

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Ciencias de la Tierra**

Modificación de los Modos de Vibración de Dos Edificaciones Aporticadas  
de Hormigón Armado con Irregularidad en Planta para Reducir la  
Respuesta Torsional

### **PROYECTO DE GRADUACIÓN**

Previo la obtención del Título de:

**Máster en Ingeniería Civil con Mención en Estructuras  
Sismorresistentes**

Presentado por:

Ángela María Gracia Villao Maruri

GUAYAQUIL - ECUADOR

Año: 2026

## **DEDICATORIA**

A todas las personas que han formado parte de mi vida y de alguna manera han contribuido con la mujer que soy ahora. Dedico este trabajo especialmente a mis padres, por formarme para ser un adulto que aporte a donde vaya, por enseñarme los valores que poseo y por estar presentes en mis victorias y fracasos.

## **AGRADECIMIENTOS**

Quiero expresar mi sincero agradecimiento a mi tutor, Daniel Gómez Pizano, por su apoyo y orientación a lo largo de este proceso. Su conocimiento y experiencia han sido fundamentales para el desarrollo de esta investigación.

Asimismo, a todos aquellos que han contribuido directa o indirectamente a este proyecto, les agradezco por su apoyo y aliento.

## **DECLARACIÓN EXPRESA**

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; *Angela Maria Gracia Villao Maruri*, doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

---

Angela Villao Maruri

## **EVALUADORES**

.....  
**Nadia Rosaura Quijano**

.....  
**Samantha Hidalgo Astudillo**

**Natividad García Troncoso**

**Daniel Gómez Pizano**

## RESUMEN

La respuesta torsional en edificaciones aporticados de hormigón armado con irregularidad en planta tipo 1 se asocia a excentricidades entre el centro de masa y el centro de rigidez, lo que genera incremento de derivas en pórticos extremos y rotaciones bajo acción sísmica, condición que puede evidenciarse en los modos de vibración del sistema. En esta investigación se tiene como objetivo modificar los modos de vibración fundamentales en dos estructuras aporticadas de hormigón armado con irregularidad en planta tipo 1, con el fin de mejorar su comportamiento estructural ante cargas sísmicas mediante un algoritmo de optimización basado en análisis modal. Se plantea como hipótesis que una redistribución localizada de rigideces en el sistema de resistencia sísmica, orientada a reducir la excentricidad estática, disminuye la respuesta torsional y favorece modos dominantes traslacionales, aportando además a la eficiencia del dimensionamiento estructural. Para ello, se modelaron dos edificaciones y se caracterizó su comportamiento dinámico inicial mediante análisis modal. Posteriormente, se implementó un procedimiento de optimización que ajustó de forma localizada la rigidez de elementos verticales, empleando Optimización por Enjambre de Partículas (PSO) y evaluando parámetros estructurales conforme a los criterios normativos aplicables. Como resultado, se obtuvieron reducciones de la excentricidad estática y de indicadores asociados a la torsión, junto con una modificación de participación modal, con predominio de las masas participativas traslacionales en los modos principales, evidenciándose mejoras en la respuesta global frente a cargas sísmicas en comparación con los modelos originales. En conclusión, la optimización basada en análisis modal permite controlar el efecto torsional mediante la rigidización estratégica de los elementos verticales y, por ende, el ajuste de sus modos predominantes, contribuyendo a una respuesta sísmica más favorable sin requerir incrementos generalizados de secciones, lo que la convierte en una alternativa útil para el ajuste del comportamiento dinámico en etapa de diseño o evaluación.

**Palabras Clave:** torsión; análisis modal; optimización por enjambre de partículas (PSO); excentricidad estática; irregularidad en planta.

## **ABSTRACT**

*Torsional response in reinforced-concrete moment-resisting frame buildings with Type 1 plan irregularity is associated with eccentricities between the center of mass and the center of rigidity, which increases drifts in perimeter frames and rotations under seismic loading—an effect that can be observed in the system’s vibration modes. This research aims to modify the fundamental vibration modes of two reinforced-concrete frame structures with Type 1 plan irregularity in order to improve their structural behavior under seismic actions by means of an optimization algorithm based on modal analysis. The hypothesis is that a localized redistribution of stiffness in the seismic-force-resisting system, aimed at reducing static eccentricity, decreases torsional response and promotes dominant translational modes, while also improving the efficiency of structural sizing. To this end, two buildings were modeled and their initial dynamic behavior was characterized through modal analysis. Subsequently, an optimization procedure was implemented to locally adjust the stiffness of vertical elements, using Particle Swarm Optimization (PSO) and evaluating structural parameters in accordance with applicable code-based criteria. As a result, reductions in static eccentricity and torsion-related indicators were achieved, together with changes in modal participation, with translational effective modal mass predominating in the main modes, demonstrating improved overall response to seismic loading compared with the original models. In conclusion, modal-analysis-based optimization makes it possible to control torsional effects through the strategic stiffening of vertical elements and, consequently, the adjustment of dominant modes, contributing to a more favorable seismic response without requiring generalized increases in member sizes, making it a useful alternative for tuning dynamic behavior at the design or assessment stage.*

*Keywords: seismic torsion; modal analysis; eccentricity; Particle Swarm Optimization (PSO); plan irregularity.*

# ÍNDICE GENERAL

EVALUADORES.....	5
RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS .....	VI
SIMBOLOGÍA .....	VII
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS .....	XI
1. Introducción.....	1
1.1 Antecedentes.....	2
1.2 Problemática por resolver .....	4
1.3 Justificación .....	6
1.4 Objetivos.....	7
1.4.1 Objetivo General .....	7
1.4.2 Objetivos Específicos .....	7
2. Desarrollo del proyecto .....	8
2.1 Marco conceptual .....	8
2.1.1 Configuración estructural .....	8
2.1.2 Propiedades dinámicas de las estructuras.....	10
2.1.3 Análisis de estructuras en 3D.....	12
2.1.4 Análisis Sísmico de Sistemas Lineales .....	15
2.1.5 Algoritmo de optimización global en Matlab .....	18
2.1.5.1 Algoritmo de Optimización por Enjambre de Partículas (PSO).....	18
2.2 Marco metodológico .....	19

2.3	Definición de las edificaciones de estudio .....	22
2.3.1	Descripción de los modelos estructurales .....	22
2.3.2	Casos de análisis .....	23
2.3.3	Dimensiones de los elementos estructurales .....	24
2.3.4	Propiedades mecánicas del modelo estructural .....	25
2.3.5	Cargas aplicadas en los casos de análisis .....	25
2.3.6	Modos de vibración y Porcentajes de Masa Participativa de las estructuras de estudio .....	28
2.3.7	Cálculo de Relación de Irregularidad torsional (RIT) .....	31
2.3.8	Cálculo del Centro de Rigidez .....	32
2.4	Cálculo del Centro de Masa .....	37
2.5	Cálculo de la excentricidad .....	38
2.6	Desarrollo e implementación del algoritmo de optimización para dimensiones de columnas .....	39
2.6.1	Definición de la función objetivo .....	40
2.6.2	Variables de diseño y restricciones .....	42
2.6.3	Algoritmo de optimización .....	43
2.6.4	Convergencia de resultados .....	43
2.6.5	Verificación de la solución: Cálculo de RIT y excentricidad de las estructuras de estudio optimizadas .....	47
3.	RESULTADOS Y DISCUSIÓN .....	48
3.1	Resultados .....	48
3.1.1	Comparación de ubicación del C.R. obtenidos mediante el Método Analítico y el PSO .....	48
3.1.2	Secciones de columnas propuestas por el algoritmo de optimización ...	51
3.1.3	Comparación de derivas máximas de piso .....	57

3.1.4	Comparación de fuerzas internas en elementos .....	60
3.1.5	Comparación de demanda de cuantías de acero en elementos .....	67
3.1.6	Comparación de modos de vibración mediante porcentajes de masa participativa .....	71
3.1.7	Comparación de excentricidades estáticas .....	74
3.1.8	Comparación de la Relación de Irregularidad Torsional (RIT) .....	76
3.2	Análisis de Resultados.....	79
3.2.1	Implicaciones económicas de la optimización .....	80
4.	Conclusiones Y Recomendaciones.....	82
4.1	Conclusiones .....	82
4.2	Recomendaciones .....	84
BIBLIOGRAFÍA.....		86
ANEXOS .....		89

## ABREVIATURAS

NEC-15	Norma Ecuatoriana de la Construcción 2015
C.M.	Centro de Masa
C.R.	Centro de Rigidez
XCM	Coordenada X del Centro de Masa
YCM	Coordenada Y del Centro de Masa
XCR	Coordenada X del Centro de Rigidez
YCR	Coordenada Y del Centro de Rigidez
RIT	Relación de Irregularidad Torsional
PSO	Optimización por Enjambre de Partículas (Particle Swarm Optimization)
3D	Análisis/trabajo en tres dimensiones
CV	Coeficiente de variación (medida de dispersión; usado para controlar la variabilidad de dimensiones entre columnas de un mismo eje)
MDOF	Multiple Degrees of Freedom (sistema de múltiples grados de libertad).
PMT	Porcentaje de Masa Participativa Torsional.
GDL	Grado(s) de libertad.
SRSS	Square Root of the Sum of the Squares (raíz cuadrada de la suma de los cuadrados).
CQC	Complete Quadratic Combination (combinación cuadrática completa).

## SIMBOLOGÍA

$m$	Metro
$cm$	Centímetro
$Tonf$	Tonelada fuerza
$kgf$	Kilogramo fuerza
$Hz$	Hertz (unidad de frecuencia)
$seg$	Segundos (unidad de período)
$f$	Frecuencia natural
$\xi$	Porcentaje de amortiguamiento
$\Gamma_n$	factor de participación modal del modo $n$
$n$	número total de modos (en combinación modal)
$u(t)$	respuesta de desplazamientos del sistema, dependiente del tiempo
$\Phi_P$	Factor de irregularidad en planta
$A_x$	Factor de amplificación torsional
$\Phi$	Matriz modal (contiene los modos de vibración)
$\phi_n$	modo de vibración $n$ -ésimo
$q(t)$	Coordenadas modales (ecuaciones de movimiento modales); dependen del tiempo
$S_a$	Ordenada del espectro de desplazamientos para un período $T_n$ y amortiguamiento modal definido
$S_d$	Ordenada del espectro de aceleraciones para un período $T_n$ y amortiguamiento modal definido
$\omega_n$	Frecuencia (angular) asociada al modo $n$
$r$	Respuesta combinada de todos los modos
$r_i, r_j$	Respuestas máximas asociadas a los modos $i$ y $j$ (para CQC)
$\rho_{ij}$	Coefficiente de correlación entre modos $i$ y $j$
$\beta_{ij}$	Relación entre las frecuencias de los modos $i$ y $j$ .
$M$	Masa total del sistema
$i$	Vector que indica la dirección en la que se aplica el sismo al sistema
$M_n$	Masa asociada a cada grado de Libertad del sistema

$\Delta X_A, \Delta Y_A$	Desplazamiento en dirección "X" y "Y" del punto "A" del pórtico, respectivamente.
$\theta_{zz}$	Rotación del diafragma alrededor del eje "Z".
$\Delta X, \Delta Y$	Desplazamiento en dirección "X" y "Y" en el Centro de Masa del diafragma, respectivamente.
b	Ancho de la sección transversal de la columna
h	Profundidad de la sección transversal de la columna
x	Vector de variables de diseño (dimensiones de ancho b y profundidad h de la sección transversal de las columnas de la estructura).
$N_p$	número de pisos de la edificación.
$e_x$	Excentricidad en dirección X
$e_y$	Excentricidad en dirección Y
$w_{exc}$	peso asociado al término de excentricidades.
$w_{PMT}$	peso asociado al porcentaje de masa participativa torsional.
$w_{der}$	peso asociado al término de derivas en cada dirección.
$w_{CV}$	peso asociado al término de variabilidad de dimensiones de columnas dentro de un mismo eje.
$P_{ex}$	penalización de la excentricidad en la dirección X
$P_{ey}$	penalización de la excentricidad en la dirección Y
$P_{PMT}$	penalización del porcentaje de masa participativa torsional
$P_{\Delta x}$	penalización de la deriva en la dirección X
$P_{\Delta y}$	penalización de la deriva en la dirección Y
$P_{CV}$	penalización asociada a la variabilidad entre dimensiones de columnas de un mismo eje

## ÍNDICE DE FIGURAS

Figura 2.1 Irregularidad en planta tipo 1 según NEC-15.....	9
Figura 2.2 Excentricidad estática debido a diferencia de coordenadas entre centro de masa y centro de resistencia o rigidez.....	11
Figura 2.3 Representación visual del comportamiento del Algoritmo por PSO. Obtenido en la web. ....	19
Figura 2.4 Vista en planta y en 3D de modelo estructural Caso 1. ....	23
Figura 2.5 Vista en planta y en 3D de modelo estructural Caso 2. ....	24
Figura 2.6 Espectro de Respuesta de Aceleraciones NEC-15. ....	27
Figura 2.7 Representación visual de los resultados de C.R. por el método Analítico y PSO para el Caso 1 (Edificio de 3 pisos).....	36
Figura 2.8 Representación visual de los resultados de C.R. por el método Analítico y PSO para el Caso 2 (Edificio de 6 pisos).....	36
Figura 2.9 Excentricidades por piso, Caso 1. ....	38
Figura 2.10 Excentricidades por piso, Caso 2. ....	39
Figura 2.11 Función de costo de la OPT de dimensiones de columnas, caso 1.....	44
Figura 2.12 Función de costo de la OPT de dimensiones de columnas, caso 2.....	44
Figura 2.13 Convergencia del algoritmo PSO para dimensiones de columna del Caso 1.....	45
Figura 2.14 Convergencia del algoritmo PSO para dimensiones de columna del Caso 2.....	46
Figura 3.1 Ubicación del Centro de Masa y Centro de Rigidez en cada piso del Caso 1.....	49
Figura 3.2 Ubicación del Centro de Masa y Centro de Rigidez en cada piso del Caso 2.....	50
Figura 3.3 Comparación de Áreas de Columnas, caso 1.....	53
Figura 3.4 Comparación de Áreas de Columnas, caso 2.....	53
Figura 3.5 Comparación de Secciones de Columnas antes y después de la optimización, caso 1. ....	54

Figura 3.6 Comparación de Secciones de Columnas antes y después de la optimización, caso 2. ....	55
Figura 3.7 Comparación de derivas máximas inelásticas de cada piso, caso 1. ....	58
Figura 3.8 Comparación de derivas máximas inelásticas de cada piso, caso 2. ....	59
Figura 3.9 Comparación de las excentricidades antes y después de la optimización, caso 1. ....	74
Figura 3.10 Comparación de las excentricidades antes y después de la optimización, caso 2. ....	75
Figura 3.12 Comparación de RIT antes y después de la optimización, caso 1.....	77
Figura 3.13 Comparación de RIT antes y después de la optimización, caso 2.....	78

## ÍNDICE DE TABLAS

Tabla 2.2.1. Dimensiones de elementos estructurales de Caso 1. ....	24
Tabla 2.2.2. Dimensiones de elementos estructurales de Caso 2. ....	24
Tabla 2.2.3. Propiedades de los materiales de los elementos estructurales. ....	25
Tabla 2.2.4. Peso propio más sobrecarga muerta consideradas en los modelos estructurales. ....	26
Tabla 2.2.5. Cargas vivas consideradas en los modelos estructurales.....	26
Tabla 2.2.6. Parámetros para obtención del espectro de diseño de la NEC-15. ....	28
Tabla 2.2.7. Resultados de modos de vibración y masa participativa iniciales, Caso 1. ....	29
Tabla 2.2.8. Resultados de modos de vibración y masa participativa iniciales, Caso 2. ....	30
Tabla 2.2.9. Coordenadas del C.R. por el método analítico del modelo Caso 1.....	34
Tabla 2.2.10. Coordenadas del C.R por el método analítico del modelo Caso 2.....	34
Tabla 2.2.11. Coordenadas del C.M. del modelo Caso 1. ....	37
Tabla 2.2.12. Coordenadas del C.M del modelo Caso 2. ....	37
Tabla 3.1 Centros de Rigidez de Edificio Caso 1. Método Analítico. ....	48
Tabla 3.2 Centros de Rigidez de Edificio Caso 2. Método Analítico. ....	48
Tabla 3.3 Comparación de secciones de Columnas, Caso 1. ....	51
Tabla 3.4 Comparación de secciones de Columnas, caso 2. ....	52
Tabla 3.5 Fuerzas internas en columnas del primer piso antes y después de la optimización, caso 1. ....	62
Tabla 3.6 Fuerzas internas en vigas del primer piso antes y después de la optimización, caso 1. ....	63
Tabla 3.7 Fuerzas internas en columnas del primer piso antes y después de la optimización, caso 2. ....	65
Tabla 3.8 Fuerzas internas en vigas del primer piso antes y después de la optimización, caso 2. ....	66
Tabla 3.9 Cuantías de acero longitudinal en columnas, caso 1.....	67
Tabla 3.10 Cuantías de acero a flexión en vigas, caso 1.....	69
Tabla 3.11 Cuantías de acero longitudinal en columnas, caso 2.....	70

Tabla 3.12 Cuantías de acero a flexión en vigas, caso 2.....	70
Tabla 3.13 Porcentajes de masa participativa antes y después de la optimización, caso 1.....	72
Tabla 3.14 Porcentajes de masa participativa antes y después de la optimización, caso 2.....	73

# CAPÍTULO 1

## 1. INTRODUCCIÓN

Los países con alto riesgo sísmico, como Ecuador, ubicado en el denominado Cinturón del Pacífico, requieren que los diseños de sus edificaciones sean apropiados para resistir cargas inducidas por sismos. En este contexto, es fundamental comprender los modos de vibración, que son las formas características en las que una estructura oscila cuando es sometida a fuerzas dinámicas como los sismos.

Con el fin de reducir el impacto que los sismos pueden ocasionar en las estructuras, se ha desarrollado una filosofía de diseño para edificaciones sismo resistentes que obedece a ciertos criterios aplicables durante su vida útil. Estos incluyen que las estructuras resistan sismos de baja intensidad sin sufrir daños, sismos de intensidad moderada con daños mínimos en elementos estructurales, y que las estructuras no esenciales resistan sismos severos con daños significativos en sus elementos estructurales sin llegar al colapso, mientras que las estructuras esenciales resistan sismos severos sin sufrir daños importantes (Awad, 2012).

Para cumplir estos criterios, la Norma Ecuatoriana de la Construcción vigente (NEC-15), recomienda no utilizar configuraciones estructurales que presentan irregularidades ya sea en planta como en elevación (Ministerio de Desarrollo Urbano y Vivienda, 2014a). Sin embargo, las irregularidades en planta y elevación a menudo surgen por necesidades arquitectónicas o funcionales, lo que puede llevar a un comportamiento inadecuado de las estructuras bajo cargas sísmicas. Las edificaciones con irregularidades en planta o elevación tienden a presentar un menor desempeño en comparación con aquellas que tienen configuraciones simétricas y regulares, aumentando así la probabilidad de daños severos e incluso colapsos durante sismos (Özmen et al., 2014).

Un aspecto crítico de las irregularidades estructurales es su tendencia a generar excentricidades, que producen rotaciones importantes alrededor del eje vertical. Estas rotaciones torsionales pueden producir esfuerzos adicionales en el sistema resistente a

cargas laterales y pueden comprometer la integridad de la edificación (Balaji Rao et al., 2022).

Ante esta situación, la presente investigación se centra en la evaluación y modificación de los modos de vibración en estructuras con irregularidades torsionales. A través de simulaciones numéricas, se buscará caracterizar el comportamiento dinámico de estas estructuras, identificando los modos de vibración predominantes y desarrollando una optimización en la geometría de las columnas para transformar los modos torsionales en traslacionales.

Esta investigación no solo contribuirá al conocimiento en el ámbito de la ingeniería civil, sino que también responde a la necesidad de diseñar estructuras más seguras y eficientes. Además, se alinea con los Objetivos de Desarrollo Sostenible (ODS), específicamente con el ODS 11 "Ciudades y comunidades sostenibles" y el ODS 9 "Industria, innovación e infraestructura", promoviendo comunidades resilientes y una infraestructura sólida (Asamblea General ONU, 2015).

## **1.1 Antecedentes**

Existe evidencia de que edificios con irregularidades en su configuración estructural, ya sea en planta, como en elevación, han exhibido un comportamiento inadecuado o un desempeño deficiente durante eventos sísmicos en distintas partes del mundo como Venezuela, Nicaragua, México, Chile, Haití, Europa y Asia (Arnold & Reitherman, 1982).

Ejemplos notables incluyen:

- El terremoto de Caracas en 1967, donde edificios con irregularidades en planta sufrieron daños severos.
- El sismo de Managua en 1972, que reveló la vulnerabilidad de estructuras con pisos blandos.
- El terremoto de México en 1985, donde edificios con irregularidades en elevación colapsaron.
- El terremoto de Chile en 2010, que afectó significativamente a estructuras con asimetrías.

Estos casos han enfatizado la necesidad de comprender la respuesta de las edificaciones con características irregulares ante fuerzas sísmicas (Awad, 2012).

Cuando una edificación es sometida a fuerzas sísmicas, las cuales son de tipo dinámicas, se desarrollan fuerzas de inercia en la estructura, que se concentran en un punto denominado “centro de masa”. A su vez, las fuerzas sísmicas representadas como fuerzas externas horizontales son resistidas por los elementos estructurales verticales de la edificación. Su resultante es concentrada en un punto de la estructura denominado “centro de rigidez” (Naveen E et al., 2019).

Cuando la ubicación del centro de masa y el centro de rigidez de una estructura se encuentran alejados, se producen excentricidades, lo que resulta en efectos torsionales (Naveen E et al., 2019). Esta situación es particularmente problemática, ya que las torsiones estructurales son una de las causas más relevantes de daños severos e incluso colapsos durante eventos sísmicos (De-la-Colina & Almeida, 2004).

Es importante recalcar que los principales factores que contribuyen en el desplazamiento entre las ubicaciones de los centros de masa y rigidez son las irregularidades geométricas en planta y elevación. Estas son producidas por distribuciones no uniformes de masas y rigideces, es decir, el arreglo de los elementos estructurales y no estructurales dentro de la edificación. Las estructuras irregulares que presentan torsión tendrán el modo torsional como modo fundamental, que representa la forma de vibración predominante (De la Llera & Chopra, 1994). Esto es significativo debido a que las torsiones estructurales son una de las causas más relevantes que conducen a daños severos en estructuras, e inclusive, el colapso (Özmen et al., 2014).

