

CAPÍTULO I

INTRODUCCIÓN

1.1 El Problema

1.1.1 Planteamiento del Problema

En la práctica contemporánea de la ingeniería civil, el diseño estructural ha experimentado una transición tecnológica irreversible hacia la automatización digital. Si bien la complejidad de las edificaciones modernas exige el uso de herramientas computacionales de alto rendimiento, los métodos tradicionales de verificación manual resultan inviables para el contexto actual. Estos procedimientos no solo son técnicamente insuficientes para capturar con precisión fenómenos complejos como la torsión o la interacción espacial de pórticos, sino que su ejecución demanda una inversión de tiempo excesiva.

Esta limitación operativa conduce al ingeniero a depender de software comercial que, aunque agiliza el trabajo, opera bajo una arquitectura de "caja negra". En este escenario, la lógica de cálculo intermedia permanece oculta, impidiendo al profesional auditar la procedencia matemática de los resultados o diagnosticar el origen de anomalías estructurales. Se identifica, por tanto, un vacío tecnológico: la carencia de una herramienta que sea capaz de analizar la complejidad de estructuras tridimensionales de manera fácil, eficiente y rápida, pero que, a diferencia del software comercial, garantice la transparencia total del proceso matricial para corroborar la convergencia numérica sin sacrificar productividad.

Por ende, resulta imperativo desarrollar una alternativa tecnológica orientada a la validación. Una herramienta que, manteniendo la capacidad de resolver pórticos espaciales mediante el método matricial de rigidez, ofrezca simultáneamente la transparencia documental necesaria. Esto permitiría al ingeniero contrastar y certificar la fiabilidad de los resultados obtenidos en software comercial, fundamentando sus decisiones de diseño en una evidencia matemática clara y trazable, más allá de la reputación del software utilizado.

1.1.2 Formulación del problema

¿De qué manera el desarrollo de una herramienta de código abierto basada en el método matricial contribuye a la auditoría transparente y a la validación de resultados estructurales en edificaciones tridimensionales frente a la opacidad del software comercial?

1.1.3 Sistematización

Para responder al problema, se presenta la solución sistematizada de la siguiente manera:

1. **Integración tecnológica:** Implementación de Python y sus librerías científicas como motor de cálculo para superar la barrera de software cerrado.
2. **Transparencia de cálculo:** Programación del Método Matricial de Rigidez con reporte detallado del proceso de cálculo.
3. **Validación técnica:** Comparación rigurosa de los resultados obtenidos (VORTEX 3D) con software de referencia para certificar la fiabilidad de la herramienta propuesta.

1.2 Justificación

Para responder a la problemática planteada, se propone el desarrollo de VORTEX 3D, escrito en Python, una herramienta de código abierto para el análisis estructural de pórticos tridimensionales. Esta iniciativa se justifica desde tres perspectivas complementarias:

1.2.1 Técnica

El desarrollo de VORTEX 3D se justifica por su capacidad de implementar de manera transparente los fundamentos del análisis matricial de estructuras. Al estar escrito en Python y apoyarse en la librería NumPy, los procedimientos matemáticos se representan de forma directa y verificable, diferenciándose del software de código cerrado. Python, como lenguaje de programación de alto nivel ampliamente utilizado

en la computación científica, aporta claridad sintáctica y la integración de librerías especializadas, lo que permite al ingeniero revisar la formulación de matrices y vectores de fuerza con mayor control y precisión en la verificación estructural.

1.2.2 Académica

Desde el punto de vista pedagógico, VORTEX 3D se constituye en una herramienta didáctica que promueve la comprensión profunda del análisis matricial. Al exponer de forma clara la relación entre los conceptos teóricos y su implementación computacional, permite a estudiantes e investigadores explorar el funcionamiento interno del método, visualizar los pasos intermedios del cálculo y experimentar con modificaciones al código. Esto contribuye a desmitificar el uso del software estructural, fomentando una formación más crítica, autónoma y orientada al entendimiento.

1.2.3 Social

El acceso abierto a herramientas como VORTEX 3D democratiza el conocimiento técnico y fortalece la práctica profesional y docente en contextos donde el acceso a software comercial es limitado. Tanto ingenieros como educadores pueden utilizar esta plataforma para realizar verificaciones independientes, compartir modelos reproducibles y fomentar una cultura de transparencia en el análisis estructural. En regiones donde los recursos tecnológicos son escasos, este tipo de iniciativas contribuye a reducir brechas de acceso y a potenciar el desarrollo local en ingeniería.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar la herramienta VORTEX 3D de código abierto basada en el Método Matricial de Rigidez, con la finalidad de proveer un instrumento de validación transparente que garantice la trazabilidad y auditoría de los resultados estructurales en edificaciones.

1.3.2 Objetivos Específicos

Implementar los fundamentos teóricos del Método Matricial de la Rigidez para elementos de barra (viga, columna) en un entorno tridimensional.

Desarrollar un motor de cálculo matricial en lenguaje Python incorporando algoritmos de rigidez y ensamblaje global para elementos tridimensionales.

Implementar rutinas de distribución geométrica de cargas superficiales para simular la transferencia de fuerzas de losas a barras.

Validar la precisión de la herramienta mediante análisis comparativo de desplazamientos y fuerzas internas respecto a software patrón (CYPE/Ftool).

Desarrollar un módulo generador de reportes que documente la memoria de cálculo detallada para fines de auditoría.

1.4 Alcance y Limitaciones

El alcance del proyecto abarca el desarrollo de una aplicación funcional (VORTEX 3D) capaz de realizar el ciclo completo de análisis estático-lineal para pórticos 3D: Definición de geometría, materiales y secciones, aplicación de cargas (nodales, lineales y superficiales), cálculo de envolventes de esfuerzos.

Se establecen las siguientes limitaciones:

- El análisis implementado es estático-lineal. No se considera el análisis dinámico, el análisis de estabilidad (pandeo) ni el análisis no lineal (material o geométrico).
- El programa se enfoca en elementos tipo barra (pórticos). Aunque se implementa un distribuidor de cargas de losas, las losas no son analizadas por elementos finitos, sino que tributan cargas a las vigas de forma unidireccional o bidireccional.

1.5. Formulación de hipótesis

Se establece la siguiente hipótesis de trabajo:

La implementación de una herramienta de software basada en el método matricial de rigidez y código abierto permite obtener una precisión numérica equivalente al software comercial, ofreciendo simultáneamente la trazabilidad completa de los algoritmos de resolución de las estructuras.

1.6. Identificación y Operacionalización de Variables

Para dar cumplimiento a los objetivos planteados y contrastar la hipótesis de investigación, es necesario traducir los conceptos teóricos en elementos medibles y observables. A continuación, se identifican las variables que intervienen en el estudio y se presenta su operacionalización, estableciendo las dimensiones e indicadores que permitirán evaluar el desempeño de la herramienta desarrollada.

1.6.1. Identificación de Variables

- **Variable Independiente (V.I.):** Herramienta de Software de Código Abierto VORTEX 3D.
 - *Definición Conceptual:* Aplicación computacional desarrollada bajo una arquitectura de software libre, que implementa algoritmos del Método Matricial de Rigidez para el análisis de estructuras tridimensionales, permitiendo el acceso irrestricto a su código fuente y lógica de cálculo.
- **Variable Dependiente (V.D.):** Validación y Auditoría del Cálculo Estructural.
 - *Definición Conceptual:* Capacidad técnica para verificar la exactitud de los resultados numéricos (desplazamientos y fuerzas internas) y rastrear la secuencia lógica del proceso matemático utilizado para obtenerlos, garantizando la fiabilidad del diseño.

1.6.2. Matriz de Operacionalización de Variables

En la **Tabla 1** se detalla la descomposición de las variables en sus dimensiones físicas y lógicas, así como los indicadores específicos que servirán como instrumentos de medición durante la fase de resultados.

Tabla 1: Operacionalización de Variables

Variable	Definición Operacional	Dimensiones	Indicadores	Escala
Independiente Herramienta VORTEX 3D (Código Abierto)	Desarrollo e implementación de módulos de programación en lenguaje Python orientados al análisis matricial y la gestión de datos estructurales.	1. Motor de Cálculo Matricial	• Algoritmos de ensamblaje de la Matriz de Rigidez Global [K]. • Rutinas de resolución del sistema de ecuaciones $[F]=[K][d]$.	Nominal (Dicotómica: <i>Cumple / No Cumple</i>)
		2. Gestión de Cargas	• Algoritmos de distribución de cargas superficiales. • Vectores de fuerzas nodales equivalentes.	Nominal (Dicotómica: <i>Cumple / No Cumple</i>)
		3. Transparencia Documental	• Módulo generador de reportes. • Acceso al código fuente.	Nominal (Dicotómica: <i>Presente / Ausente</i>)

Variable	Definición Operacional	Dimensiones	Indicadores	Escala
Dependiente Validación y Auditoría Estructural	Proceso de verificación mediante el cual se compara el desempeño de la herramienta propuesta frente a un software patrón y se examina la trazabilidad de los datos.	1. Precisión Numérica	• Porcentaje de Error Relativo (%) en desplazamientos nodales comparado con CYPE/Ftool. • Convergencia de fuerzas internas (Momentos y Cortantes).	De Razón (Cuantitativa Continua)
		2. Trazabilidad (Auditoría)	• Disponibilidad de matrices intermedias en los reportes. • Verificación "paso a paso" del flujo de cálculo.	Nominal (Cualitativa: Verificable / No Verificable)

CAPÍTULO II

FUNDAMENTO TEÓRICO

Para desarrollar una herramienta computacional de análisis estructural, deben abordarse dos campos teóricos, el primero es el del análisis de estructuras por método de rigidez directo, el segundo es el de ciencias de computación. Es así que, en el presente capítulo, se detalla la fundamentación teórica de VORTEX 3D.

2.1. Análisis Estructural Computacional

2.1.1 Contexto General y Evolución

El análisis estructural ha evolucionado partiendo de métodos manuales y geometrías simples, hasta llegar a métodos numéricos con geometrías complejas. Fue la necesidad de calcular estructuras cada vez más complejas, sometidas a escenarios de carga igual de elaborados lo que nos lleva al análisis estructural moderno. Esta transición fue facilitada por el desarrollo y masificación de la computación a partir de la segunda mitad del siglo XX

2.1.2 Relevancia de los Métodos Numéricos en Ingeniería Civil

El método matricial de rigidez, un método numérico, es ampliamente relevante por su capacidad de sistematizar y automatizar el cálculo de estructuras reticulares como los pórticos. Este es la base del análisis estructural computacional, teniendo un enfoque algorítmico capaz de resolver de forma eficiente, sistemas hiperestáticos complejos.

2.1.3 Rol del Software en el Análisis y Diseño Estructural Moderno

Considerando software comercial (como CYPECAD, SAP2000, ETABS) y software de código abierto (como VORTEX 3D). El software no reemplaza el criterio del ingeniero, sino que funciona como una herramienta de cálculo estructural intensivo que automatiza la resolución de los sistemas de ecuaciones matriciales. El papel del software es, por tanto, ejecutar el análisis (cálculo de desplazamientos y esfuerzos) y, en muchos casos, asistir en el diseño (verificación y dimensionamiento de elementos según normativas), permitiendo al profesional centrarse en la idealización del modelo y la interpretación de los resultados.

2.2 El Método Matricial de Rigidez para Estructuras Aporticadas en 3D

El Método Matricial de Rigidez es un procedimiento algorítmico que representa un estándar para el análisis de estructuras aporticadas en dos y tres dimensiones formadas por elementos lineales (unidimensionales). Este es el fundamento sobre el cual se construye el motor de cálculo de VORTEX 3D.

El método sigue una secuencia algorítmica definida, que se detalla del punto 2.2.1 al 2.2.12. Esta secuencia permite ensamblar las propiedades de los elementos individuales en un sistema global utilizado para establecer el equilibrio de la estructura, este sistema se resuelve para los desplazamientos nodales, y se realiza el post-proceso para obtener las sollicitaciones internas y reacciones.

2.2.1 Idealización y Discretización de la Estructura

El primer paso del método consiste en la idealización de la estructura física. Esta se representa como un conjunto de elementos unidimensionales conectados en sus nodos.

En el método, el ensamblaje de la matriz de rigidez global a partir de las matrices locales o elementales, “dependerá sólo de la topología o modo en que las barras están conectadas entre sí”¹.

2.2.2 Grados de Libertad Nodales en 3D

Para el análisis espacial tridimensional, cada nodo posee "seis grados de libertad, tres traslaciones ($\delta_x, \delta_y, \delta_z$) y tres rotaciones ($\theta_x, \theta_y, \theta_z$)”² respecto a un sistema de coordenadas global.

Es importante distinguir entre GDL totales “teóricos” y GDL totales “reales” donde los totales teóricos definen el tamaño de la matriz de rigidez global, y los grados de libertad reales, es decir, los grados de libertad no restringidos, definen el tamaño de la matriz

¹ David Sánchez y Ramón González, *Cálculo de elementos estructurales*, 1ra ed. (Oficina de Publicacions Acadèmiques Digitals de la UPC, 2011), 13.

² Sánchez y González, *Cálculo de elementos estructurales*, 22.

de rigidez global reducida. Según Kardestuncer, “El grado de libertad de la estructura, es el número total de componentes de las deflexiones desconocidas de los nodos libres.”³.

2.2.3 Sistemas de Coordenadas Globales y Locales

La formulación matricial requiere la definición de dos sistemas ortogonales de referencia:

1. **Sistema Global (X, Y, Z):** Un sistema cartesiano único para definir la geometría de la estructura, las fuerzas externas y los desplazamientos de los nodos. Es el sistema de referencia amigable con el usuario y que sirve para el ensamblaje y la resolución de la estructura.

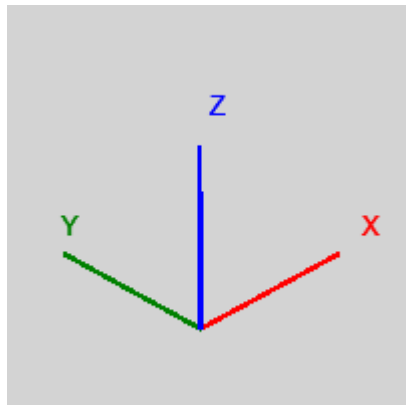


Figura 1: Sistema de Coordenadas Global

Nota. Elaboración propia, captura de pantalla de VORTEX 3D

³ Hayrettin Kardestuncer, *Introducción al análisis estructural con matrices* (McGRAW-HILL, 1975), 14.

2. **Sistema Local (x' , y' , z'):** Un sistema propio de cada elemento, donde el eje x' se alinea con el eje longitudinal del elemento y su sentido va del nodo inicial “i” al nodo final “j”, luego, por la regla de la mano derecha se define que y' es el eje lateral y z' es el eje vertical.

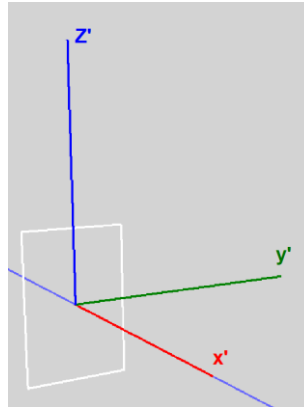


Figura 2: Sistema de Coordenadas Local

Nota. Elaboración propia, captura de pantalla de VORTEX 3D

2.2.4 Matriz de Rigidez del Elemento de Barra 3D (Coordenadas Locales)

En el sistema local, la relación entre las fuerzas nodales (f') y los desplazamientos nodales (d') de un elemento se establece por medio de la matriz de rigidez elemental local (k')

Para un elemento de pórtico 3D con tres traslaciones y tres rotaciones por nodo, “al considerar las dos secciones extremas de cada barra, la matriz de rigidez elemental tiene, en una matriz de $12 \times 12 (= 2 \cdot 6 \times 2 \cdot 6)$ ”⁴. La matriz de rigidez “Elemental” o “Local” para satisfacer $f' = k' \cdot d'$ tendrá la forma:

⁴ Sánchez y González, *Cálculo de elementos estructurales*, 22.

D_y : Rigidez mixta a corte y flexión

- Rigideces en el plano lateral (x'-y'):

B_z : Rigidez a Cortante

C_z : Rigidez a Flexión

D_z : Rigidez mixta a corte y flexión

2.2.4.1 Rigidez Axial (Componente A)

La rigidez axial del elemento representa su resistencia a la deformación longitudinal sea de alargamiento o encogimiento. Corresponde al componente A en la matriz de rigidez local de la *Figura 3*.

Vincula los desplazamientos axiales con los grados de libertad 1 y 7, se expresa de la siguiente forma

$$k_{ax} = \frac{EA}{L}$$

Donde

k_{ax} : Rigidez Axial.

E : Módulo de Young o Módulo de Elasticidad.

A : Área de la sección del elemento.

L : Longitud del elemento.

2.2.4.2 Rigidez a Torsión

La rigidez torsional del elemento representa su resistencia a la rotación sobre su propio eje longitudinal (x'). Este componente vincula el momento torsor aplicado en los nodos con las rotaciones torsionales en los mismos nodos.

Logan⁵ presenta la deducción de la expresión de la rigidez torsional de manera análoga a la de la rigidez axial, sustituyendo E (módulo de elasticidad) por G (módulo de rigidez al corte) y A (área de la sección) por J (constante torsional de la sección de Saint Venant). Expresándose como sigue:

$$k_t = \frac{GJ}{L}$$

Donde:

k_t : Rigidez a torsión.

G : Módulo de rigidez al corte.

J : Constante torsional de la sección.

2.2.4.3 Rigideces a Flexión y Cortante

Para las rigideces a flexión y cortante se consideran dos escenarios, el ensamblaje de la matriz por teoría de Euler-Bernoulli y por teoría de Timoshenko que considera la deformación por cortante.

Euler-Bernoulli

- Rigidez a Cortante: La rigidez a cortante se define como la "relación T_c/δ_c entre la fuerza vertical T_c aplicada en un extremo y el desplazamiento subsiguiente δ_c del mismo extremo"⁶, bajo un esquema de viga doblemente empotrada sometida a un cortante T_c . Tras resolver el problema hiperestático mediante la ecuación de la elástica, los autores obtienen la fórmula de rigidez:

$$k_{cort} = \frac{12EI}{L^3}$$

⁵ Daryl Logan, *A First Course in the Finite Element Method*, Cuarta Edición (Thompson Canada Limited, 2007), 241.

⁶ Sánchez y González, *Cálculo de elementos estructurales*, 17.

- Rigidez a Flexión: Sánchez y González⁷ definen la rigidez a flexión como la "relación M_f/θ_f entre el momento extremo M_f y el giro de la sección θ_f del mismo extremo", demostrando mediante la resolución de la ecuación de la elástica que esta relación resulta en:

$$k_{flex} = \frac{4EI}{L}$$

- Rigidez mixta a corte y flexión: Esta se define como la “relación M_E/δ entre el momento de empotramiento M_E cuando se aplica una fuerza vertical T en el extremo que produzca un desplazamiento δ de dicho extremo”⁸. Resultando en la siguiente expresión:

$$k_{cf} = \frac{6EI}{L^2}$$

Timoshenko

La teoría de Euler-Bernoulli es correcta, sin embargo, contiene una simplificación importante “despreciando la deformación transversal por cortante”⁹, cosa que Timoshenko corrige. Para elementos “gruesos” es imperativo considerar este tipo de deformaciones.

La base conceptual de esta teoría, presentada por Logan, radica en que “la deformación por cortante (debida a la fuerza cortante V), ahora está incluida.”¹⁰ A diferencia de Euler-Bernoulli, en la viga de Timoshenko la sección transversal “permanece plana, pero ya no es perpendicular al eje neutro”.

La teoría se construye sobre la relación entre el momento flector y la curvatura:

$$M_{(\hat{x})} = EI \frac{d\hat{\phi}(\hat{x})}{d(\hat{x})}$$

y la relación entre la fuerza cortante y la deformación por cortante:

⁷ Sánchez y González, *Cálculo de elementos estructurales*, 15.

⁸ Sánchez y González, *Cálculo de elementos estructurales*, 18.

⁹ Logan, *A First Course in the Finite Element Method*, 161.

¹⁰ Logan, *A First Course in the Finite Element Method*, 158.

$$V_{(\hat{x})} = k_s AG \beta(\hat{x})$$

explicadas en ¹¹.

Logan desarrolla la deducción completa para obtener la matriz de rigidez del elemento que incorpora tanto deformación flexional como deformación por cortante. En este proceso, presenta la submatriz flexo-cortante [figura 4] “para ver más claramente el efecto del factor de corrección por cortante,”¹² introduciendo un factor adimensional de corrección por cortante ϕ descrito por la siguiente expresión:

$$\phi = \frac{12EI}{A_v GL^2}$$

Donde:

A_v : Área resistente al corte, dada por: $A_v = k_v A$, donde $k_v = \frac{5}{6}$ para secciones rectangulares.

$$\hat{k} = \frac{EI}{L^3(1 + \phi)} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & (4 + \phi)L^2 & -6L & (2 - \phi)L^2 \\ -12 & -6L & 12 & -6L \\ 6L & (2 - \phi)L^2 & -6L & (4 + \phi)L^2 \end{bmatrix} \quad (4.1.15o)$$

Figura 4: Matriz de Rigidez a Flexión que Contempla la Deformación por Cortante

Nota. Figura tomada de “A First Course in the Finite Element Method”, por D. Logan, 2007, Thompson Canada Limited.

A partir de los términos de la matriz de la **figura 4**, se extraen los coeficientes de rigidez modificados:

Rigidez a Cortante:

¹¹ Logan, *A First Course in the Finite Element Method*, 159.

¹² Logan, *A First Course in the Finite Element Method*, 161.

$$k_{cort} = \frac{12EI_y}{L^3(1 + \phi)}$$

Rigidez a Flexión:

$$C = k_{flex} = \frac{(4 + \phi)EI_y}{L(1 + \phi)}$$

$$\text{Además, el componente "C/2"} = \frac{(2-\phi)EI_y}{L(1+\phi)}$$

Rigidez mixta a corte y flexión:

$$k_{cf} = \frac{6EI_y}{L^2(1 + \phi)}$$

En caso de establecer el factor de corrección a corte con valor cero, entonces la matriz resultante será exactamente la misma que la descrita por Euler-Bernoulli.

2.2.5 Formulación de la Matriz de Transformación de Coordenadas (T)

Para unificar el análisis, es indispensable una matriz de transformación de coordenadas (T) que relacione las componentes de un vector en el sistema local con las componentes del mismo vector en el sistema global. Esta matriz de rotación, de 12×12 , se construye a partir de los cosenos directores que definen la orientación de la barra en el espacio y permite “transformar fuerzas y vectores de desplazamiento, y por extensión lógica, matrices de rigidez, de coordenadas locales a globales y viceversa”¹³. Esta operación es fundamental para el correcto ensamblaje de la matriz de rigidez global de la estructura, dado que ambos sistemas de coordenadas son ortogonales y las acciones pueden representarse indistintamente en cualquiera de ellos.

2.2.5.1 La Matriz de Rotación (R)

La transformación fundamental es realizada a nivel vectorial por medio de la matriz de rotación R (3×3) mostrada a continuación. Esta correlaciona componentes de un vector en el sistema global con sus componentes en el sistema local.

¹³ William McGuire y col., *Matrix Structural Analysis*, 2nd Edition (John Wiley & Sons, 2000), 94.

