

E= 1 KN/m²
 A= 1 m²
 L= 5,00 m
 AE/L= 0,20 KN/m
 XF= 4,00 m
 XN= 0,00 m

$\lambda_X = 0,80$
 YF= 0,00 m
 YN= 3,00 m
 $\lambda_Y = -0,60$
 $\lambda_x^2 = 0,64$

MATRIZ 5

$$\begin{bmatrix} 0,13 & -0,10 & -0,13 & 0,10 \\ -0,10 & 0,07 & 0,10 & -0,07 \\ -0,13 & 0,10 & 0,13 & -0,10 \\ 0,10 & -0,07 & -0,10 & 0,07 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 3 \\ 4 \end{matrix}$$

$$\lambda_y^2 = 0,36$$

$$\lambda_x \lambda_y = -0,48$$

BARRA 6

MATRIZ 6

| | | | | | | | | | |
|-------------------------|---------------------|--|--|--|--|--|--|--|--|
| E= | 1 KN/m ² | | | | | | | | |
| A= | 1 m ² | | | | | | | | |
| L= | 3,00 m | | | | | | | | |
| AE/L= | 0,33 KN/m | | | | | | | | |
| XF= | 4,00 m | | | | | | | | |
| XN= | 4,00 m | | | | | | | | |
| $\lambda_x =$ | 0,00 | | | | | | | | |
| YF= | 0,00 m | | | | | | | | |
| YN= | 3,00 m | | | | | | | | |
| $\lambda_y =$ | -1,00 | | | | | | | | |
| $\lambda_x^2 =$ | 0,0 | | | | | | | | |
| $\lambda_y^2 =$ | 1,0 | | | | | | | | |
| $\lambda_x \lambda_y =$ | 0,0 | | | | | | | | |

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

APLICACIÓN DE LA TEORIA DEL METODO DE LA RIGIDEZ

$$\begin{bmatrix} Q_K \\ Q_U \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} D_U \\ D_K \end{bmatrix}$$

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

CALCULO DE DESPLAZAMIENTOS EN LOS NUDOS

$$D_U = [K_{11}]^{-1} Q_K$$

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

CALCULO DE REACCIONES

$$Q_U = K_{21}D_U$$

$$\begin{bmatrix} R_{x5} \\ R_{y6} \\ R_{x7} \\ R_{y8} \end{bmatrix} = \begin{bmatrix} -0,25 & 0,00 & -0,13 & 0,10 \\ 0,00 & 0,00 & 0,10 & -0,07 \\ -0,13 & -0,10 & -0,25 & 0,00 \\ -0,10 & -0,07 & 0,00 & 0,00 \end{bmatrix} \begin{bmatrix} 37,80 \\ -75,91 \\ -16,87 \\ -66,42 \end{bmatrix} = \begin{bmatrix} -13,67 \text{ KN} \\ 3,16 \text{ KN} \\ 6,67 \text{ KN} \\ 1,84 \text{ KN} \end{bmatrix}$$

ESFUERZOS EN LOS MIEMBROS

$$q = \frac{AE}{L} \begin{bmatrix} -\lambda_x & -\lambda_y & \lambda_x & \lambda_y \end{bmatrix} \begin{bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{bmatrix}$$

FUERZAS INTERNAS BARRA 1

$$q = \begin{bmatrix} -0,25 & 0,00 & 0,25 & 0,00 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ -16,87 \\ -66,42 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 3 \\ 4 \end{matrix}$$

q= -4,22 kn

FUERZAS INTERNAS BARRA 2

$$q = \begin{bmatrix} -0,16 & -0,12 & 0,16 & 0,12 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ 37,80 \\ -75,91 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 1 \\ 2 \end{matrix}$$

q= -3,06 kn

FUERZAS INTERNAS BARRA 3

$$q = \begin{bmatrix} 0,00 & -0,33 & 0,00 & 0,33 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ 0,00 \\ 0,00 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 5 \\ 6 \end{matrix}$$

q= 0,0 kn

FUERZAS INTERNAS BARRA 4

$$q = \begin{bmatrix} -0,25 & 0,00 & 0,25 & 0,00 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ 37,8 \\ -75,9 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 1 \\ 2 \end{matrix}$$