Algunos autores han realizado investigaciones enfocadas en comprender los efectos de las irregularidades estructurales bajo cargas sísmicas. Guevara et al., (1992) analizaron la influencia de la forma en planta de las edificaciones en la respuesta a terremotos. Su investigación concluyó que las edificaciones con plantas en forma de “L” o “H” debían ser subdivididos en bloques de formas regulares separados por juntas sísmicas. Valmundsson & Nau, (1997) estudiaron la respuesta sísmica de estructuras aporricadas con irregularidades en elevación. Su conclusión fue que un decremento del 30% en las rigideces de piso generaba un incremento entre 20-40% en las derivas de piso. Gokdemir

et al., (2013), investigaron el efecto de las irregularidades torsionales en el desempeño sísmico de estructuras. Ellos corroboraron que la magnitud del momento torsor dependía de la excentricidad estática. Estos estudios han proporcionado valiosas perspectivas sobre cómo las irregularidades afectan el comportamiento dinámico de las estructuras y han sentado las bases para investigaciones posteriores (Stathopoulos & Anagnostopoulos, 2005).

El presente trabajo se centrará en desarrollar una estrategia para garantizar que los modos de vibración que más contribuyen a la respuesta de las edificaciones de estudio no sean torsionales, sino traslacionales.

## **1.2 Problemática por resolver**

Los países con alto riesgo sísmico, como Ecuador, que se encuentra dentro del denominado Cinturón de Fuego del Pacífico, requieren que los diseños de sus edificaciones sean apropiados para resistir cargas inducidas por sismos. La filosofía de diseño sismo resistente establece criterios específicos para la vida útil de las estructuras, incluyendo la resistencia a sismos de diversas intensidades con diferentes niveles de daño aceptable (Beauval et al., 2018). Para cumplir estos criterios, la Norma Ecuatoriana de la Construcción (NEC 2015) recomienda no utilizar configuraciones estructurales con irregularidades en planta o elevación (Villalobos et al., 2018).

Se ha demostrado que el comportamiento de estructuras irregulares posee un desempeño inferior frente al comportamiento de estructuras simétricas y regulares durante sismos (Sim et al., 2017). Las irregularidades estructurales promueven el desplazamiento entre centros de masa y rigidez, alejándolos entre sí y generando rotaciones significativas alrededor del eje vertical de la edificación. Estas rotaciones producen esfuerzos adicionales en el sistema que pueden conducir a daños en la estructura (Makarios, 2008).

Las torsiones son generalmente inducidas en una estructura debido a necesidades arquitectónicas (Naveen E et al., 2019). Entonces, eliminar las fuentes de esfuerzos torsionales es en ocasiones difícil en diseños arquitectónicos, donde se busca el confort

y la funcionalidad de en la edificación. Una de estas situaciones se presenta cuando, debido a demandas de tipo arquitectónica o de funcionalidad, una estructura sufre modificaciones en la forma de la planta o en la distribución de espacios, lo que se traduce en irregularidades en planta. A su vez, la configuración de la planta conduce a problemas de tipo torsional en el sistema, y, a nivel de fuerzas internas en los elementos, las irregularidades en planta producen distribuciones no uniformes de relaciones Demanda/Capacidad en las columnas del piso (Pujar & Amaresh, 2017).

El comportamiento torsional de este tipo de estructuras es manifestado mediante su modo de vibración fundamental, que refleja la deformación principal que tendrá. Es de esperarse, entonces, que se desee tener un modo fundamental traslacional antes que uno torsional, pues reflejará el comportamiento predominante en la estructura (Balaji Rao et al., 2022).

Los diseñadores estructurales enfrentan el desafío de seleccionar configuraciones estructurales en planta idóneas para los sistemas resistentes a fuerzas sísmicas. Sin embargo, cuando esto no es posible debido a la arquitectura de la edificación, deben encontrar la manera de que la estructura conserve los modos de vibración traslacionales como sus modos fundamentales, por encima de los modos de vibración torsionales (Paulay & Priestly, 1992).

Para abordar esta problemática, se propone una técnica basada en un algoritmo de optimización implementado en Matlab. Este algoritmo, que se desarrollará a partir del análisis modal de las estructuras en estudio, buscará modificar las secciones y aumentar la rigidez de columnas específicas. El objetivo es acercar el centro de rigidez al centro de masa de cada piso, para transformar los modos de vibración fundamentales torsionales en traslacionales. Esta aproximación permite una solución sistemática y cuantificable al problema de las irregularidades torsionales, ofreciendo una herramienta para los diseñadores estructurales.

Considerando esta problemática, surge la siguiente pregunta de investigación:

¿Cómo modificar los modos de vibración fundamentales en estructuras con irregularidades torsionales para mejorar su comportamiento estructural ante cargas sísmicas?

### **1.3 Justificación**

La importancia de resolver la problemática de las irregularidades torsionales en las estructuras sometidas a cargas sísmicas recae en la necesidad de garantizar la seguridad de vida y proteger los bienes materiales en países de alto riesgo sísmico como Ecuador. Debido a su ubicación dentro del Cinturón de Fuego del Pacífico, Ecuador es vulnerable a sismos frecuentes y de intensidades considerables. Por lo tanto, es fundamental que las edificaciones cumplan con los criterios de diseño sismo-resistentes minimizando el riesgo de colapsos y daños (Ballesteros-Salazar et al., 2022).

El presente trabajo de investigación referente a cómo modificar los modos de vibración de estructuras con irregularidades torsionales tiene varias implicaciones prácticas dentro de la práctica de la Ingeniería Civil:

- Seguridad de vidas: Se busca mejorar el comportamiento estructural ante cargas sísmicas para reducir la posibilidad de colapsos durante un evento sísmico, protegiendo así a los ocupantes de las edificaciones.
- Minimización de daños materiales: Cuando una estructura tiene un buen desempeño durante un sismo, se pueden evitar daños en los elementos estructurales y no estructurales, lo que reduce costos de reparación y reconstrucción.
- Innovación en diseño estructural: Abordar esta problemática fomenta la investigación y la innovación en el campo de la ingeniería civil, alentando el desarrollo de nuevas soluciones que pueden ser aplicadas no solo en Ecuador, sino también en otros países del mundo que enfrentan situaciones similares.
- Desarrollo Sostenible: Al promover soluciones para edificaciones más seguras desde el punto de vista estructural contribuimos con los Objetivos de Desarrollo Sostenible (ODS); el ODS N°11 “Ciudades y comunidades sostenibles” y el ODS N°9 “Industria, innovación e infraestructura”.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Modificar los modos de vibración fundamentales en dos estructuras aporticadas de concreto armado con irregularidad en planta tipo 1, con el fin de mejorar su comportamiento estructural ante cargas sísmicas, utilizando un algoritmo de optimización basado en análisis modal.

### **1.4.2 Objetivos Específicos**

- Caracterizar el comportamiento dinámico inicial de dos estructuras aporticadas de hormigón armado con irregularidad en planta tipo 1.
- Implementar un procedimiento de optimización para la modificación de la rigidez del sistema de resistencia sísmica de las edificaciones que minimice el efecto torsional en la respuesta estructural.
- Comparar la respuesta ante cargas sísmicas de las edificaciones de estudio antes y después de la modificación de sus modos fundamentales, evaluando diferentes parámetros estructurales.

# CAPÍTULO 2

## 2. DESARROLLO DEL PROYECTO

### 2.1 Marco conceptual

#### 2.1.1 Configuración estructural

La configuración estructural de un sistema hace referencia a la disposición y arreglo de los elementos que componen la estructura y que servirán como sistema resistente a cargas impuestas. Incluye la forma general o geometría, las dimensiones de cada elemento y su distribución en el espacio. La configuración estructural incluye:

- Geometría: Se refiere a la forma de la estructura en planta y en elevación, que puede ser regular o irregular.
- Distribución de elementos: Se refiere a la ubicación en el espacio de todos los elementos que la conforman (columnas, vigas, muros, losas) y cómo se encuentran conectados entre sí. La distribución de elementos afecta a la estabilidad y rigidez de la estructura.
- Materiales: Aunque estos no afectan de manera directa a la forma de la estructura, sí afectan su resistencia y su comportamiento ante cargas.
- Sistema de resistencia: se refiere a los elementos y mecanismos con los que la estructura resiste las cargas impuestas, por ejemplo, sistemas de pórticos, muros de corte, y sistemas duales.

##### 2.1.1.1 Estructuras tipo pórticos

Son sistemas estructurales compuestos por vigas y columnas conectadas por nodos rígidos, diseñados para resistir cargas verticales y laterales principalmente por medio de la flexión de sus elementos.

### 2.1.1.2 Irregularidad en planta de estructuras

Son estructuras que presentan discontinuidades en la forma de su planta, o en la disposición de sus elementos estructurales. Son ejemplos de estas estructuras, las edificaciones que poseen formas de “T”, “E”, “L”, “U”, cruz, o edificaciones que poseen una disposición no uniforme de columnas y vigas.

### 2.1.1.3 Irregularidad en planta tipo 1 (Irregularidad torsional) según la NEC 2015

Se presenta cuando la deriva máxima de piso en un extremo (incluyendo la torsión accidental) y medida perpendicularmente a un eje de referencia, es mayor que 1.2 veces la deriva promedio de los dos extremos más alejados entre sí con respecto al mismo eje de referencia. Se define esta irregularidad en la sección 5.2.3 de la Norma de Peligro Sísmico. Diseño Sismo Resistente. NEC 2015.

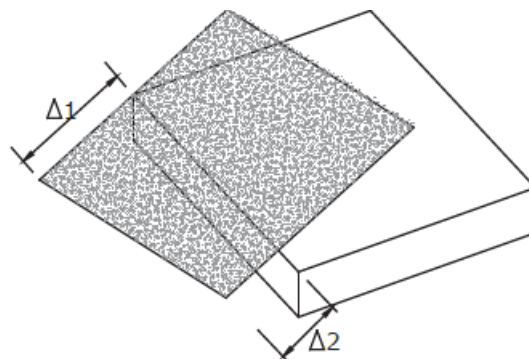


Figura 2.1 Irregularidad en planta tipo 1 según NEC-15.

### 2.1.1.4 Relación de irregularidad torsional RIT

Es la relación que existe entre la deriva máxima de piso de un extremo de la estructura medida perpendicularmente a un eje determinado y la deriva promedio de los dos extremos de la estructura con respecto al mismo eje de referencia. Esta relación debe ser menor a 1.2 para que la estructura no sea considerada con irregularidad en planta tipo 1.

$$RIT = \frac{\Delta_1}{\frac{(\Delta_1 + \Delta_2)}{2}} < 1.2 \quad (2.1)$$

### **2.1.1.5 Diafragma rígido**

Un diafragma puede considerarse rígido cuando posee una rigidez determinada a flexión, pero una rigidez axial en su plano que tiende a infinito de tal manera que no presenta deformaciones diferenciales entre sus nodos, y sus propiedades de masa y rigidez se puedan considerar concentradas en el centro de masa y rigidez, respectivamente (Awad, 2012).

## **2.1.2 Propiedades dinámicas de las estructuras**

Se refieren a las características que describen un sistema y cómo este responde a cargas o excitaciones que varían en el tiempo. En el contexto de la ingeniería estructural y la dinámica de estructuras, estas propiedades son fundamentales para entender y predecir el comportamiento de una estructura bajo cargas dinámicas como sismos, viento, o vibraciones inducidas por maquinaria. Las principales propiedades son: masa, rigidez y amortiguamiento.

### **2.1.2.1 Grado de libertad**

Se refiere al número de desplazamientos independientes requeridos para conocer la posición desplazada de un sistema con respecto a su posición original (Cimellaro & Marasco, 2018).

### **2.1.2.2 Centro de masa**

El Centro de Masa C.M. de un piso es el punto en el cual puede considerarse concentrada la masa total correspondiente a ese nivel y constituye el punto de aplicación de la fuerza sísmica en el piso considerado (Awad, 2012).

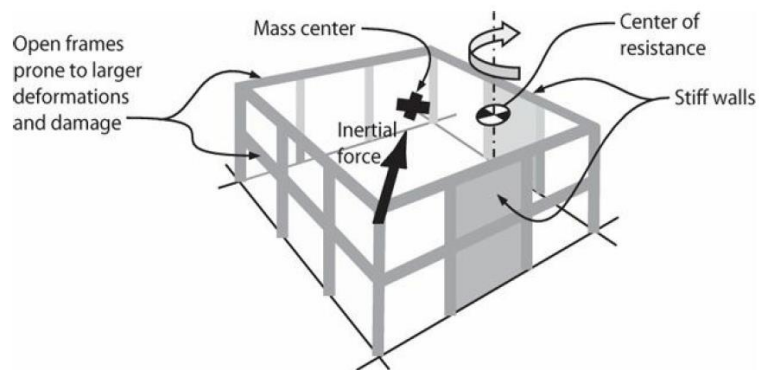
Su ubicación depende de la distribución de masas en la planta, que a su vez está determinada por la geometría de la estructura, y las cargas permanentes y superpuestas asignadas al piso.

### 2.1.2.3 Centro de rigidez

Se refiere al punto en el piso donde se concentra la resultante de las fuerzas resistentes generadas por los elementos estructurales verticales cuando la estructura es sometida a fuerzas horizontales, como las producidas por un sismo. Este punto depende de la distribución de la rigidez de la estructura (Awad, 2012). También puede definirse como el punto en el piso donde al aplicar una fuerza horizontal, este solo experimenta traslación.

### 2.1.2.4 Excentricidad estática

Se define como la diferencia entre las coordenadas del centro de masa y el centro de rigidez del piso (Awad, 2012). En la Figura 2.2 se muestra la ubicación del centro de masa y centro de rigidez o resistencia de la estructura; la separación entre ellos se conoce como excentricidad estática.



**Figura 2.2 Excentricidad estática debido a diferencia de coordenadas entre centro de masa y centro de resistencia o rigidez.**

### **2.1.2.5 Frecuencia natural**

Es el número de ciclos u oscilaciones que un sistema realiza en un segundo cuando vibra libremente. Su unidad está expresada en Hertz (Hz).

### **2.1.2.6 Periodo natural**

Es el tiempo que tarda un sistema en completar una oscilación en el modo que moviliza la mayor parte de su masa. Es el inverso de su frecuencia.

### **2.1.2.7 Amortiguamiento**

Se refiere al proceso por el cual la amplitud de la oscilación de un sistema disminuye de manera constante, a la vez que el sistema disipa energía (Chopra, 2014).

## **2.1.3 Análisis de estructuras en 3D**

### **2.1.3.1 Análisis Modal**

Es un método de análisis del comportamiento dinámico de estructuras que consiste en convertir un sistema de n ecuaciones diferenciales de movimiento que se encuentran acopladas, donde n es el número de grados de libertad, en un conjunto de ecuaciones de movimiento independientes. Este método es aplicado solo a sistemas lineales (García Reyes, 1998).

Se muestra la forma matricial de la ecuación de movimiento general para un sistema de Múltiples grados de libertad (MDOF):

$$[M].\{\ddot{u}(t)\} + [c].\{\dot{u}(t)\} + [K].\{u(t)\} = \{0\} \quad (2.2)$$

Como la matriz de rigidez no es diagonal, las ecuaciones que describen el movimiento para cada grado de libertad se encuentran acopladas, es decir, dependen la una de la otra. Desde el punto de vista matemático, la solución de estas ecuaciones diferenciales

es complicada, por lo que se recurre a una transformación de coordenadas geométricas a coordenadas modales (Cimellaro & Marasco, 2018):

$$\{u(t)\} = [\Phi].\{q(t)\} \quad (2.3)$$

Donde,

$u(t)$  = Respuesta de desplazamientos del sistema. Es dependiente del tiempo.

$\Phi$  = matriz modal que contiene los modos de vibración asociados a cada grado de libertad del sistema. No depende del tiempo.

$q(t)$  = coordenadas modales. Son las ecuaciones de movimiento asociadas a cada grado de libertad del sistema. Son dependientes del tiempo.

Sustituyendo la expresión previa, se obtiene:

$$[M].[\Phi].\{\ddot{q}(t)\} + [c].[\Phi].\{\dot{q}(t)\} + [K].[\Phi].\{q(t)\} = \{0\} \quad (2.4)$$

Como  $[\Phi]$  es independiente del tiempo, es consideradaa una constante dentro de las ecuaciones diferenciales. Esta transformación de coordenadas geométricas a modales tiene como objetivo desacoplar el sistema de ecuaciones, de manera que pueda obtenerse la respuesta de cada grado de libertad del sistema para distintos instantes de tiempo.

### 2.1.3.2 Modos de vibración

Es la deformada característica en la que un sistema dinámico oscila cuando se somete a fuerzas dinámicas. El modo de vibración es una propiedad que depende de la rigidez y la masa del sistema dinámico (Garcia Reyes, 1998).

### 2.1.3.3 Factor de participación modal ( $\Gamma_n$ )

Es el factor de participación del modo n-ésimo. Puede tener distintos valores para el mismo modo n-ésimo dependiendo de la normalización que se haya aplicado al modo. No confundir con el factor de participación de masa.

$$\Gamma_n = \frac{\phi_n \cdot M \cdot i}{M_n} \quad (2.5)$$

Donde,

$\phi_n$  = modo de vibración n-ésimo

$M$  = masa total del sistema

$i$  = vector que indica la dirección en la que se aplica el sismo al sistema

$M_n$  = Masa asociada a cada grado de Libertad del sistema

#### 2.1.3.4 Factor de participación de la masa

Es un factor de participación que representa la cantidad de masa de la estructura que participa en un modo de vibración particular. Se expresa generalmente como un porcentaje de la masa total de la estructura. Este factor ayuda a identificar los modos de vibración predominantes de la estructura. Su importancia recae en la determinación del número de modos de vibración que deben considerarse en el análisis dinámico para capturar de manera adecuada el comportamiento de la estructura.

$$Masa\ total = \sum_{n=1}^N \frac{(\phi_n \cdot M \cdot i)^2}{M_n} \quad (2.6)$$

$$Masa\ efectiva\ del\ modo = \frac{(\sum_{n=1}^N \phi_n \cdot M_n)^2}{\sum_{n=1}^N (\phi_n)^2 \cdot M_n} \quad (2.7)$$

$$Factor\ de\ participación\ de\ masa = \frac{Masa\ efectiva\ del\ modo}{Masa\ total} \quad (2.8)$$

#### 2.1.3.5 Porcentaje de participación de la masa

Es el Factor de participación de masa expresado en porcentaje. Según la NEC-15, en el análisis modal se debe tomar en cuenta la cantidad de modos de vibración que exciten como mínimo el 90% de la masa total del sistema.

## 2.1.4 Análisis Sísmico de Sistemas Lineales

Son métodos usados para evaluar la respuesta de estructuras ante cargas sísmicas, asumiendo que el comportamiento del sistema se mantiene en el rango elástico lineal.

### 2.1.4.1 Análisis Modal Espectral

El análisis modal espectral es un método de análisis sísmico que combina la respuesta de varios modos de vibración utilizando un espectro de diseño para estimar la respuesta máxima de la estructura.

Dado que los valores que indica el espectro corresponden al valor máximo que puede tener la respuesta de un sistema dinámico de un solo grado de libertad de un periodo natural y un amortiguamiento determinado, se puede obtener la respuesta máxima por modo de cada grado de libertad desacoplado y luego combinarlas para determinar la respuesta máxima de todo el sistema (García Reyes, 1998).

La respuesta máxima en coordenadas modales del modo n-ésimo, está dada por la ecuación:

$$\{q_n\}_{max} = |\Gamma_n| \cdot S_d(T_n, \xi) \quad (2.9)$$

Donde,

$S_d$  = representa la ordenada del espectro de desplazamientos para un periodo  $T_n$  y un porcentaje de amortiguamiento modal  $\xi$  definido.

Cuando se dispone del espectro de aceleraciones, la ecuación previa puede expresarse como:

$$\{q_n\}_{max} = |\Gamma_n| \cdot \frac{1}{\omega_n^2} \cdot S_a(T_n, \xi) \quad (2.10)$$

Donde,

$S_a$  = representa la ordenada del espectro de aceleraciones para un periodo  $T_n$  y un factor de amortiguamiento modal  $\xi$  definido.

$\omega_n$  = corresponde a la frecuencia del n-ésimo modo.

Como este análisis arroja los valores máximos de respuesta para cada modo, pero estas respuestas no se dan simultáneamente en el tiempo, sería errado hacer una sumatoria simple de los desplazamientos. Por esta razón, se aplican métodos estadísticos de combinación, conocidos como métodos de combinación modal espectral (Garcia Reyes, 1998). Uno de ellos se basa en la raíz de la sumatoria de los cuadrados de las respuestas asociadas a cada modo de vibración.

#### **2.1.4.2 Espectro de respuesta**

Los espectros de respuesta son gráficas que muestran las respuestas dinámicas máximas (desplazamientos, velocidades, aceleraciones) que sistemas de 1 GDL (grado de libertad) caracterizado por un período natural (T) y un porcentaje de amortiguamiento ( $\xi$ ) determinado experimentarán durante la excitación (Villao, 2018).

#### **2.1.4.3 Método de la raíz cuadrada de la suma de los cuadrados (SRSS) para combinación de respuesta modal**

Es un método de combinación modal desarrollado por E. Rosenblueth en su tesis doctoral, e indica que para cualquier parámetro de respuesta modal  $r$ , el valor máximo factible del parámetro, al tomar en cuenta las respuestas máximas de las distintas componentes modales  $r_i$ , se obtiene a través de la siguiente ecuación (Garcia Reyes, 1998):

$$r = \sqrt{\sum_{i=1}^n r_i^2} \quad (2.11)$$

donde,

$r$  = respuesta combinada de todos los modos.

$r_i$  = respuesta máxima asociada al modo  $i$ .

$n$  = número total de modos de vibración considerados.

Este método se basa en la suposición de que los valores modales máximos son independientes y que la probabilidad de que las respuestas máximas de diferentes

modos ocurran simultáneamente es baja. Bajo estas condiciones, la combinación cuadrática proporciona un valor representativo de la respuesta máxima total.

#### 2.1.4.4 Método de la raíz cuadrada de la suma de los cuadrados (CQC) para combinación de respuesta modal

Es una técnica de combinación modal planteado por primera vez por Rosenblueth en 1969, y desarrollado posteriormente por Der Kiureghian en 1981. Este método parte del método de combinación SRSS incorporando el efecto de correlación entre modos con frecuencias cercanas. Este efecto es relevante en estructuras con modos de vibración acoplados o con amortiguamientos significativos, donde las respuestas de modos adyacentes no son completamente independientes entre sí (Garcia Reyes, 1998).