McGuire y col., definen esta matriz (a la que llaman “ γ ”), de la siguiente forma: “La matriz $[\gamma]$, compuesta por los nueve cosenos directores que relacionan el sistema de coordenadas local con el global, es una matriz de rotación”¹⁴. Pues las filas de esta matriz R , son en realidad, los vectores unitarios de los ejes locales x' , y' y z' , expresados en coordenadas globales.

$$R = \begin{bmatrix} vx \\ vy \\ vz \end{bmatrix} = \begin{bmatrix} X_{x'} & Y_{x'} & Z_{x'} \\ X_{y'} & Y_{y'} & Z_{y'} \\ X_{z'} & Y_{z'} & Z_{z'} \end{bmatrix}$$

Donde:

R : Matriz de rotación.

vx : Vector unitario que representa el eje x' en coordenadas globales.

vy : Vector unitario que representa el eje y' en coordenadas globales.

vz : Vector unitario que representa el eje z' en coordenadas globales.

X, Y, Z : Coordenadas globales de cada vector unitario.

2.2.5.2 Matriz de Transformación (T)

La transformación de los 12 grados de libertad del elemento barra unidimensional (3 traslaciones y 3 rotaciones en cada nudo) se realiza a partir de la matriz de rotación R (3×3), la cual establece la orientación el sistema local respecto al global. Ya que las fuerzas, traslaciones, momentos y rotaciones se transforman de manera idéntica pero independiente en cada nudo. La matriz de transformación se reparte de la siguiente forma:

¹⁴ McGuire y col., *Matrix Structural Analysis*, 96.

$$\begin{Bmatrix} F_{x'1} \\ F_{y'1} \\ F_{z'1} \\ M_{x'1} \\ M_{y'1} \\ M_{z'1} \\ F_{x'2} \\ F_{y'2} \\ F_{z'2} \\ M_{x'2} \\ M_{y'2} \\ M_{z'2} \end{Bmatrix} = \begin{bmatrix} [\gamma] & 0 & 0 & 0 \\ 0 & [\gamma] & 0 & 0 \\ 0 & 0 & [\gamma] & 0 \\ 0 & 0 & 0 & [\gamma] \end{bmatrix} \begin{Bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ F_{x2} \\ F_{y2} \\ F_{z2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \end{Bmatrix} \quad (5.7)$$

Figura 5: Matriz de Transformación aplicada a Fuerzas y Momentos

Nota. Figura tomada de “Matrix Structural Analysis”, por McGuire, W., Gallagher, R., & Ziemian, R., 2000, John Wiley & Sons.

En la **figura 5**, la matriz γ es para nosotros la matriz de rotación R y como se puede ver, se aplica por bloques a cada conjunto de grados de libertad en los dos nodos del elemento. Cada nodo presenta 6 grados de libertad (3 traslaciones asociadas a fuerzas y 3 rotaciones asociadas a momentos). Por ello, la matriz de transformación T adquiere una forma diagonal por bloques 3×3 y tiene dimensión 12×12 , compatible con la matriz de rigidez local del elemento barra, que también considera 12 grados de libertad distribuidos entre sus extremos.

Finalmente tenemos:

$$T = \begin{bmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{bmatrix}$$

Donde:

T : Matriz de transformación.

R : Matriz de rotación.

0 : Matriz de ceros.

2.2.6 Matriz de Rigidez del Elemento en Coordenadas Globales

Utilizando la matriz de transformación T recientemente definida, es posible transformar la matriz de rigidez del elemento en coordenadas locales k' al sistema global de coordenadas. La matriz de rigidez global del elemento k se obtiene por medio de la transformación de congruencia¹⁵:

$$k = T^T k' T$$

McGuire y col., afirman: “Esta es la ecuación más comúnmente utilizada para encontrar la matriz de rigidez en coordenadas globales una vez que uno la ha formulado en coordenadas locales y conoce la relación geométrica entre los dos sistemas de ejes.”¹⁶

2.2.7 Ensamblaje de la Matriz de Rigidez Global

El Método Matricial de Rigidez, inicialmente calcula matrices de rigidez (k) para cada elemento y de forma aislada, cada una de estas está relacionada únicamente con las solicitaciones y deformaciones de sus nodos, sin embargo, para representar el comportamiento de la estructura en su totalidad, estas matrices deben ensamblarse en una matriz de rigidez global (K)

Este ensamblaje se realiza mediante la superposición de las contribuciones de rigidez de cada elemento. Como señala Logan: “La matriz K se construye simplemente sumando directamente los términos asociados a los grados de libertad en $k(1)$ y $k(2)$ en sus posiciones correspondientes e idénticas dentro de K ”¹⁷.

Ya que cada matriz elemental k está formulada en términos de grados de libertad (GDL) locales, no pueden sumarse directamente, por lo que se mapean los GDL locales de cada elemento a los GDL globales de la estructura. Para esto, se expande cada matriz k al orden de la matriz global K , añadiendo ceros en las posiciones correspondientes a los GDL que no estén asociados al elemento.

¹⁵ Transformación de Congruencia: Operación matemática de la forma $B = P^T A P$

¹⁶ McGuire y col., *Matrix Structural Analysis*, 98.

¹⁷ Logan, *A First Course in the Finite Element Method*, 38.

Una vez alineadas, las matrices elementales expandidas se suman término a término. Si varios elementos comparten un nodo, sus contribuciones se acumulan en el mismo GDL global. Como explican Sánchez & González, esto implica “sumas matriciales de los bloques cuando un nodo es común a varias barras”¹⁸. Desde el punto de vista físico, esta superposición refleja que “cuando en un nodo o concurrencia de barras inciden más barras, será tanto más difícil provocar desplazamientos en él, ya que habrá que vencer las rigideces de todas las barras a deformarse para lograr desplazarlo”¹⁹

2.2.8 Vector de Cargas Nodales Globales

En el análisis estructural, es normal considerar cargas aplicadas al elemento “entre nodos”, un enfoque riguroso y adecuado para trabajar con estas acciones en elementos unidimensionales, superior a la creación de nodos artificiales, “emplea los conceptos relacionados de restricción ficticia de nodos, fuerzas en los extremos fijos y cargas nodales equivalentes”²⁰.

Este método modifica la relación fundamental de rigidez de la siguiente manera:

$$P = k d + P^F$$

Donde:

P : Vector de cargas externas aplicadas directamente en los nodos.

k : Matriz de rigidez elemental global.

d : Vector de desplazamientos.

P^F : Vector de Fuerzas de Empotramiento Perfecto (FEPs) ensambladas en los grados de libertad globales. Este representa las fuerzas de reacción que se generarían si los nodos estuvieran restringidos contra todo desplazamiento.

¹⁸ *Cálculo de elementos estructurales*, 34.

¹⁹ Sánchez y González, *Cálculo de elementos estructurales*, 35.

²⁰ McGuire y col., *Matrix Structural Analysis*, 108.

Para resolver los desplazamientos, la ecuación se reordena. “La transferencia del vector P^F ... al lado izquierdo... es el equivalente algebraico de aplicar las fuerzas de empotramiento perfecto invertidas como cargas nodales”²¹ resultando en:

$$k d = P - P^F$$

Donde $P - P^F$ representará al vector de cargas nodales globales (efectivas o equivalentes) F .

2.2.8.1 Cargas Nodales Directas

Las acciones (fuerzas P_x, P_y, P_z o momentos M_x, M_y, M_z) que se aplican directamente sobre un nodo, se incorporan directamente en el vector global P en las posiciones que coincidan con los grados de libertad del nodo en el que actúan.

2.2.8.2 Cargas sobre Elementos

Las cargas actuantes a lo largo de un elemento se manejan calculando el vector de Fuerzas de Empotramiento Perfecto (FEPs) en coordenadas locales (P^F). Este vector representa las reacciones (fuerzas y momentos) que se generarían en los nodos, si el elemento estuviese perfectamente empotrado mientras se aplican las cargas, esta es la “restricción ficticia de nodos” de la que habla McGuire.

A continuación se distingue entre la forma de obtener FEPs para cargas uniformes distribuidas y para casos generales (distribuidas parciales y triangulares parciales que son utilizadas para formar cargas triangulares y trapezoidales)

Cargas Uniformes Distribuidas

Para calcular los FEPs de este tipo de cargas, se usan fórmulas tabuladas y de dominio popular, en este caso, para obtener los “valores de las reacciones en los extremos y los momentos de empotramiento son $\frac{qL}{2}$ y $\frac{qL^2}{12}$, respectivamente”²².

²¹ McGuire y col., *Matrix Structural Analysis*, 112.

²² McGuire y col., *Matrix Structural Analysis*, 110.

Casos Generales (Método de Equivalencia de Trabajo)

En escenarios de carga compleja se requiere un enfoque general como el Método de Equivalencia de Trabajo, que reemplaza la carga distribuida por un conjunto de cargas nodales energéticamente equivalentes.

Como señala Hutton, “El objetivo aquí es determinar las cargas nodales equivalentes para que el trabajo sea el mismo que el realizado por cargas nodales directas”²³.

Esta equivalencia conduce a la formulación integral de cada componente del vector de cargas nodales equivalentes F_i ²⁴:

$$F_i = \int_0^L q(x) \cdot N_i(x) dx$$

Donde:

$q(x)$: Función de carga distribuida general.

$N_i(x)$: Función de forma asociada al grado de libertad correspondiente.

Para elementos de viga en flexión, se emplean cuatro funciones de forma polinómicas “Hermitianas”, independientes del tipo de carga. Estas relacionan los desplazamientos y rotaciones nodales $\delta_i, \theta_i, \delta_j, \theta_j$ con el campo de desplazamientos transversales $\delta(x)$:

- $N_1(x) = 1 - 3\left(\frac{x}{L}\right)^2 + 2\left(\frac{x}{L}\right)^3$, asociada a δ_i
- $N_2(x) = x - 2\frac{x^2}{L} + \frac{x^3}{L^2}$, asociada a θ_i
- $N_3(x) = 3\left(\frac{x}{L}\right)^2 - 2\left(\frac{x}{L}\right)^3$, asociada a δ_j
- $N_4(x) = \frac{x^3}{L^2} - \frac{x^2}{L}$, asociada a θ_j

La versatilidad del método radica en cómo se define $q(x)$ y los límites de integración [a,b]:

²³ Hutton, *Fundamentals of Finite Element Analysis*, International Edition (McGRAW-HILL, 2004), 106.

²⁴ Hutton, *Fundamentals of Finite Element Analysis*, ec. 4.54-4.57.

- **Carga uniforme parcial:** $q(x) = w$ constante en $[a,b]$

$$F_i = \int_a^b w \cdot N_i(x) dx$$

- **Carga triangular parcial:** $q(x) = k(x - a)$, con $k = \frac{w_{pico}}{b-a}$

$$F_i = \int_a^b k(x - a) \cdot N_i(x) dx$$

La resolución analítica de estas integrales para cada $N_i(x)$ proporciona las fórmulas exactas de las Fuerzas de Empotramiento Perfecto (FEPs) en casos de carga generalizada.

2.2.8.3 Ensamblaje Final

El vector de FEPs local de cada elemento $P^{F'}$, debe ser transformado a coordenadas globales usando la matriz de transformación transpuesta definida anteriormente como T :

$$P^F = T^T \cdot P^{F'}$$

Finalmente restamos este vector de FEPs global del vector de cargas nodales directa P para obtener el valor del vector de fuerzas nodales globales efectivas F

$$F = P - P^F$$

2.2.9 Sistema Fundamental de Ecuaciones

El equilibrio estático de la estructura completa, se expresa mediante el sistema fundamental de ecuaciones lineales:

$$F = K \cdot d$$

Considerando que todos sus componentes son aquellos referidos a la estructura completa, es decir, que contempla todos los grados de libertad existentes en la misma y ordenados.

Donde:

F : Vector de fuerzas nodales globales efectivas.

K : Matriz de rigidez global ensamblada.

d : Vector de desplazamientos nodales globales (incógnitas)

Antes de aplicar apoyos, esta matriz K es singular, ya que no se han restringido los movimientos de cuerpo rígido.

2.2.10 Imposición de condiciones de borde (apoyos)

Las condiciones de borde (apoyos estructurales) son restricciones que imponen valores de desplazamiento conocidos (generalmente nulos) a grados de libertad (GDL) específicos del sistema.

El sistema global de rigidez, $[F] = [K] \cdot [d]$, en su forma original, es matemáticamente singular. Esto se debe a que la estructura, al no estar restringida, es susceptible a movimientos de cuerpo rígido (traslaciones o rotaciones sin deformación), lo que impide obtener una solución matemática única.

La imposición de apoyos elimina esta singularidad. Para incorporar estas condiciones, se aplica un procedimiento de partición y eliminación, que consiste en:

- Identificar las filas y columnas de la matriz global K asociadas a los GDL con desplazamiento conocido (restringidos).
- Eliminar dichas filas y columnas del sistema de ecuaciones.

Este proceso genera un sistema reducido:

$$F_{reducido} = K_{reducida} \cdot d_{desconocidos}$$

Cuya matriz es estable (no singular) y, por tanto, matemáticamente resoluble.

2.2.11 Resolución del Sistema y Obtención de Desplazamientos [d]

Una vez establecido el sistema reducido, que contiene únicamente los grados de libertad libres de la estructura, se procede a resolverlo para obtener el vector de desplazamientos nodales desconocidos $d_{desconocidos}$.

Este vector constituye la incógnita principal del método de la rigidez. Su solución permite conocer la respuesta deformada de la estructura bajo las cargas aplicadas.

Finalmente, los desplazamientos desconocidos se combinan con los valores conocidos (generalmente nulos) impuestos por los apoyos, reconstruyendo así el vector completo de desplazamientos globales d del sistema.

2.2.12 Cálculo de Reacciones y Esfuerzos Internos (Post-proceso)

Una vez resuelto el sistema reducido y obtenido el vector de desplazamientos nodales desconocidos $d_{desconocidos}$, se reconstruye el vector global de desplazamientos d , que incluye tanto los valores calculados como los desplazamientos prescritos (generalmente nulos) en los grados de libertad restringidos por los apoyos.

Con el vector d completo, se inicia la fase de postproceso, en la que se determinan las fuerzas de reacción en los apoyos y los esfuerzos internos en los elementos estructurales.

2.2.12.1 Cálculo de Reacciones

Las reacciones se obtienen a partir de la ecuación global de equilibrio:

$$F = K \cdot d$$

El vector F representa las fuerzas internas inducidas por los desplazamientos en todos los grados de libertad del sistema. En los GDL libres, estas fuerzas deben coincidir con las cargas externas aplicadas directamente en los nodos, verificando así el equilibrio. En los GDL restringidos, las componentes de F corresponden a las reacciones R generadas por los apoyos.

2.2.12.2 Cálculo de Fuerzas Internas Locales

El cálculo de esfuerzos internos (axiales, cortantes y momentos) se realiza individualmente para cada elemento, siguiendo estos pasos:

- **Extracción de desplazamientos del elemento:**

Se extrae el sub vector de desplazamientos nodales del elemento d_e a partir del vector global d .

- **Transformación a coordenadas locales:**

Los desplazamientos globales se transforman al sistema de coordenadas locales del elemento mediante la matriz de transformación T :

$$d'_e = T \cdot d_e$$

- **Cálculo de esfuerzos internos:**

Los esfuerzos en los extremos del elemento se obtienen aplicando la ecuación de rigidez local, que considera los desplazamientos nodales locales y las Fuerzas de Empotramiento Perfecto (FEPs) previamente aplicadas:

$$f' = k' \cdot d'_e + f_{FEP}'$$

El vector f' contiene las fuerzas internas en los nudos i y j del elemento, y constituye la base para la construcción de los diagramas de esfuerzos.

Donde:

f' : Vector de fuerzas internas finales (fuerzas axiales, cortantes y momentos) en los extremos del elemento, expresado en el sistema de coordenadas locales del mismo.

k' : Es la matriz de rigidez local del elemento, que relaciona fuerzas y desplazamientos en el sistema local.

d'_e : Es el vector de desplazamientos nodales del elemento en coordenadas locales.

f_{FEP}' : Es el vector de **Fuerzas de Empotramiento Perfecto (Acciones)** en el sistema local. Representa las fuerzas aplicadas (acciones equivalentes a la carga) que se

generan en los nudos del elemento. Este vector es el negativo de las reacciones de empotramiento (es decir, $f'_{FEP(acción)} = -f_{FEP(reacción)}$) y se suma en esta formulación.

2.3 Fundamentos Computacionales del Motor de Análisis Estructural

El desarrollo de software contemporáneo para ingeniería, especialmente en el ámbito de la simulación y el análisis numérico, se fundamenta en una separación crucial: la distinción entre el frontend y el backend.

El frontend, también conocido como interfaz gráfica de usuario (GUI), comprende todos los elementos visuales e interactivos que el usuario manipula. Su propósito es facilitar la entrada de datos y presentar los resultados de manera clara e intuitiva. Es la capa que “conecta e interactúa con los usuarios”²⁵, y de su diseño depende la percepción funcional del sistema.

Por su parte, el backend constituye el núcleo funcional del software. Es responsable de procesar las solicitudes del usuario, ejecutar los algoritmos computacionales y gestionar la lógica del análisis estructural. En este contexto, el backend recibe la definición del modelo (nudos, barras, cargas), realiza el ensamblaje matricial, resuelve el sistema de ecuaciones y devuelve los resultados numéricos. Es decir, el backend “es todo el conjunto del desarrollo que se encarga de que [el sistema] funcione... pero que es totalmente invisible para el usuario”²⁶.

Separar estas capas ofrece ventajas clave señaladas por xenonfactory²⁷. Por ejemplo, “se trata de un entorno mucho más escalable”, ya que cada parte puede evolucionar con recursos específicos. Además, “las migraciones y actualizaciones son más sencillas”, dado que el núcleo de cálculo puede mantenerse estable mientras la interfaz se adapta a nuevas necesidades o plataformas.

²⁵ xenonfactory, «Ventajas de separar backend y frontend en tu proyecto | Xenon Factory», *Diseño web en Asturias desde 2007*, 16 de diciembre de 2020, <https://xenonfactory.es/blog/conoce-las-ventajas-de-separar-backend-y-frontend-en-tu-proyecto/>.

²⁶ xenonfactory, «Ventajas de separar backend y frontend en tu proyecto | Xenon Factory».

²⁷ «Ventajas de separar backend y frontend en tu proyecto | Xenon Factory».

De acuerdo con esta delimitación, esta sección se centrará exclusivamente en los fundamentos computacionales que estructuran el backend del motor de análisis. Se excluyen las tecnologías de visualización para abordar únicamente los paradigmas de programación y las librerías numéricas que permiten implementar de forma eficiente el método matricial de rigidez.

2.3.1 Paradigmas de Programación: Fundamento del Motor Computacional

Antes de seleccionar un lenguaje, la arquitectura de un motor de software se define por el paradigma de programación adoptado. Este paradigma constituye un modelo conceptual que orienta la forma en que se estructura y se implementa un programa. Tal como lo describe AcademiaLab, “los paradigmas de programación son una forma de clasificar los lenguajes de programación en función de sus características”²⁸. Esta clasificación no solo afecta la sintaxis o el estilo, sino que determina el modelo de ejecución, el orden de operaciones, y la manera en que se organiza el código.

La elección del paradigma es una decisión arquitectónica fundamental, pues define la principal herramienta cognitiva que utiliza el desarrollador para traducir un problema del mundo real (como un sistema estructural físico) en un modelo computacional. En este sentido, el paradigma actúa como una abstracción que filtra y estructura la realidad técnica, permitiendo que el software represente, simule o interactúe con sistemas complejos de forma coherente y reproducible.

El paradigma de programación adoptado en este trabajo es el de la programación orientada a objetos, cuya estructura y fundamentos se desarrollan en el apartado siguiente.

2.3.1.1. Programación Orientada a Objetos (POO) en la Modelación Estructural

La Programación Orientada a Objetos (POO) es un paradigma que organiza el software en torno a entidades llamadas *objetos*, cada uno con sus propios datos y

²⁸ «Paradigmas de la programación – AcademiaLab», 2024, <https://academia-lab.com/enciclopedia/paradigmas-de-la-programacion/>.

comportamientos. En lugar de centrarse en funciones aisladas, este enfoque propone que el sistema se construya como un conjunto de objetos que interactúan entre sí para cumplir tareas específicas.

Según Sommerville, “Un sistema orientado a objetos está compuesto por objetos que interactúan entre sí, mantienen su propio estado local y ofrecen operaciones sobre ese estado.”²⁹. Esto implica que cada objeto gestiona su propia información y ofrece métodos para manipularla, sin exponer directamente cómo está organizada internamente.

Uno de los principios clave de este paradigma es el **encapsulamiento**, que puede entenderse como el concepto de “caja negra”: otros componentes del sistema pueden utilizar un objeto sin necesidad de conocer su funcionamiento interno. Sommerville lo describe así: “La representación del estado es privada y no puede ser accedida directamente desde fuera del objeto.”³⁰.

En el contexto de la modelación estructural, este paradigma permite representar entidades físicas (como nodos, elementos o materiales) como objetos con atributos propios y operaciones específicas. Estas entidades se gestionan mediante una clase central que organiza la información en colecciones internas, utilizando estructuras de datos como diccionarios. La interacción con estas colecciones se realiza a través de una interfaz pública de métodos, lo que garantiza la modularidad y facilita el mantenimiento del sistema.

Gracias a esta organización, el software puede evolucionar sin comprometer su estructura general: cambios en la forma de almacenar o procesar los datos pueden implementarse localmente, sin afectar los demás componentes que interactúan con el modelo.

²⁹ Sommerville, *Software Engineering*, 10th Edition (Pearson Education Limited, 2016), 198.

³⁰ Sommerville, *Software Engineering*, 198.

2.3.2 Lenguajes de Programación para Computación Científica

La implementación de un paradigma computacional, como la programación orientada a objetos (POO), requiere de un lenguaje de programación adecuado. En el ámbito de la ingeniería y la computación científica, la elección del lenguaje constituye una decisión fundamental que debe equilibrar dos factores fundamentales: la velocidad de ejecución y la velocidad de desarrollo. Los lenguajes de programación pueden clasificarse en dos grandes categorías: compilados e interpretados.

- **Lenguajes compilados:**

Como C, C++ y Fortran, se caracterizan por traducir el código fuente a lenguaje máquina antes de su ejecución. Esta característica permite alcanzar el máximo rendimiento posible, lo que los convierte en la opción preferida para aplicaciones donde la velocidad es crítica, como los algoritmos de resolución en análisis estructural o los motores de videojuegos. Sin embargo, presentan una curva de aprendizaje más pronunciada, una sintaxis compleja y requieren gestión manual de memoria³¹, lo que puede ralentizar el desarrollo. En particular, Fortran ha sido históricamente el estándar en análisis estructural debido a su eficiencia en cálculos numéricos. Como se señala en la página de Fortran, “Fortran es un lenguaje muy rápido y eficiente en el cálculo de matrices y en la resolución de ecuaciones diferenciales, lo que lo hace ideal para computación científica”³²

Lenguajes interpretados:

Como Python, MATLAB y R, ejecutan el código línea por línea mediante un intérprete en tiempo de ejecución. Su principal ventaja radica en la simplicidad y rapidez de desarrollo, gracias a una sintaxis más legible y cercana al pseudocódigo, así como a la gestión automática de memoria. Python, en particular, ha ganado popularidad como

³¹ **Memoria:** en este contexto, es el área de trabajo temporal donde un programa guarda los datos que necesita mientras se ejecuta. No se trata de almacenamiento permanente, sino de variables, arreglos, objetos y cualquier información que se manipula en tiempo real.