q= 9,4 kn

FUERZAS INTERNAS BARRA 5

$$q = \begin{bmatrix} -0,16 & 0,12 & 0,16 & -0,12 \end{bmatrix} \begin{bmatrix} 0,00 \\ 0,00 \\ -16,87 \\ -66,42 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 3 \\ 4 \end{matrix}$$

q= 5,3 kn

FUERZAS INTERNAS BARRA 6

$$q = \begin{bmatrix} 0,00 & 0,33 & 0,00 & -0,33 \end{bmatrix} \begin{bmatrix} 37,80 \\ -75,91 \\ -16,87 \\ -66,42 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

q= -3,2 kn

5.6 ANÁLISIS COMPARATIVO

Tabla 5.1 Análisis Comparativo de Ejercicio 1

| Resultados | Desplazamientos | | Desplazamientos | | | | Fuerzas Internas | | Promedio del Margen |
|----------------------|-----------------|----------|-----------------|------|-------|------|------------------|--------|---------------------|
| | D1 | D2 | R3 | R4 | R5 | R6 | Barra1 | Barra2 | |
| Calculo del Software | -0,00094 | -0,00396 | 1,5 | 0 | -1,5 | 2 | -1,5 | 2,5 | |
| Calculos Manuales | -0,0009 | -0,004 | 1,50 | 0,00 | -1,50 | 2,00 | -1,50 | 2,50 | |
| MARGEN DE DIFERENCIA | -4,3% | 1,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | -0,406% |

Elaboración Propia

Tabla 5.2 Análisis Comparativo del Ejercicio 2

| Resultados | Desplazamientos | | Desplazamientos | | | | | | Fuerzas Internas | | | Promedio del Margen |
|----------------------|-----------------|----------|-----------------|------|------|-------|--------|------|------------------|--------|--------|---------------------|
| | D1 | D2 | R3 | R4 | R5 | R6 | R7 | R8 | Barra1 | Barra2 | Barra3 | |
| Calculo Software | 0,00556 | -0,02188 | 11,11 | 8,33 | 0 | -8,33 | -11,11 | 0 | -13,89 | 8,33 | 11,11 | |
| Calculos Manuales | 0,0056 | -0,0219 | 11,11 | 8,33 | 0,00 | -8,33 | -11,11 | 0,00 | -13,89 | 8,33 | 11,10 | |
| MARGEN DE DIFERENCIA | 0,7% | 0,1% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | -0,1% | 0,066% |

Elaboración Propia

Tabla 5.3 Análisis Comparativo del Ejercicio 3

| Resultados | Desplazamientos | | | | Reacciones | | | | Fuerzas Internas | | | | | | Promedio del Margen |
|----------------------|-----------------|----------|----------|-----------|------------|------|------|------|------------------|--------|--------|--------|--------|--------|---------------------|
| | D1 | D2 | D3 | D4 | R5 | R6 | R7 | R8 | Barra1 | Barra2 | Barra3 | Barra4 | Barra5 | Barra6 | |
| Calculo Software | 37,79754 | -75,9112 | -16,8691 | -66,42218 | -13,67 | 3,16 | 6,67 | 1,84 | -4,22 | -3,06 | 0 | 9,45 | 5,27 | -3,16 | |
| Calculos Manuales | 37,8 | -75,91 | -16,87 | -66,42 | -13,67 | 3,16 | 6,67 | 1,84 | -4,22 | -3,06 | 0 | 9,40 | 5,30 | -3,20 | |
| MARGEN DE DIFERENCIA | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | -0,5% | 0,6% | 1,3% | 0,094% |

Elaboración Propia

Como se puede apreciar en las Tablas 16, 17 y 18, el software computarizado arroja resultados sorprendentemente similares, con una diferencia promedio de menos del 1.00% en comparación con los cálculos manuales. Esta pequeña discrepancia se debe en parte a las diferencias en la forma en que se redondean los decimales en el cálculo manual, lo cual difiere de la metodología empleada por el software "Análisis de Cerchas". Este hallazgo refuerza la confiabilidad del Método de la Matriz de Rigidez y respalda aún más su eficacia en la implementación de este software.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Luego de completar el desarrollo del software de análisis de cerchas, se pueden extraer las siguientes conclusiones:

- ❖ El software ha demostrado la capacidad de generar resultados precisos y coherentes en el análisis de cerchas en 2D. La comparación con un software comercial establecido respalda la fiabilidad de los cálculos obtenidos, lo que es esencial para tomar decisiones informadas en proyectos de ingeniería.
- ❖ La automatización del análisis de cerchas ha agilizado considerablemente el proceso. Lo que antes requería cálculos manuales complejos y demorados ahora se realiza de manera eficiente y rápida a través del software, lo que ahorra tiempo y recursos valiosos.
- ❖ La interfaz de usuario del software se ha diseñado de manera amigable y accesible. Esto permite a los usuarios, incluso aquellos sin experiencia técnica avanzada, ingresar datos y obtener resultados sin dificultades.
- ❖ Además de su utilidad en la práctica profesional, el software puede ser una herramienta educativa valiosa. Su capacidad para visualizar el proceso de cálculo paso a paso y generar resultados claros es un recurso valioso para estudiantes de ingeniería civil.
- ❖ El éxito del software en la generación de resultados confiables reafirma la eficacia del Método de la Matriz de Rigidez como enfoque para el análisis de cerchas en 2D. Esto tiene implicaciones más amplias para la comunidad de ingenieros y para futuros desarrollos en el campo de la ingeniería estructural.
- ❖ El software puede considerarse como una base sólida para futuras mejoras y expansiones. La incorporación de más funciones y la adaptación para abordar otros tipos de estructuras podrían aumentar aún más su utilidad y alcance.
- ❖ En conjunto, el desarrollo de este software de análisis de cerchas en 2D ha demostrado su potencial para mejorar tanto la eficiencia como la precisión en el

análisis estructural, allanando el camino para soluciones más avanzadas y eficaces en el campo de la ingeniería civil.

- ❖ La interfaz gráfica del programa se ha desarrollado de manera intuitiva y sencilla para facilitar la comprensión del método de análisis de cerchas. Esto asegura que tanto estudiantes como docentes puedan utilizar el programa de forma efectiva y que resulte una herramienta de apoyo valiosa en el proceso de enseñanza y aprendizaje.

RECOMENDACIONES

Este estudio ha generado valiosos hallazgos que pueden guiar el uso efectivo del software tanto para estudiantes como para profesionales en la ingeniería civil. A continuación, se presentan recomendaciones clave basadas en los resultados obtenidos:

- ❖ Se recomienda que los estudiantes utilicen el software después de haber comprendido los principios fundamentales del análisis de cerchas. Esto asegurará que aprovechen al máximo la herramienta y comprendan la lógica detrás de los resultados generados.
- ❖ Los estudiantes deben utilizar el software como una herramienta de refuerzo para sus estudios. Se sugiere que realicen análisis comparativos entre los resultados del software y los cálculos manuales para una mejor comprensión.
- ❖ Recordarles que, aunque el software agiliza el proceso, aún deben comprender los fundamentos del análisis de cerchas. No deben depender completamente del software sin entender los principios subyacentes.

- ❖ Se recomienda utilizar el software como una herramienta de verificación rápida para resultados obtenidos manualmente. Esto es especialmente útil para proyectos más complejos donde la verificación manual puede ser extensa.
- ❖ Para profesionales pueden utilizar el software durante la capacitación para mostrar visualmente cómo se realizan los análisis de cerchas. Esto es particularmente efectivo en entornos educativos y de formación.

- ❖ Independientemente del nivel de experiencia, tanto estudiantes como profesionales deben recordar que el software es una herramienta para complementar su conocimiento y experiencia en análisis estructural.
- ❖ En resumen, estas recomendaciones buscan orientar la utilización eficaz del software de análisis de cerchas en 2D en un contexto académico y profesional. Los usuarios deben recordar que el software es un apoyo valioso para el análisis, pero no reemplaza la comprensión profunda de los principios subyacentes del análisis de cerchas.