La expresión general de la combinación modal CQC es:

$$r = \sqrt{\sum_{i=1}^n \sum_{j=1}^n \rho_{ij} r_i r_j} \quad (2.12)$$

donde,

$r$  = respuesta combinada de todos los modos.

$r_i, r_j$  = respuesta máxima asociada al modo  $i$  y al modo  $j$ , respectivamente.

$\rho_{ij}$  = coeficiente de correlación entre los modos  $i$  y  $j$ , calculado en función de sus frecuencias y del amortiguamiento modal.

$n$  = número total de modos de vibración considerados.

El coeficiente de correlación  $\rho_{ij}$  se obtiene a partir de la siguiente expresión:

$$\rho_{ij} = \frac{8\xi^2(1 + \beta_{ij})(\beta_{ij})^{3/2}}{(1 - \beta_{ij}^2)^2 + 4\xi^2\beta_{ij}(1 + \beta_{ij}^2)^2}$$

donde,

$\xi$  = razón de amortiguamiento modal (adimensional, típicamente 0.05 para 5% de amortiguamiento).

$\beta_{ij}$  = relación entre las frecuencias de los modos i y j.

Cuando los modos tienen frecuencias muy diferentes ( $\rho_{ij} \approx 0$ ), el método CQC se reduce prácticamente al SRSS. En cambio, cuando las frecuencias son cercanas,  $\rho_{ij}$  toma valores significativos y la contribución combinada aumenta respecto al SRSS.

### **2.1.5 Algoritmo de optimización global en Matlab**

Son rutinas que ofrece Matlab como herramienta para hallar parámetros que minimizan o maximizan una función objetivo mientras respetan restricciones. Para resolver problemas con restricciones, Matlab ofrece los siguientes algoritmos de optimización global:

- Algoritmo genético
- Recocido simulado
- Enjambre de partículas

Estos algoritmos están implementados en diferentes funciones de optimización de MATLAB como `fminunc`, `fmincon`, `lsqnonlin`, `ga`, `simulannealbnd`, etc. El usuario puede seleccionar el algoritmo más apropiado según el tipo de problema de optimización que esté resolviendo en ingeniería civil (*MathWorks*, s/f).

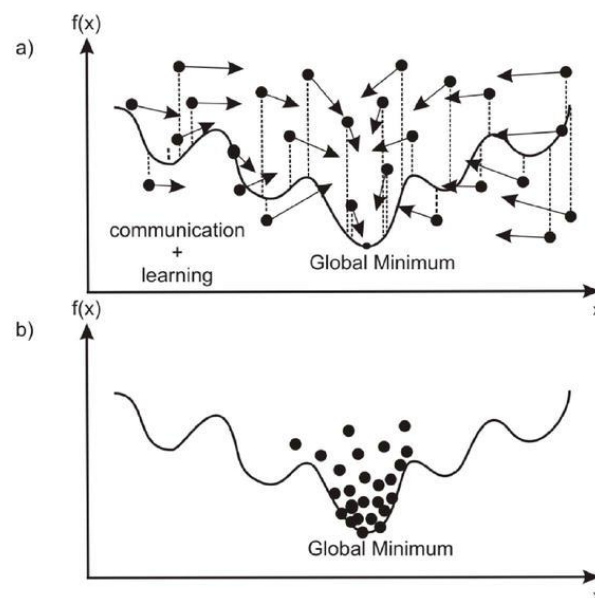
#### **2.1.5.1 Algoritmo de Optimización por Enjambre de Partículas (PSO)**

El algoritmo de optimización por Enjambre de Partículas (Particle Swarm Optimization PSO) es un método de optimización global que se inspira en la naturaleza de organismos vivos e imita su comportamiento en su búsqueda de alimento o lugares específicos (Kennedy & Eberhart, s/f).

El PSO busca una solución óptima a un problema iterando distintas posiciones en un espacio de búsqueda mediante la interacción de un conjunto de posibles soluciones, denominadas partículas. Cada partícula es una posible solución a la variable del problema y se caracteriza por tener una posición en el espacio y una velocidad. La posición de la partícula corresponde a los valores iterativos de las variables del problema y la velocidad determina cómo se desplaza la partícula en el espacio de búsqueda o

“vecindad” (Akemi Ikegawa, Crespo Chacón, Rodriguez, Montesinos Barrios, & García Morillo, 2019).

En cada iteración, las partículas del enjambre ajustan su posición o valor basándose en su propio resultado obtenido para el problema y en el resultado del resto de partículas. En el literal a) de la Figura 2.3. se muestra el comportamiento del enjambre de partículas buscando alcanzar el valor mínimo que puede tener la función objetivo representada por la curva mostrada, mediante la comunicación entre partículas. En el literal b) se muestra el avance de las partículas hacia la convergencia de resultados.



**Figura 2.3 Representación visual del comportamiento del Algoritmo por PSO. Obtenido en la web.**

## 2.2 Marco metodológico

Para el desarrollo del presente trabajo se divide la metodología en tres fases con base en los objetivos específicos planteados: (1) caracterización del comportamiento dinámico de dos estructuras aporticadas tipo de hormigón armado (una de tres pisos y otra de seis pisos) con irregularidad en planta tipo 1, (2) desarrollo e implementación de un algoritmo de optimización para la modificación de la rigidez del sistema de resistencia sísmica de

las edificaciones que minimice el efecto torsional en la respuesta estructural, (3) comparación de la respuesta ante cargas sísmicas de las edificaciones de estudio antes y después de la modificación de sus modos fundamentales, evaluando diferentes parámetros estructurales. A su vez, cada fase de la metodología está dividida en una serie de actividades, lo que permitirá el cumplimiento del objetivo general y los objetivos específicos de la investigación.

### **Fase 1. Caracterización del comportamiento dinámico de dos estructuras aporticadas tipo de hormigón armado con irregularidad en planta tipo 1.**

En la primera fase del proyecto se caracterizarán las propiedades dinámicas de dos estructuras de hormigón tipo pórtico con diagonales en uno de sus pórticos para analizar el comportamiento estructural de ellas. Estas estructuras permitirán representar irregularidades en planta del tipo 1 indicadas en la NEC 2015. En primera instancia se definirá la geometría y el material de los elementos estructurales de las edificaciones. Luego, se obtendrá la matriz de masa, rigidez y amortiguamiento de los sistemas de estudio, y se obtendrá el centro de masas y rigidez de cada piso. Posteriormente, se aplicará en las estructuras un análisis modal para obtener sus modos de vibración, frecuencias asociadas a cada modo y porcentajes de participación de masa por modo. Finalmente, se aplicará un análisis modal espectral para obtener las respuestas de desplazamientos y rotaciones máximas asociadas a cada grado de libertad del sistema. Esta información se obtendrá mediante el desarrollo de un algoritmo en Matlab.

#### **A1.1. Definir geometría y material de las estructuras de estudio**

Se definirá las dimensiones y propiedades de columnas, vigas y losas que conformarán los sistemas dinámicos de estudio. Se incluirá diagonales en pórticos perimetrales para inducir irregularidad torsional.

#### **A1.2. Obtención de propiedades dinámicas de las estructuras de estudio**

Se ensamblarán las matrices de masa, rigidez y amortiguamiento de las estructuras. Además, se obtendrán las coordenadas del centro de masa y el centro de rigidez de cada piso, que serán relevantes en la posterior implementación del algoritmo de optimización.

#### **A1.3. Análisis modal de las estructuras de estudio**

Se resolverá la ecuación de valores y vectores propios para obtener los modos de vibración, sus frecuencias asociadas y se determinará la masa participativa por modo.

Se obtendrá las respuestas máximas de los sistemas ante cargas sísmicas de estudio asociadas a cada grado de libertad.

#### **A1.4. Cálculo de la relación de irregularidad torsional (RIT)**

Con base en las respuestas obtenidas del análisis previo, se determinará la razón entre la deriva máxima en un extremo y el promedio de las derivas de los pórticos extremos (RIT). Este índice debe ser mayor que 1.2 para que la estructura sea considerada como irregular (Ministerio de Desarrollo Urbano y Vivienda, 2014a).

### **Fase 2. Desarrollo e implementación de un algoritmo de optimización para la modificación de la rigidez del sistema de resistencia sísmica de las edificaciones que minimice el efecto torsional en la respuesta estructural.**

En la segunda fase del proyecto, se procesará la información obtenida previamente para desarrollar e implementar un algoritmo de optimización de la estructura. El algoritmo de optimización buscará determinar las secciones de columnas mínimas necesarias para acercar el centro de rigidez de piso al centro de masa lo suficiente para transformar los modos de vibración torsionales en traslacionales, reduciendo el índice de irregularidad por torsión por debajo del límite de 1.2 indicado por la norma NEC-15. Una vez que el algoritmo haya detectado las secciones óptimas que cumplan los requerimientos antes mencionados, se hará un análisis modal final para confirmar que los modos fundamentales han pasado de ser torsionales a traslacionales.

#### **A2.1. Definición de la función objetivo**

Se establece una función que cuantifica la diferencia entre el centro de rigidez y el centro de masa de cada piso. El objetivo será minimizar esta función.

#### **A2.2. Establecimiento de variables de diseño**

Se definen las dimensiones de las columnas como variables a optimizar.

#### **A2.3. Implementación de restricciones**

Se establecen restricciones para asegurar que las modificaciones cumplan con los requisitos de resistencia y normativas de construcción.

#### **A2.4. Implementación de algoritmo de optimización**

Seleccionar y programar el método de optimización en Matlab.

#### **A2.5. Iteración y convergencia**

El algoritmo ajusta iterativamente las dimensiones de las columnas, recalculando en cada paso el análisis modal y la función objetivo hasta alcanzar una convergencia o un número máximo de iteraciones.

#### **A2.6. Verificación de resultados**

Se realiza un análisis modal final para confirmar que los modos fundamentales del sistema se han reordenado de manera que los que contribuyan con mayor masa participativa a la respuesta sean traslacionales.

#### **Fase 3. Comparación de la respuesta ante cargas sísmicas de las edificaciones de estudio antes y después de la modificación de sus modos fundamentales, evaluando diferentes parámetros estructurales.**

En la última fase del proyecto se hará una comparación de respuestas dinámicas de los sistemas de estudio antes y después de la modificación de sus modos de vibración.

##### **A3.1. Comparación de resultados de las estructuras antes y después de la modificación de sus modos de vibración**

Se tabularán los resultados de las respuestas de desplazamientos, derivas de piso, fuerzas internas. Se comparará la demanda de acero en las columnas antes y después de la modificación de sus modos de vibración.

##### **A3.2. Análisis de cambios en los modos de vibración**

Se analizará cambios en los modos de vibración mediante el porcentaje de participación de masa.

##### **A3.3. Evaluación de la mejora en el comportamiento estructural**

Se calculará el porcentaje de reducción en derivas de entrepiso y se compararán excentricidades antes y después de la optimización. Finalmente, se verificará si se logró reducir la Relación de Irregularidad Torsional por debajo de 1.2.

### **2.3 Definición de las edificaciones de estudio**

#### **2.3.1 Descripción de los modelos estructurales**

Se propuso dos modelos estructurales irregulares como base, uno de 3 pisos y otro de 6 pisos. Los modelos están conformados por 5 ejes en dirección "X" (A, B, C, D y E)

separados 7.0 m entre sí y 5 ejes en dirección “Y” (1, 2, 3, 4 y 5) separados 5.5 m entre sí. Las alturas de entrepiso de ambos modelos son de 3 m.

### 2.3.2 Casos de análisis

Para demostrar la irregularidad torsional tipo 1 referida en la NEC-15, se optó por incorporar en las estructuras objetos tipo “frame” que simulen el efecto de una escalera en uno de los pórticos paralelo al eje “X”. Esto ocasionó excentricidades en los pisos de las edificaciones, proporcionando los modelos estructurales irregulares en planta (Caso 1 y Caso 2).

#### 2.3.2.1 Caso 1 (Edificio de 3 pisos)

Se denominó Caso 1 al modelo estructural irregular de tres pisos que cuenta con la escalera (diagonal), como se muestra en la figura 2.4.

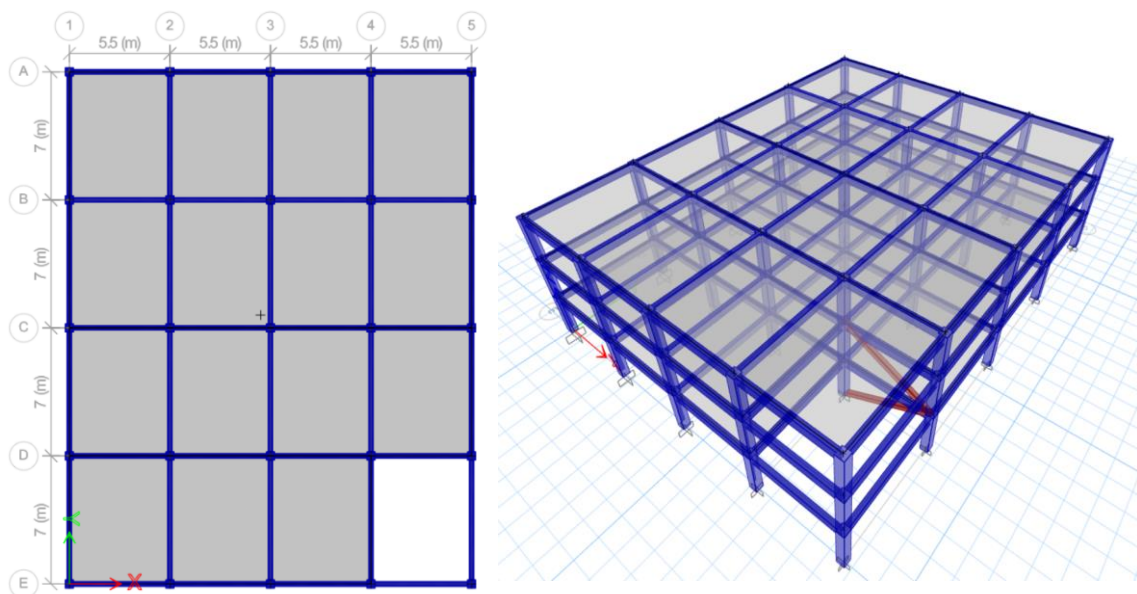
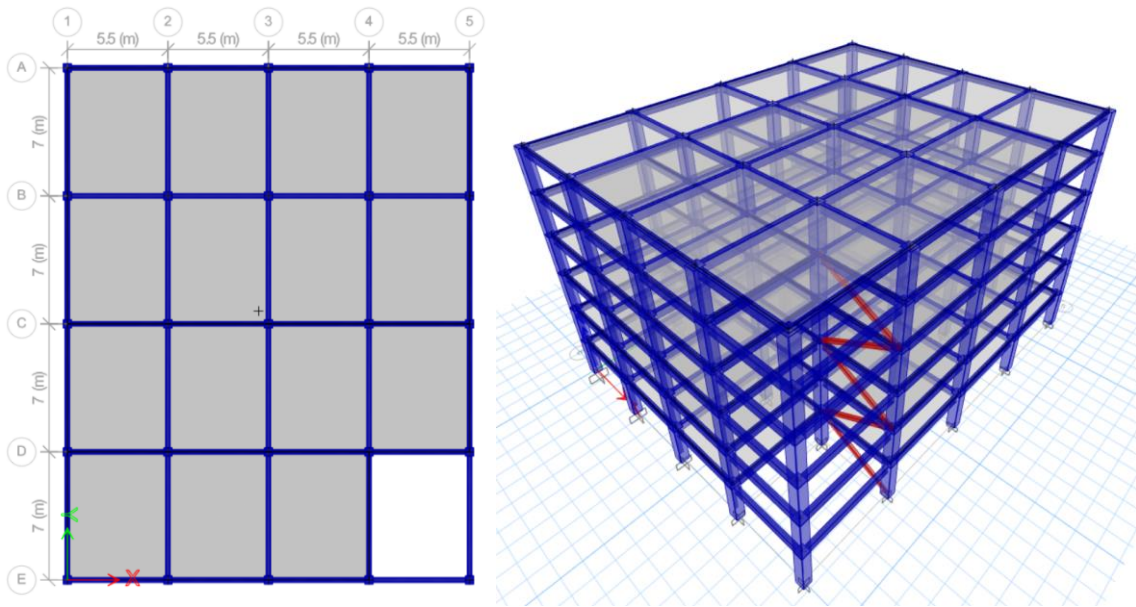


Figura 2.4 Vista en planta y en 3D de modelo estructural Caso 1.

#### 2.3.2.2 Caso 2 (Edificio de 6 pisos)

Se denominó Caso 2 al modelo estructural irregular de seis pisos que cuenta con la escalera (diagonal), como se muestra en la figura 2.5.



**Figura 2.5 Vista en planta y en 3D de modelo estructural Caso 2.**

### 2.3.3 Dimensiones de los elementos estructurales

En las Tabla 1 y Tabla 2, se detalla las dimensiones de los elementos estructurales considerados en el modelo estructural regular 1 y 2.

**Tabla 2.2.1. Dimensiones de elementos estructurales de Caso 1.**

Piso	Elementos	Dimensiones [cm]
3	Columnas	45x45
	Vigas	25x45
	Losa maciza	e = 20
	Diagonal	25x25
2	Columnas	45x45
	Vigas	25x45
	Losa maciza	e = 20
	Diagonal	25x25
1	Columnas	45x45
	Vigas	25x45
	Losa maciza	e = 20
	Diagonal	25x25

**Tabla 2.2.2. Dimensiones de elementos estructurales de Caso 2.**

Piso	Elementos	Dimensiones [cm]
6	Columnas	60x60
	Vigas	25x45

	Losa maciza	e = 20
5	Columnas Vigas Losa maciza Diagonal	60x60 25x45 e = 20 20x20
4	Columnas Vigas Losa maciza Diagonal	60x60 25x45 e = 20 20x20
3	Columnas Vigas Losa maciza Diagonal	60x60 25x45 e = 20 20x20
2	Columnas Vigas Losa maciza Diagonal	60x60 25x45 e = 20 20x20
1	Columnas Vigas Losa maciza Diagonal	60x60 25x45 e = 20 20x20

### 2.3.4 Propiedades mecánicas del modelo estructural

Las propiedades mecánicas de los materiales considerados en los dos modelos estructurales regulares se detallan en la Tabla 3.

**Tabla 2.2.3. Propiedades de los materiales de los elementos estructurales.**

<b>Materiales</b>	<b>Peso específico [kgf/m<sup>3</sup>]</b>	<b>Resistencia [kgf/cm<sup>2</sup>]</b>	<b>Módulo de elasticidad [kgf/cm<sup>2</sup>]</b>
Hormigón	2400.00	280.00	252671.33
Acero de refuerzo	7850.00	4200.00	2100000.00

### 2.3.5 Cargas aplicadas en los casos de análisis

#### 2.3.5.1 Carga Muerta

Las cargas permanentes (o cargas muertas) están constituidas por los pesos de todos los elementos estructurales y no estructurales, tales como: muros, paredes, recubrimientos, instalaciones sanitarias, eléctricas, mecánicas, máquinas y todo artefacto integrado permanentemente a la estructura (Norma Ecuatoriana De La

Construcción, 2015). Se consideró el peso propio de la estructura más una sobrecarga muerta (SD) en cada piso, la cual se detalla en la Tabla 4.

**Tabla 2.2.4. Peso propio más sobrecarga muerta consideradas en los modelos estructurales.**

Piso	Sobrecarga muerta (Sd) [kgf/m <sup>2</sup> ]
Todos	940

### 2.3.5.2 Carga Viva

Las sobrecargas que se utilicen en el cálculo dependen de la ocupación a la que está destinada la edificación y están conformadas por los pesos de personas, muebles, equipos y accesorios móviles o temporales, mercadería en transición, y otras (Norma Ecuatoriana De La Construcción, 2015). Se consideró como edificio de oficinas y las cargas fueron obtenidos del ASCE 7-16 y se detallan en la Tabla 5.

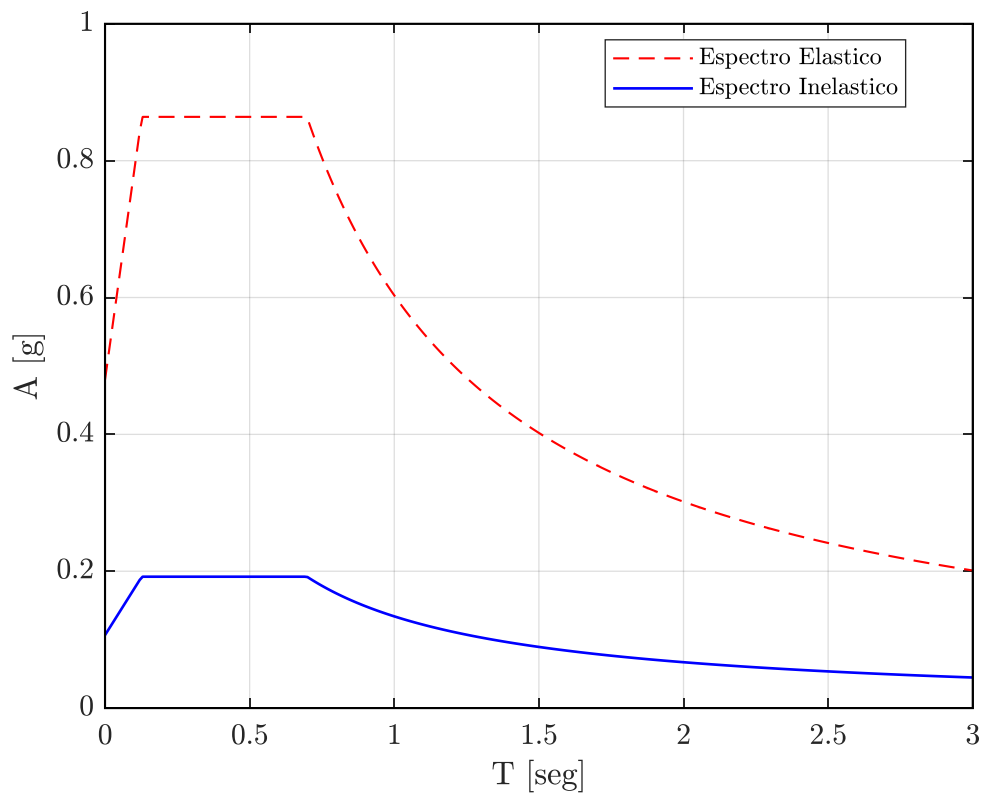
**Tabla 2.2.5. Cargas vivas consideradas en los modelos estructurales.**

Piso	Carga Viva (L) [kgf/m <sup>2</sup> ]
Todos excepto cubierta	240
Cubierta	70

### 2.3.5.3 Carga sísmica

Las cargas sísmicas son producto del movimiento del suelo, las cuales generan fuerzas como respuesta de la estructura, donde dicha respuesta depende de las características dinámicas de la misma como la masa, rigidez y amortiguamiento.