³² «Fortran - Ventajas y desventajas», Fortran, 2023, <https://sites.google.com/view/fortrangrupo9p/ventajas-y-desventajas>.

lenguaje de orquestación en entornos científicos, permitiendo estructurar la lógica principal de forma clara mientras delega los cálculos intensivos a librerías optimizadas en C o FORTRAN. Como lo señala CACheME (Computer-Aided Chemical Engineering), “lenguajes de alto nivel como Python o MATLAB están optimizados para humanos, mientras que lenguajes, como Fortran y C, están optimizados para ordenadores”³³.

La computación científica moderna ha adoptado un enfoque híbrido, donde lenguajes como Python actúan como “lenguajes pegamento” que integran módulos (librerías) de alto rendimiento escritos en lenguajes compilados. Esta estrategia permite combinar la eficiencia de ejecución con la agilidad en el desarrollo. En este sentido, ProgramaciónPro destaca que Fortran “ha sido optimizado para realizar operaciones matemáticas de manera rápida y precisa”, lo que lo convierte en “una opción ideal para aplicaciones que requieren un alto grado de precisión en los cálculos”³⁴. Esto refuerza su rol complementario en arquitecturas mixtas, donde la velocidad de ejecución es crítica.

2.3.3 El Lenguaje Python como Ecosistema de Computación Científica

El paradigma POO descrito anteriormente se implementa utilizando el lenguaje de programación Python. La documentación oficial lo define como un lenguaje "interpretado, interactivo y orientado a objetos" que "combina un poder destacado, con una sintaxis muy clara"³⁵

La elección de Python para el desarrollo completo del programa, responde a su doble rol como orquestador lógico y como interfaz hacia librerías de alto rendimiento. Esta

³³ «Ventajas de Python frente a Fortran / C ¿Cuál es más rápido?», *CACheME*, 1 de octubre de 2013, <https://www.cacheme.org/ventajas-python-vs-fortran-c/>.

³⁴ programacionpro.com, «Lenguaje de Programación Fortran: Ventajas y Desventajas», *ProgramaciónPro*, 8 de abril de 2024, <https://programacionpro.com/lenguaje-de-programacion-fortran-ventajas-y-desventajas/>.

³⁵ Python Software Foundation, «Preguntas frecuentes generales sobre Python — documentación de Python - 3.10.19», Python, 2020, <https://docs.python.org/es/3.10/faq/general.html#what-is-python>.

capacidad es una característica central del diseño del lenguaje, el cual “tiene interfaces a muchas llamadas de sistema y bibliotecas... y es extensible en C o C++”³⁶

Esta configuración permite que la sintaxis limpia de Python se utilice para implementar la lógica POO y estructurar el flujo computacional, delegando simultáneamente las operaciones numéricamente intensivas a librerías compiladas (como C o FORTRAN). Por lo tanto, Python se selecciona como lenguaje principal por ofrecer esta simbiosis óptima: un paradigma potente para la organización del modelo, combinado con el acceso directo a un ecosistema de librerías científicas de alto rendimiento.

2.3.4 NumPy como Núcleo Computacional del Método Matricial

En el núcleo del ecosistema científico de Python, NumPy (Numerical Python) se erige como la librería fundamental para el cálculo numérico. Se la describe como "la librería principal de programación de arreglos (matriciales) para el lenguaje Python" y, de forma más crucial, como "la base sobre la cual se construye el ecosistema científico de Python"³⁷. Para el método matricial de rigidez, que se fundamenta en la resolución de problemas de álgebra lineal a gran escala, NumPy no representa una opción, sino un requisito computacional.

2.3.4.1 El “ndarray”: Representación de K , d y F

En el núcleo del paquete NumPy se encuentra el objeto ndarray, definido como un objeto que “encapsula arreglos (matriciales) n-dimensionales de tipos de datos homogéneos”³⁸. Esta estructura constituye la base para representar datos numéricos de forma eficiente y consistente. Harris et al. lo caracterizan como “una estructura de datos que almacena y accede eficientemente a arreglos multidimensionales... [consiste en] un puntero a memoria, junto con metadatos”³⁹, destacando que el ndarray combina acceso

³⁶ Python Software Foundation, «Preguntas frecuentes generales sobre Python — documentación de Python - 3.10.19».

³⁷ Charles R. Harris et al., «Array Programming with NumPy», *Nature* 585, n.º 7825 (2020): 357-62, <https://doi.org/10.1038/s41586-020-2649-2>.

³⁸ NumPy Developers, «What is NumPy? — NumPy v2.3 Manual», 2025, <https://numpy.org/doc/stable/user/whatisnumpy.html>.

³⁹ «Array Programming with NumPy».

directo a los datos con información estructural que permite interpretarlos y manipularlos eficientemente, lo que permite comprender su diseño orientado al rendimiento y su papel central en el ecosistema científico de Python.

Este objeto permite la implementación computacional directa de las tres entidades matemáticas (y todas las utilizadas en el proceso) que definen el sistema de equilibrio del método matricial:

- La matriz de rigidez global K se instancia como un ndarray de dos dimensiones (con forma $N_{GDL} \times N_{GDL}$, donde N_{GDL} es el número total de GDL de la estructura).
- Los vectores de desplazamientos d y de fuerzas F se instancian como ndarray de una dimensión (con forma (N_{GDL})).

A diferencia de las listas estándar de Python (colecciones heterogéneas con alta sobrecarga de memoria), el objeto ndarray de NumPy representa un bloque contiguo de memoria con elementos homogéneos. Según la definición oficial, “Se requiere que todos los elementos en un arreglo NumPy sean del mismo tipo de dato y, por lo tanto, tengan el mismo tamaño en memoria.”⁴⁰, lo que permite una gestión eficiente del espacio y acceso rápido a los datos. Esta organización es clave para habilitar la “vectorización: operar sobre arreglos enteros en lugar de sus elementos individuales.”⁴¹, considerada esencial para la programación por medio de arreglos matriciales. Además, NumPy “maneja los bucles sobre los elementos del arreglo de forma bastante óptima.”⁴² mediante rutinas precompiladas en C, lo que refuerza su rendimiento en tareas numéricas intensivas.

2.3.4.2 Resolución del Sistema: Delegación de la Complejidad Numérica

El paso computacionalmente más intensivo del análisis matricial es la resolución del sistema de ecuaciones lineales $F = K \cdot d$. En este punto converge el requisito teórico-

⁴⁰ NumPy Developers, «What is NumPy? — NumPy v2.3 Manual».

⁴¹ Harris y col., «Array Programming with NumPy».

⁴² Harris y col., «Array Programming with NumPy».

estructural (un sistema estable y adecuadamente restringido) con su implementación matemática. La teoría exige que la matriz de rigidez global (K) sea cuadrada y de rango completo (no singular), lo que garantiza una solución única para el vector de desplazamientos d .

Desde el punto de vista computacional, el backend del software no implementa algoritmos de resolución desde cero, sino que se encarga de construir correctamente K y F a partir del modelo. La resolución se delega a la función `numpy.linalg.solve(a, b)`, que según la documentación oficial “calcula la solución ‘exacta’, x , de la ecuación matricial lineal bien determinada, es decir, de rango completo, $ax=b$ ”⁴³.

Esta función opera como una interfaz de alto nivel, mientras que la optimización ocurre en capas inferiores mediante rutinas LAPACK. Tal como indica la documentación general, “las funciones de álgebra lineal de NumPy dependen de BLAS y LAPACK para proporcionar implementaciones eficientes de bajo nivel”, utilizando backends como “OpenBLAS, MKL (TM) y ATLAS”⁴⁴.

Las precondiciones de la función —que la matriz a sea cuadrada y de rango completo— no constituyen una limitación impuesta por la librería, sino una validación computacional de los principios teóricos del método de rigidez. La documentación lo confirma al señalar que a “debe ser cuadrada y de rango completo”, y que se genera una excepción o error `LinAlgError` “si a es singular o no es cuadrada”⁴⁵.

2.4 Criterios de Simultaneidad y Combinación de Acciones (NB 1225002)

El análisis estructural moderno no se limita a estudiar la estructura bajo una única condición de carga, sino que evalúa la probabilidad de que diferentes acciones actúen al mismo tiempo sobre el sistema.

⁴³ NumPy Developers, «`numpy.linalg.solve` — NumPy v2.3 Manual», 2025, <https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html>.

⁴⁴ NumPy Developers, «Linear algebra — NumPy v2.3 Manual», 2025, <https://numpy.org/doc/stable/reference/routines.linalg.html>.

⁴⁵ NumPy Developers, «`numpy.linalg.solve` — NumPy v2.3 Manual».

2.4.1 Concepto de Simultaneidad de Cargas

La simultaneidad de acciones es el principio normativo que reconoce que es estadísticamente improbable que todas las cargas máximas variables (viento, sismo, sobrecarga de uso, nieve) ocurran en su valor pico en el mismo instante de tiempo. Por esta razón, la norma boliviana NB 1225002⁴⁶, alineada con los estándares internacionales (ASCE 7, ACI 318), utiliza factores de combinación que reducen la magnitud de ciertas cargas cuando se consideran actuando conjuntamente con otras.

El objetivo es definir un escenario de diseño (Estado Límite Último) que tenga una probabilidad de excedencia aceptable y uniforme, evitando el sobredimensionamiento que resultaría de sumar todos los valores máximos teóricos de las cargas.

2.4.2 Concurrencia de Hipótesis de Carga

Se define como hipótesis concurrentes a aquel conjunto de casos de carga que, por su naturaleza física o definición normativa, actúan de manera permanente y aditiva en la estructura. Estas hipótesis no compiten entre sí, sino que se superponen obligatoriamente en todas las combinaciones de diseño.

El ejemplo paradigmático es la **Carga Muerta (D)**. Esta hipótesis general suele estar compuesta (pero no limitada) por dos sub-hipótesis concurrentes:

1. **Peso Propio (PP):** La fuerza gravitatoria generada por la masa de los elementos estructurales.
2. **Cargas Muertas (CM):** El peso de elementos no estructurales (pisos, muros, instalaciones).

Dado que ambas están presentes permanentemente debido a la gravedad, se consideran concurrentes y se suman algebraicamente en el vector de fuerzas antes de aplicar cualquier factor de mayoración. Sus efectos son siempre simultáneos.

⁴⁶ IBNORCA, NB 1225002 - *Acciones sobre las estructuras* (2013).

2.4.3 Hipótesis Alternantes y Mutuamente Excluyentes

A diferencia de las anteriores, las hipótesis alternantes o mutuamente excluyentes son aquellos estados de carga que, por lógica física o requerimiento de envolvente, no pueden o no deben considerarse actuando simultáneamente en su totalidad.

Este concepto se aplica fundamentalmente en dos escenarios:

1. **Incompatibilidad Física:** Casos donde la presencia de una acción anula físicamente la otra. Por ejemplo, el viento actuando en dirección +X es mutuamente excluyente con el viento en dirección -X; la estructura debe verificarse para ambos casos por separado, pero nunca para la suma de ambos.
2. **Alternancia de Sobrecargas:** Basado en la teoría de líneas de influencia, este criterio establece que la carga viva (L) no actúa necesariamente en toda la estructura al mismo tiempo. Para maximizar los esfuerzos en estructuras continuas (vigas o losas), la norma requiere aplicar la carga viva en patrones alternados.

2.5 Distribución de Cargas en Losas

La transformación de cargas superficiales a cargas lineales sobre vigas se sustenta en la Teoría de Líneas de Fluencia. Esta teoría asume que, en el estado límite último, la losa se fisura siguiendo patrones geométricos donde los segmentos resultantes "pueden considerarse como cuerpos rígidos que rotan en el espacio alrededor de estos ejes de rotación"⁴⁷, los cuales están definidos por los apoyos y las líneas de fluencia.

2.5.1. Distribución en Losas Bidireccionales (Método de las Áreas Tributarias)

El patrón de fisuración define el área tributaria que "descarga" sobre cada viga perimetral. La geometría de estas áreas depende de la restricción al giro (continuidad) de los bordes:

⁴⁷ David Darwin y Charles Dolan, *Design of Concrete Structures*, 16th ed. (McGRAW-HILL, 2021), 764.

- **Bordes Simétricos (45°):** Cuando dos bordes adyacentes tienen la misma condición de apoyo (continuo cuando existe otra losa adyacente al borde; discontinuo cuando no existe otra losa adyacente.), los ejes de rotación se cruzan en la esquina del paño. En consecuencia, la teoría dicta que las “líneas de rotura diagonales deben pasar por las esquinas”⁴⁸, lo que geométricamente corresponde a una bisectriz a 45°, generando áreas tributarias triangulares o trapezoidales equiláteras.
- **Bordes Asimétricos (30°/60°):** Cuando interactúan un borde continuo y uno discontinuo, la asimetría en la rigidez rotacional modifica la ubicación de los ejes de giro y, consecuentemente, el patrón de falla. Si bien las líneas de fluencia se mantienen teóricamente como "líneas rectas, ya que representan la intersección de dos planos", su orientación debe ajustarse para converger con la intersección desplazada de los ejes. En la práctica, esto resulta en una inclinación asimétrica (modelada simplificada como 60° respecto al borde discontinuo y 30° respecto al continuo) que otorga una mayor área tributaria al borde más rígido, reflejando su capacidad para atraer mayor carga antes de la fluencia.

⁴⁸ Darwin y Dolan, *Design of Concrete Structures*, 765.

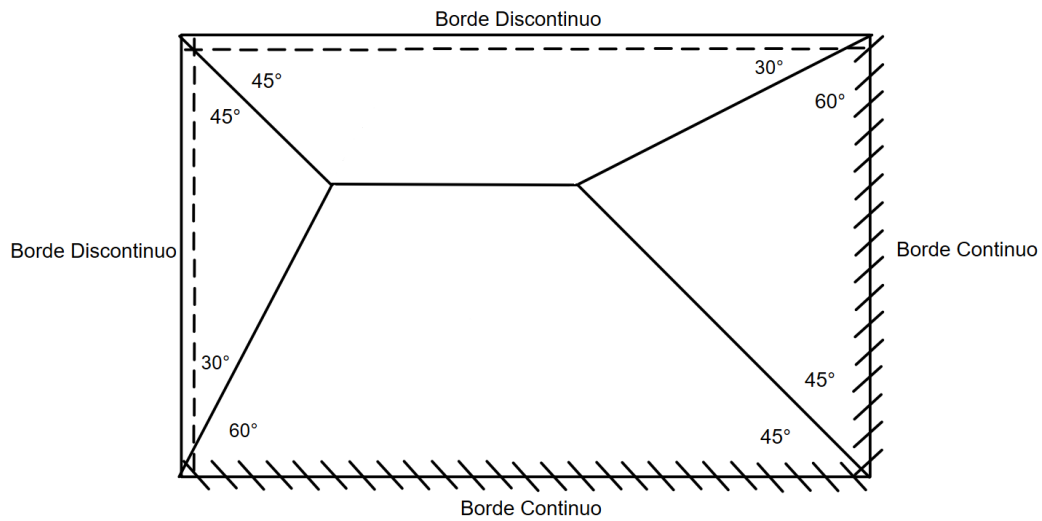


Figura 6: Representación gráfica de las líneas de fluencia en losas bidireccionales

Nota. Elaboración propia.

2.5.2. Distribución en Losas Unidireccionales

Para losas unidireccionales, la teoría de líneas de fluencia indica que las áreas triangulares en los extremos perpendiculares a la dirección de la losa son despreciables en comparación con el área rectangular central. El mecanismo de falla predominante es una fisura central paralela a los apoyos largos. En consecuencia, se asume una simplificación teórica donde la carga se distribuye equitativamente 50% a cada viga de borde principal, despreciando el aporte a las vigas de borde secundario.

CAPÍTULO III

METODOLOGÍA

El presente capítulo detalla la estructura lógica y los procedimientos computacionales implementados en la herramienta VORTEX 3D. Se describe la secuencia de operaciones que el software ejecuta para transformar los datos de entrada en resultados estructurales válidos, fundamentándose en el Método Matricial de la Rigidez.

3.1 Algoritmo de Resolución Estructural

A continuación, se presenta la secuencia lógica de ejecución que realiza el software desde la recepción de datos hasta la obtención de los esfuerzos finales:

1. **Recepción y validación de datos geométricos y propiedades:** Se identifican las coordenadas espaciales de cada nudo y la conectividad de los elementos barra. Simultáneamente, se asignan las propiedades mecánicas (Módulo de Elasticidad, Módulo de Corte) y geométricas (Área, Inercias, Constante de Torsión) a cada elemento.
2. **Definición de grados de libertad y condiciones de borde:** Se establece el número total de grados de libertad del sistema. Se procesa la información de los apoyos para identificar los GDL restringidos (desplazamientos o giros nulos) y los GDL libres (incógnitas a calcular).
3. **Ensamblaje del vector de cargas:** El programa procesa las distintas hipótesis de carga (fuerzas nodales, cargas distribuidas en elementos y cargas superficiales en losas). Las cargas distribuidas y superficiales son transformadas estáticamente a fuerzas y momentos equivalentes aplicados directamente sobre los nudos, ensamblando así el vector de fuerzas globales.
4. **Generación de la matriz de rigidez local:** Para cada elemento barra individual, se calcula su matriz de rigidez local k' en su propio sistema de referencia, basándose en sus propiedades mecánicas y longitud.
5. **Transformación de coordenadas:** Se calcula la matriz de transformación para cada elemento, la cual relaciona el sistema local con el sistema global. Mediante

operaciones matriciales, la matriz local k' es rotada para obtener la matriz de rigidez del elemento en coordenadas globales k .

6. **Ensamblaje de la matriz de rigidez global:** Se construye la matriz de rigidez total de la estructura K mediante la superposición de las contribuciones de rigidez de cada elemento en las posiciones correspondientes a los grados de libertad que conectan.
7. **Aplicación de condiciones de borde y reducción estática:** Se impone la partición de la matriz global K . Se segregan las filas y columnas correspondientes a los apoyos (desplazamientos conocidos), aislando la sub-matriz de rigidez reducida asociada exclusivamente a los grados de libertad libres.
8. **Resolución del sistema de ecuaciones:** Se resuelve el sistema lineal de ecuaciones, obteniendo como resultado el vector de desplazamientos nodales (traslaciones y rotaciones) para todos los nudos libres de la estructura.
9. **Cálculo de reacciones en los apoyos:** Utilizando los desplazamientos calculados y la sub-matriz correspondiente a los vínculos, se determinan las fuerzas de reacción y momentos en los nudos restringidos¹⁰.
10. **Obtención de fuerzas internas:** Finalmente, para cada elemento, se transforman los desplazamientos globales de sus nudos extremos de vuelta a su sistema local. Con estos desplazamientos locales, se calculan las fuerzas internas definitivas: carga axial, fuerzas cortantes en dos direcciones, momentos flectores en dos ejes y momento torsor.

3.1.1 Diagramas de Flujo

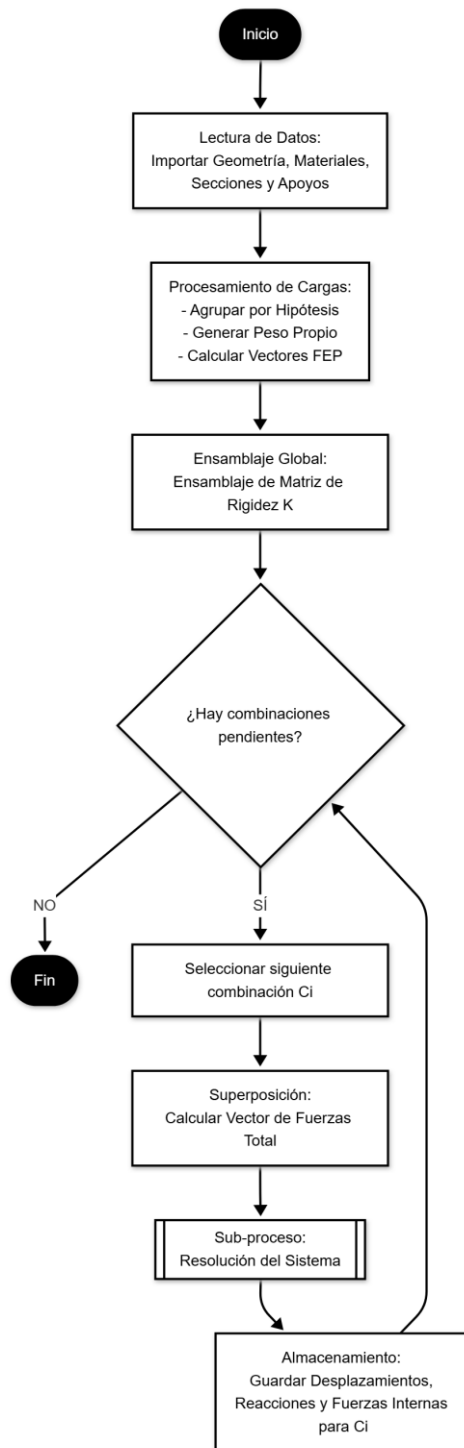


Figura 7: Flujo general del procesamiento de la estructura

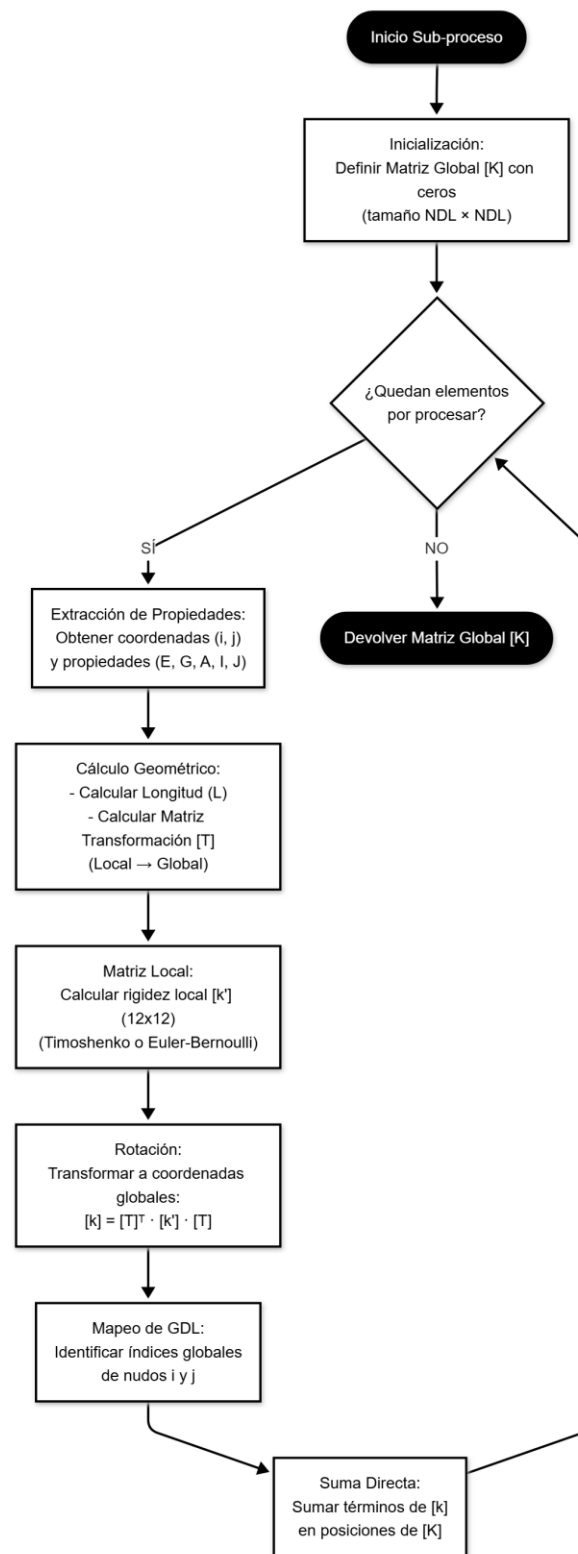


Figura 8: Ensamblaje de la matriz de rigidez global

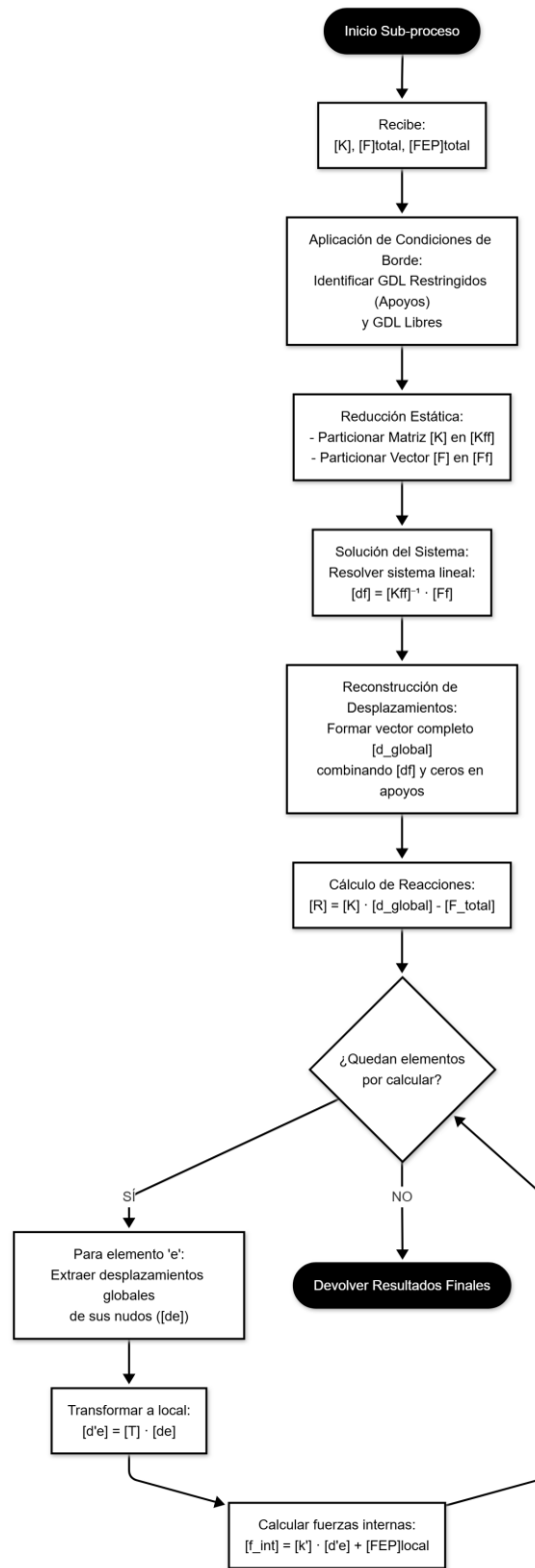


Figura 9: Cálculo de fuerzas internas

3.2 Algoritmo de Distribución de Cargas en Losas

Este módulo automatiza la transferencia de solicitaciones superficiales aplicadas sobre losas hacia vigas. El procedimiento evita la asignación manual de cargas trapezoidales o triangulares, realizando un análisis geométrico para determinar las áreas tributarias exactas.

El algoritmo procesa cada panel de losa mediante un esquema de decisión bifurcada, adaptando la estrategia de distribución según la tipología estructural definida:

1. Análisis Topológico y Clasificación:

Se recupera la geometría del panel y se identifican los elementos viga perimetrales. El algoritmo verifica la existencia de losas adyacentes para establecer automáticamente las condiciones de borde (continuo o discontinuo) y clasifica el panel según su distribución (unidireccional o bidireccional).

2. Rama A: Distribución Unidireccional:

Si la losa trabaja en una dirección, el algoritmo identifica el par de bordes portantes paralelos al eje de carga. Se calcula un ancho tributario constante ($L_{tributario} = L_{vano}/2$) y se genera una carga lineal uniformemente distribuida sobre las vigas de soporte.

3. Rama B: Distribución Bidireccional (Líneas de Rotura):

Para losas armadas en dos direcciones, se generan vectores de las líneas de rotura desde los vértices. Los ángulos de proyección se ajustan dinámicamente según la rigidez de los apoyos (45° para bordes simétricos, ángulos asimétricos para bordes continuos). La intersección de estos vectores define polígonos tributarios (triángulos y trapecios) que se traducen en perfiles de carga variable $q(x)$ sobre el perímetro.

4. Discretización y Mapeo:

Finalmente, las cargas calculadas (rectangulares, triangulares o trapezoidales) se proyectan y segmentan para coincidir con la discretización de elementos finitos de las vigas receptoras, incorporándose al vector de fuerzas global.

3.2.1 Diagrama de Flujo

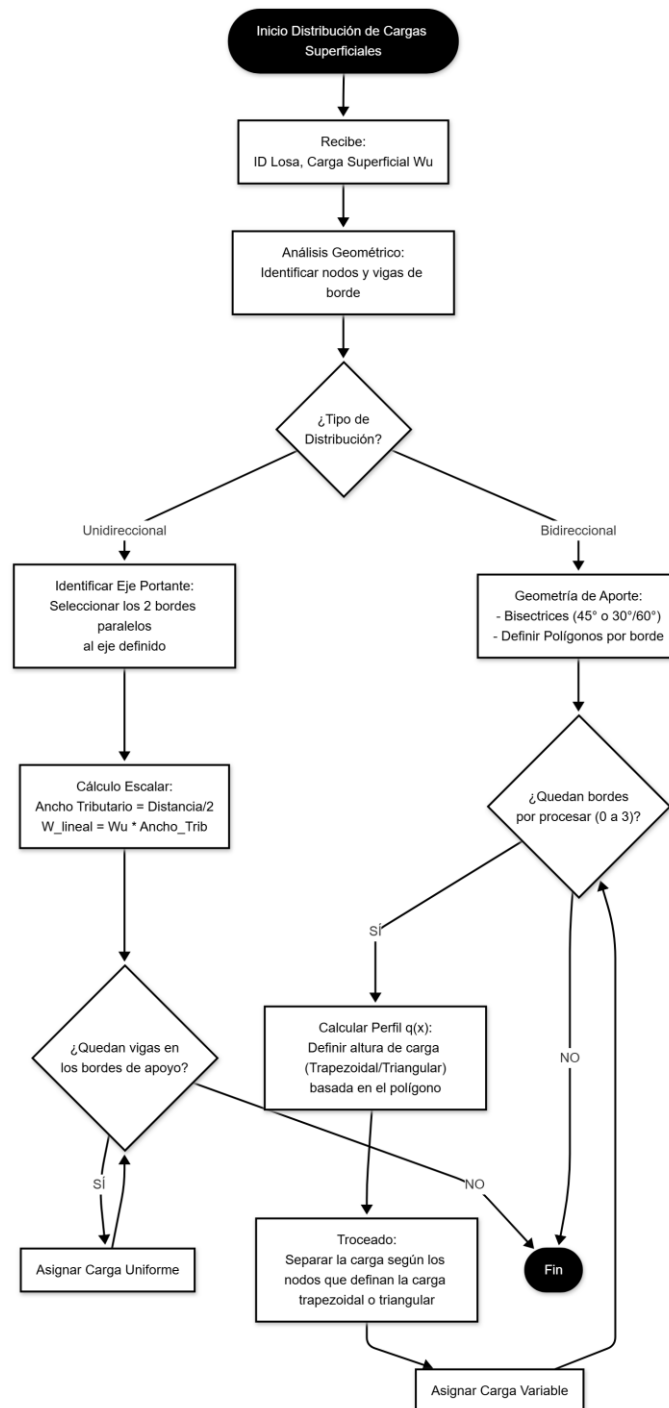


Figura 10: Distribución de cargas superficiales

3.3 Algoritmo de Procesamiento de Cargas

3.3.1 Generación Automática de Peso Propio

Para minimizar el error humano en la entrada de datos y garantizar que las masas del modelo sean consistentes con la geometría definida, el software incorpora un sub-sistema de cálculo automático del Peso Propio (PP). Este algoritmo opera bajo un paradigma "Just-In-Time" (JIT), lo que significa que el vector de fuerzas inerciales no se almacena estáticamente, sino que se recalcula dinámicamente en cada ejecución del análisis, asegurando que cualquier cambio en las secciones o materiales se refleje inmediatamente en las cargas.

A. Lógica de Cálculo y Concurrencia

El procedimiento distingue entre elementos lineales y superficiales, unificando sus efectos en un único vector de fuerzas inerciales (F_{PP}) que actúa concurrentemente con las cargas muertas definidas por el usuario.

1. **Elementos 1D (Vigas y Columnas):** El algoritmo recorre la topología de barras. Para cada elemento, recupera el área de la sección transversal (A) y el peso específico del material asignado (γ). Se genera una carga lineal uniforme distribuida $q_{PP} = -(\gamma \cdot A)$ orientada en el eje global gravitacional (Z^-), la cual se transforma a coordenadas locales y se convierte en fuerzas nodales y momentos de empotramiento.
2. **Elementos 2D (Losas):** Para los elementos de área, se obtiene el espesor (h) y el peso específico (γ). Se calcula una carga superficial uniforme $w_{PP} = -(\gamma \cdot h)$. Críticamente, esta carga no se aplica directamente a los nudos de la losa, sino que se inyecta en el Algoritmo de Distribución (descrito en la sección 3.2.1), el cual la descompone en cargas lineales sobre las vigas perimetrales.
3. **Superposición con Hipótesis de tipo D (cargas muertas):** En la etapa de combinaciones, el vector resultante F_{PP} se trata como una hipótesis de carga muerta base. El sistema aplica el principio de superposición para fusionarlo con

las cargas muertas adicionales introducidas por el usuario $F_{D,usuario}$, tal que la carga muerta total de diseño se define como:

$$F_{D,total} = F_{pp} + F_{D,usuario}$$

3.3.2 Gestión de Solicitaciones y Superposición

Aquí se organizarán las múltiples fuentes de carga que actúan sobre la estructura y se las preparará para el análisis matricial, aplicando el Principio de Superposición válido para sistemas de comportamiento lineal elástico.

El procedimiento no resuelve la estructura para cada hipótesis individualmente. En su lugar, opta por una estrategia de superposición de vectores de fuerza, donde las acciones se combinan matemáticamente antes de entrar al sistema de ecuaciones. Esto optimiza el proceso de cálculo, reduciendo el problema a resolver un único sistema

$$K \cdot d = F_{comb}$$

por cada combinación de diseño requerida.

El algoritmo se estructura en tres etapas fundamentales:

1. **Generación de Vectores Base (F_i):** Se recorre el modelo y se agrupan todas las cargas según su naturaleza (Muerta, Viva, Viento, etc.). Las cargas superficiales y distribuidas se transforman estáticamente en Fuerzas de Empotramiento Perfecto (FEP).
 - *Resultado:* Un vector de fuerzas independiente para cada hipótesis elemental ($F_D, F_L, F_W, \dots$).
2. **Composición Vectorial por Combinación:** Para cada combinación de carga definida por la normativa, el algoritmo realiza una suma ponderada de los vectores base y sus FEPs correspondientes utilizando los factores de mayoración (γ).

$$F_{comb} = \gamma_1 F_1 + \gamma_2 F_2 + \gamma_3 F_3 + \dots + \gamma_n F_n$$

3. **Resolución Directa:** El vector resultante F_{comb} se ingresa en la ecuación fundamental de rigidez. Se obtienen los desplazamientos totales y, posteriormente,

se calculan los esfuerzos internos finales sumando las contribuciones locales y los términos de FEP ya mayorados.

3.3.3 Diagramas de Flujo

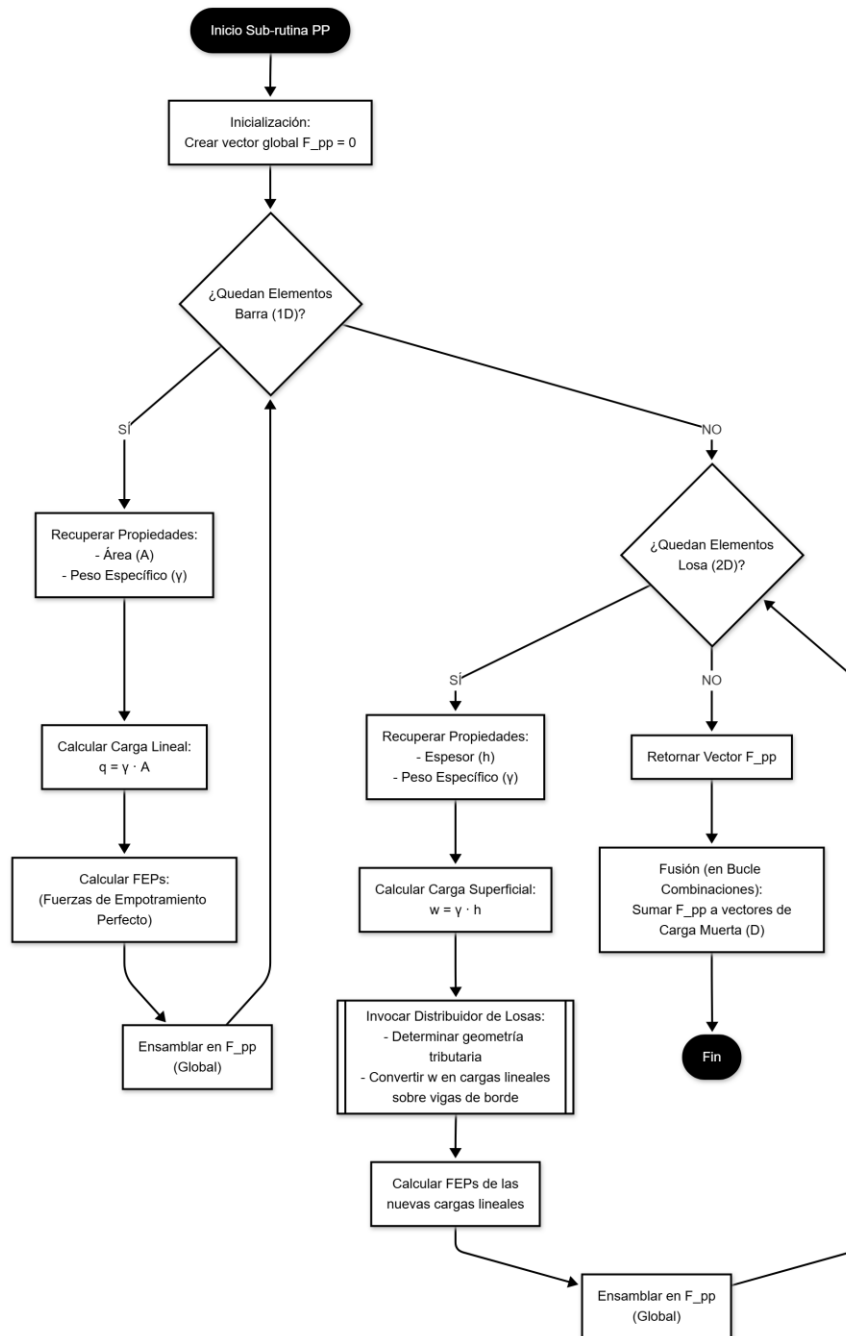


Figura 11: Cálculo de Peso Propio

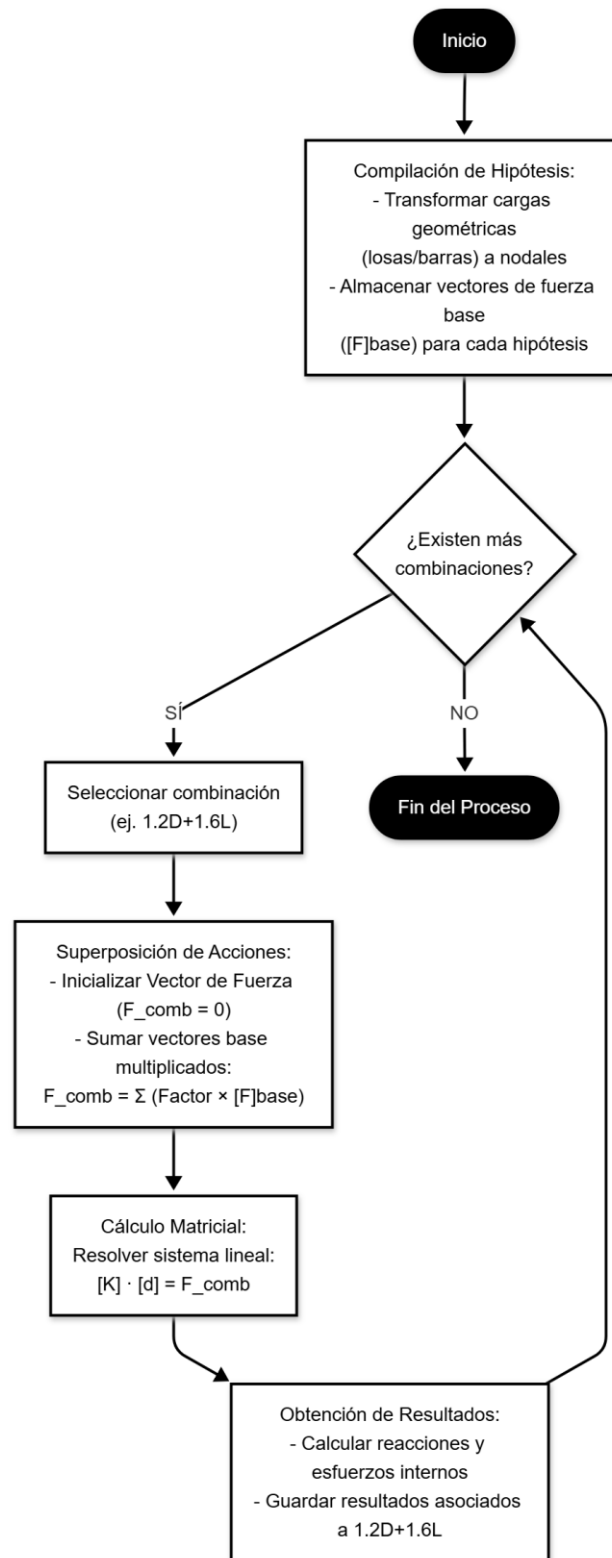


Figura 12: Resolución del sistema por combinaciones

3.4 Algoritmo de Cálculo de Envolventes

El análisis estructural requiere la determinación de los efectos máximos a los que estará sometido el sistema. Dado que las cargas actúan en distintas combinaciones, es imprescindible calcular la Envolvente de Esfuerzos, definida como el lugar geométrico de los valores extremos (máximos positivos y negativos) de fuerzas internas (Axial, Cortante, Momento Flector y Torsor) para cada sección de los elementos estructurales.

El procedimiento computacional implementado es el siguiente:

1. **Discretización del Elemento:** Cada elemento barra se divide en una serie de estaciones (puntos de control) a lo largo de su eje longitudinal ($L/50$).
2. **Barrido de Combinaciones:** Para cada estación de análisis x_i , el sistema itera sobre el conjunto total de combinaciones de carga resueltas ($C_1, C_2, C_3, \dots, C_n$).
3. **Selector de Extremos:** Se inicializan dos acumuladores para cada tipo de esfuerzo, por ejemplo, Momento M_y : $M_{max+}(x_i)$ y $M_{max-}(x_i)$.

En cada iteración, se compara el valor de la combinación actual $M_{comb}(x_i)$ con los acumuladores:

- Si $M_{comb} > M_{max+}$, se actualiza el Máximo positivo.
 - Si $M_{comb} < M_{max-}$, se actualiza el Máximo negativo.
4. **Construcción de la Envolvente:** El resultado es una curva compuesta que conecta todos los puntos M_{max+} y otra para M_{max-} .

3.4.1 Diagrama de Flujo

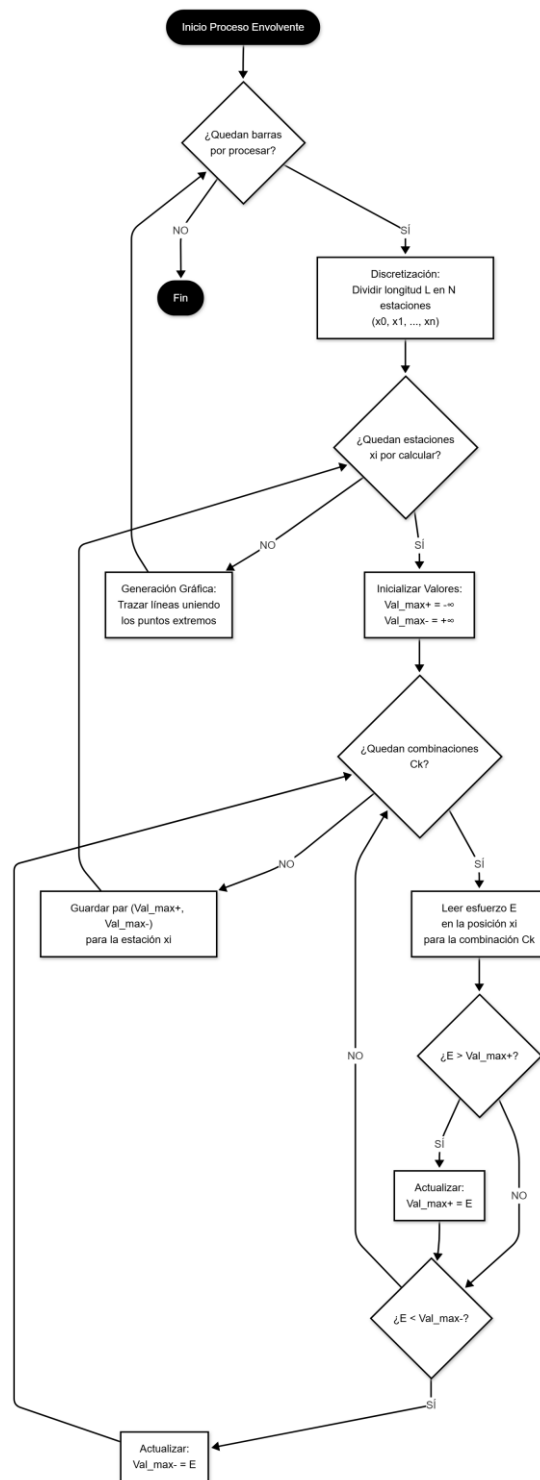


Figura 13: Cálculo de envolventes

3.5 Generación Automatizada de Memoria de Cálculo

En consonancia con el objetivo de transparencia y validación del software, se define la lógica para un módulo de documentación exhaustiva orientado a la "Auditoría del Análisis". Su propósito fundamental es exponer la trazabilidad completa del Método Matricial de Rigidez, permitiendo al ingeniero verificar los procesos intermedios de ensamblaje, las transformaciones de coordenadas, la resolución del sistema de ecuaciones, la obtención de fuerzas internas y envolventes.

El algoritmo de generación de reportes se estructura linealmente en tres fases de procesamiento de información:

1. Auditoría de Topología y Cargas (Pre-Cálculo):

En esta primera fase, se documentan los datos de entrada.

- Geometría y Materiales: Se tabulan las coordenadas nodales y se reportan las propiedades mecánicas derivadas (E , A , I , G , J , etc) de cada sección.
- Trazabilidad de Cargas: Se recopilan los registros de la distribución de cargas. Para cargas superficiales en losas, el reporte debe detallar explícitamente la geometría de las áreas tributarias y el valor resultante de las cargas lineales transmitidas a los pórticos, justificando el origen de las fuerzas actuantes.