En el presente trabajo, para obtener las respuestas de los modelos de estudio, se desarrolló un análisis modal espectral para el cual se utilizó el espectro de aceleraciones propuesto por la NEC-15 para un suelo de la Costa de tipo D. Se optó por un Factor de importancia igual a 1 debido al uso de la estructura, y un Factor de ductilidad de R=5 para obtener el espectro inelástico. El análisis modal se realizó mediante un algoritmo desarrollado en Matlab. En la Figura 2.6, se muestra la gráfica del espectro de respuesta elástico (R=1) e inelástico (R=5) seleccionados para el análisis.



**Figura 2.6 Espectro de Respuesta de Aceleraciones NEC-15.**

Previamente al análisis modal espectral de esta tesis, a manera de verificación, se comparó los resultados que proporcionaron dos tipos de análisis modal desarrollados en el algoritmo; espectral y cronológico utilizando el acelerograma de una de las componentes ortogonales de un sismo de prueba, el Sismo de “El Centro”. Una vez que se verificó la confiabilidad de los resultados de los dos análisis, se procedió a ingresar el espectro de respuesta seleccionado para este trabajo de tesis.

### **2.3.5.3.1 Análisis modal espectral RSA**

El Análisis dinámico espectral es un procedimiento en el que la respuesta de la estructura se obtiene mediante una combinación específica de las contribuciones modales, las cuales están caracterizadas por la máxima respuesta de cada modo de vibración arraigado en función del distinto nivel de participación modal, el cual la contribución de cada modo a la respuesta total de la estructura. Para estructuras irregulares la NEC-15 indica que este procedimiento es obligatorio a realizarse para cualquier tipo de análisis.

Para efectuar el análisis espectral modal, se tiene que tomar en consideración la combinación de un número de modos, tal que la participación de la masa total sea al menos el 90% de la masa de la estructura en la dirección de análisis considerada (Ministerio de Desarrollo Urbano y Vivienda, 2014a).

La NEC-15 indica que se puede utilizar cualquier metodología permitida por la dinámica estructural (SRSS, CQC, etc.). En este trabajo se utilizó la combinación de modos CQC.

El espectro de diseño elástico e inelástico de aceleraciones se obtuvo según la NEC, a partir de los siguientes parámetros mostrados en la Tabla 6.

**Tabla 2.2.6. Parámetros para obtención del espectro de diseño de la NEC-15.**

Parámetros para espectro de diseño	
Tipo de suelo	D
Z (zona sísmica)	0.40
Fa	1.2
Fd	1.19
Fs	1.28
r	1
$\eta$	1.80
R	5
$\Phi_p$	0.9

### **2.3.6 Modos de vibración y Porcentajes de Masa Participativa de las estructuras de estudio**

El análisis modal permite identificar las formas características de vibración de una estructura, sus periodos y frecuencias asociadas, y el porcentaje de masa del sistema que participa en cada modo. El porcentaje de masa participativa nos indica qué fracción de la masa total del sistema está siendo excitada en cada modo, por lo que es un indicador de la contribución de dicho modo a la respuesta dinámica total.

De acuerdo con la Norma Ecuatoriana de la Construcción (NEC-15), el análisis modal debe considerar un número de modos suficiente para movilizar, como mínimo, el 90%

de la masa total en la dirección analizada (Ministerio de Desarrollo Urbano y Vivienda, 2014a).

De igual manera, la identificación del tipo de modo; traslacional o torsional está dada por el porcentaje de masa traslacional o rotacional que dicho modo estaría movilizándolo. Lo último es relevante, ya que las estructuras con irregularidad torsional presentan un modo fundamental de tipo torsional con una participación modal significativa, lo que deriva en una irregularidad en planta de tipo torsional, según la NEC-15, con la consecuente penalización del espectro de diseño y aumento de la demanda sísmica.

En la Tabla 2.7. y 2.8. se presentan los porcentajes de masa participativa obtenidos para el Caso 1 (Edificio de 3 pisos) y el Caso 2 (Edificio de 6 pisos) antes de la aplicación de la Optimización de rigideces del sistema. Para cada modo se muestra el período T, el tipo de modo y los porcentajes de masa participativa en las direcciones X, Y y alrededor de Z.

La comparación de los modos antes y después de la Optimización se presentan en el Capítulo 3.

**Tabla 2.2.7. Resultados de modos de vibración y masa participativa iniciales, Caso 1.**

Modo	Período [seg]	Ux [%]	Uy [%]	Rz [%]	Tipo de Modo
1	0.85	0.02	81.01	0.06	Traslacional
<b>2</b>	<b>0.71</b>	<b>42.12</b>	<b>0.05</b>	<b>40.35</b>	<b>Torsional-traslacional</b>
3	0.51	37.06	0.00	39.96	Torsional-traslacional
4	0.23	0.02	14.40	0.00	Traslacional
5	0.21	14.57	0.04	0.46	Traslacional
6	0.17	2.02	0.03	14.96	Torsional
7	0.11	0.04	4.38	0.07	Traslacional
8	0.10	3.50	0.07	0.72	Traslacional
9	0.08	0.66	0.01	3.43	Torsional

**Tabla 2.2.8. Resultados de modos de vibración y masa participativa iniciales, Caso 2.**

Modo	Período [seg]	Ux [%]	Uy [%]	Rz [%]	Tipo de Modo
1	1.61	0.06	75.25	0.15	Traslacional
<b>2</b>	<b>1.35</b>	<b>43.30</b>	<b>0.16</b>	<b>33.34</b>	<b>Torsional-traslacional</b>
3	0.96	34.11	0.00	42.72	Torsional-traslacional
4	0.44	0.02	13.26	0.00	Traslacional
5	0.38	9.38	0.04	2.48	Traslacional
6	0.31	2.42	0.01	10.35	Torsional
7	0.19	0.02	5.81	0.06	Traslacional
8	0.18	5.10	0.03	0.60	Traslacional
9	0.15	0.44	0.02	5.05	Torsional
10	0.11	0.03	3.08	0.01	Traslacional
11	0.10	2.81	0.04	0.10	Traslacional
12	0.08	0.14	0.01	2.94	Torsional
13	0.07	0.04	1.64	0.03	Traslacional
14	0.07	1.51	0.05	0.08	Traslacional
15	0.05	0.05	0.00	1.54	Torsional
16	0.05	0.03	0.57	0.00	Traslacional
17	0.05	0.52	0.03	0.01	Traslacional
18	0.04	0.02	0.00	0.56	Torsional

A partir de estos resultados se observa que, en ambos casos de estudio, el modo fundamental 2 es de tipo torsional-traslacional, con un porcentaje de masa participativa torsional superior al 40%. Esto implica que la respuesta sísmica inicial está condicionada por la torsión, lo cual es característico de las edificaciones con irregularidad en planta tipo 1.

Asimismo, se puede apreciar que para alcanzar el 90% de masa participativa acumulada en cada dirección principal es necesario considerar al menos seis modos en el edificio de 3 pisos (Caso 1) y un número similar en el edificio de 6 pisos (Caso 2). Sin embargo,

el predominio de un modo torsional como uno de los fundamentales justifica la aplicación de la estrategia de optimización propuesta en esta investigación, cuyo objetivo es transformar los modos fundamentales torsionales en traslacionales, reduciendo así la respuesta torsional y mejorando el desempeño sísmico de las edificaciones.

### 2.3.7 Cálculo de Relación de Irregularidad torsional (RIT)

Para el cálculo de la Relación de Irregularidad torsional, se realizó el análisis modal espectral de la edificación de tres pisos (Caso 1) y seis pisos (Caso 2). Mediante el análisis se obtuvo como resultado los desplazamientos de diafragma de cada piso que luego fueron transformadas en desplazamientos de cada pórtico. Las ecuaciones 2.12 y 2.13 relacionan a los desplazamientos de cada diafragma con los desplazamientos de un punto de cada pórtico mediante el ángulo de rotación de cada diafragma y su distancia hasta el pórtico en cuestión.

$$\Delta X_A = \Delta X + \theta_{zz} \cdot (Y_A + Y_{CM}) \quad (2.12)$$

$$\Delta Y_A = \Delta Y + \theta_{zz} \cdot (X_A + X_{CM}) \quad (2.13)$$

Donde,

$\Delta X_A, \Delta Y_A$  = Desplazamiento en dirección "X" y "Y" del punto "A" del pórtico, respectivamente.

$\theta_{zz}$  = Rotación del diafragma alrededor del eje "Z".

$\Delta X, \Delta Y$  = Desplazamiento en dirección "X" y "Y" en el Centro de Masa del diafragma, respectivamente.

Una vez que se han obtenido los desplazamientos relativos laterales de los pórticos más alejados entre sí de cada piso, se obtiene el RIT usando la ecuación 2.1. Este valor clasificará a las edificaciones como estructuras con regularidad o irregularidad torsional, según la NEC-15.

Para las estructuras de estudio del Caso 1 y Caso 2 sin optimizar, estos índices alcanzaron valores máximos mayores a 1.70, es decir, las estructuras presentan una irregularidad torsional extrema.

### **2.3.8 Cálculo del Centro de Rigidez**

La obtención del centro de rigidez de cada piso de las estructuras es una parte fundamental para la implementación de la optimización de las rigideces localizadas de los sistemas de estudio. A partir de la diferencia entre las coordenadas del centro de rigidez C.R. y el centro de masa C.M., se obtiene la excentricidad estática. Mientras menor sea la excentricidad, la estructura experimentará menores rotaciones alrededor del eje vertical ante cargas sísmicas, dando como resultado en algún punto, una estructura cuyos modos de vibración fundamentales, ya no serán torsionales sino traslacionales.

Debido a que la masa traslacional y rotacional de cada piso depende principalmente de la geometría de la planta y las cargas asignadas de piso, la reducción de la excentricidad se logra desplazando el C.R. hacia el C.M. correspondiente. A su vez, para modificar la ubicación del C.R. es necesario alterar localmente las rigideces del sistema resistente a cargas laterales; en este estudio, dichas rigideces se modifican mediante cambios en las columnas.

Para el cálculo del centro de rigidez de cada piso se realizó dos procedimientos; un procedimiento analítico y un procedimiento basado en optimización. El C.R. (Centro de rigidez), calculado de forma analítica, es el método principal utilizado para resultados de esta tesis. La obtención del C.R. mediante optimización es un procedimiento implementado a manera de verificación de los valores calculados mediante el método analítico.

#### **2.3.8.1 Procedimiento Analítico de cálculo de C.R.**

Este procedimiento se basa en la rigidez a la torsión del entrepiso alrededor del eje vertical. En primera instancia, se aplica un Momento torsor  $M_{Tz}$  en el C.R. en sentido horario alrededor del eje que sale del plano del entrepiso, lo que produce una rotación  $\theta_{zz}$ . Asumiendo comportamiento lineal elástico, por la ley de Hooke, podemos relacionar ambas magnitudes con la rigidez a la rotación del entrepiso:

$$k_{rot} = M_{Tz} / \theta_{zz} \quad (2.13)$$

Luego, se aplica una fuerza lateral  $F_x$  en dirección "X" al entrepiso en un punto arbitrario distinto del C.R. De manera conveniente, se aplicará en el origen de coordenadas definido para el entrepiso, lo que dará como resultado una rotación  $\theta_{zx}$  en sentido antihorario alrededor del eje vertical debido a la fuerza aplicada. Si se multiplica la fuerza aplicada por la distancia desde el C.R. hasta el punto de aplicación (origen de coordenadas) YCR, se obtiene un Momento torsor  $M_{Tx}$ :

$$M_{Tx} = F_x \cdot YCR \quad (2.14)$$

De igual forma, se puede relacionar el Momento  $M_{Tx}$ , la rotación resultante  $\theta_{zx}$  y la rigidez rotacional de piso  $k_{rot}$ , mediante la expresión:

$$k_{rot} = M_{Tx} / \theta_{zx} \quad (2.15)$$

Se igualan las ecuaciones 2.13 y 2.15:

$$M_{Tz} / \theta_{zz} = M_{Tx} / \theta_{zx} \quad (2.16)$$

Se sustituye  $M_{Tx}$  en la expresión 2.16 y se despeja la coordenada YCR que representa a la coordenada en "Y" del C.R.:

$$YCR = \frac{M_{Tz}}{F_x} \cdot \frac{\theta_{zx}}{\theta_{zz}} \quad (2.17)$$

Si las magnitudes del momento y la fuerza aplicados son iguales, la expresión 2.17 se simplificaría aún más, quedando como resultado de YCR la relación entre las rotaciones  $\theta_{zx}$  y  $\theta_{zz}$ .

Para la obtención de la coordenada en "X" del C.R. de cada piso XCR, se realiza un procedimiento análogo, quedando como resultado la expresión:

$$XCR = \frac{M_{Tz}}{F_y} \cdot \frac{\theta_{zy}}{\theta_{zz}} \quad (2.18)$$

Una vez que se implementó el método analítico de cálculo en la rutina, se obtuvieron los resultados descritos en la Tabla 2.9 y Tabla 2.10.

**Tabla 2.2.9. Coordenadas del C.R. por el método analítico del modelo Caso 1.**

Nº Piso	XCR [m]	YCR [m]
1	11.00	11.47
2	11.00	10.57
3	11.00	11.30

**Tabla 2.2.10. Coordenadas del C.R. por el método analítico del modelo Caso 2.**

Nº Piso	XCR [m]	YCR [m]
1	11.00	12.78
2	11.00	11.97
3	11.00	11.47
4	11.00	11.15
5	11.00	11.00
6	11.00	11.21

### **2.3.8.2 Procedimiento basado en optimización de cálculo de C.R.**

A modo de verificación de los resultados obtenidos del Método previo (método analítico), se desarrolló un algoritmo de optimización en Matlab que busca el punto dentro de la planta de cada piso en el cual, al aplicar una fuerza lateral, el piso solo experimenta traslación sin rotaciones, o con rotaciones despreciables, es decir, el punto que representa el Centro de Rigidez C.R.

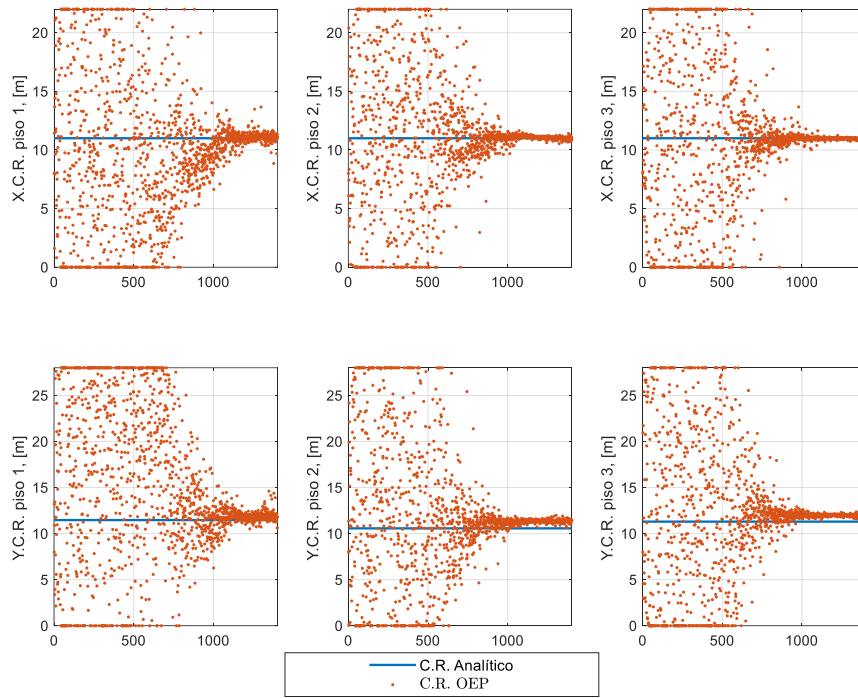
En la optimización en Matlab se utilizó el algoritmo de Optimización por Enjambre de Partículas (PSO), el cual imita el comportamiento de un enjambre de organismos en la naturaleza que van en búsqueda de un sitio específico. En este tipo de optimización se simula un enjambre de partículas donde cada partícula representa una solución potencial al problema (valores posibles de las variables). Estas partículas se desplazan por el

espacio de búsqueda ajustando sus posiciones en cada iteración con base en; su experiencia personal y la experiencia de todo el enjambre.

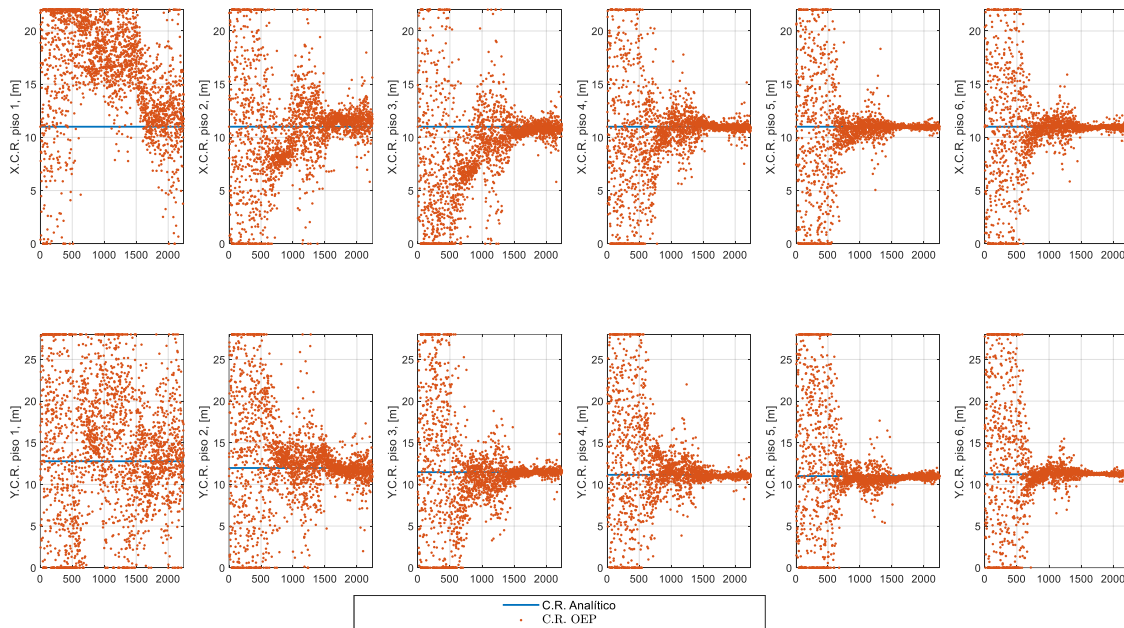
Ajustando esta herramienta que ofrece Matlab al caso del cálculo del C.R. se ha formulado el problema de optimización de la siguiente manera:

1. Descripción del problema de optimización: Hallar el punto dentro del espacio del entrepiso en el cual, al aplicar una fuerza horizontal arbitraria, el entrepiso experimente una rotación despreciable.
2. Función objetivo: Valores de rotación del diafragma debido a una fuerza aplicada en sentido "X" y en sentido "Y".
3. Sentido del objetivo: Minimizar la función objetivo.
4. Variables: Coordenada "X" del C.R. y coordenada "Y" del C.R.
5. Restricciones: Valores mínimos y máximos que puede tener el C.R. En este caso los valores de C.R. están limitados por dimensiones de la planta del entrepiso.

En la Figura 2.7 y 2.8 se muestra la convergencia del algoritmo PSO hacia las coordenadas de C.R. para el Caso 1 y Caso 2, respectivamente. Una vez que se implementó el método de PSO, se realizó una comparación de ambos métodos, obteniendo los resultados descritos en el subcapítulo 3.1.1.



**Figura 2.7 Representación visual de los resultados de C.R. por el método Analítico y PSO para el Caso 1 (Edificio de 3 pisos).**



**Figura 2.8 Representación visual de los resultados de C.R. por el método Analítico y PSO para el Caso 2 (Edificio de 6 pisos).**

## 2.4 Cálculo del Centro de Masa

Para determinar las coordenadas (XCM, YCM) del Centro de Masa se empleó el método de promedio ponderado en función de las masas parciales entre nodos debido al peso propio de los elementos estructurales, cargas sobrecargas muertas y cargas vivas asignadas a la estructura y transformadas a unidades de masa. La expresión general es:

$$X_{CM} = \frac{\sum_{i=1}^n m_i \cdot x_i}{\sum_{i=1}^n m_i} \quad (2.19)$$

$$Y_{CM} = \frac{\sum_{i=1}^n m_i \cdot y_i}{\sum_{i=1}^n m_i} \quad (2.20)$$

donde:

$m_i$  = masa concentrada  $i$  entre nodos del piso considerado.

$x_i, y_i$  = coordenadas del centro de la masa concentrada en planta.

$n$  = Número total de masas parciales en el piso.

En el presente estudio, las coordenadas del C.M. de cada piso fueron calculadas a partir de cada modelo en el análisis estructural realizado en el algoritmo de Matlab, utilizando las cargas definidas en la Sección 2.3.5. Los valores obtenidos se emplearon posteriormente para determinar la excentricidad al compararlos con las coordenadas del C.R. (Capítulo 2.3.8).

En la Tabla 2.11. y 2.12 se muestran las coordenadas del C.M. (XCM, YCM) para los modelos de estudio Caso 1 y Caso 2, respectivamente.