2. Auditoría del Ensamblaje Matricial (Núcleo del Método):

Se accede a los datos internos del motor de resolución para exponer la construcción del modelo matemático.

- Matrices Elementales: Para cada elemento, se formatean y presentan las matrices de rigidez local k' y las matrices de transformación T utilizadas.
- Matriz Global: Se documenta la estructura de la Matriz de Rigidez Global K , permitiendo visualizar la rigidez del sistema ensamblado antes de la resolución.

3. Tabulación de Resultados y Esfuerzos (Post-Cálculo):

Para cada combinación de carga procesada, el algoritmo tabula los vectores de respuesta numérica.

- **Estado Nodal:** Se listan los desplazamientos ($\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z$) y las reacciones en los apoyos.
- **Esfuerzos Internos:** En lugar de generar gráficos visuales, el algoritmo invoca a las rutinas de discretización para obtener los valores numéricos de los esfuerzos (Axial, Cortante, Momento) en múltiples estaciones a lo largo del eje de la barra (ej. décimos de la luz). Esto genera tablas de valores paso a paso y resúmenes de máximos y mínimos (envolventes numéricas), facilitando la validación manual por parte del usuario.

3.5.1 Diagrama de Flujo

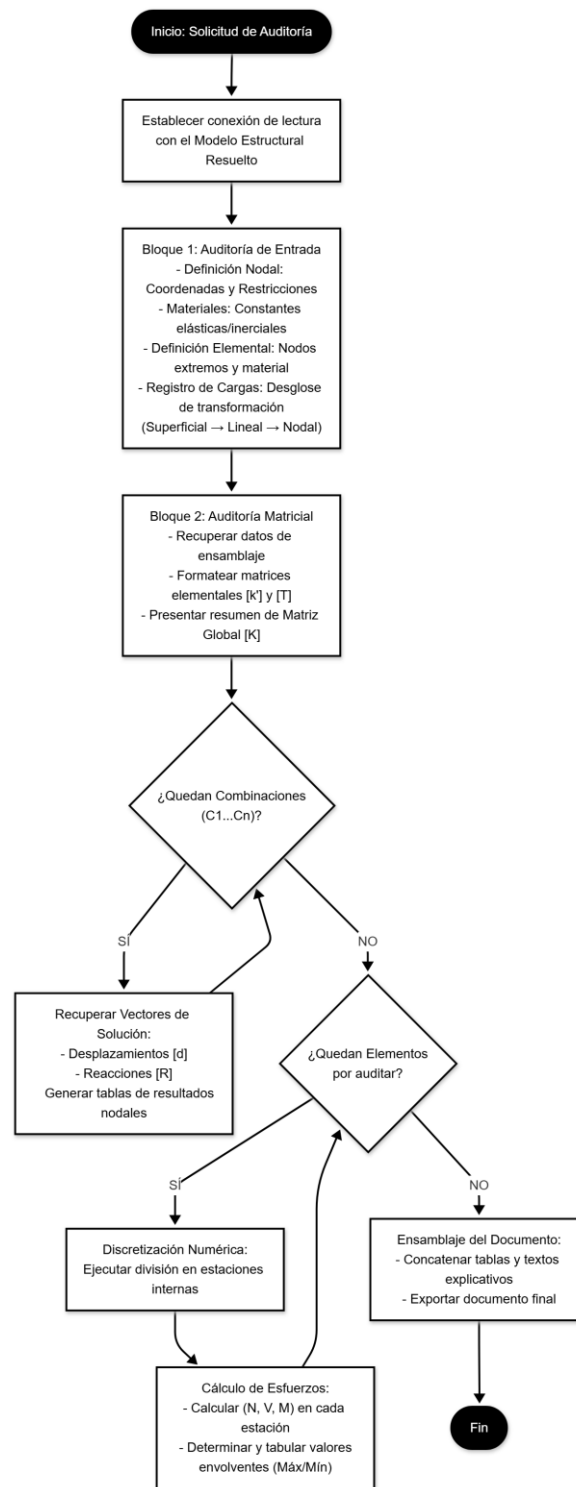


Figura 14: Generación de reporte de cálculo

3.6 Arquitectura del Programa

Para garantizar la escalabilidad y el mantenimiento del código, VORTEX 3D fue desarrollado bajo el paradigma de Programación Orientada a Objetos (POO). Esta arquitectura permite encapsular la complejidad del comportamiento estructural en entidades lógicas (Clases) que interactúan entre sí, replicando la naturaleza física de los objetos estudiados.

Habiendo definido los algoritmos lógicos y matemáticos, esta sección describe la estructura de software construida para soportarlos, justificando la organización del código fuente en núcleos funcionales.

La arquitectura del sistema se divide en tres núcleos funcionales principales:

1. **Núcleo de Datos (ModeloEstructura):** Es la clase contenedora principal que actúa como la "memoria" del programa. Gestiona el estado completo del proyecto, almacenando diccionarios de Nodos, Elementos, Materiales y Cargas.
2. **Núcleo de Gestión de Solicitaciones (ProcesadorCargas):** Este módulo actúa como el orquestador de las acciones externas. Prepara los vectores de fuerza. Es responsable de interpretar las hipótesis de carga, ejecutar los algoritmos de distribución geométrica de losas y aplicar el principio de superposición para construir los vectores de carga combinados (F_{total}) que alimentarán al motor matemático.
3. **Núcleo de Cálculo (Solucionador3D):** Clase especializada que hereda la geometría del modelo para ejecutar el análisis matemático. Sus métodos internos (`ensamblar_K`, `resolver`) son puros y no modifican la interfaz gráfica, garantizando la separación entre el motor de cálculo y la visualización.

3.6.1 Organización del Código Fuente (Módulos de Análisis)

A continuación, se detalla la estructura de archivos del código fuente (*scripts*) desarrollado exclusivamente para el motor de análisis matricial:

- **modelo_estructura.py:** Su clase principal es “Estructura” y su función es la definición de la topología (Nodos, barras), propiedades de sección y almacenamiento de resultados brutos.
- **calc.py:** Su clase principal es “Solucionador3D” y su función es la implementación numérica del Método Matricial de Rigidez, lo que incluye el cálculo de matrices elementales, matrices de transformación, ensamblaje de la matriz de rigidez global y la resolución del sistema fundamental de ecuaciones.
- **procesador_cargas.py:** Su clase principal es “ProcesadorCargas” y su función es el cálculo de fuerzas nodales equivalentes, fuerzas de empotramiento perfecto y superposición de vectores de carga por combinaciones.
- **distribuidor_losas.py:** Script funcional completo, cuya función es el algoritmo geométrico encargado de la identificación de áreas tributarias y conversión de cargas superficiales en cargas lineales sobre pórticos.
- **diagramas.py:** Su clase principal es “GeneradorDiagramas”, su función es el algoritmo de discretización e integración numérica para obtener los valores de esfuerzos internos en cualquier punto x del elemento.
- **generador_reporte.py:** Su clase principal es “GeneradorReporte” y funciona como un módulo de auditoría que extrae y formatea las matrices y vectores del análisis para la generación de una memoria de cálculo transparente.

Es posible descargar el programa desde el siguiente repositorio de github:

<https://github.com/CoderBadger/VORTEX-3D.git>

3.7 Protocolo de Validación de Resultados

Para garantizar la fiabilidad del software desarrollado, se ha diseñado un esquema de validación escalonado que aísla progresivamente las variables del cálculo. El protocolo inicia con la verificación de la formulación matemática fundamental y culmina con la simulación integral de un proyecto de edificación real.

3.7.1 Nivel I: Validación Analítica en Sistemas Planos

Esta etapa tiene como propósito certificar la precisión del motor matricial en su forma más pura, eliminando variables espaciales y efectos de corte complejos para contrastar directamente contra la teoría base. Se analizarán tres casos de estudio de complejidad incremental: viga continua, cercha sometida a cargas nodales y pórtico plano sometido a cargas distribuidas en elementos.

- **Alcance:** Elementos viga continuos y pórticos planos (2D) sometidos a cargas en el plano.
- **Método de Cálculo:** El cálculo de estos elementos será realizado por medio de la teoría de Bernoulli
- **Método de Control:** Herramienta Ftool para Validación cruzada de desplazamientos y reacciones.
- **Justificación de la Herramienta:** Se selecciona **Ftool** por ser un *solucionador* matricial de elementos de barra puro, lo que permite una comparación "uno a uno" con la formulación de VORTEX 3D sin la intervención de consideraciones constructivas automáticas en su proceso de cálculo.

3.7.2 Nivel II: Validación Espacial y Distribución de Cargas

En este nivel se introduce la tercera dimensión para validar dos aspectos fundamentales: la transformación de coordenadas espaciales y el Algoritmo de Distribución de Cargas de Losas. Se han definido **dos prototipos de prueba:** el

primero evaluando la respuesta espacial ante cargas nodales y lineales directas, y el siguiente incorporando losas unidireccionales y bidireccionales respectivamente.

- **Alcance:** Pórticos espaciales (Prototipos 3D) sometidos a cargas sobre barras y cargas gravitacionales transferidas desde paneles de losa.
- **Objetivo de la Prueba:** Verificar que la transferencia de solicitaciones (de superficiales a lineales) genere las reacciones y diagramas de esfuerzos correctos en el esqueleto de barras, sin modificar la matriz de rigidez del sistema.
- **Método de Control:** Análisis Comparativo Numérico.
- **Herramienta de Referencia: CYPE 3D.**
 - *Estrategia:* Se modelará la geometría idéntica en CYPE 3D. Para validar la distribución, se introducirán en la herramienta de referencia las cargas lineales (rectangulares, triangulares o trapezoidales) de forma manual, replicando la salida teórica del algoritmo de áreas tributarias. En esta etapa, se activará la consideración de deformación por corte (Teoría de Timoshenko) en ambos programas.
- **Variables de Control:**
 - Diagramas de Momento Flector y Cortante en vigas principales.
- **Análisis de Divergencia:** Se reportará el error relativo porcentual entre los valores obtenidos para cuantificar la precisión del cálculo.

3.7.3 Nivel III: Validación Sistémica en Proyecto de Edificación

La fase culminante del protocolo contrasta los resultados del software frente a una situación de diseño real, integrando la totalidad de los módulos desarrollados (análisis de cargas, combinaciones y resolución global) en la simulación de una estructura de hormigón armado completa.

- **Alcance del Modelo:** Edificio Multifamiliar "Concordia", constituido por un sistema aporticado de Hormigón Armado de 6 niveles (Planta Baja + 4 Pisos + Azotea Accesible).
- **Definición de Cargas:** Se aplicará el análisis de cargas completo según norma **NB 1225002**, incluyendo Peso Propio, Carga Muerta (CM), Sobrecarga de Uso (SC), Viento (W) y Granizo (S).
- **Herramienta de Referencia: CYPECAD.**
- **Justificación del Cambio de Software:** Se utiliza CYPECAD por ser el estándar de la industria local para edificación. A diferencia de CYPE 3D, este software introduce numerosas hipótesis constructivas aplicadas automáticamente.
- **Enfoque de la Comparación:** No se busca una igualdad matemática absoluta, dado que los modelos numéricos difieren conceptualmente en el tratamiento de diseño constructivo realizado por CYPECAD. El objetivo es validar la coherencia ingenieril de los resultados, analizando si las diferencias se encuentran dentro de rangos aceptables para la práctica profesional.
- **Variables de Control Críticas:**
 1. **Esfuerzo Axial Máximo en Columnas:** Se seleccionan los elementos verticales críticos en la base para comparar la carga axial máxima resultante de las combinaciones de carga, validando la precisión del descenso de cargas y la resolución matricial.
 2. **Envoltentes de Diseño:** Comparación de las curvas de esfuerzos máximos (Momentos y Cortantes) en los elementos identificados como críticos por VORTEX 3D.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

En este capítulo se realizan las comprobaciones y validaciones del programa VORTEX 3D con otros tales como ser Ftool, CYPE 3D y CYPECAD. Las comprobaciones serán realizadas en tres niveles, como fue detallado en 3.7.3.

4.1 Resultados Nivel I: Validación Analítica en Sistemas Planos

En esta sección se presentan los resultados comparativos para elementos 2D, contrastando la solución obtenida por VORTEX 3D frente al software de control Ftool, bajo la hipótesis de Bernoulli (sin deformación por corte).

4.1.1 Caso de Estudio 1: Viga Continua Hiperestática

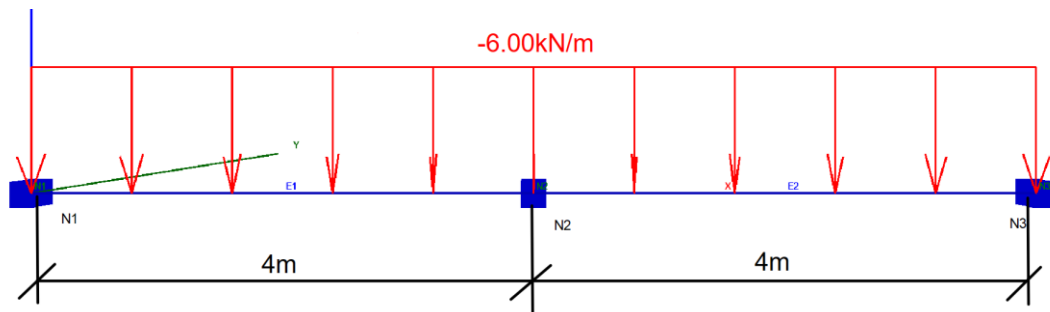


Figura 15: Caso de Estudio 1 - Viga Continua Hiperestática

El Caso 1 consta de una viga continua hiperestática, compuesta por 2 vanos de 4 metros cada uno. Además, está sometida a una carga lineal de 6kN/m. A continuación, se procede con la resolución manual y la presentación de resultados en VORTEX 3D y Ftool. En este, ambos tramos son de la misma sección y el mismo material. Con fines de mantener los mismos valores, tanto en VORTEX como en Ftool se maneja una Sección Rectangular $0.2 \times 0.3m$, misma que tiene las siguientes propiedades:

$$E=21000 \text{ MPa}; I_y = 4.5 \times 10^{-4} m^4$$

Todos sus apoyos únicamente restringen traslaciones en X y en Z.

La sencillez del presente caso, permite mostrar la memoria de cálculo completa de la resolución de la estructura, que por si misma ya es muy extensa, por lo que sería inviable mostrarla completa en caso de analizar una estructura un poco mayor.

4.1.1.1 Resultados VORTEX 3D

En el siguiente texto, proveniente de los reportes generados por VORTEX 3D, se presentan los desplazamientos nodales y las reacciones en los apoyos

```

--- Desplazamientos (Globales) para 'Caso 1' ---
Nodo Ux Uy Uz Rx Ry Rz
-----
1 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 -8.4656e-04 0.0000e+00
3 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 8.4656e-04 0.0000e+00
--- Reacciones en Apoyos (Globales) ---
Nodo Fx Fy Fz Mx My Mz
-----
1 0.0000 0.0000 9.0000 0.0000 0.0000 0.0000
2 0.0000 0.0000 30.0000 0.0000 0.0000 0.0000
3 0.0000 0.0000 9.0000 0.0000 0.0000 0.0000

```

4.1.1.2 Memoria de cálculo de VORTEX 3D

```

=====
MEMORIA DE CÁLCULO ESTRUCTURAL - VORTEX 3D
=====
Fecha: 2025-11-21 05:26:19
Archivo: C:/Users/user/Documents/DIEGO/10mo Semestre/Tesis/Comparaciones Cap4/Vortex/Nivel1_Caso1.votx

1. DATOS DE ENTRADA
-----
1.1 Nodos:
Nodo      X (m)      Y (m)      Z (m)
1          0.000      0.000      0.000
2          4.000      0.000      0.000
3          8.000      0.000      0.000

1.2 Materiales (Propiedades Calculadas):
ID  Tipo      E (kPa)      G (kPa)      A (m²)      J (m⁴)      Iy (m⁴)      Iz (m⁴)      Avy (m²)      Avz (m²)      PE (kN/m³)
1   rectangular  2.100e+04    8.077e+03    6.000e-02    4.695e-04    4.500e-04    2.000e-04    5.000e-02    5.000e-02    24.00

1.3 Elementos (Pórtico):
ID   Nodo i   Nodo j   Material   Longitud (m)
1     1       2       1          4.000
2     2       3       1          4.000

1.4 Losas (Definición):
ID   Nodos      Distribución   Eje Uni.   Espesor (m)   PE (kN/m³)

1.5 Apoyos (R=Restringido, L=Libre):
Nodo  Tx   Ty   Tz   Rx   Ry   Rz
1     R   R   R   R   L   R
2     R   R   R   R   L   R
3     R   R   R   R   L   R

1.6 Hipótesis y Cargas Asignadas:
Hipótesis 1: "Caso 1" (Tipo: D)
Cargas en Elementos:
- Elem 1 (uniforme): [-6.00]
- Elem 2 (uniforme): [-6.00]

1.7 Cargas de Peso Propio (1D Automáticas):
(No se calcularon Cargas PP 1D para este análisis)

```

2. PROCESAMIENTO DE CARGAS DE LOSA

(No se registraron distribuciones de carga por losas para este cálculo)

3. ANÁLISIS MATRICIAL

3.1 Logs de Ensamblaje:

--- INICIO: Ensamblaje de Matriz de Rigidez Global ---
 -> Ensamblando Pórtico E1 (Nodos 1-2)
 -> Ensamblando Pórtico E2 (Nodos 2-3)
 --- FIN: Ensamblaje explícito completado. Dimensión de K_global: (18, 18) ---

3.2 Matrices Locales y de Transformación (Ejemplos):

Elemento 1:

Matriz T:

GDL	1	2	3	4	5	6	7	8	9	10	11	12
1	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
6	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
7	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
8	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00
10	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00
11	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00
12	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00

Matriz k_local:

GDL	1	2	3	4	5	6	7	8	9	10	11	12
1	3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	-3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	7.88e+02	0.00e+00	0.00e+00	0.00e+00	1.58e+03	0.00e+00	-7.88e+02	0.00e+00	0.00e+00	0.00e+00	1.58e+03
3	0.00e+00	0.00e+00	1.77e+03	0.00e+00	-3.54e+03	0.00e+00	0.00e+00	0.00e+00	-1.77e+03	0.00e+00	-3.54e+03	0.00e+00
4	0.00e+00	0.00e+00	0.00e+00	9.48e+02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	-9.48e+02	0.00e+00	0.00e+00
5	0.00e+00	0.00e+00	-3.54e+03	0.00e+00	9.45e+03	0.00e+00	0.00e+00	0.00e+00	3.54e+03	0.00e+00	4.72e+03	0.00e+00
6	0.00e+00	1.58e+03	0.00e+00	0.00e+00	0.00e+00	4.20e+03	0.00e+00	-1.58e+03	0.00e+00	0.00e+00	0.00e+00	2.10e+03
7	-3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
8	0.00e+00	-7.88e+02	0.00e+00	0.00e+00	0.00e+00	-1.58e+03	0.00e+00	7.88e+02	0.00e+00	0.00e+00	0.00e+00	-1.58e+03
9	0.00e+00	0.00e+00	-1.77e+03	0.00e+00	3.54e+03	0.00e+00	0.00e+00	0.00e+00	1.77e+03	0.00e+00	3.54e+03	0.00e+00
10	0.00e+00	0.00e+00	0.00e+00	-9.48e+02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	9.48e+02	0.00e+00	0.00e+00
11	0.00e+00	0.00e+00	-3.54e+03	0.00e+00	4.72e+03	0.00e+00	0.00e+00	0.00e+00	3.54e+03	0.00e+00	9.45e+03	0.00e+00
12	0.00e+00	1.58e+03	0.00e+00	0.00e+00	0.00e+00	2.10e+03	0.00e+00	-1.58e+03	0.00e+00	0.00e+00	0.00e+00	4.20e+03

Elemento 2:

Matriz T:

GDL	1	2	3	4	5	6	7	8	9	10	11	12
1	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
6	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
7	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
8	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00
10	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00
11	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00
12	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00

Matriz k_local:

GDL	1	2	3	4	5	6	7	8	9	10	11	12
1	3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	-3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	7.88e+02	0.00e+00	0.00e+00	0.00e+00	1.58e+03	0.00e+00	-7.88e+02	0.00e+00	0.00e+00	0.00e+00	1.58e+03
3	0.00e+00	0.00e+00	1.77e+03	0.00e+00	-3.54e+03	0.00e+00	0.00e+00	0.00e+00	-1.77e+03	0.00e+00	-3.54e+03	0.00e+00
4	0.00e+00	0.00e+00	0.00e+00	9.48e+02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	-9.48e+02	0.00e+00	0.00e+00
5	0.00e+00	0.00e+00	-3.54e+03	0.00e+00	9.45e+03	0.00e+00	0.00e+00	0.00e+00	3.54e+03	0.00e+00	4.72e+03	0.00e+00
6	0.00e+00	1.58e+03	0.00e+00	0.00e+00	0.00e+00	4.20e+03	0.00e+00	-1.58e+03	0.00e+00	0.00e+00	0.00e+00	2.10e+03
7	-3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.15e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
8	0.00e+00	-7.88e+02	0.00e+00	0.00e+00	0.00e+00	-1.58e+03	0.00e+00	7.88e+02	0.00e+00	0.00e+00	0.00e+00	-1.58e+03
9	0.00e+00	0.00e+00	-1.77e+03	0.00e+00	3.54e+03	0.00e+00	0.00e+00	0.00e+00	1.77e+03	0.00e+00	3.54e+03	0.00e+00
10	0.00e+00	0.00e+00	0.00e+00	-9.48e+02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	9.48e+02	0.00e+00	0.00e+00
11	0.00e+00	0.00e+00	-3.54e+03	0.00e+00	4.72e+03	0.00e+00	0.00e+00	0.00e+00	3.54e+03	0.00e+00	9.45e+03	0.00e+00
12	0.00e+00	1.58e+03	0.00e+00	0.00e+00	0.00e+00	2.10e+03	0.00e+00	-1.58e+03	0.00e+00	0.00e+00	0.00e+00	4.20e+03

K_global (Formato Simplificado):

GDL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.15e+05	0	0	0	0	0	-3.15e+05	0	0	0	0	0	0	0	0	0	0	0
2	0	7.88e+02	0	0	0	1.58e+03	0	-7.88e+02	0	0	0	1.58e+03	0	0	0	0	0	0
3	0	0	1.77e+03	0	-3.54e+03	0	0	0	-1.77e+03	0	-3.54e+03	0	0	0	0	0	0	0
4	0	0	0	9.48e+02	0	0	0	0	0	-9.48e+02	0	0	0	0	0	0	0	0
5	0	0	-3.54e+03	0	9.45e+03	0	0	0	3.54e+03	0	4.72e+03	0	0	0	0	0	0	0
6	0	1.58e+03	0	0	0	4.20e+03	0	-1.58e+03	0	0	0	2.10e+03	0	0	0	0	0	0
7	-3.15e+05	0	0	0	0	0	6.30e+05	0	0	0	0	0	-3.15e+05	0	0	0	0	0
8	0	-7.88e+02	0	0	0	-1.58e+03	0	1.58e+03	0	0	0	0	0	-7.88e+02	0	0	0	1.58e+03
9	0	0	-1.77e+03	0	3.54e+03	0	0	0	3.54e+03	0	0	0	0	0	-1.77e+03	0	-3.54e+03	0
10	0	0	0	-9.48e+02	0	0	0	0	0	1.90e+03	0	0	0	0	0	-9.48e+02	0	0
11	0	0	-3.54e+03	0	4.72e+03	0	0	0	0	0	1.89e+04	0	0	0	3.54e+03	0	4.72e+03	0
12	0	1.58e+03	0	0	0	2.10e+03	0	0	0	0	0	8.40e+03	0	-1.58e+03	0	0	0	2.10e+03
13	0	0	0	0	0	0	-3.15e+05	0	0	0	0	0	3.15e+05	0	0	0	0	0
14	0	0	0	0	0	0	0	-7.88e+02	0	0	0	-1.58e+03	0	7.88e+02	0	0	0	-1.58e+03
15	0	0	0	0	0	0	0	0	-1.77e+03	0	3.54e+03	0	0	0	1.77e+03	0	3.54e+03	0
16	0	0	0	0	0	0	0	0	0	-9.48e+02	0	0	0	0	0	9.48e+02	0	0
17	0	0	0	0	0	0	0	0	-3.54e+03	0	4.72e+03	0	0	0	3.54e+03	0	9.45e+03	0
18	0	0	0	0	0	0	0	1.58e+03	0	0	0	2.10e+03	0	-1.58e+03	0	0	0	4.20e+03

4. RESOLUCIÓN POR COMBINACIÓN DE CARGA

=====

4.1 Combinación: "Cálculo Simple"

Caso: "Caso 1"

4.x.1 Sistema de Ecuaciones (Reducido - Completo):
GDL Libres (3): [5, 11, 17]

K_reducida:

GDL	5	11	17
5	9.45e+03	4.72e+03	0.00e+00
11	4.72e+03	1.89e+04	4.72e+03
17	0.00e+00	4.72e+03	9.45e+03

F_reducido (3x1):

GDL 5 : 8.00000e+00
GDL 11 : 0.00000e+00
GDL 17 : -8.00000e+00

4.x.2 Resultados:

Vector Desplazamientos [U] (m, rad) (18x1):

GDL 1 : 0.00000e+00
GDL 2 : 0.00000e+00
GDL 3 : 0.00000e+00
GDL 4 : 0.00000e+00
GDL 5 : 8.46561e-04
GDL 6 : 0.00000e+00
GDL 7 : 0.00000e+00
GDL 8 : 0.00000e+00
GDL 9 : 0.00000e+00
GDL 10 : 0.00000e+00
GDL 11 : 0.00000e+00
GDL 12 : 0.00000e+00
GDL 13 : 0.00000e+00
GDL 14 : 0.00000e+00
GDL 15 : 0.00000e+00
GDL 16 : 0.00000e+00
GDL 17 : -8.46561e-04
GDL 18 : 0.00000e+00

Vector Reacciones [R] (kN, kN-m) (18x1):

```
GDL 1 : 0.0000e+00
GDL 2 : 0.0000e+00
GDL 3 : 9.0000e+00
GDL 4 : 0.0000e+00
GDL 5 : 0.0000e+00
GDL 6 : 0.0000e+00
GDL 7 : 0.0000e+00
GDL 8 : 0.0000e+00
GDL 9 : 3.0000e+01
GDL 10 : 0.0000e+00
GDL 11 : 0.0000e+00
GDL 12 : 0.0000e+00
GDL 13 : 0.0000e+00
GDL 14 : 0.0000e+00
GDL 15 : 9.0000e+00
GDL 16 : 0.0000e+00
GDL 17 : 0.0000e+00
GDL 18 : 0.0000e+00
```

5. FUERZAS INTERNAS POR ELEMENTO (LOCALES)

Modo: Detalle de fuerzas nodales por caso seleccionado

Caso: "Cálculo Simple / Caso 1"

Elemento 1:

	Px	Py	Pz	Mx	My	Mz
Nodo i:	0.0000e+00	0.0000e+00	9.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
Nodo j:	0.0000e+00	0.0000e+00	1.5000e+01	0.0000e+00	1.2000e+01	0.0000e+00

Elemento 2:

	Px	Py	Pz	Mx	My	Mz
Nodo i:	0.0000e+00	0.0000e+00	1.5000e+01	0.0000e+00	-1.2000e+01	0.0000e+00
Nodo j:	0.0000e+00	0.0000e+00	9.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00

6. RESUMEN DE MÁXIMOS GLOBALES (ENVOLVENTE REAL)

Efecto	Valor Máx (+)	Elemento	Caso	Valor Máx (-)	Elemento	Caso
Px	-0.0000 Elem 1	Cálculo Simple/Caso 1		-0.0000 Elem 1	Cálculo Simple/Caso 1	
Py	-0.0000 Elem 1	Cálculo Simple/Caso 1		-0.0000 Elem 1	Cálculo Simple/Caso 1	
Pz	15.0000 Elem 2	Cálculo Simple/Caso 1		-15.0000 Elem 1	Cálculo Simple/Caso 1	
Mx	0.0000 Elem 1	Cálculo Simple/Caso 1		0.0000 Elem 1	Cálculo Simple/Caso 1	
My	6.7488 Elem 1	Cálculo Simple/Caso 1		-12.0000 Elem 2	Cálculo Simple/Caso 1	
Mz	-0.0000 Elem 1	Cálculo Simple/Caso 1		-0.0000 Elem 1	Cálculo Simple/Caso 1	

FIN DEL REPORTE

4.1.1.3 Resultados Ftool

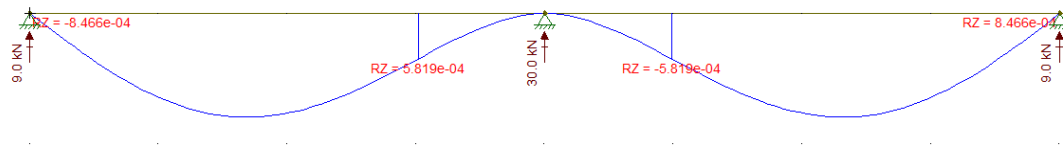


Figura 16: Caso de Estudio 1 - Resultados de Ftool

4.1.1.4 Comparación de Reacciones en Apoyos

	VORTEX 3D	Ftool	Variación %
V1[kN]	9	9	0.00
V2[kN]	30	30	0.00
V3[kN]	9	9	0.00

Tabla 2: Caso de Estudio 1 - Comparación de reacciones

Como se puede ver, las únicas reacciones generadas son Reacciones verticales en los tres nodos. Todas coinciden.

4.1.1.5 Comparación de desplazamientos

En este caso se analizan los *desplazamientos* nodales, mismos que solo ocurren en los nodos 1 y 3, pues aquellas deformaciones producidas en los tramos se anulan completamente en el nodo 2.

	VORTEX 3D	Ftool	Variación %
$\theta 1[\text{rad}]$	-8.47	-8.47	0.00
$\theta 3[\text{rad}]$	8.47	8.47	0.00

Tabla 3: Caso de Estudio 1 - Comparación de desplazamientos

Como se puede ver, las únicas reacciones generadas son Reacciones verticales en los tres nodos, siendo simétricas en los apoyos extremos. Esta simetría se refleja en las deformaciones rotacionales producidas en los mismos apoyos.

Los valores en ambos casos han resultado idénticos.

4.1.2 Caso de Estudio 2: Cercha con carga nodal

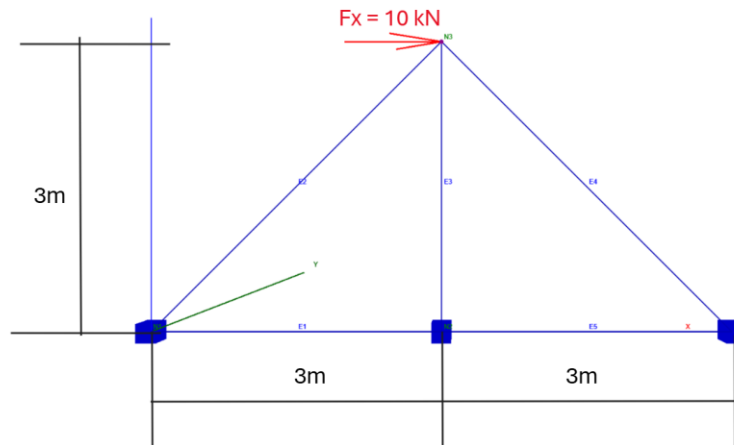


Figura 17: Caso de Estudio 2 – Cercha con Carga Nodal

El Caso 2 se compone de una cercha triangular sometida a una carga nodal horizontal de 10 kN, con el objetivo de evidenciar su desplazamiento lateral, esta tiene una altura de 3 metros y una base de 6 metros. Todos sus apoyos únicamente restringen traslaciones en X y en Z.

4.1.2.1 Resultados VORTEX 3D

En el siguiente texto, proveniente de los reportes generados por VORTEX 3D, se presentan los desplazamientos nodales y las reacciones en los apoyos

--- Desplazamientos (Globales) para 'Caso 2' ---

Nodo Ux Uy Uz Rx Ry Rz

```
-----
1 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 1.4158e-07 0.0000e+00
2 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 5.5826e-06 0.0000e+00
3 5.0400e-05 0.0000e+00 -4.5330e-25 -0.0000e+00 1.6621e-05 0.0000e+00
4 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 1.4158e-07 0.0000e+00
```

--- Reacciones en Apoyos (Globales) ---

Nodo Fx Fy Fz Mx My Mz

```
-----
1 -4.9894 0.0000 -5.0000 0.0000 -0.0000 0.0000
2 -0.0213 0.0000 -0.0000 0.0000 0.0000 0.0000
4 -4.9894 0.0000 5.0000 0.0000 0.0000 0.0000
```

4.1.2.2 Resultados Ftool

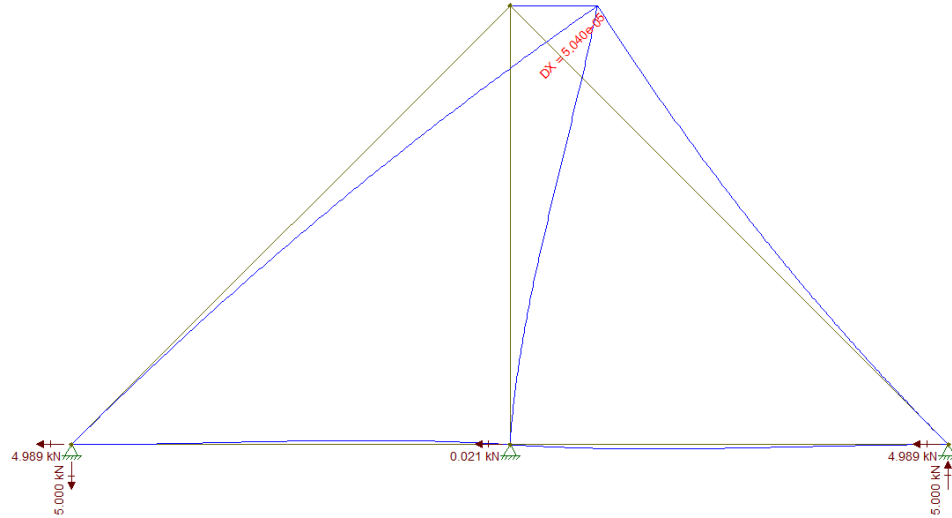


Figura 18: Caso de Estudio 2 - Resultados de Ftool

4.1.2.3 Comparación de Reacciones en Apoyos

	VORTEX 3D	Ftool	Variación %
V1[kN]	-5	-5	0.00
V4[kN]	5	5	0.00
H1[kN]	-4.99	-4.99	0.00
H2[kN]	-0.021	-0.021	0.00
H4[kN]	-4.99	-4.99	0.00

Tabla 4: Caso de Estudio 2 - Comparación de reacciones

4.1.2.3 Comparación de desplazamientos

En este caso se analiza el único desplazamiento lateral en el nodo 3 (superior)

	VORTEX 3D	Ftool	Variación %
$\delta 3[m]$	5.04E-05	5.04E-05	0.00

Tabla 5: Caso de Estudio 2 - Comparación de desplazamientos

Nuevamente, los resultados obtenidos tanto para reacciones como para desplazamientos traslacionales son idénticos en ambos programas.

4.1.3 Caso de Estudio 3: Pórtico Plano con Cargas Distribuidas

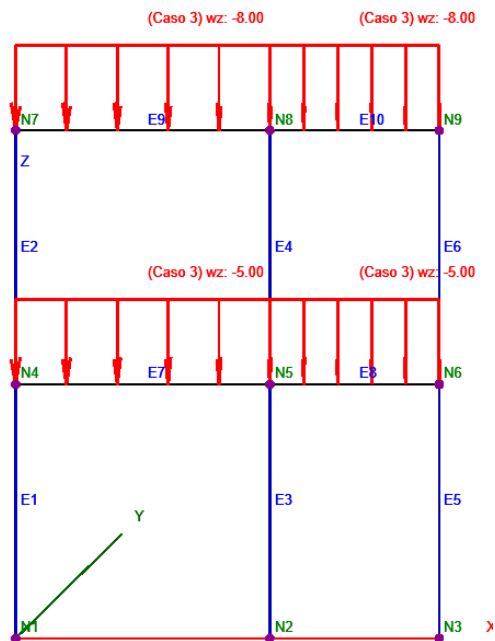


Figura 19: Caso de Estudio 3 – Pórtico Plano con Cargas Distribuidas en Elementos

A continuación, se presentan los datos de entrada extraídos de la memoria de cálculo de VORTEX 3D:

1. DATOS DE ENTRADA

1.1 Nodos:

Nodo X (m) Y (m) Z (m)

1 0.000 0.000 0.000
 2 3.000 0.000 0.000
 3 5.000 0.000 0.000
 4 0.000 0.000 3.000
 5 3.000 0.000 3.000
 6 5.000 0.000 3.000
 7 0.000 0.000 6.000
 8 3.000 0.000 6.000
 9 5.000 0.000 6.000

1.2 Materiales (Propiedades Calculadas):

ID	Tipo	E (kPa)	G (kPa)	A (m ²)	J (m ⁴)	Iy (m ⁴)	Iz (m ⁴)	Avy (m ²)	Avz (m ²)	PE (kN/m ³)
1	rectangular	2.100e+04	8.077e+03	4.000e-02	2.253e-04	1.333e-04	1.333e-04	3.333e-02	3.333e-02	24.00
2	rectangular	2.100e+04	8.077e+03	6.000e-02	4.695e-04	4.500e-04	2.000e-04	5.000e-02	5.000e-02	24.00

1.3 Elementos (Pórtico):

ID	Nodo i	Nodo j	Material	Longitud (m)
1	1	4	1	3.000
2	4	7	1	3.000
3	2	5	1	3.000
4	5	8	1	3.000
5	3	6	1	3.000
6	6	9	1	3.000
7	4	5	2	3.000
8	5	6	2	2.000
9	7	8	2	3.000
10	8	9	2	2.000

1.4 Losas (Definición):

ID	Nodos	Distribución	Eje Uni.	Espesor (m)	PE (kN/m ³)
----	-------	--------------	----------	-------------	-------------------------

1.5 Apoyos (R=Restringido, L=Libre):

Nodo	Tx	Ty	Tz	Rx	Ry	Rz
1	R	R	R	R	R	R
2	R	R	R	R	R	R
3	R	R	R	R	R	R

4.1.3.1 Resultados VORTEX 3D

--- Desplazamientos (Globales) para 'Caso 3' ---

Nodo Ux Uy Uz Rx Ry Rz

4 2.6702e-05 0.0000e+00 -6.3038e-05 0.0000e+00 2.0266e-04 0.0000e+00
 5 2.8534e-05 0.0000e+00 -1.3268e-04 0.0000e+00 -7.8384e-05 0.0000e+00
 6 2.9071e-05 0.0000e+00 -3.6428e-05 0.0000e+00 -6.8069e-05 0.0000e+00
 7 1.4019e-04 0.0000e+00 -1.0104e-04 0.0000e+00 4.7013e-04 -0.0000e+00
 8 1.3754e-04 0.0000e+00 -2.1642e-04 0.0000e+00 -1.7751e-04 -0.0000e+00
 9 1.3674e-04 0.0000e+00 -5.7540e-05 -0.0000e+00 -1.2872e-04 0.0000e+00

--- Reacciones en Apoyos (Globales) ---

Nodo Fx Fy Fz Mx My Mz

1 0.3451 0.0000 17.6506 0.0000 0.3285 0.0000

2 -0.1818 0.0000 37.1496 0.0000 -0.1996 0.0000
 3 -0.1632 0.0000 10.1997 0.0000 -0.1813 0.0000

Informe de Envolventes:

6. RESUMEN DE MÁXIMOS GLOBALES (ENVOLVENTE REAL)

Efecto	Valor Máx (+)	Elemento	Caso	Valor Máx (-)	Elemento	Caso
Px	0.7696	Elem 7	Cálculo Simple/Caso 3	-37.1496	Elem 3	Cálculo Simple/Caso 3
Py	0.6133	Elem 4	Cálculo Simple/Caso 3	-1.1146	Elem 2	Cálculo Simple/Caso 3
Pz	10.6411	Elem 9	Cálculo Simple/Caso 3	-13.3589	Elem 9	Cálculo Simple/Caso 3
Mx	0.0000	Elem 1	Cálculo Simple/Caso 3	0.0000	Elem 1	Cálculo Simple/Caso 3
My	5.1551	Elem 9	Cálculo Simple/Caso 3	-5.9983	Elem 9	Cálculo Simple/Caso 3
Mz	1.9216	Elem 2	Cálculo Simple/Caso 3	-1.4223	Elem 2	Cálculo Simple/Caso 3

Según el informe de envolventes, el valor máximo de Momento flector tanto negativo, como positivo se encuentra en el elemento 9, resultando de la siguiente manera:

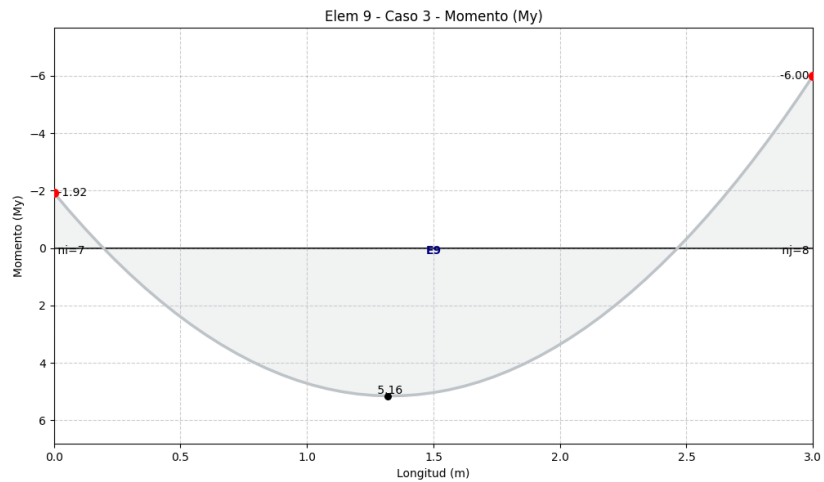


Figura 20: Caso de Estudio 3 – Momento máximo de la estructura

4.1.3.1 Resultados Ftool

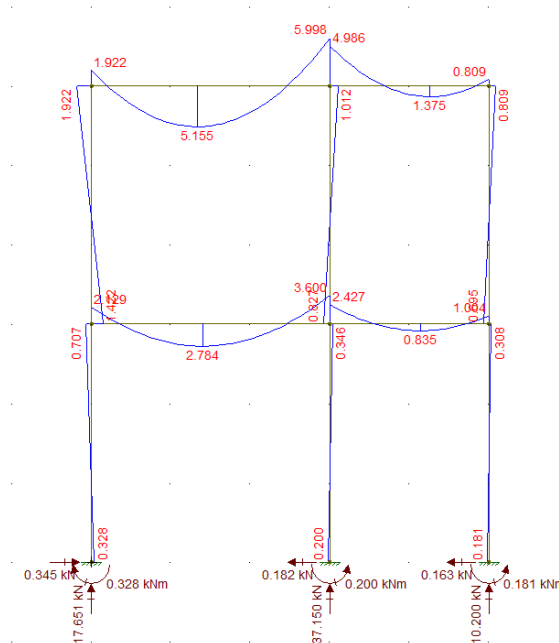


Figura 21: Caso de Estudio 3 – Resultados Ftool

4.1.2.3 Comparación de Reacciones en Apoyos

	VORTEX 3D	Ftool	Variación %
V1[kN]	17.65	17.65	0.00
V2[kN]	37.15	37.15	0.00
V3[kN]	10.2	10.2	0.00
H1[kN]	0.35	0.35	0.00
H2[kN]	-0.18	-0.18	0.00
H3[kN]	-0.16	-0.16	0.00
M1[kN*m]	0.32	0.32	0.00
M2[kN*m]	-0.2	-0.2	0.00
M3[kN*m]	-0.18	-0.18	0.00

Tabla 6: Caso de Estudio 3 - Comparación de resultantes

4.1.2.3 Comparación de Momentos máximos

	VORTEX 3D	CYPE 3D	Variación %
Mmax+	5.16	5.16	0.00
Mmax-	-6	-6	0.00

Tabla 7: Caso de Estudio 3 - Comparación de Momentos máximos

Nuevamente, todos los valores dan una precisión del 100%

4.2 Resultados Nivel II: Validación Espacial y Distribución de Cargas

Se evalúa el desempeño del motor de cálculo en entornos tridimensionales (6 GDL por nudo) y la precisión del algoritmo de áreas tributarias para losas, considerando la deformación por corte (Timoshenko). Debido a lo extenso del procedimiento, en esta sección solamente se compararán Momentos y Cortantes máximos en la viga más solicitada. La comparación se realiza con Cype 3D

4.2.1 Caso de Estudio 4: Pórtico 3D con cargas distribuidas sobre elementos

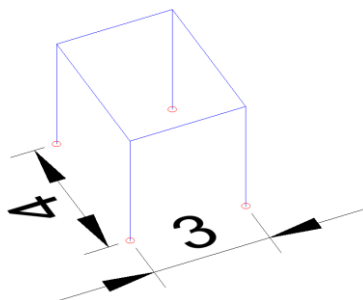


Figura 22: Caso de Estudio 4 – Dimensiones del pórtico

Nota. Se consideran 3 metros de altura

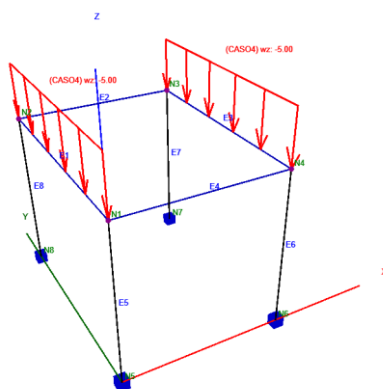


Figura 23: Caso de Estudio 4 – Estructura Cargada

La geometría de las secciones es la siguiente:

ID	Tipo	Desc.	E	v	b	h
1	Rectangular	Vig	2.10e+07	0.300	0.2000	0.3000
2	Rectangular	Col	2.10e+07	0.300	0.2000	0.2000

Figura 24: Caso de Estudio 4 – Geometría de las secciones

Valores calculados de las propiedades de la sección:

ID	Tipo	E (kPa)	G (kPa)	A (m ²)	J (m ⁴)	Iy (m ⁴)	Iz (m ⁴)	Avy (m ²)	Avz (m ²)
1	rectangular	2.10E+07	8.08E+06	6.00E-02	4.70E-04	4.50E-04	2.00E-04	5.00E-02	5.00E-02
2	rectangular	2.10E+07	8.08E+06	6.00E-02	4.70E-04	4.50E-04	2.00E-04	5.00E-02	5.00E-02