**Tabla 2.2.11. Coordenadas del C.M. del modelo Caso 1.**

Nº Piso	XCM [m]	YCM [m]
1	10.45	14.70
2	10.45	14.70
3	11.00	14.00

**Tabla 2.2.12. Coordenadas del C.M del modelo Caso 2.**

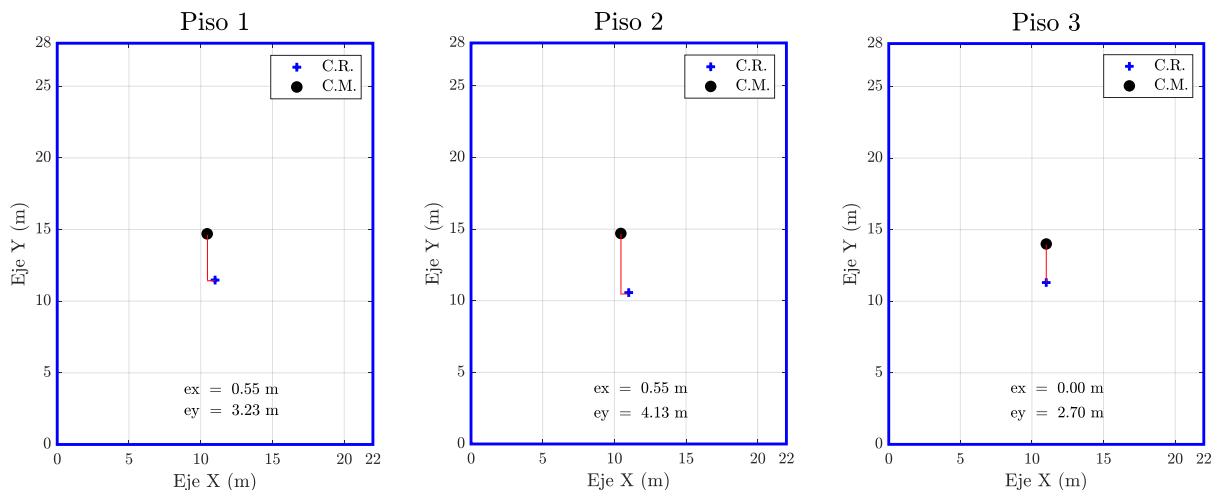
Nº Piso	XCM [m]	YCM [m]
1	10.45	14.70
2	10.45	14.70
3	10.45	14.70

<b>4</b>	10.45	14.70
<b>5</b>	10.45	14.70
<b>6</b>	11.00	14.00

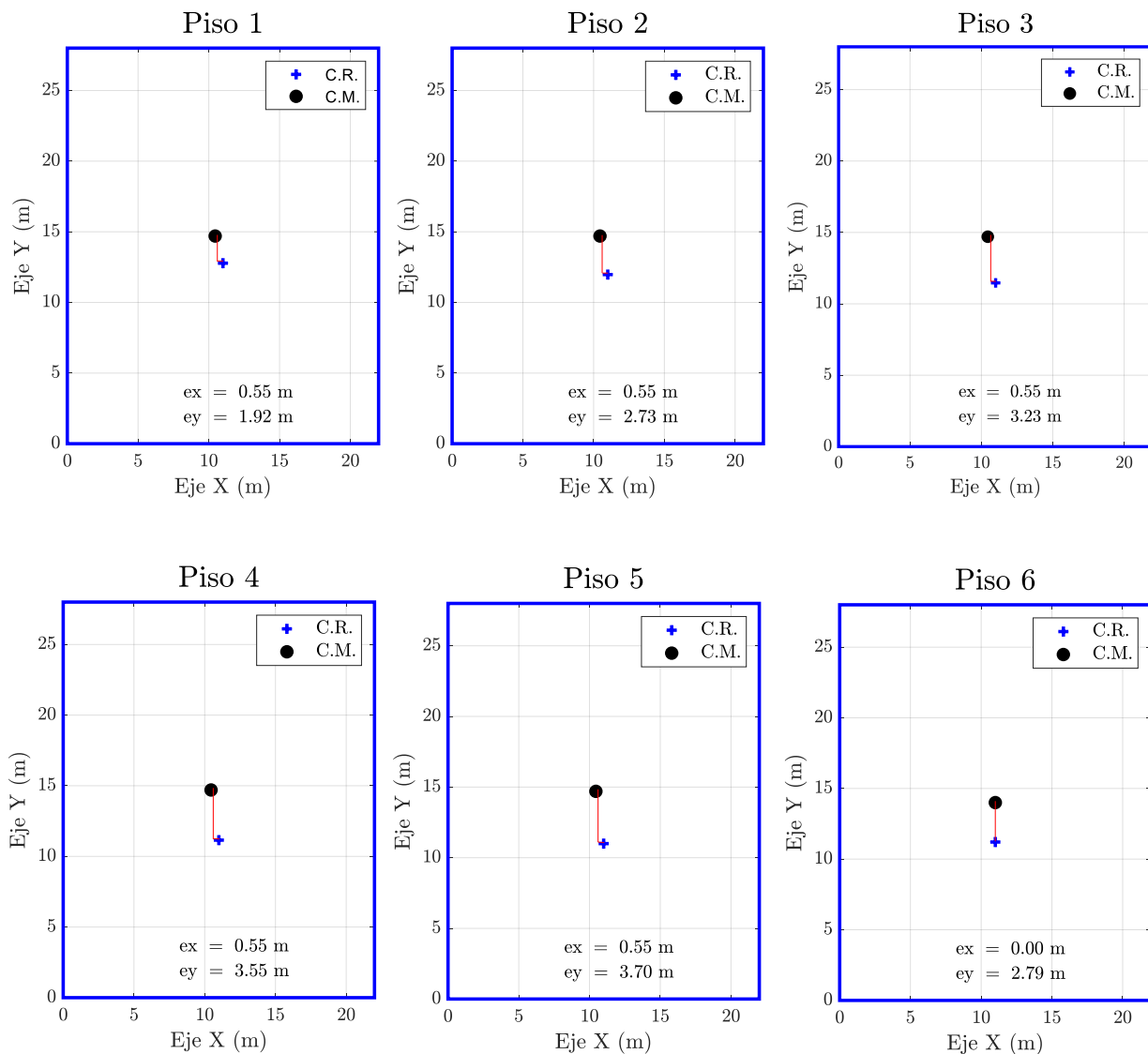
El conocimiento preciso de la ubicación del Centro de Masa en cada piso es fundamental para la fase de optimización, ya que el objetivo de la estrategia propuesta es reducir la excentricidad entre el C.M. y el C.R., disminuyendo así la respuesta torsional de las estructuras de estudio.

## 2.5 Cálculo de la excentricidad

Una vez que se obtuvo el Centro de Masa y el Centro de rigidez de cada piso de las estructuras de estudio, se realizó el cálculo de la excentricidad. Esta es la diferencia entre las coordenadas del Centro de Masa ( $X_{CM}$ ,  $Y_{CM}$ ) y las coordenadas del Centro de Rigidez ( $X_{CR}$ ,  $Y_{CR}$ ) de cada piso de las edificaciones. A continuación, en las figuras 2.9 y 2.10, se visualiza las excentricidades en dirección "X" ( $e_x$ ) y en dirección "Y" ( $e_y$ ) de cada piso del Caso 1 y Caso 2.



**Figura 2.9 Excentricidades por piso, Caso 1.**



**Figura 2.10 Excentricidades por piso, Caso 2.**

## 2.6 Desarrollo e implementación del algoritmo de optimización para dimensiones de columnas

Con el fin de reducir la respuesta torsional de las estructuras de estudio, se desarrolló e implementó en Matlab un algoritmo de Optimización por Enjambre de Partículas (PSO) con el objetivo de modificar estratégicamente las dimensiones de columnas de los modelos de estudio. El objetivo de esta estrategia es desplazar el Centro de Rigidez (C.R.) de cada piso hacia la posición del Centro de Masa (C.M.), reduciendo así la

excentricidad y transformando los modos fundamentales torsionales en modos traslacionales.

El proceso de optimización se estructuró en cinco etapas: 1) definición de la función objetivo, 2) selección de las variables de diseño y establecimiento de restricciones, 3) elección e implementación del algoritmo de optimización, 4) seguimiento de la convergencia de los resultados, y 5) verificación de la eficacia de la solución mediante el recálculo del RIT y de las excentricidades de las estructuras de estudio optimizadas.

### 2.6.1 Definición de la función objetivo

La función objetivo planteada inicialmente en esta investigación buscaba minimizar la distancia entre el C.R. y el C.M. en cada piso de la estructura. Matemáticamente, esta formulación se expresó como:

$$f_{obj}(x) = \sum_{j=1}^{N_p} [ e_x^2(j; x) + e_y^2(j; x) ] \quad (2.21)$$

donde

$x$  = vector de variables de diseño (dimensiones de ancho  $b$  y profundidad  $h$  de la sección transversal de las columnas de la estructura).

$N_p$  = número de pisos de la edificación.

$e_x(j; x)$  = Excentricidad en dirección X en el piso  $j$  para un conjunto de variables  $x$ .

$e_y(j; x)$  = Excentricidad en dirección Y en el piso  $j$  para un conjunto de variables  $x$ .

El objetivo era encontrar el conjunto de variables  $x$  (ancho y profundidad de columna) que minimizara  $f_{obj}(x)$  acercando en lo posible el C.R. al C.M. en cada piso de la estructura.

Sin embargo, la ejecución del algoritmo con la Ecuación (2.21) mostró que la minimización exclusiva de la excentricidad no resultaba eficiente para el problema de optimización, ya que tendía a rigidizar la estructura de manera excesiva. En la práctica, el algoritmo intentaba llevar las excentricidades a valores cercanos a cero incrementando

las dimensiones de las columnas hasta el límite máximo permitido por las restricciones de diseño.

Para evitar este efecto no deseado, se adoptaron dos medidas:

- 1) Flexibilización del objetivo de excentricidad: se permitió que la excentricidad máxima en cada dirección pudiera ser del orden del 10% de la longitud del piso en la dirección analizada, en lugar de forzarla a cero.
- 2) Incorporación de parámetros adicionales: se incluyeron el Porcentaje de Masa Participativa Torsional (PMT) y la Deriva inelástica  $\Delta$ .

El primer parámetro induce a que los modos fundamentales presenten valores de PMT cercanos a 0%, mientras que el segundo limita la reducción excesiva de derivas, evitando que la estructura se rigidice más allá de lo necesario para satisfacer los demás objetivos.

Adicionalmente, se observó que el optimizador modificaba las secciones de columna de manera arbitraria entre columnas pertenecientes a una misma fila. Por esta razón, se incorporó un término que controla la variabilidad entre columnas de un mismo eje mediante el coeficiente de variación (CV), con el objetivo de promover transiciones más suaves entre secciones.

Finalmente, cada término se ponderó mediante un coeficiente de importancia ( $w$ ) que representa su relevancia relativa dentro del objetivo global. De este modo, la función objetivo final se formuló como:

$$f_{obj}(x) = \sum_{j=1}^{N_p} \left[ \left( w_{exc} \cdot P_{ex}^2(j; x) \right) + \left( w_{exc} \cdot P_{ey}^2(j; x) \right) + \left( w_{PMT} \cdot P_{PMT}^2(j; x) \right) \right. \\ \left. + \left( w_{der} \cdot P_{\Delta x}^2(j; x) \right) + \left( w_{der} \cdot P_{\Delta y}^2(j; x) \right) \right] + \left( w_{CV} \cdot P_{CV}^2(x) \right) \quad (2.22)$$

donde:

$w_{exc}$  = peso asociado al término de excentricidades.

$w_{PMT}$  = peso asociado al porcentaje de masa participativa torsional.

$w_{der}$  = peso asociado al término de derivas en cada dirección.

$w_{CV}$  = peso asociado al término de variabilidad de dimensiones de columnas dentro de un mismo eje.

$P_{ex}(j; x)$  = penalización de la excentricidad en la dirección X en cada piso j.

$P_{ey}(j; x)$  = penalización de la excentricidad en la dirección Y en cada piso j.

$P_{PMT}(j; x)$  = penalización del porcentaje de masa participativa torsional en cada piso j.

$P_{\Delta x}(j; x)$  = penalización de la deriva en la dirección X en cada piso j.

$P_{\Delta y}(j; x)$  = penalización de la deriva en la dirección Y en cada piso j.

$P_{CV}(x)$  = penalización asociada a la variabilidad entre dimensiones de columnas de un mismo eje para el conjunto de variables x.

La inclusión de estos términos permitió equilibrar la reducción de la excentricidad con la preservación de un nivel de flexibilidad estructural adecuado, contribuyendo a una mejora integral del comportamiento dinámico de las estructuras optimizadas.

## 2.6.2 Variables de diseño y restricciones

Las variables de diseño corresponden a las dimensiones de las secciones transversales de columnas en cada modelo estructural. Cada variable está definida por el ancho (b) y profundidad (h) de la sección de hormigón armado de la columna, medidas en metros.

Se establecieron las siguientes restricciones:

- 1) Límites mínimos de dimensiones: las dimensiones no pueden ser menores a las requeridas para cumplir con los requisitos mínimos para pórticos especiales resistentes a momento según el ACI 318.
- 2) Límites máximos de dimensiones: se fijaron valores máximos para evitar secciones sobredimensionadas que incrementen innecesariamente el peso y el costo.

### **2.6.3 Algoritmo de optimización**

El problema se resolvió empleando el algoritmo de Optimización por Enjambre de Partículas (PSO) dada su eficacia en problemas de optimización no lineales con más de una variable.

En el PSO, cada partícula representa un conjunto potencial de dimensiones de columnas ( $x$ ), con una posición y velocidad inicial definidas dentro de los límites de diseño. En cada iteración, las partículas actualizan sus posiciones combinando:

- 1) Experiencia individual: mejor solución encontrada por la partícula hasta el momento.
- 2) Experiencia global: mejor solución encontrada por todo el enjambre.

La iteración se ejecutó hasta que se cumplió alguno de los siguientes criterios de convergencia:

- Alcanzar un valor mínimo predefinido de la función objetivo.
- No detectar mejoras significativas en un número consecutivo de iteraciones.
- Llegar al número máximo de iteraciones definido.

### **2.6.4 Convergencia de resultados**

Durante la ejecución del PSO, se registró la evolución de la función objetivo en cada iteración para evaluar la tendencia de convergencia. En ambos casos de estudio, el valor de la función objetivo mostró una disminución progresiva, alcanzando una meseta estable tras un número determinado de iteraciones.

En la Figura 2.11 y Figura 2.12 se presentan las curvas de convergencia para el Caso 1 y el Caso 2, respectivamente, evidenciando que el algoritmo logró estabilizarse en un valor mínimo que representa una reducción de la Excentricidad, el Porcentaje de Masa

Participativa torsional en los modos de vibración fundamentales, una reducción necesaria en las derivas y el cumplimiento del criterio de variabilidad aplicado.

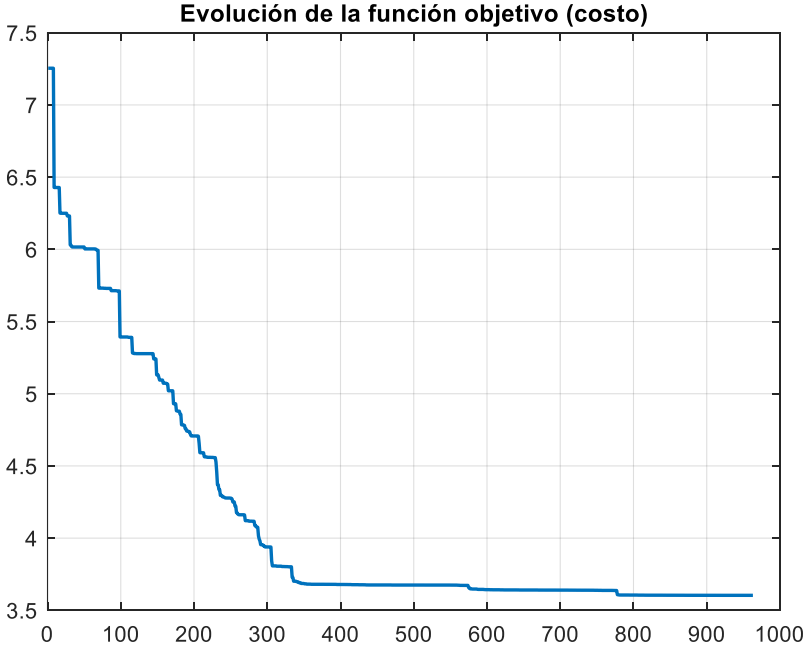


Figura 2.11 Función de costo de la OPT de dimensiones de columnas, caso 1.

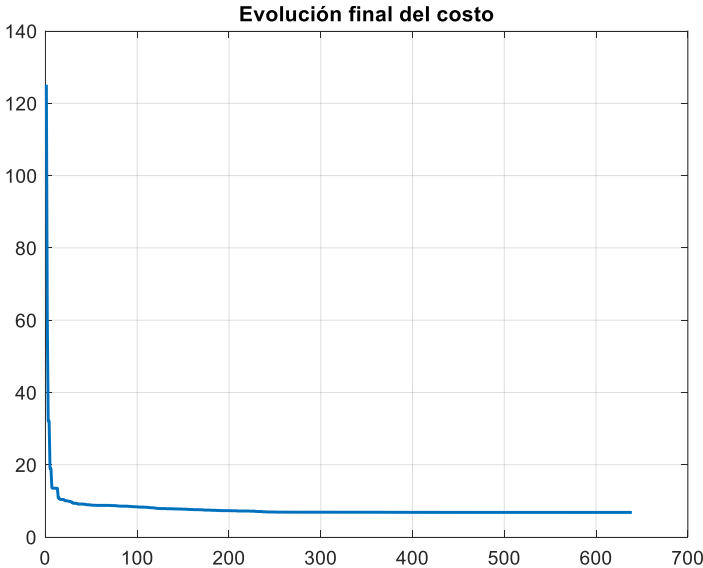
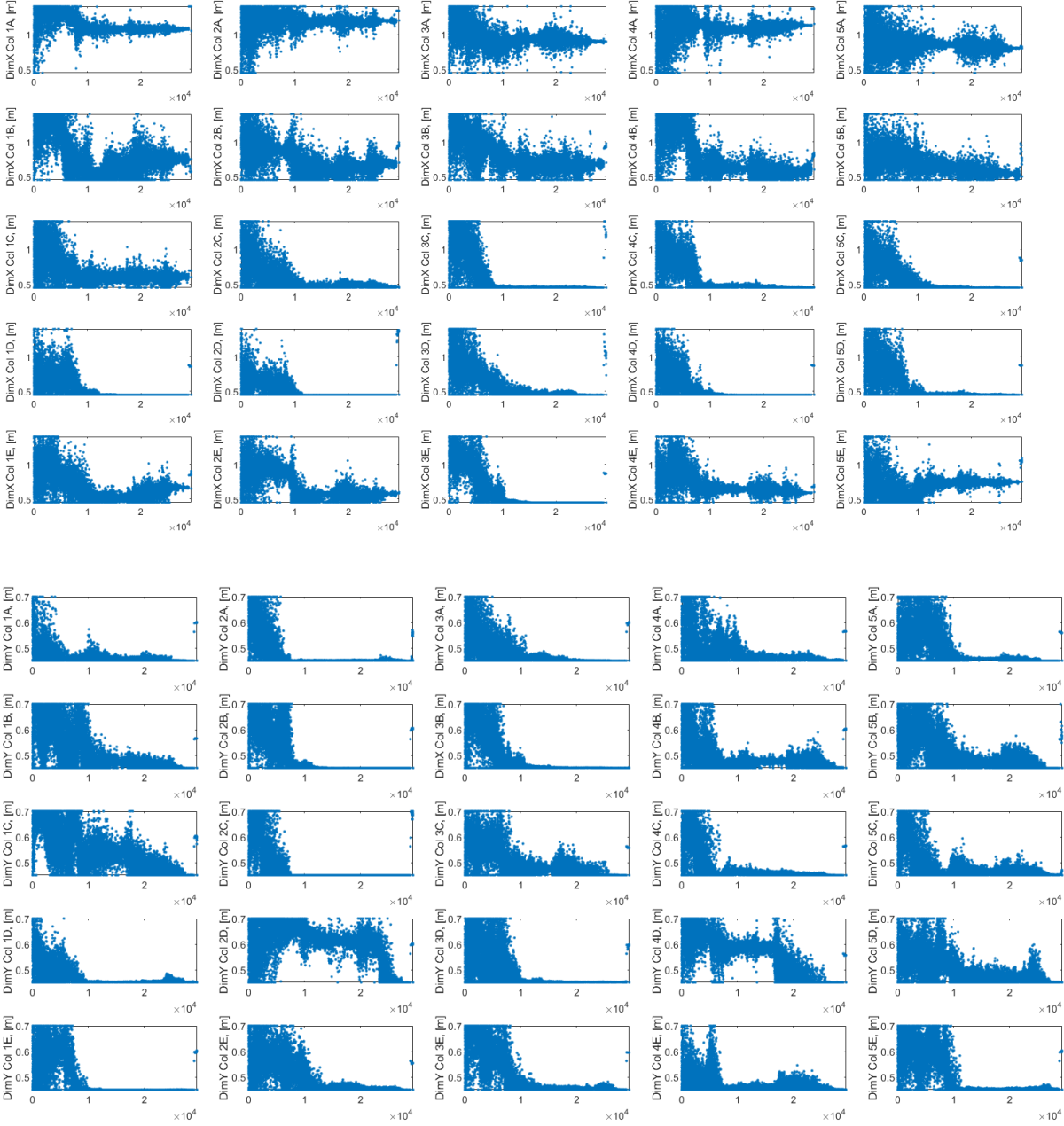
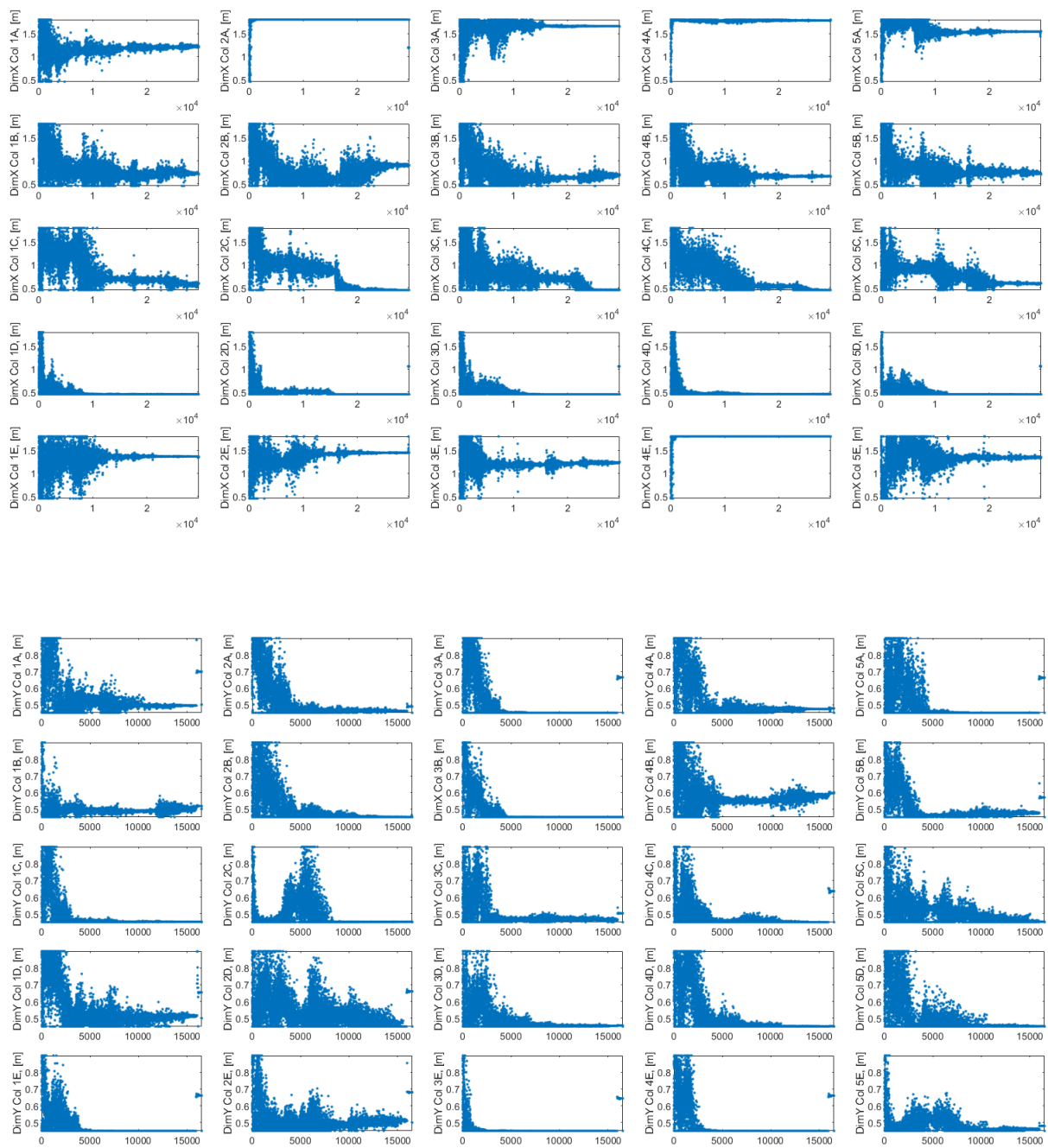


Figura 2.12 Función de costo de la OPT de dimensiones de columnas, caso 2.