4.2.1.1 Elementos más solicitados

6. RESUMEN DE MÁXIMOS GLOBALES (ENVOLVENTE REAL)

Efecto	Valor Máx (+)	Elemento	Caso	Valor Máx (-)	Elemento	Caso
Px	0.0000	Elem 4	Cálculo Simple/CAS04	-10.0000	Elem 5	Cálculo Simple/CAS04
Py	0.0000	Elem 2	Cálculo Simple/CAS04	-0.0000	Elem 1	Cálculo Simple/CAS04
Pz	10.0000	Elem 1	Cálculo Simple/CAS04	-10.0000	Elem 1	Cálculo Simple/CAS04
Mx	0.0000	Elem 2	Cálculo Simple/CAS04	-0.0000	Elem 5	Cálculo Simple/CAS04
My	7.0768	Elem 3	Cálculo Simple/CAS04	-2.9232	Elem 1	Cálculo Simple/CAS04
Mz	0.0000	Elem 2	Cálculo Simple/CAS04	-0.0000	Elem 2	Cálculo Simple/CAS04

Figura 25: Caso de Estudio 4 – Elementos más solicitados

4.2.1.2 Diagrama de Momentos de VORTEX 3D

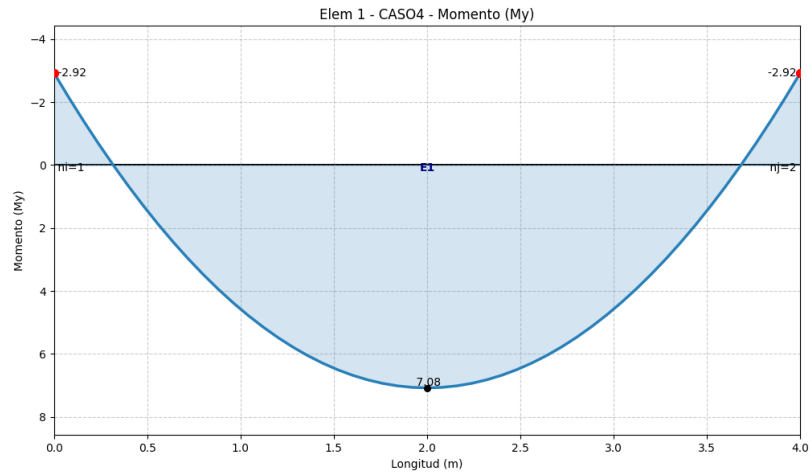


Figura 26: Caso de Estudio 4 – Momentos VORTEX 3D

4.2.1.3 Diagrama de Momentos de CYPE 3D

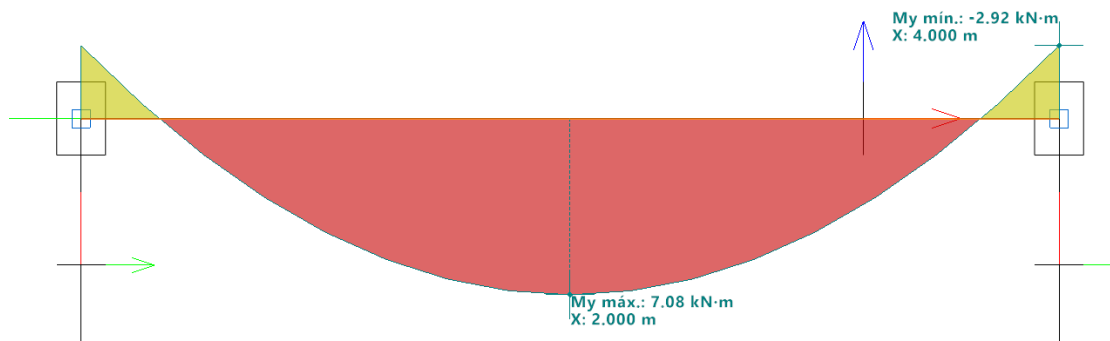


Figura 27: Caso de Estudio 4 – Momentos CYPE 3D

4.2.1.4 Comparación de Momentos

Los valores de CYPE 3D y VORTEX 3D son idénticos, estos consideran teoría de timoshenko y son los siguientes:

	VORTEX 3D	CYPE 3D	Variación %
M_{max+} [kN*m]	7.08	7.08	0.00
M_{max-} [kN*m]	-2.92	-2.92	0.00

Tabla 8: Caso de Estudio 4 - Comparación de Momentos máximos

4.2.1.5 Verificación con Teoría de Bernoulli

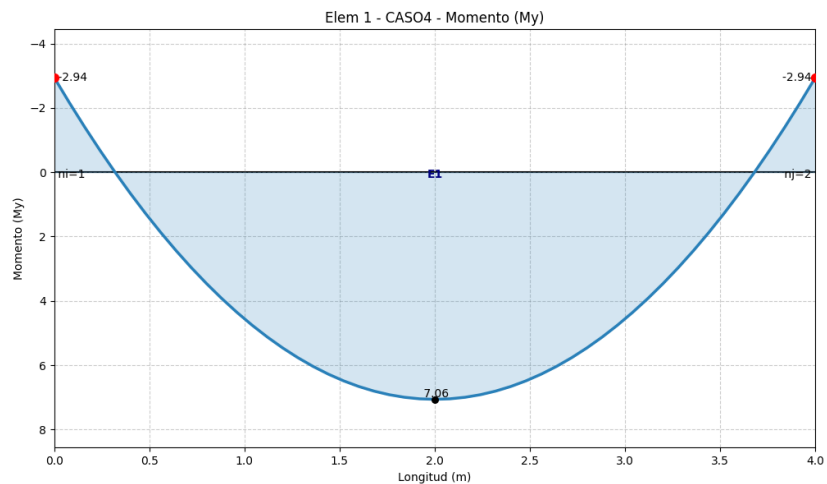


Figura 28: Caso de Estudio 4 – Momentos con teoría de Bernoulli en VORTEX 3D

Los momentos resultantes ahora tienen valores de $M_{max+} = 7.06 \text{ kN} \cdot \text{m}$ y $M_{max-} = -2.94 \text{ kN} \cdot \text{m}$, resultando en una leve desviación de resultados que según Timoshenko se hace más evidente en elementos de gran canto.

4.2.2 Caso de Estudio 5: Pórtico 3D con distribución de cargas superficiales en losa bidireccional con cuatro bordes simétricos

Este se basa en la misma estructura anterior, sin embargo, esta vez la definición de cargas se realiza mediante una losa bidireccional

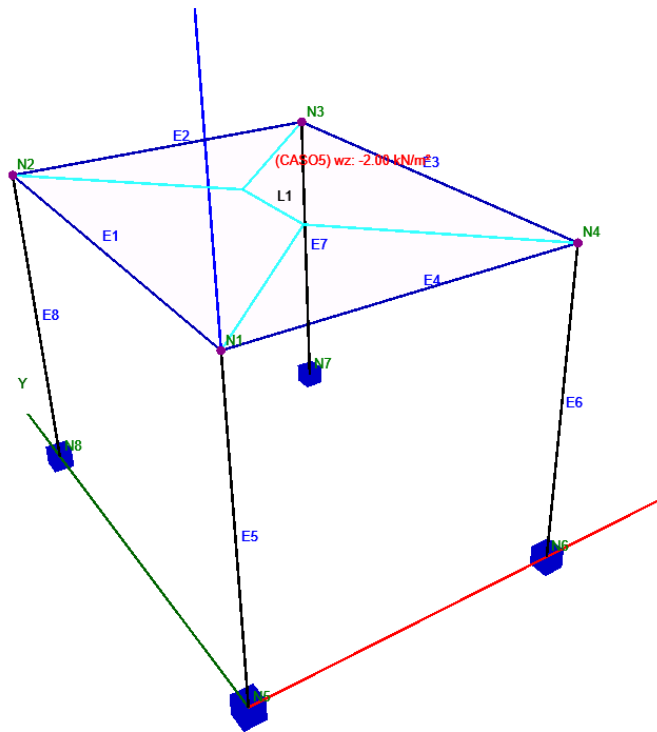


Figura 29: Caso de Estudio 5 – Estructura con cargas superficiales distribuidas en losas bidireccionales

4.2.2.1 Análisis de geometría tributaria

El distribuidor de losas integrado de VORTEX 3D, se encarga de definir las áreas tributarias según la teoría de líneas de fluencia analizando condiciones de borde en losas bidireccionales. En la losa 1 (L1) de la figura 27, se tienen 4 bordes discontinuos, por lo que todos los ángulos serán de 45° por simetría.

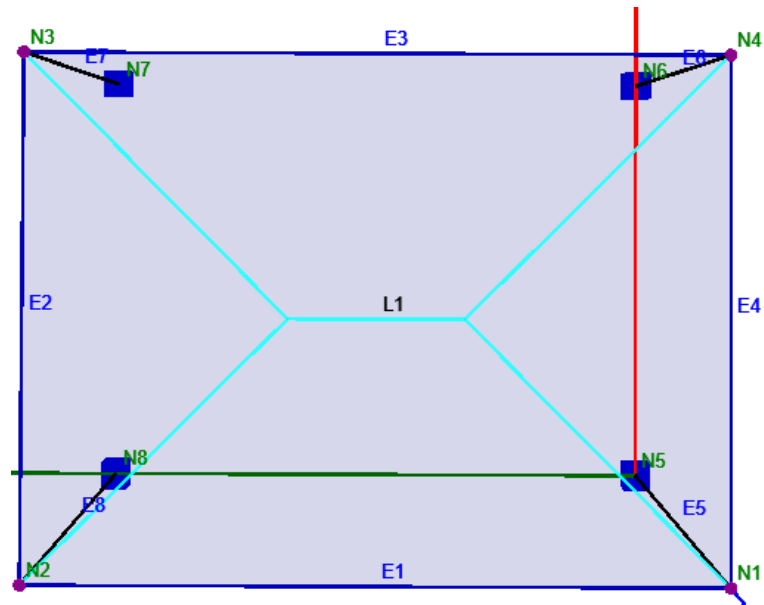


Figura 30: Caso de Estudio 5 – Distribución de cargas superficiales en losas bidireccionales con 3 bordes discontinuos y 1 continuo

Luego se desarrolla la siguiente lógica de distribución, extraída del reporte generado por VORTEX 3D:

2. PROCESAMIENTO DE CARGAS DE LOSA

```
[Losa 1 - bidireccional]: Iniciando Distribución (wz = -2.000 kPa)
-> Procesando Viga ID: 4 (Borde 1-4)
- Long. Viga: 3.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.50000, -3.00000), (1.00000, 0.00000)
-> Procesando Viga ID: 3 (Borde 4-3)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.37500, -3.00000), (0.62500, -3.00000), (1.00000, 0.00000)
-> Procesando Viga ID: 2 (Borde 3-2)
- Long. Viga: 3.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.50000, -3.00000), (1.00000, 0.00000)
-> Procesando Viga ID: 1 (Borde 2-1)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.37500, -3.00000), (0.62500, -3.00000), (1.00000, 0.00000)
[Losa 1 - bidireccional]: Fin de distribución.
```

Es así que obtenemos valores de cargas variables trapezoidales y triangulares que serán entregadas a procesador_cargas para obtener los FEPs que estas descargan en los nodos

que correspondan. Las longitudes a las que varía la carga están expresadas en una fracción de la longitud de la viga a la que tributa, es decir, para la viga 1, la carga llega a -3 kN/m en $x'=0.375L$ donde L es 4 metros, por tanto este punto de carga se encuentra a 1.5m de n_i (nodo inicial).

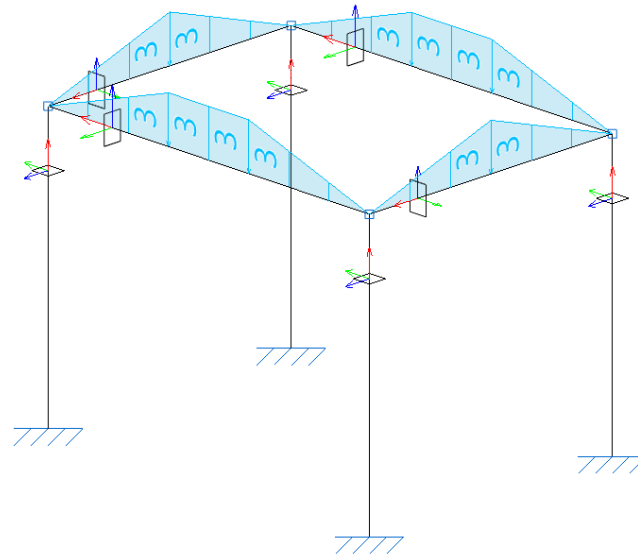


Figura 31: Caso de Estudio 5 – Idealización de la distribución de cargas lineales provenientes de Losas Bidireccionales

4.2.2.2 Diagrama de Momentos de VORTEX 3D

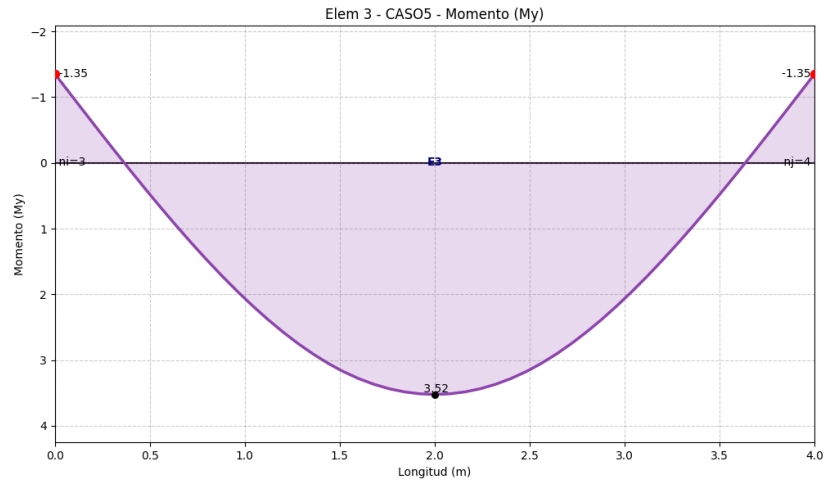


Figura 32: Caso de Estudio 5 – Momentos VORTEX 3D

4.2.2.3 Diagrama de Momentos de CYPE 3D

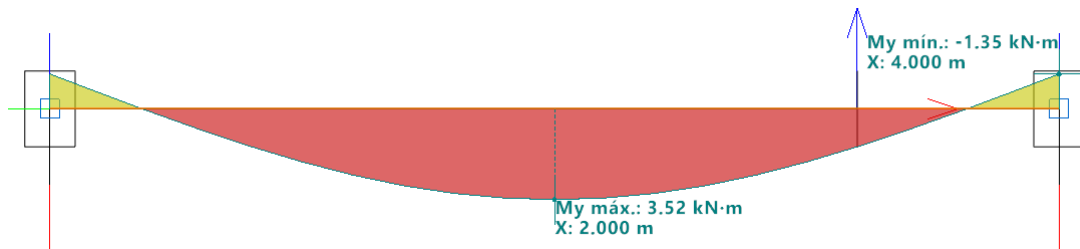


Figura 33: Caso de Estudio 5 – Momentos CYPE 3D

Los diagramas de momentos son idénticos, esto indica que el procesador de cargas triangulares y trapezoidales funciona correctamente.

	VORTEX 3D	CYPE 3D	Variación %
Mmax+[kN*m]	3.52	3.52	0.00
Mmax-[kN*m]	-1.35	-1.35	0.00

Tabla 9: Caso de Estudio 4 - Comparación de Momentos máximos

4.2.3 Caso de Estudio 6: Pórtico 3D con distribución de cargas superficiales en losa bidireccional con 1 borde asimétrico

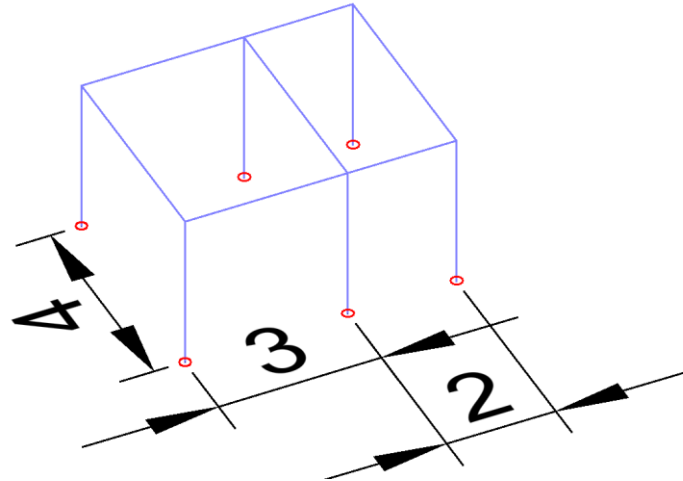


Figura 34: Caso de Estudio 6 – Dimensiones – Altura: 3m

El caso 6 es una variante del caso 5 en el que se añade una losa continua a uno de los lados para poder evidenciar la distribución de cargas en bordes asimétricos.

4.2.3.1 Análisis de geometría tributaria

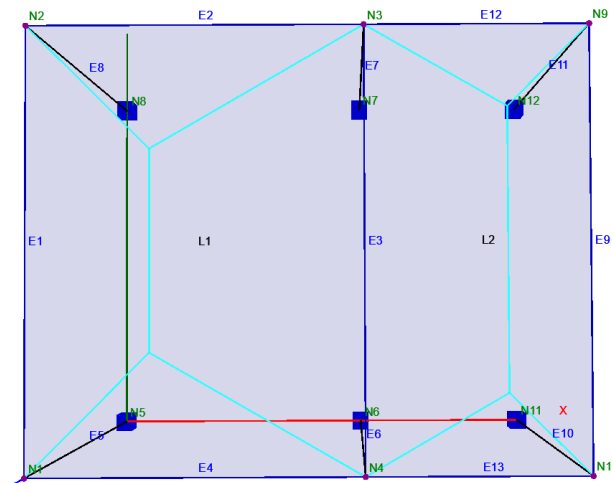


Figura 35: Caso de Estudio 6 – Distribución con un borde asimétrico en ambas losas

Se carga cype 3D en función de la siguiente distribución:

2. PROCESAMIENTO DE CARGAS DE LOSA

```
-----
[Losa 1 - bidireccional]: Iniciando Distribución (wz = -2.000 kPa)
-> Procesando Viga ID: 4 (Borde 1-4)
- Long. Viga: 3.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.36603, -2.19615), (1.00000, 0.00000)
-> Procesando Viga ID: 3 (Borde 4-3)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.27452, -3.80385), (0.72548, -3.80385), (1.00000, 0.00000)
-> Procesando Viga ID: 2 (Borde 3-2)
- Long. Viga: 3.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.63397, -2.19615), (1.00000, 0.00000)
-> Procesando Viga ID: 1 (Borde 2-1)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.27452, -2.19615), (0.72548, -2.19615), (1.00000, 0.00000)
[Losa 1 - bidireccional]: Fin de distribución.
-----
[Losa 2 - bidireccional]: Iniciando Distribución (wz = -2.000 kPa)
-> Procesando Viga ID: 13 (Borde 4-10)
- Long. Viga: 2.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.63397, -1.46410), (1.00000, 0.00000)
-> Procesando Viga ID: 9 (Borde 10-9)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.18301, -1.46410), (0.81699, -1.46410), (1.00000, 0.00000)
-> Procesando Viga ID: 12 (Borde 9-3)
- Long. Viga: 2.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.36603, -1.46410), (1.00000, 0.00000)
-> Procesando Viga ID: 3 (Borde 3-4)
- Long. Viga: 4.00 m
- Carga Aplicada: tramos_locales (Eje: z)
- Puntos (p_norm, q): (0.00000, 0.00000), (0.18301, -2.53590), (0.81699, -2.53590), (1.00000, 0.00000)
[Losa 2 - bidireccional]: Fin de distribución.
-----
```

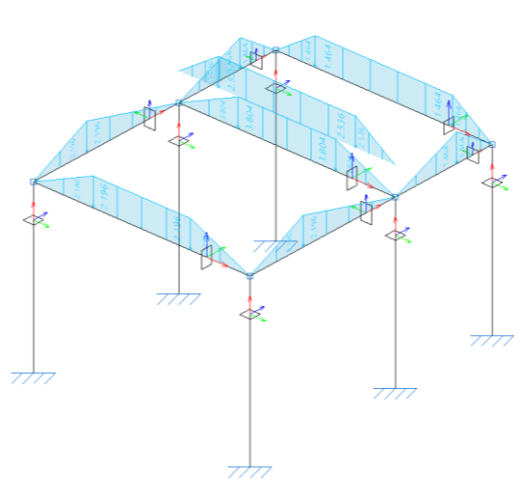


Figura 36: Caso de Estudio 6 – Idealización de la distribución de cargas lineales provenientes de Losas Bidireccionales

Aquí a simple vista puede saberse que el elemento 3 será el más solicitado, pues este tiene dos losas tributarias cargándole simultáneamente.

4.2.3.2 Diagrama de momentos de VORTEX 3D

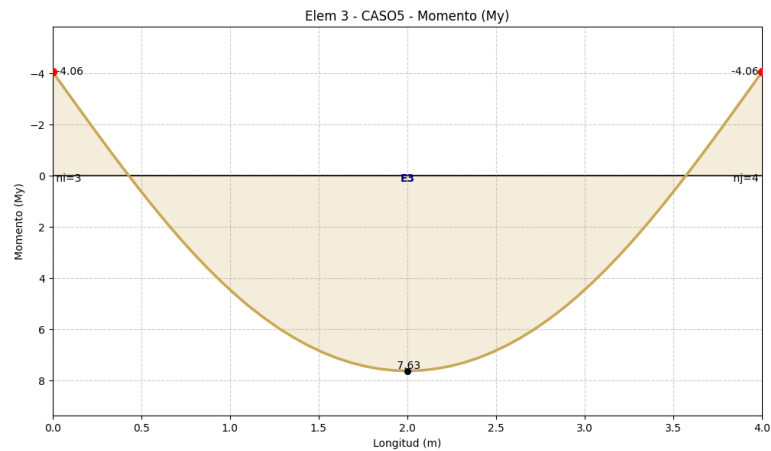


Figura 37: Caso de Estudio 6 –Momentos VORTEX 3D

4.2.3.3 Diagrama de momentos de CYPE 3D

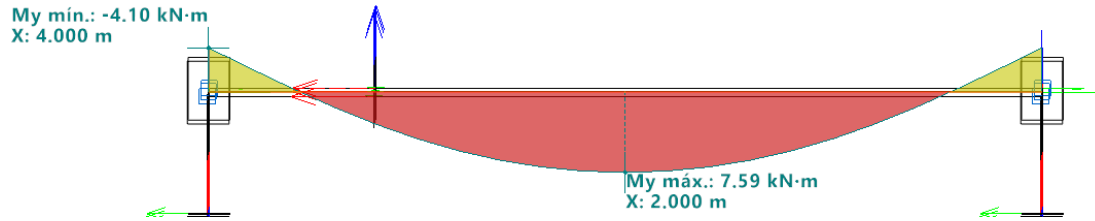


Figura 38: Caso de Estudio 6 –Momentos CYPE 3D

En este caso si se tiene una diferencia entre los resultados de VORTEX 3D y CYPE 3D, esta podría deberse a la cantidad de decimales que permite ingresar CYPE 3D, ya que este tipo de cargas suelen ser *números irracionales* como lo muestra el procedimiento de distribución de cargas superficiales con 1 tramo asimétrico.