En las Figuras 2.13 y 2.14 se muestra la convergencia del algoritmo PSO hacia las dimensiones (ancho y profundidad) de columnas optimizadas para el Caso 1 y Caso 2, respectivamente.



**Figura 2.13** Convergencia del algoritmo PSO para dimensiones de columna del Caso 1.



**Figura 2.14** Convergencia del algoritmo PSO para dimensiones de columna del Caso 2.

### **2.6.5 Verificación de la solución: Cálculo de RIT y excentricidad de las estructuras de estudio optimizadas**

Una vez obtenidas las dimensiones óptimas de las columnas, se actualizaron los modelos estructurales en la rutina de Matlab desarrollada para este trabajo de investigación para recalcular:

- 1) Coordenadas del C.R. optimizado en cada piso.
- 2) Excentricidad entre C.R. y C.M. optimizados.
- 3) Relación de Irregularidad Torsional (RIT) para cada piso, aplicando la ecuación (2.1).

# CAPÍTULO 3

## 3. RESULTADOS Y DISCUSIÓN

### 3.1 Resultados

#### 3.1.1 Comparación de ubicación del C.R. obtenidos mediante el Método Analítico y el PSO

El cálculo de los Centros de Rigidez (C.R.) se realizó de dos formas: el procedimiento analítico (método principal de esta tesis) y una verificación por optimización (PSO), obteniendo los resultados descritos en la tabla 3.1 y 3.2 para el edificio de tres pisos (Caso 1) y seis pisos (Caso 2), respectivamente.

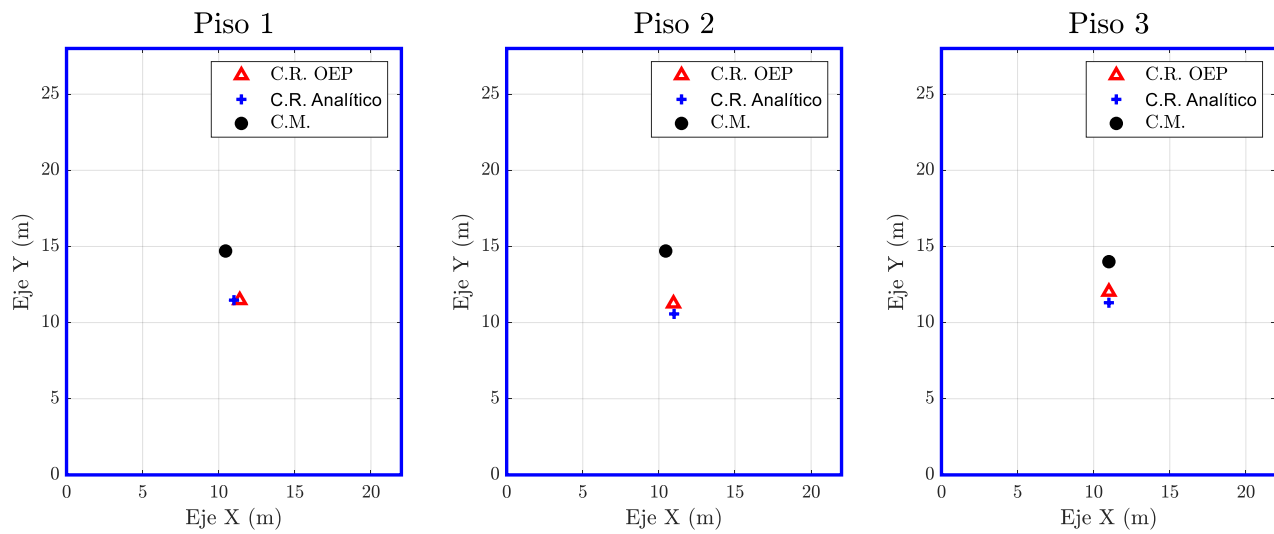
**Tabla 3.1 Centros de Rigidez de Edificio Caso 1. Método Analítico.**

N° Piso	XCR [m]			YCR [m]		
	Analítico	PSO	Error [%]	Analítico	PSO	Error [%]
1	11.00	11.38	3.45%	11.47	11.47	0.00%
2	11.00	10.96	0.36%	10.57	11.25	6.43%
3	11.00	11	0.00%	11.30	12.02	6.37%

**Tabla 3.2 Centros de Rigidez de Edificio Caso 2. Método Analítico.**

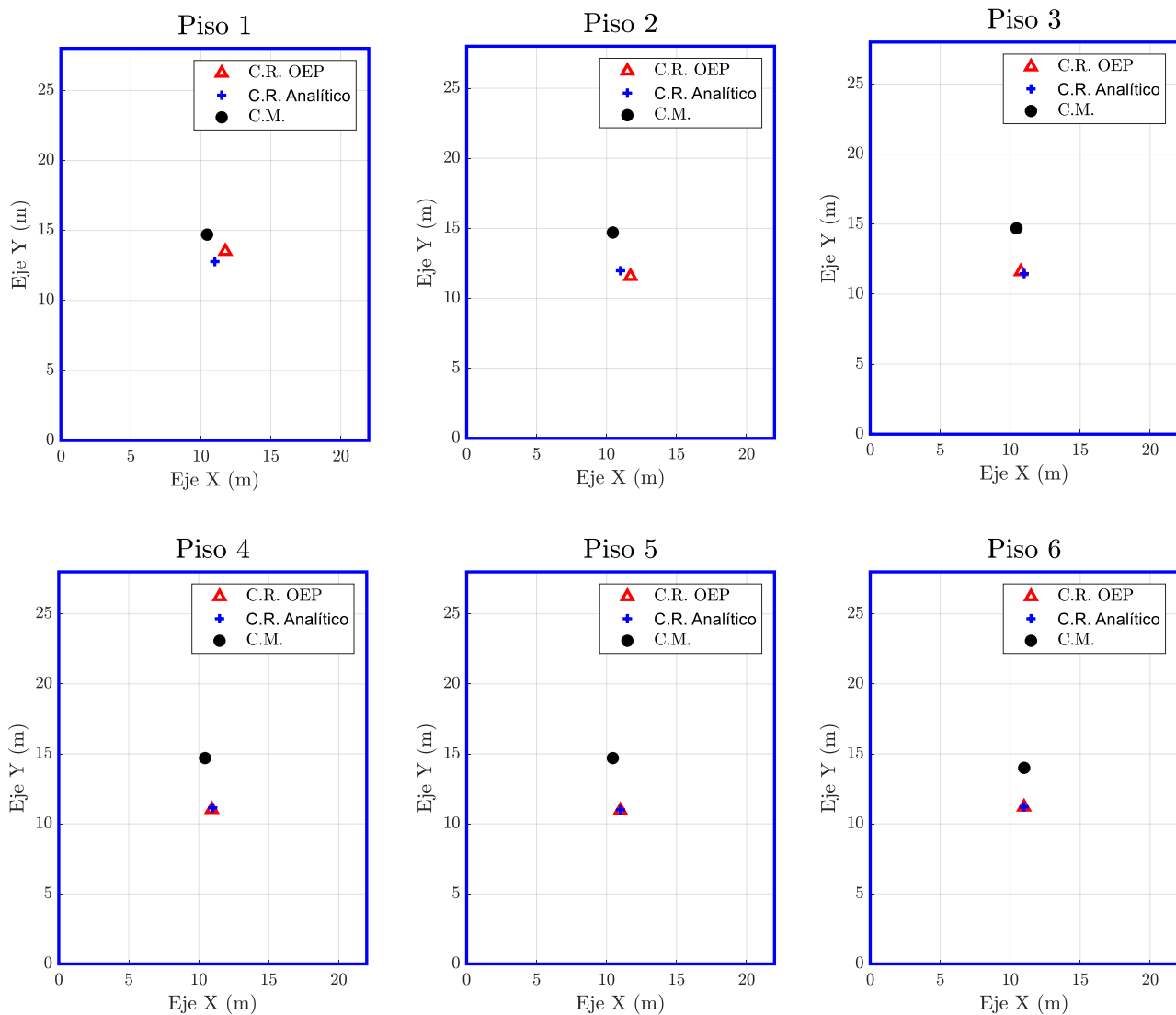
N° Piso	XCR [m]			YCR [m]		
	Analítico	PSO	Error [%]	Analítico	PSO	Error [%]
1	11.00	11.75	6.82%	12.78	13.51	5.71%
2	11.00	11.71	6.45%	11.97	11.57	3.34%
3	11.00	10.76	2.18%	11.47	11.61	1.22%
4	11.00	10.94	0.55%	11.15	11.03	1.08%
5	11.00	10.99	0.09%	11.00	10.96	0.36%
6	11.00	10.99	0.09%	11.21	11.21	0.00%

En la Figura 3.1, se grafican los resultados de ambos métodos empleados para el cálculo del C.R. del Edificio de tres pisos (Caso 1).



**Figura 3.1 Ubicación del Centro de Masa y Centro de Rigidez en cada piso del Caso 1.**

En la Figura 3.2 se muestra los resultados de ambos métodos empleados para el cálculo del C.R. del Edificio de seis pisos (Caso 2).



**Figura 3.2 Ubicación del Centro de Masa y Centro de Rigidez en cada piso del Caso 2.**

Los resultados muestran que las coordenadas del C.R. obtenidas por el método analítico presentan una muy buena concordancia con las calculadas mediante el método PSO, con errores relativos generalmente inferiores al 5% en la mayoría de los pisos.

Esta consistencia entre métodos valida la implementación del procedimiento analítico y proporciona confianza en los valores obtenidos, que son utilizados posteriormente para el cálculo de la excentricidad estática y para la formulación de la función objetivo en el algoritmo de optimización de rigideces.

### 3.1.2 Secciones de columnas propuestas por el algoritmo de optimización

Una vez que se ejecutó la optimización por Enjambre de Partículas (PSO) para rigidizar convenientemente las columnas de los modelos de manera que se reduzca su respuesta torsional, se obtuvieron los resultados de las Tablas 3.3 y 3.4 para el Caso 1 y Caso 2, respectivamente:

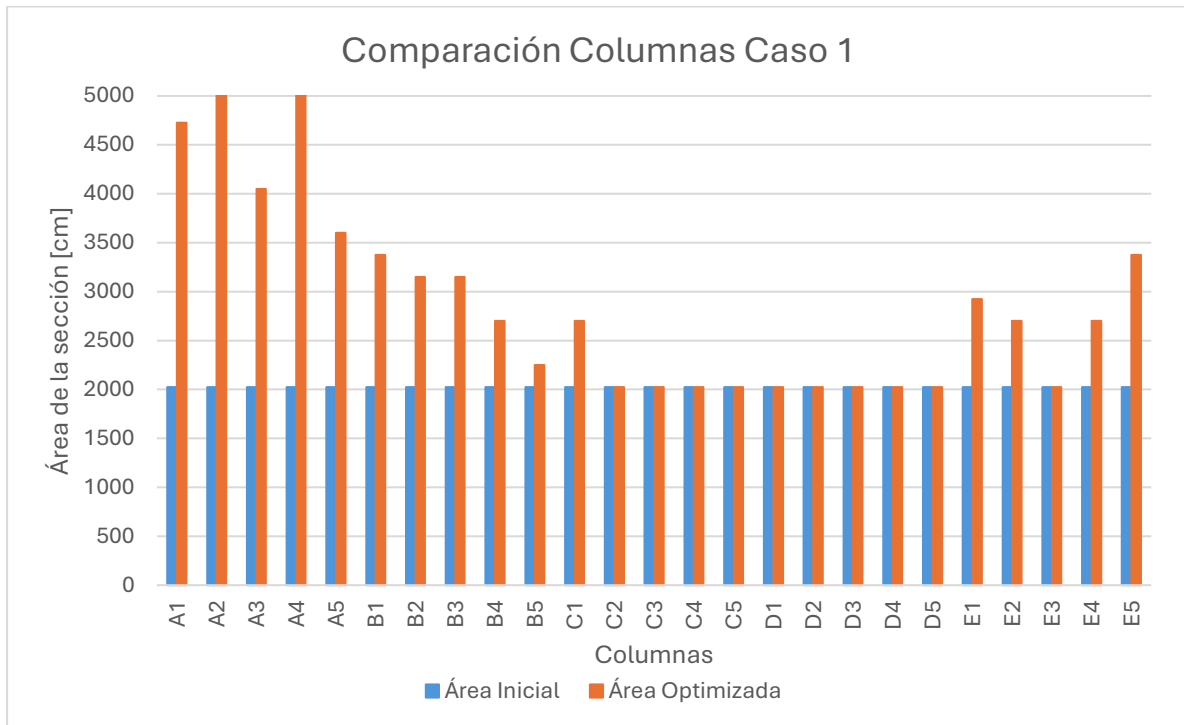
**Tabla 3.3 Comparación de secciones de Columnas, Caso 1.**

Columna	b inicial [cm]	h inicial [cm]	Área inicial [cm <sup>2</sup> ]	b Opt [cm]	h Opt [cm]	Área Opt [cm <sup>2</sup> ]	Variación [%]
A1	45	45	2025	105	45	4725	133%
A2	45	45	2025	120	45	5400	167%
A3	45	45	2025	90	45	4050	100%
A4	45	45	2025	115	45	5175	156%
A5	45	45	2025	80	45	3600	78%
B1	45	45	2025	75	45	3375	67%
B2	45	45	2025	70	45	3150	56%
B3	45	45	2025	70	45	3150	56%
B4	45	45	2025	60	45	2700	33%
B5	45	45	2025	50	45	2250	11%
C1	45	45	2025	60	45	2700	33%
C2	45	45	2025	45	45	2025	0%
C3	45	45	2025	45	45	2025	0%
C4	45	45	2025	45	45	2025	0%
C5	45	45	2025	45	45	2025	0%
D1	45	45	2025	45	45	2025	0%
D2	45	45	2025	45	45	2025	0%
D3	45	45	2025	45	45	2025	0%
D4	45	45	2025	45	45	2025	0%
D5	45	45	2025	45	45	2025	0%
E1	45	45	2025	65	45	2925	44%
E2	45	45	2025	60	45	2700	33%
E3	45	45	2025	45	45	2025	0%
E4	45	45	2025	60	45	2700	33%
E5	45	45	2025	75	45	3375	67%

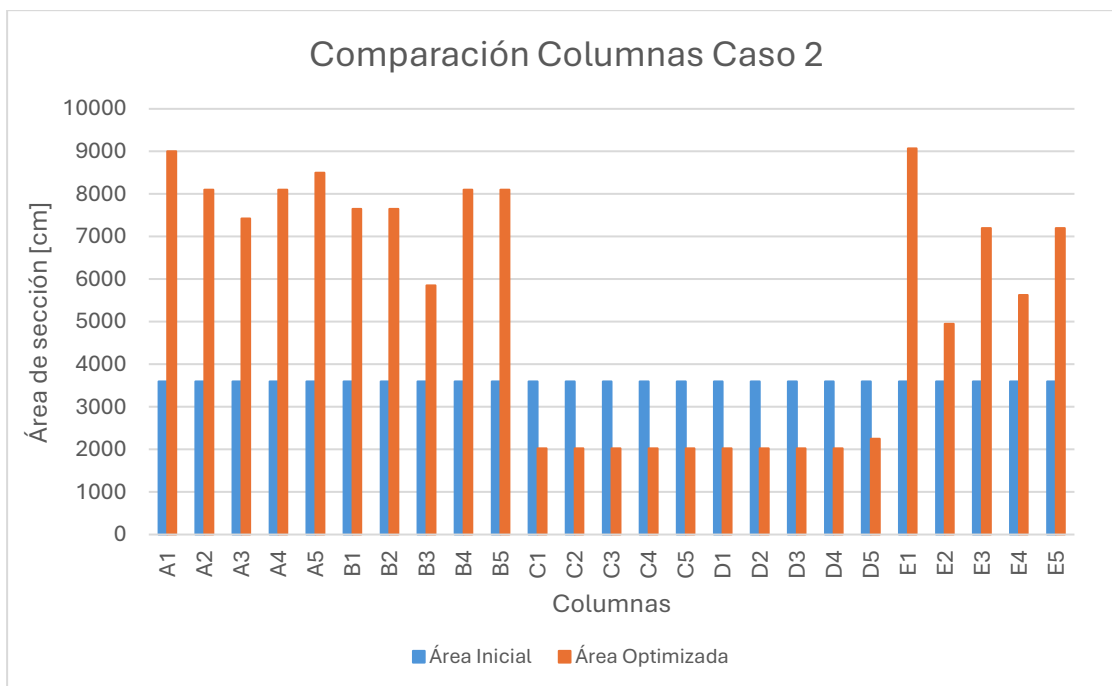
**Tabla 3.4 Comparación de secciones de Columnas, caso 2.**

<b>Columna</b>	<b>b inicial [cm]</b>	<b>h inicial [cm]</b>	<b>Área inicial [cm<sup>2</sup>]</b>	<b>b Opt [cm]</b>	<b>h Opt [cm]</b>	<b>Área Opt [cm<sup>2</sup>]</b>	<b>Variación [%]</b>
<b>A1</b>	60	60	3600	180	50	9000	150%
<b>A2</b>	60	60	3600	180	45	8100	125%
<b>A3</b>	60	60	3600	165	45	7425	106%
<b>A4</b>	60	60	3600	180	45	8100	125%
<b>A5</b>	60	60	3600	170	50	8500	136%
<b>B1</b>	60	60	3600	170	45	7650	113%
<b>B2</b>	60	60	3600	170	45	7650	113%
<b>B3</b>	60	60	3600	130	45	5850	63%
<b>B4</b>	60	60	3600	180	45	8100	125%
<b>B5</b>	60	60	3600	180	45	8100	125%
<b>C1</b>	60	60	3600	45	45	2025	-44%
<b>C2</b>	60	60	3600	45	45	2025	-44%
<b>C3</b>	60	60	3600	45	45	2025	-44%
<b>C4</b>	60	60	3600	45	45	2025	-44%
<b>C5</b>	60	60	3600	45	45	2025	-44%
<b>D1</b>	60	60	3600	45	45	2025	-44%
<b>D2</b>	60	60	3600	45	45	2025	-44%
<b>D3</b>	60	60	3600	45	45	2025	-44%
<b>D4</b>	60	60	3600	45	45	2025	-44%
<b>D5</b>	60	60	3600	50	45	2250	-38%
<b>E1</b>	60	60	3600	165	55	9075	152%
<b>E2</b>	60	60	3600	110	45	4950	38%
<b>E3</b>	60	60	3600	160	45	7200	100%
<b>E4</b>	60	60	3600	125	45	5625	56%
<b>E5</b>	60	60	3600	160	45	7200	100%

Las Figuras 3.3 y 3.4, muestran la comparación entre el área de las secciones de Columnas Iniciales (edificio con torsión) y las Columnas Optimizadas del Caso 1 y Caso 2, respectivamente.



**Figura 3.3 Comparación de Áreas de Columnas, caso 1.**



**Figura 3.4 Comparación de Áreas de Columnas, caso 2.**

Las Figuras 3.5 y 3.6 muestran la comparación visual de las secciones antes y después de la Optimización para el Caso 1 y Caso 2.

PLANTA - EDIFICIO DE 3 PISOS  
S/E.

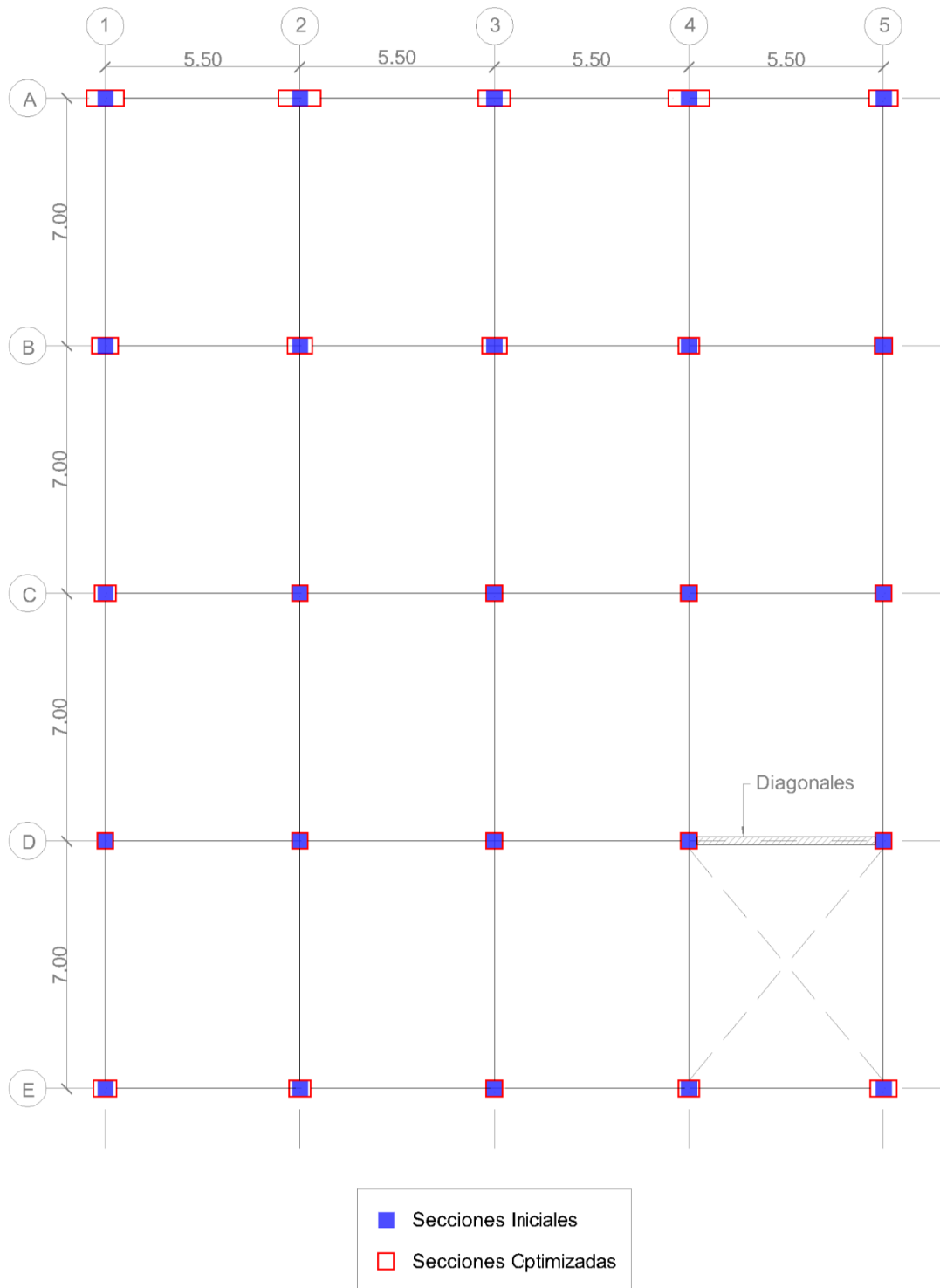


Figura 3.5 Comparación de Secciones de Columnas antes y después de la optimización, caso 1.

PLANTA - EDIFICIO DE 6 PISOS  
S/E.

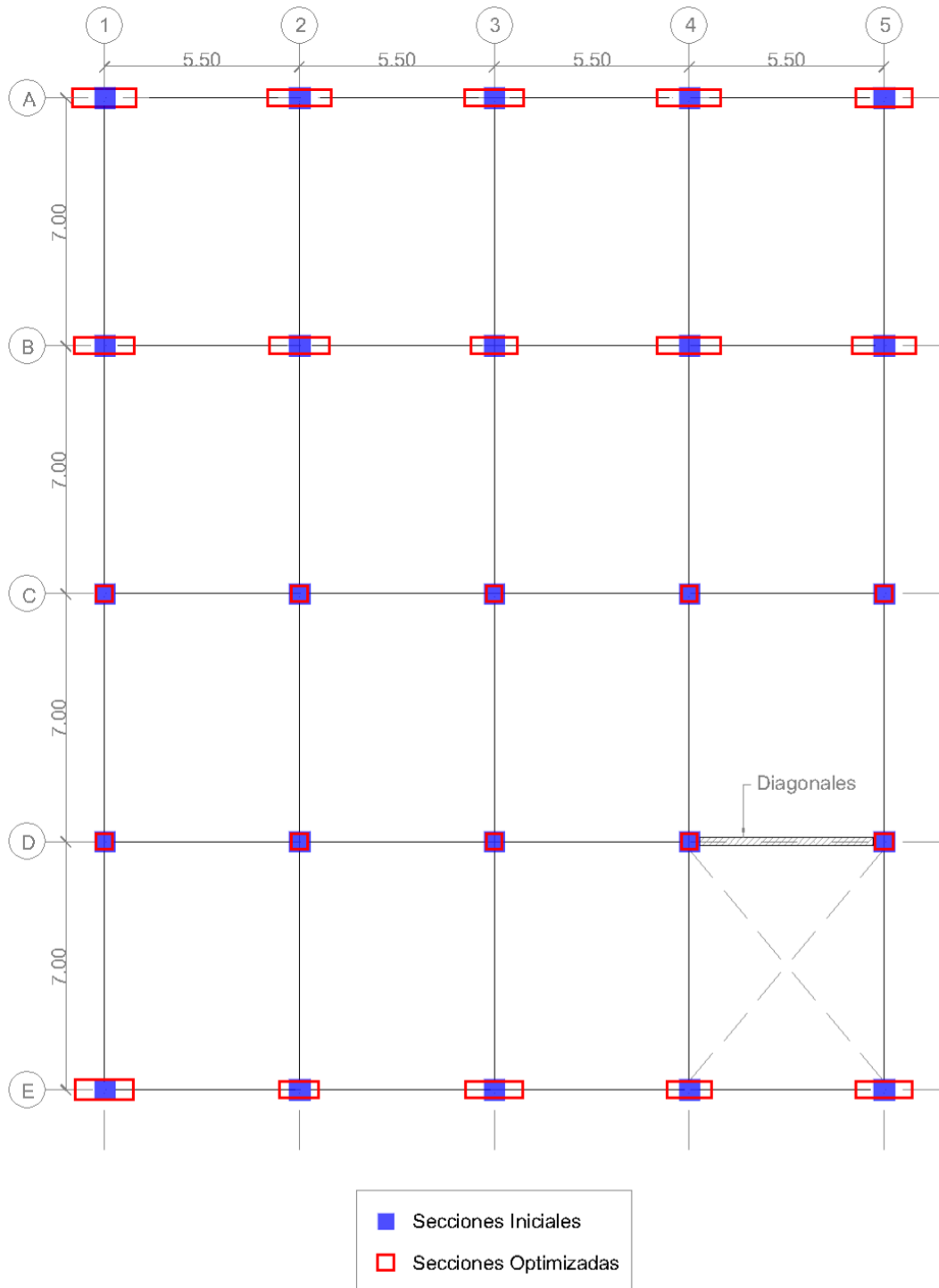


Figura 3.6 Comparación de Secciones de Columnas antes y después de la optimización, caso 2.

De las Tablas y Figuras precedentes se observa que, en la mayoría de los casos, el área de las secciones de columnas presenta incrementos significativos, alcanzando valores de hasta un 152% para el caso 1 y una 167 % para el caso 2, respecto a las dimensiones originales. Dicho crecimiento se refleja principalmente en el ancho  $b$  (dimensión en “X”) de columnas. Esto se debe a que el algoritmo de Optimización por Enjambre de Partículas (PSO) actúa rigidizando selectivamente las columnas con el fin de aproximar el Centro de Rigidez (C.R.) al Centro de Masa (C.M.).

Como la mayor excentricidad se presenta en el sentido “Y”, el algoritmo tiende a incrementar la dimensión en “X” de las columnas pertenecientes a los pórticos más alejados del C.R., logrando así “atraerlo” hacia dichas posiciones.

En general, el algoritmo logró una redistribución estratégica de rigideces: aumentó la sección en las columnas necesarias para corregir la respuesta torsional y mantuvo sin cambios aquellas cuya contribución era marginal. Esto confirma que la metodología implementada no solo persigue la minimización de la excentricidad, sino que también introduce un control sobre el sobredimensionamiento, garantizando un diseño estructural más balanceado y eficiente.

Es evidente que el aumento de las secciones de columnas implica un mayor consumo de hormigón, lo cual podría incrementar los costos constructivos de la estructura optimizada. No obstante, la reducción lograda en el comportamiento torsional de la edificación genera beneficios importantes. En particular, si el RIT (Relación de Irregularidad Torsional) disminuye a valores inferiores a 1.20, la edificación dejaría de clasificarse como irregular en planta. En tal caso, la demanda sísmica ya no estaría penalizada por el coeficiente de Irregularidad en Planta establecido en la NEC-15 ni se aplicaría el factor de amplificación torsional ( $A_x$ ) para el diseño de los elementos.

Este efecto repercute directamente en las demandas de diseño de los elementos, que en una estructura regular pueden reducirse aproximadamente en un 20 %. En consecuencia, la resistencia requerida por la estructura también disminuye,

traduciéndose en un ahorro potencial de costos del acero de refuerzo que podría compensar, al menos parcialmente, el mayor volumen de hormigón utilizado en las columnas optimizadas.

### 3.1.3 Comparación de derivas máximas de piso

Se realizó el cálculo de las derivas máximas inelásticas ( $\Delta_M$ ) de cada piso según lo determina la NEC-15 tanto para los modelos sin optimizar como para los modelos optimizados, mediante la expresión:

$$\Delta_M = 0.75 \cdot R \cdot \Delta_E \quad (3.1)$$

donde

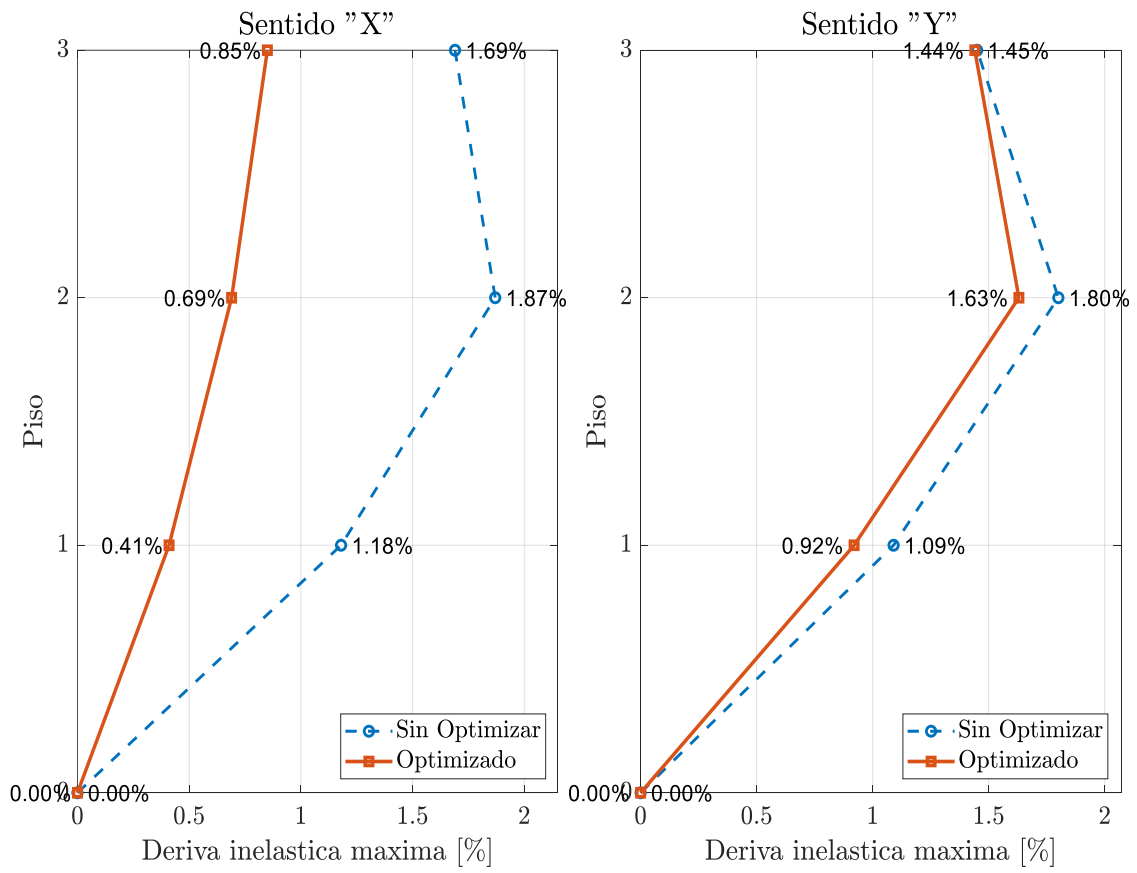
$\Delta_M$ : Deriva máxima inelástica

$\Delta_E$ : Desplazamiento obtenido en aplicación de las fuerzas laterales de diseño reducidas.

R: Factor de reducción de resistencia determinado a partir de la ductilidad esperada del sistema estructural.

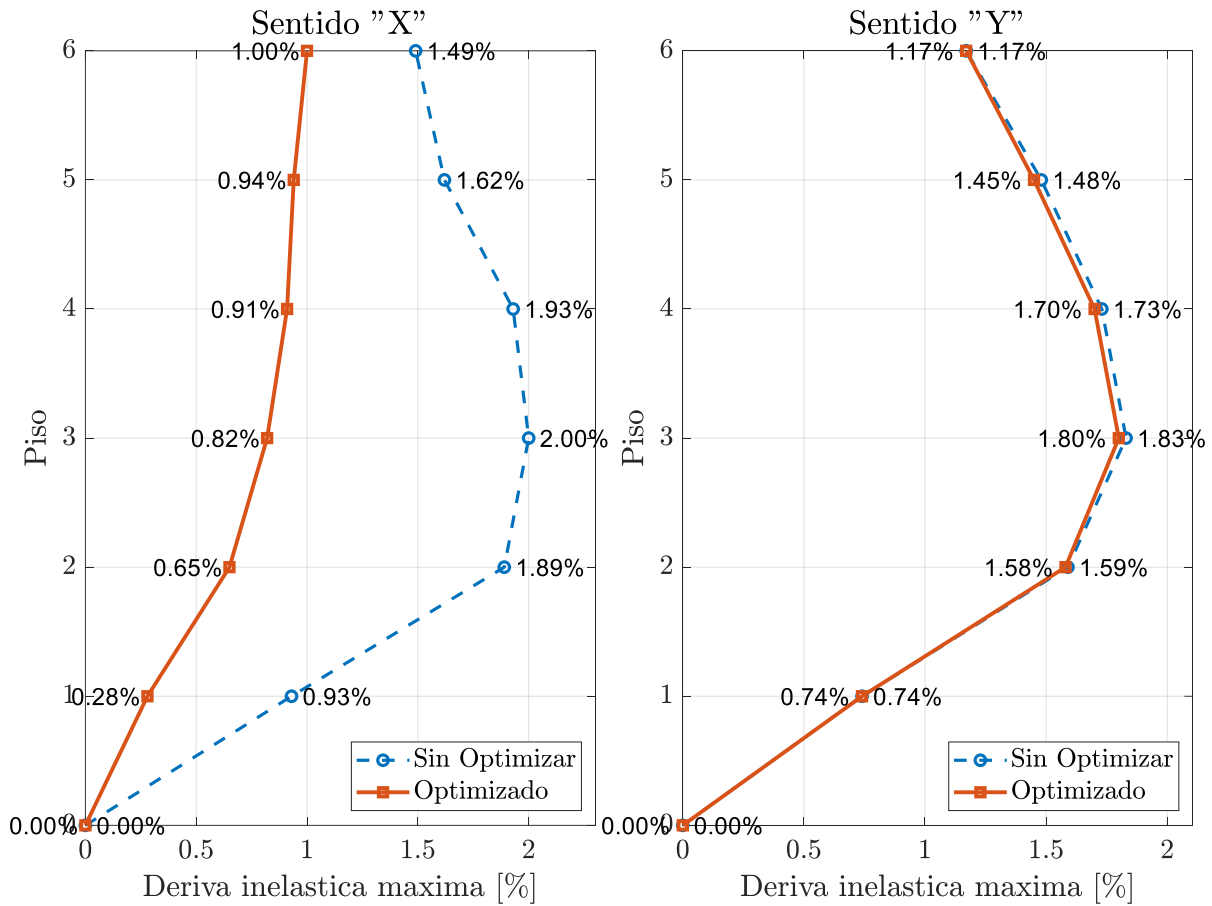
En las Figuras 3.7 y 3.8 se muestran las derivas máximas inelásticas de los modelos del Caso 1 y Caso 2, respectivamente.

### Derivas de piso



**Figura 3.7 Comparación de derivas máximas inelásticas de cada piso, caso 1.**

### Derivas de piso



**Figura 3.8 Comparación de derivas máximas inelásticas de cada piso, caso 2.**

Como se observa en la Figura 3.7, en dirección X se obtiene una disminución de hasta 63% entre las derivas antes y después de la optimización, mientras que en dirección Y la disminución alcanza un 9%. De forma parecida, para la Figura 3.8, en dirección X se observó una disminución de derivas de 61%, mientras que en la dirección Y la disminución es insignificante. Este comportamiento indica que el algoritmo de optimización rigidizó en mayor magnitud en el sentido X, debido a que los pórticos en esa dirección contribuyen de manera significativa en la variación de la coordenada Y del centro de rigidez (YCR). En otras palabras, el algoritmo de optimización tiende a modificar principalmente la rigidez en el sentido que reduce la excentricidad crítica. Tal como se constató en el subcapítulo 2.5, la mayor excentricidad inicial se presenta en la dirección Y, por lo tanto, el procedimiento de optimización prioriza ajustes en la rigidez

de los pórticos en dirección X, buscando acercar el centro de rigidez al centro de masa y, de esta forma, mitigar la respuesta torsional de la estructura.

Los resultados evidencian que la implementación del algoritmo de optimización no solo logra reducir las excentricidades críticas entre el centro de rigidez y el centro de masa, sino que también repercute directamente en la respuesta global de derivas. En particular, las diferencias significativas observadas en dirección X confirman que el procedimiento concentra la rigidización en el sentido más influyente para corregir la excentricidad dominante, mientras que en dirección Y las variaciones más pequeñas muestran un ajuste complementario que contribuye al balance estructural. Esta redistribución de rigideces mejora la capacidad del sistema para controlar desplazamientos laterales y, especialmente, para disminuir la demanda torsional en los pisos superiores, donde la interacción entre traslación y rotación suele ser más crítica. En consecuencia, la estrategia de optimización no solo cumple con el objetivo de acercar el C.R. al C.M., sino que también produce un efecto favorable en la respuesta dinámica global, garantizando un comportamiento más estable y eficiente frente a acciones sísmicas.

#### **3.1.4 Comparación de fuerzas internas en elementos**

Las estructuras de estudio se diseñaron de acuerdo con los requisitos del ACI-318, considerando un sistema de pórticos intermedios resistentes a momento. A partir del análisis estructural, se obtuvieron las fuerzas internas (axiales P, cortantes V y momentos flectores M) en vigas y columnas bajo las combinaciones de carga de diseño establecidas por la normativa.

Con el fin de enfocar el análisis en la condición más desfavorable, en este apartado se presentan únicamente los resultados de las envolventes de fuerzas internas correspondientes al primer piso, dado que este nivel concentra las mayores demandas de la estructura. Se presentan las envolventes de fuerza axial P, cortante V y momentos flectores M en columnas, y de cortante V y momentos positivos y negativos en vigas (Tablas 3.5 a la 3.8)., para los modelos sin optimizar (con torsión) y los modelos optimizados, bajo las combinaciones de carga de diseño indicadas en la norma. Los

pisos superiores presentan demandas menores y tendencias similares, por lo que su inclusión no aporta información adicional relevante para la comparación.

En las Tablas 3.5 y 3.6, se presentan los resultados de fuerzas internas en los elementos vigas y columnas del caso 1. Se ha aplicado una escala de color en las celdas, para resaltar los valores en función de su magnitud.

**Tabla 3.5 Fuerzas internas en columnas del primer piso antes y después de la optimización, caso 1.**

Columna	P inicial [tonf]	P Optimizado [tonf]	V 2-2 inicial [tonf]	V 2-2 Optimizado [tonf]	V 3-3 inicial [tonf]	V 3-3 Optimizado [tonf]	M 2-2 inicial [tonf-m]	M 2-2 Optimizado [tonf-m]	M 3-3 inicial [tonf-m]	M 3-3 Optimizado [tonf-m]
A1	25.20	21.10	10.75	11.80	12.15	16.90	33.60	79.60	29.10	37.50
A2	34.10	32.40	9.75	13.35	15.40	24.40	36.20	117.30	25.20	39.70
A3	32.40	32.00	8.25	10.40	15.30	12.00	36.10	51.85	21.00	31.00
A4	34.00	32.35	9.60	13.05	15.40	21.90	36.20	103.90	24.90	40.00
A5	25.00	20.90	10.40	9.60	12.85	8.60	36.80	36.80	28.20	29.20
B1	37.00	32.40	12.40	9.95	9.40	8.50	23.80	30.90	30.70	28.20
B2	57.45	57.10	10.40	9.10	10.75	7.40	25.10	25.60	25.80	25.70
B3	57.30	57.20	8.50	8.75	10.65	7.40	25.00	25.60	21.30	25.00
B4	57.45	57.20	10.20	8.20	10.70	5.30	25.00	16.85	25.45	22.50
B5	36.90	32.00	11.90	7.50	9.50	3.90	24.00	10.50	29.70	19.80
C1	32.60	31.50	12.30	8.45	6.15	6.10	14.70	17.60	30.60	23.10
C2	57.20	57.00	10.30	6.70	6.50	3.40	15.20	8.30	25.75	17.36
C3	57.00	57.00	8.40	6.45	6.55	3.40	15.10	8.30	21.20	16.80
C4	57.30	57.20	10.10	6.70	6.60	3.40	15.10	8.30	25.30	17.40
C5	32.70	31.20	11.85	7.00	6.20	3.60	15.00	8.40	29.60	18.00
D1	32.55	31.80	12.35	7.00	5.70	4.10	12.80	9.75	30.65	18.00
D2	57.50	57.40	10.35	6.70	5.90	4.15	13.00	9.70	25.80	17.40
D3	57.30	57.30	8.50	6.50	5.90	4.15	13.00	10.00	21.30	17.00
D4	108.20	103.50	10.70	7.25	6.15	4.40	13.20	9.50	25.90	17.10
D5	134.30	119.90	12.50	7.60	5.50	3.90	12.60	8.75	30.20	17.80
E1	21.60	21.00	10.80	8.15	7.50	10.00	18.40	29.00	29.20	24.00
E2	33.90	32.30	9.75	8.10	8.60	9.00	19.60	24.00	25.20	22.30
E3	32.20	31.60	8.22	6.45	8.60	4.90	19.50	11.25	24.30	16.80
E4	24.70	23.20	9.00	7.50	9.00	9.40	20.00	24.00	20.00	21.80
E5	12.50	12.00	9.80	8.50	7.30	14.30	18.00	43.20	27.60	27.00

En la tabla 3.5, en el estado inicial (con torsión), todas las columnas del edificio del caso 1 (3 pisos) poseían la misma sección 45x45 cm, por lo que la distribución de fuerzas internas dependía principalmente de la configuración en planta y de la distancia de los marcos respecto al centro de rigidez (CR). En consecuencia, los marcos que experimentaban mayores deformaciones (los más alejados del CR) concentraban una mayor participación en la resistencia lateral, lo cual se refleja en las envolventes de V y M de los ejes A, B y E.