	VORTEX 3D	CYPE 3D	Variación %
Mmax+[kN*m]	7.63	7.59	0.52%
Mmax-[kN*m]	4.06	4.10	0.98%

Tabla 10: Caso de Estudio 6 - Comparación de Momentos máximos

4.3 Resultados Nivel III: Validación Integral en Edificación

Presentación de resultados para el "Edificio Multifamiliar Concordia", comparando el desempeño de VORTEX 3D frente al estándar industrial CYPECAD bajo condiciones de carga normativas (NB 1225002) presentadas en el Anexo 5.

4.3.1 Descripción del Caso de Estudio

El edificio concordia cuenta con Planta Baja, 4 plantas tipo y una azotea accesible. Este fue modelado en VORTEX 3D y en CYPECAD a partir de los planos arquitectónicos presentes en el Anexo 6.

La profundidad de desplante de 2 metros fue obtenida en base al estudio de suelos presente en el Anexo 4.

4.3.2 Estructuras Modeladas

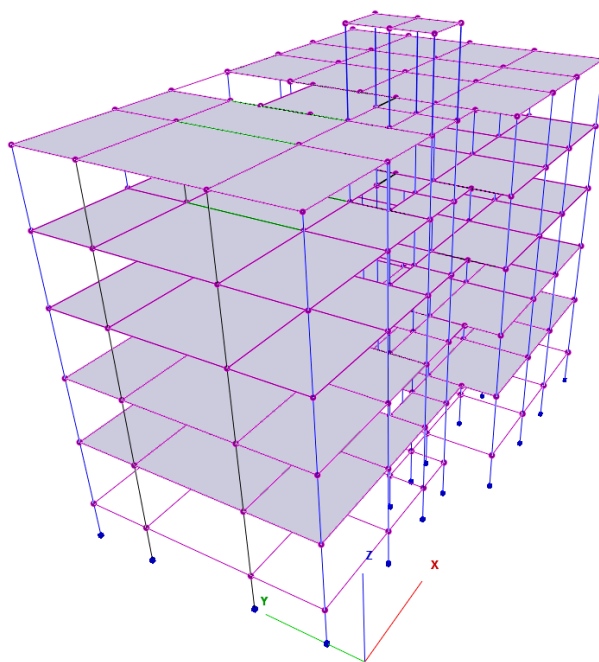


Figura 39: Estructura modelada en VORTEX 3D

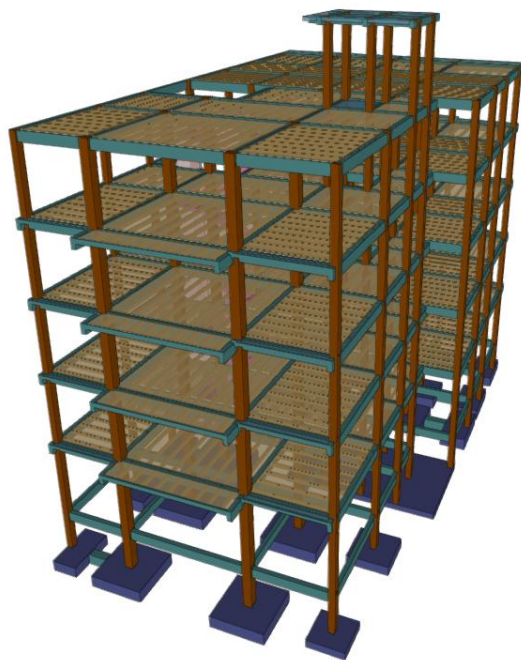


Figura 40: Estructura modelada en CYPECAD

4.3.3 Identificación de efectos máximos

VORTEX 3D otorga la posibilidad de identificar efectos máximos de la estructura según combinación y elemento

6. RESUMEN DE MÁXIMOS GLOBALES (ENVOLVENTE REAL)							
Efecto	Valor Máx (+)	Elemento	Caso	Valor Máx (-)	Elemento	Caso	
Px	30.2097	Elem 564	Granizo Cubierta/Granizo; Muerta;..	-1239.1256	Elem 31	Granizo Sobrecarga/Granizo; Muert..	
Py	19.8125	Elem 42	Granizo Sobrecarga/Granizo; Muert..	-24.5219	Elem 12	Granizo Sobrecarga/Granizo; Muert..	
Pz	150.2919	Elem 309	Granizo Sobrecarga/Granizo; Muert..	-150.3131	Elem 309	Granizo Sobrecarga/Granizo; Muert..	
Mx	16.0513	Elem 605	Granizo Sobrecarga/Granizo; Muert..	-15.2920	Elem 332	Granizo Sobrecarga/Granizo; Muert..	
My	105.8195	Elem 36	Granizo Sobrecarga/Granizo; Muert..	-143.0882	Elem 309	Granizo Sobrecarga/Granizo; Muert..	
Mz	51.7021	Elem 12	Granizo Sobrecarga/Granizo; Muert..	-41.1213	Elem 42	Granizo Sobrecarga/Granizo; Muert..	

Figura 41: Efectos máximos Edificio Concordia

Por lo que los efectos máximos serán verificados en estos elementos.

4.3.3 Verificación de Axial máximo en columnas

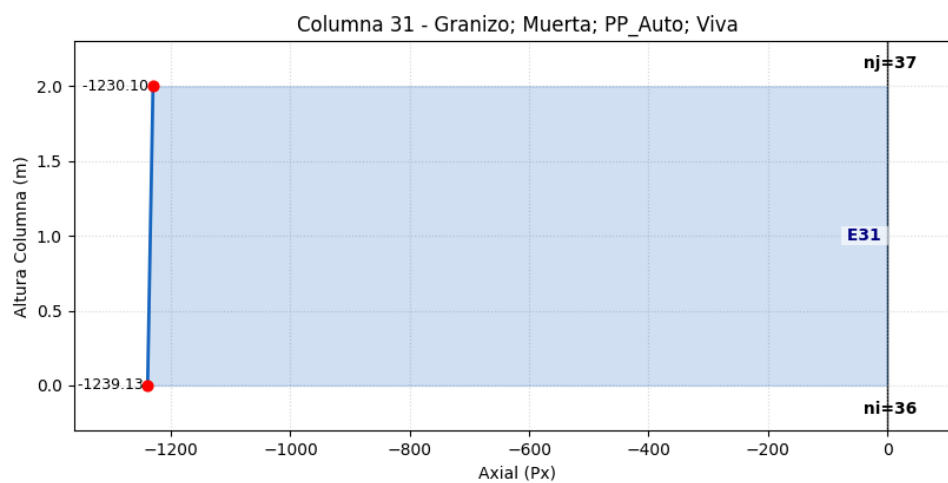


Figura 42: Cargas axiales máximas en VORTEX 3D

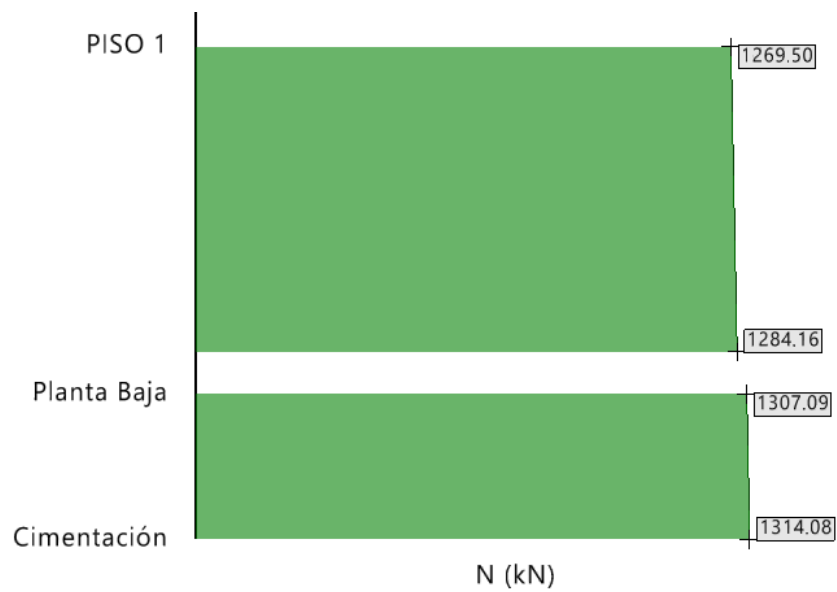


Figura 43: Carga axial máxima en CYPECAD

4.3.4 Verificación de Envoltentes de Diseño (vigas)

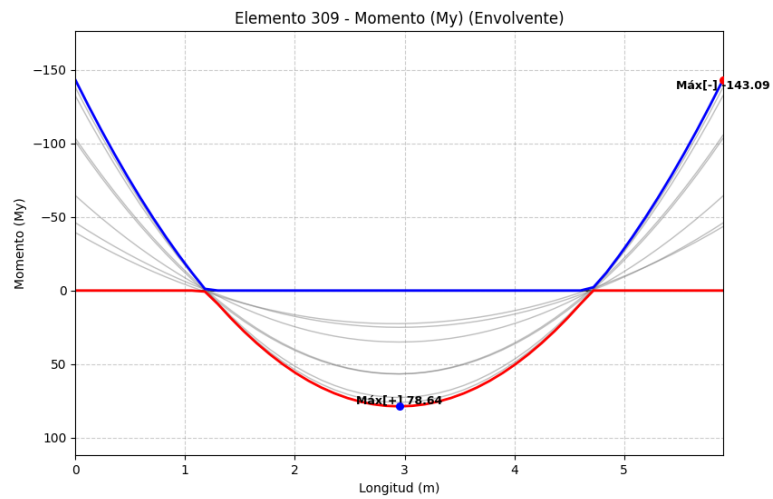


Figura 44: Envoltente de Momentos máximos en VORTEX 3D

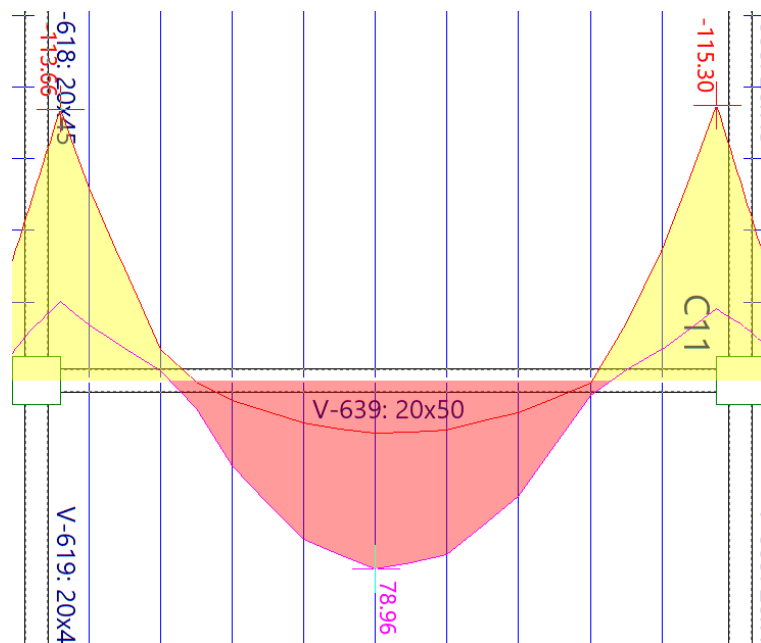


Figura 45: Envoltente de Momentos máximos en CYPECAD

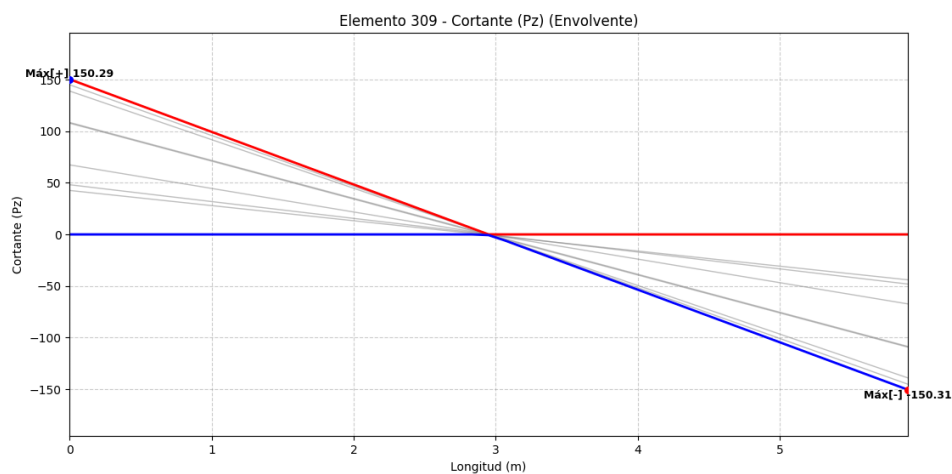


Figura 46: Envolvente de cortantes máximos en VORTEX 3D

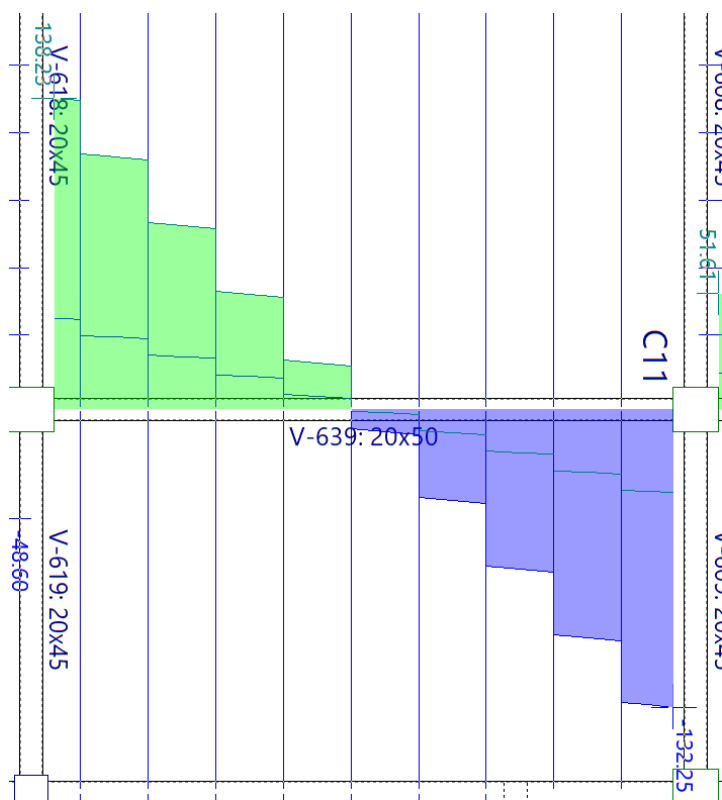


Figura 47: Envolvente de cortantes máximos en CYPECAD

Nota. $V_{max+}=138.23$ $V_{max-}=-132.25$

	VORTEX 3D	CYPECAD	Variación %	Mayor
Nmax	1239.13	1314.08	5.70	CYPECAD
Mmax+	78.64	78.96	0.41	CYPECAD
Mmax-	143.09	115.30	19.42	VORTEX 3D
Vmax+	150.29	138.23	8.02	VORTEX 3D
Vmax-	150.31	132.25	12.02	VORTEX 3D

Tabla 11: Comparación Final VORTEX 3D / CYPECAD

Como se puede observar, existe diferencia entre resultados, sin embargo, estas no son considerables y son mutuamente coherentes.

4.4 Análisis de Resultados

El proceso de validación del software VORTEX 3D, estructurado en tres niveles de complejidad creciente, ha permitido caracterizar con precisión el desempeño del motor de cálculo desarrollado. A partir de los datos expuestos en las secciones precedentes, se presentan los siguientes hallazgos:

4.4.1 Precisión en Mecánica Fundamental y Geometrías Simples

En el Nivel I (sistemas planos) y los primeros casos del Nivel II (pórticos espaciales simétricos), VORTEX 3D demostró una correlación exacta con el software de control (Ftool y CYPE 3D), lo cual valida de manera concluyente la correcta implementación de los algoritmos de la Matriz de Rigidez Directa, las matrices de transformación de coordenadas y la resolución de sistemas de ecuaciones lineales. Se confirma así la robustez del código desarrollado, pues las variaciones respecto al software comercial desaparecen cuando se simplifican las hipótesis (análisis en ejes), garantizando que la base matemática de VORTEX 3D funciona correctamente.

4.4.2 Sensibilidad en la Distribución de Cargas Bidireccionales

En el análisis de losas bidireccionales asimétricas (Caso de Estudio 6), se identificó una leve divergencia (entre 0.52% y 0.98%) en los momentos flectores respecto a

CYPE 3D. El análisis detallado sugiere que esta discrepancia no es un fallo del método matricial, sino una consecuencia de la sensibilidad numérica en la entrada de datos.

La teoría de líneas de fluencia genera cargas trapezoidales y triangulares cuyos vértices y magnitudes suelen implicar números irracionales. Al introducir estos valores en software comercial, la limitación en la cantidad de decimales significativos aceptados genera un truncamiento que, acumulado en múltiples elementos, produce variaciones en los esfuerzos finales. VORTEX 3D, al procesar estas magnitudes internamente con una precisión elevada, gracias a Python, mantiene la integridad del valor teórico, lo que explica la diferencia observada.

4.4.3 Comportamiento en Edificaciones Complejas (Edificio Concordia)

En el Nivel III, la comparación con CYPECAD para una edificación completa reveló una correlación satisfactoria. Las diferencias observadas se mantienen en un rango establecido: menos del 6% para cargas axiales y un máximo del 19.42% para momentos flectores negativos. Estas variaciones son técnicamente justificables por las distintas hipótesis de modelado asumidas por cada software:

1. **Rigidez de Nudos:** CYPECAD considera automáticamente zonas rígidas en las intersecciones viga-columna (brazos rígidos), reduciendo la luz libre de cálculo y aumentando la rigidez lateral. VORTEX 3D realiza un análisis estrictamente a ejes, sin considerar la dimensión finita del nudo. Esto resulta en una estructura teóricamente más flexible y, por tanto, genera resultados más conservadores (mayores desplazamientos y esfuerzos en ciertos elementos), lo cual es favorable desde el punto de vista de la seguridad estructural.
2. **Coefficiente de Rigidez Axial:** Se identificó que CYPECAD aplica de manera predeterminada un coeficiente de rigidez axial (típicamente igual a 2) en los pilares para mitigar los efectos del acortamiento elástico en el cálculo global. VORTEX 3D utiliza la rigidez axial teórica sin factores de amplificación. La presencia de este parámetro en el software comercial explica la tendencia a obtener esfuerzos axiales mayores en sus reportes, confirmando que la

aproximación entre ambos programas aumenta cuando este coeficiente se ajusta a 1.

3. **Coeficientes de redistribución:** CYPECAD como software de diseño de elementos estructurales, incluye en sí varias consideraciones constructivas en las que entran los coeficientes de redistribución de negativos y coeficientes de empotramiento de última planta, mismos que hacen variar los resultados finales en vigas.
4. **Metodología de cálculo:** A diferencia de VORTEX 3D, CYPECAD utiliza el método de elementos finitos en la mayor parte de sus determinaciones estructurales, esto le brinda mayor precisión cuando se tienen en cuenta elementos bidimensionales como losas.

En conclusión, VORTEX 3D se valida como una herramienta de alta precisión, manteniendo las desviaciones dentro de márgenes de seguridad coherentes y explicables por la teoría estructural.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

En correspondencia con los objetivos planteados al inicio de esta investigación y tras el análisis de los resultados obtenidos, se establecen las siguientes conclusiones:

- Cumplimiento del Objetivo General: Se ha desarrollado exitosamente VORTEX 3D, consolidándola como una herramienta de código abierto para el análisis estructural en tres dimensiones. Se demuestra que VORTEX 3D cumple su propósito de proveer transparencia y trazabilidad en los resultados, permitiendo la auditoría de procesos que generalmente permanecen ocultos en software comercial.
- Implementación de Fundamentos Teóricos (Obj. Esp. 1): Por medio de las pruebas de Nivel 1 y Nivel 2, se verificó la correcta implementación de la teoría del Método Matricial de Rigidez en un entorno computacional tridimensional, mediante la formulación de la matriz de rigidez elemental 12×12 que incluye efectos de deformación por cortante mediante la teoría de Timoshenko y matrices de rotación espacial para cualquier orientación geométrica.
- Desarrollo del Motor de Cálculo (Obj. Esp. 2): El motor de cálculo desarrollado en Python, apoyado en la librería NumPy para la realización de operaciones de álgebra lineal, demostró estabilidad y eficiencia en el ensamblaje de la matriz de rigidez global y la resolución del sistema fundamental de ecuaciones, garantizando una correcta imposición de condiciones de borde y la obtención de vectores de desplazamientos y reacciones en estructuras de múltiples grados de libertad.
- Distribución Geométrica de Cargas (Obj. Esp. 3): Se implementaron correctamente rutinas para la transferencia de cargas superficiales de losas a vigas en base a la teoría de áreas tributarias y líneas de fluencia. Los algoritmos permiten simular la distribución de cargas trapezoidales y triangulares de forma automatizada, facilitando el modelado de edificaciones complejas sin necesidad de cálculo manual de cargas sobre las barras.

- Validación y Confiabilidad (Obj. Esp. 4): El protocolo de validación comparativa frente a software patrón (CYPE 3D y Ftool) confirmó la alta confiabilidad de VORTEX 3D. Los resultados arrojan una correlación exacta en sistemas planos y variaciones controladas en modelos tridimensionales complejos, con desviaciones menores al 20% en momentos flectores críticos y una exactitud superior al 94% en cargas axiales. Estas diferencias se atribuyen a las hipótesis de modelado a ejes y a consideraciones constructivas que toma el software patrón (CYPECAD), lo que posiciona a la herramienta como un instrumento de verificación seguro y conservador.
- Documentación y Reportabilidad (Obj. Esp. 5): El módulo generador de reportes cumple con el requisito de transparencia documental al desglosar la memoria de cálculo detallada. A diferencia de las herramientas comerciales, VORTEX 3D permite al usuario inspeccionar matrices intermedias y vectores de carga, consolidando su valor como herramienta pedagógica y de auditoría profesional.

5.2 Recomendaciones

- Utilizar VORTEX 3D como herramienta pedagógica complementaria en las materias como Análisis Estructural y Hormigón Armado. Al ofrecer acceso al código fuente y a los procesos de cálculo intermedios, el software permite a los estudiantes comprender los fundamentos teóricos de forma ágil y práctica.
- Se aconseja a futuros desarrolladores mantener la estructura modular del código, especialmente en la definición de materiales y secciones, para facilitar la integración de nuevos materiales (como acero estructural o madera) sin alterar el núcleo del motor de rigidez.
- Para futuras versiones, se recomienda desarrollar e implementar elementos finitos de tipo placa o lámina con grados de libertad de flexión y membrana. Esto permitiría modelar losas de hormigón armado no solo como elementos de carga, sino como componentes estructurales que aportan rigidez al diafragma

horizontal, mejorando la simulación del comportamiento global de la estructura.

- Se recomienda la Mejora de la Interfaz Gráfica para facilitar la introducción de datos en estructuras de gran envergadura, incluyendo herramientas de replicación de niveles y visualización 3D más interactiva.
- Se recomienda incorporar la funcionalidad para definir constantes de resorte en los nudos de cimentación. Esto permitiría simular la interacción suelo-estructura, ofreciendo un análisis más realista de las cimentaciones sobre suelos deformables, superando la limitación de los apoyos infinitamente rígidos.
- Se recomienda utilizar VORTEX 3D como una herramienta de verificación en proyectos de ingeniería civil. Tras los hallazgos de esta investigación, se evidencia la utilidad de contar con un motor de cálculo independiente para auditar los "criterios de diseño" automáticos de software comercial (como los coeficientes de rigidez axial o empotramientos), garantizando que las decisiones de diseño se basen en un entendimiento profundo de la mecánica elástica.