Tras la optimización, las fuerzas internas de en columnas muestran una redistribución no uniforme, coherente con una rigidización selectiva. En particular, los valores de fuerzas axiales P tienden a mantenerse del mismo orden (con variaciones pequeñas) ya que su participación depende de la tributación de cargas gravitacionales, mientras que los cambios más relevantes se observan en los momentos flectores actuantes en la dirección de interés (M 2-2), donde las columnas más rígidas incrementan significativamente su demanda. Esto sugiere que dichas columnas pasan a desempeñar un rol más activo en el control del giro de entrepiso y en el desplazamiento del CR hacia el CM, de acuerdo con la lógica del algoritmo PSO (incremento estratégico de rigidez en elementos “más influyentes”). No obstante, después de la optimización también se observa una reducción moderada en las fuerzas cortantes de las columnas con respecto a la condición inicial.

**Tabla 3.6 Fuerzas internas en vigas del primer piso antes y después de la optimización, caso 1.**

Viga	V 2-2 inicial [tonf]	V 2-2 Opt [tonf]	M(+) 3-3 inicial [tonf-m]	M(+) 3-3 Opt [tonf-m]	M(-) 3-3 inicial [tonf-m]	M(-) 3-3 Opt [tonf-m]
Eje A	7.55	3.60	13.75	3.30	16.15	5.00
Eje B	7.65	5.30	8.30	3.00	13.10	6.80
Eje C	6.00	5.10	4.10	3.00	8.90	6.70
Eje D	5.40	5.15	3.30	2.70	5.80	5.20
Eje E	4.70	3.50	5.45	3.75	7.30	6.00
Eje 1	5.60	4.40	9.30	5.30	13.30	9.35
Eje 2	7.40	6.50	6.20	4.90	14.20	11.65
Eje 3	6.95	6.50	5.30	4.85	12.75	11.75
Eje 4	7.25	6.50	7.65	4.90	11.60	11.95
Eje 5	5.50	4.30	8.30	5.40	12.50	9.45

Por otra parte, en Tabla 3.6 correspondiente a las vigas del piso 1 se observa una reducción general de las envolventes de cortante y de los momentos positivos y

negativos en la condición optimizada. Esta disminución es consistente con un sistema que, al estar mejor balanceado torsionalmente y con una distribución de rigidez más eficiente, presenta menores rotaciones de diafragma y menores demandas en los marcos, reduciendo la transferencia de esfuerzos hacia las vigas.

Además, se puede observar que en la condición inicial (antes de la optimización) la mayor participación la tenían las vigas de los marcos más alejados del C.R., mientras que, para la condición optimizada, las fuerzas internas se reparten de manera más balanceada.

A continuación, las Tablas 3.7 y 3.8, se presentan los resultados de fuerzas internas en los elementos vigas y columnas del caso 2.

**Tabla 3.7 Fuerzas internas en columnas del primer piso antes y después de la optimización, caso 2.**

Columna	P inicial [tonf]	P Optimizado [tonf]	V 2-2 inicial [tonf]	V 2-2 Optimizado [tonf]	V 3-3 inicial [tonf]	V 3-3 Optimizado [tonf]	M 2-2 inicial [tonf-m]	M 2-2 Optimizado [tonf-m]	M 3-3 inicial [tonf-m]	M 3-3 Optimizado [tonf-m]
A1	100.00	92.20	12.90	13.80	14.70	20.30	51.65	175.75	38.85	48.30
A2	172.50	172.20	11.20	14.10	14.30	16.70	51.30	157.30	42.05	38.60
A3	172.60	172.30	11.20	13.40	14.30	13.50	51.25	122.30	42.05	35.40
A4	172.35	171.90	11.20	14.10	14.30	16.70	51.30	157.25	42.00	38.30
A5	99.10	91.65	13.00	13.10	14.60	14.25	51.30	147.50	43.75	44.85
B1	176.45	74.55	11.20	8.45	15.20	22.70	40.00	155.90	42.00	32.30
B2	345.25	347.40	9.20	8.30	15.30	18.65	36.80	153.40	34.75	31.70
B3	345.65	346.30	7.50	6.65	10.45	9.65	36.80	70.85	28.30	24.50
B4	345.25	347.55	9.10	8.65	10.50	21.40	36.85	180.60	34.30	34.00
B5	175.90	174.00	10.70	9.00	13.65	24.60	39.40	177.80	40.30	34.60
C1	173.50	71.80	11.20	3.40	12.8	5.60	28.70	8.40	42.00	9.70
C2	345.10	344.25	9.25	3.35	7.40	1.50	25.00	4.30	34.80	9.50
C3	345.50	344.00	7.50	3.30	7.40	1.50	25.00	4.40	28.25	9.40
C4	345.10	344.40	9.00	3.40	7.45	1.50	25.00	5.00	34.20	9.50
C5	172.75	171.15	10.60	3.40	13.00	5.50	27.75	8.10	40.30	9.80
D1	173.70	72.00	11.25	3.40	12.25	5.75	30.65	9.00	42.05	9.70
D2	345.25	345.15	9.30	3.30	8.40	1.60	27.00	4.90	34.85	9.50
D3	345.35	344.00	7.50	3.35	8.45	1.75	27.05	5.10	28.25	9.50
D4	411.40	358.65	12.05	6.15	10.35	3.60	28.00	6.60	37.10	12.40
D5	259.40	234.70	13.75	6.55	9.30	3.50	27.00	7.60	43.25	13.60
E1	93.80	95.50	12.85	15.80	12.45	39.40	38.75	254.50	43.30	57.80
E2	172.40	172.60	14.30	11.30	11.40	11.40	37.80	65.00	39.40	26.40
E3	172.40	174.40	12.75	13.35	11.40	28.80	37.80	190.00	33.30	35.15
E4	102.20	99.10	11.05	8.85	13.25	13.90	36.10	91.00	36.10	26.25
E5	26.30	26.45	9.40	7.20	10.00	28.15	36.45	189.60	39.05	30.00

En la tabla 3.7, en el estado inicial (con torsión), todas las columnas del edificio del caso 2 (3 pisos) poseían la misma sección 60X60 cm, por lo que, al igual que en el caso 1, la distribución de demandas tiende a concentrarse en los marcos que más participan en la resistencia lateral (los más sensibles a la torsión), lo cual se refleja en las envolventes de V y M de los ejes A, B y E.

Luego de la optimización, se observa una redistribución importante de las fuerzas internas de corte V y momento. En cambio, los valores de fuerzas axiales P tienden a mantenerse del mismo orden (con variaciones pequeñas) ya que su participación depende de la tributación de cargas gravitacionales.

Al igual que en el caso 1, los cambios más relevantes se observan en los momentos flectores actuantes en la dirección de interés (M 2-2), donde las columnas más rígidas incrementan significativamente su demanda. En M 3-3 hay una reducción (944.55 a 640.75 tonf·m; -32%), indicando que parte de la demanda que antes aparecía como momento en ese eje disminuye, compatible con una reducción del efecto torsional y/o con una redistribución hacia otros elementos/ejes.

En V 2-2, el total baja de 268.95 a 205.65 tonf (-24%), mostrando que en esa dirección la optimización logró una respuesta más eficiente (menor demanda global de corte en columnas del primer nivel). Por último, en V 3-3, el total sube de 279.8 a 331.9 tonf (+19%), lo cual sugiere que parte del “trabajo” lateral se trasladó hacia el eje asociado a esa componente.

**Tabla 3.8 Fuerzas internas en vigas del primer piso antes y después de la optimización, caso 2.**

Viga	V 2-2 inicial [tonf]	V 2-2 Opt [tonf]	M(+) 3-3 inicial [tonf-m]	M(+) 3-3 Opt [tonf-m]	M(-) 3-3 inicial [tonf-m]	M(-) 3-3 Opt [tonf-m]
Eje A	9.20	6.05	7.00	3.95	14.20	3.05
Eje B	13.30	11.50	8.70	7.70	16.00	5.25
Eje C	12.30	11.80	8.20	7.70	13.70	12.30
Eje D	12.30	11.70	8.20	7.70	14.00	12.50
Eje E	7.90	6.50	7.00	4.20	7.00	6.20
Eje 1	10.20	9.40	8.20	8.10	16.15	14.50
Eje 2	17.90	17.85	16.40	16.20	24.55	24.00
Eje 3	17.90	17.90	16.40	16.20	23.55	24.00

<b>Eje 4</b>	17.90	17.90	16.40	16.20	24.50	24.00
<b>Eje 5</b>	10.10	9.35	8.20	8.00	15.90	14.15

En las vigas (Tabla 3.8), se observa una reducción más homogénea. El cortante V 2-2, el momento positivo M(+) y el momento negativo M(-), presentan reducciones de hasta el 17% con respecto a los valores antes de la optimización.

En conjunto, los resultados indican que la optimización no produjo una rigidización global indiscriminada, sino una reconfiguración de la ruta resistente: ciertas columnas incrementan su participación (reflejada en aumentos de M y, en algunos casos, V), mientras que las vigas muestran una tendencia a disminuir sus demandas, evidenciando un comportamiento estructural más controlado frente a efectos torsionales.

### 3.1.5 Comparación de demanda de cuantías de acero en elementos

Con el fin de evaluar el impacto de la optimización sobre la demanda de refuerzo, se compararon las cuantías longitudinales requeridas en columnas, las cuantías a flexión (positiva y negativa) en vigas, y cuantías de acero transversal requerido, considerando únicamente los elementos del primer piso, por corresponder al nivel con mayores solicitaciones. Debido a que la cuantía de acero transversal requerida fue el mínimo para ambos casos antes y después de la optimización, solo se presentan y analizan los resultados de requerimiento de acero longitudinal. En las Tablas 3.9 y 3.10 se muestran las cuantías máximas de acero para el caso 1.

**Tabla 3.9 Cuantías de acero longitudinal en columnas, caso 1.**

Columna	Cuantía A. Longitudinal Inicial [%]	Cuantía A. Longitudinal Optimizado [%]
<b>A1</b>	4.40%	1.40%
<b>A2</b>	3.75%	1.35%
<b>A3</b>	2.85%	1.20%
<b>A4</b>	3.80%	1.35%
<b>A5</b>	4.50%	1.40%

<b>B1</b>	3.55%	1.30%
<b>B2</b>	2.50%	1.00%
<b>B3</b>	1.80%	1.00%
<b>B4</b>	2.55%	1.00%
<b>B5</b>	3.60%	1.30%
<b>C1</b>	2.70%	1.30%
<b>C2</b>	1.80%	1.00%
<b>C3</b>	1.50%	1.00%
<b>C4</b>	1.80%	1.00%
<b>C5</b>	2.65%	1.30%
<b>D1</b>	2.70%	1.30%
<b>D2</b>	1.80%	1.00%
<b>D3</b>	1.50%	1.00%
<b>D4</b>	2.30%	1.40%
<b>D5</b>	3.70%	1.90%
<b>E1</b>	2.85%	1.40%
<b>E2</b>	2.25%	1.20%
<b>E3</b>	1.75%	1.20%
<b>E4</b>	2.25%	1.30%
<b>E5</b>	2.85%	1.40%

En el estado inicial, la estructura presentaba una distribución de cuantías longitudinales relativamente alta en varias columnas, con valores que alcanzan aproximadamente 4.50% (por ejemplo, en columnas del eje A), y una proporción importante de columnas con cuantías en el rango de 2.0%–3.8%. Este comportamiento es coherente con lo observado en el subcapítulo anterior: al contar con secciones uniformes, la demanda interna se concentra en los marcos más comprometidos por la excentricidad y la torsión, generando mayores solicitaciones en ciertas columnas y elevando los requerimientos de acero.

Tras la optimización, se observa una reducción generalizada de la cuantía longitudinal en prácticamente todas las columnas del primer piso. La mayoría de los elementos optimizados se ubican alrededor de 1.0%–1.4%, con una excepción (D5  $\approx$  1.90%). En términos prácticos, esto indica que el algoritmo no solo modificó rigideces para acercar el Centro de Rigidez (C.R.) al Centro de Masa (C.M.), sino que además logró redistribuir más eficientemente las demandas internas, evitando concentraciones excesivas que obligaban a cuantías elevadas.

En consecuencia, el efecto global de la optimización en columnas puede interpretarse como un balance de la respuesta: al rigidizar estratégicamente elementos clave (según su aporte al control torsional), se reduce el sobreesfuerzo en columnas críticas y, con ello, disminuyen los requerimientos de acero longitudinal en el primer piso.

**Tabla 3.10 Cuantías de acero a flexión en vigas, caso 1.**

Viga	Cuantía A. Flexión (-) inicial [%]	Cuantía A. Flexión (-) Opt [%]	Cuantía A. Flexión (+) inicial [%]	Cuantía A. Flexión (+) Opt [%]
Eje A	1.25%	0.30%	1.10%	0.25%
Eje B	1.05%	0.40%	0.70%	0.25%
Eje C	0.75%	0.40%	0.45%	0.20%
Eje D	0.70%	0.45%	0.40%	0.25%
Eje E	0.75%	0.40%	0.60%	0.30%
Eje 1	1.10%	0.60%	0.75%	0.35%
Eje 2	0.95%	0.75%	0.50%	0.30%
Eje 3	0.85%	0.75%	0.40%	0.30%
Eje 4	0.75%	0.60%	0.50%	0.30%
Eje 5	0.75%	0.60%	0.75%	0.40%

En las vigas del primer piso, la comparación evidencia una reducción marcada de las cuantías requeridas tanto para momento negativo (-) como para momento positivo (+). En condición inicial, las cuantías de flexión negativa alcanzan valores del orden de 1.25%, mientras que tras la optimización descienden a valores cercanos a 0.30%–0.75%, con reducciones típicas entre -12% y -76% dependiendo del eje.

De manera análoga, las cuantías asociadas a flexión positiva también disminuyen de forma importante: en el estado inicial se observan valores de hasta 1.10%, mientras que luego de la optimización la mayoría se ubica alrededor de 0.20%–0.40%, con reducciones entre -25% y -77%.

Este resultado es consistente con el panorama observado en las envolventes de cortante y momento del subcapítulo 3.1.4: al reducirse las demandas internas en vigas (por disminución de momentos y cortantes), la cuantía requerida para satisfacer resistencia a flexión se reduce de forma proporcional.

En las Tablas 3.11 y 3.12 se muestran las cuantías máximas de acero para el caso 2.

**Tabla 3.11 Cuantías de acero longitudinal en columnas, caso 2.**

Columna	Cuantía A. Longitudinal Inicial [%]	Cuantía A. Longitudinal Optimizado [%]
A1	1.80%	1.00%
A2	1.30%	1.00%
A3	1.00%	1.00%
A4	1.35%	1.00%
A5	1.81%	1.00%
B1	1.21%	1.00%
B2	1.00%	1.00%
B3	1.00%	1.00%
B4	1.00%	1.00%
B5	1.25%	1.00%
C1	1.00%	1.00%
C2	1.00%	2.25%
C3	1.00%	2.25%
C4	1.00%	2.25%
C5	1.00%	1.00%
D1	1.00%	1.00%
D2	1.00%	2.25%
D3	1.00%	2.25%
D4	1.26%	2.60%
D5	2.10%	1.35%
E1	1.50%	1.00%
E2	1.00%	1.00%
E3	1.00%	1.00%
E4	1.00%	1.00%
E5	1.63%	1.00%

En el estado inicial, la mayoría de columnas del primer piso presenta cuantías entre 1.0% y 2.1%. Tras la optimización, se observa que la mayor parte de las columnas baja a 1%

**Tabla 3.12 Cuantías de acero a flexión en vigas, caso 2.**

Viga	Cuantía A. Flexión (-) inicial [%]	Cuantía A. Flexión (-) Opt [%]	Cuantía A. Flexión (+) inicial [%]	Cuantía A. Flexión (+) Opt [%]
Eje A	1.50%	0.65%	0.95%	0.55%
Eje B	1.30%	0.33%	0.75%	0.50%

<b>Eje C</b>	1.15%	0.80%	0.75%	0.60%
<b>Eje D</b>	1.15%	0.85%	0.75%	0.60%
<b>Eje E</b>	1.25%	0.85%	0.80%	0.60%
<b>Eje 1</b>	1.95%	1.80%	1.20%	1.15%
<b>Eje 2</b>	1.80%	1.70%	1.35%	1.35%
<b>Eje 3</b>	1.70%	1.70%	1.35%	1.35%
<b>Eje 4</b>	1.85%	1.75%	1.35%	1.35%
<b>Eje 5</b>	1.90%	1.75%	1.20%	1.20%

En las vigas en dirección X del primer piso, el cambio se observa más favorable; en flexión negativa (-) hay reducciones importantes en las vigas perimetrales (ejes A, B y E). En vigas interiores, los cambios son pequeños.

En flexión positiva (+) también se reducen sobre todo en vigas perimetrales, mientras que en vigas interiores la variación es menor.

Al comparar ambos grupos de elementos, se observa un patrón coherente:

La rigidización estratégica de columnas (realizada por el PSO) no implicó un incremento indiscriminado de demandas, sino una mejor redistribución global del sistema resistente.

Esa redistribución se tradujo en menores envolventes de momentos y cortantes en vigas, y en una disminución general de cuantías longitudinales en columnas.

En consecuencia, la estructura optimizada no solo mejora su desempeño torsional (objetivo principal), sino que también reduce de forma notable la demanda de refuerzo en el primer nivel, lo que anticipa un impacto favorable en eficiencia estructural y costos de materiales.

### **3.1.6 Comparación de modos de vibración mediante porcentajes de masa participativa**

Las Tablas 3.9 y 3.10 muestran los porcentajes de masa participativa para los modos de vibración que tienen mayor importancia en la respuesta dinámica, para el caso 1 y caso

2, respectivamente. Como información adicional, se presentan los periodos de cada modo de vibración.

**Tabla 3.13 Porcentajes de masa participativa antes y después de la optimización, caso 1.**

Modo	Modelo sin optimizar (con torsión)				Modelo Optimizado			
	Ux [%]	Uy [%]	Rz [%]	T [seg]	Ux [%]	Uy [%]	Rz [%]	T [seg]
1	0.02	81.01	0.07	0.85	0.00	79.63	0.00	0.83
2	<b>42.12</b>	<b>0.05</b>	<b>40.35</b>	<b>0.71</b>	<b>75.84</b>	<b>0.00</b>	<b>0.00</b>	<b>0.52</b>
3	37.06	0.00	39.96	0.51	0.00	0.01	76.32	0.43
4	0.02	14.48	0.00	0.23	0.00	15.56	0.04	0.21
5	14.57	0.04	0.46	0.21	14.41	0.00	5.26	0.16
6	2.02	0.03	14.96	0.17	4.71	0.00	13.26	0.10
...	...	...	...		...	...	...	

En el Caso 1 (Tabla 3.9), la comparación de los porcentajes de masa participativa evidencia un cambio claro en la naturaleza dinámica del sistema después de la optimización.

En el modelo sin optimizar, la respuesta está acoplada en los modos principales: el Modo 2 concentra simultáneamente una alta participación traslacional en X ( $U_x \approx 42\%$ ) y una participación rotacional importante ( $R_z \approx 40\%$ ), mientras que el Modo 3 también mantiene una participación rotacional elevada ( $R_z \approx 40\%$ ). Esto es consistente con un comportamiento con acoplamiento torsión–traslación en los modos predominantes.

En el modelo optimizado, ese acoplamiento se reduce totalmente. El Modo 2 pasa a ser predominantemente traslacional en X ( $U_x \approx 76\%$ ) con rotación prácticamente nula ( $R_z \approx 0\%$ ). A su vez, la participación rotacional se relocaliza principalmente en el Modo 3, donde  $R_z$  aumenta de forma significativa ( $\approx 76\%$ ), mientras que la traslación en ese modo se vuelve despreciable.

**Tabla 3.14 Porcentajes de masa participativa antes y después de la optimización, caso 2.**

Modo	Modelo sin optimizar (con torsión)				Modelo Optimizado			
	Ux [%]	Uy [%]	Rz [%]	T [seg]	Ux [%]	Uy [%]	Rz [%]	T [seg]
1	0.06	75.25	0.15	1.61	0.00	75.67	0.00	1.67
<b>2</b>	<b>43.30</b>	<b>0.16</b>	<b>33.34</b>	<b>1.35</b>	<b>69.74</b>	<b>0.00</b>	<b>0.01</b>	<b>0.84</b>
3	34.11	0.00	42.72	0.96	0.03	0.00	69.42	0.70
4	0.02	13.26	0.00	0.44	0.00	13.12	0.03	0.46
5	9.38	0.04	2.48	0.38	0.09	5.76	0.12	0.21
6	2.42	0.01	10.35	0.31	13.28	0.05	4.12	0.20
7	0.02	5.81	0.06	0.19	4.31	0.00	13.75	0.15
8	5.10	0.03	0.60	0.18	0.00	3.11	0.00	0.11
...	...	...	...		...	...	...	

En el Caso 2 (Tabla 3.10), los porcentajes de masa participativa muestran nuevamente que, antes de optimizar, los modos principales presentan acoplamiento traslación-torsión, mientras que después de la optimización la respuesta se desacopla y la torsión se concentra en un modo posterior.

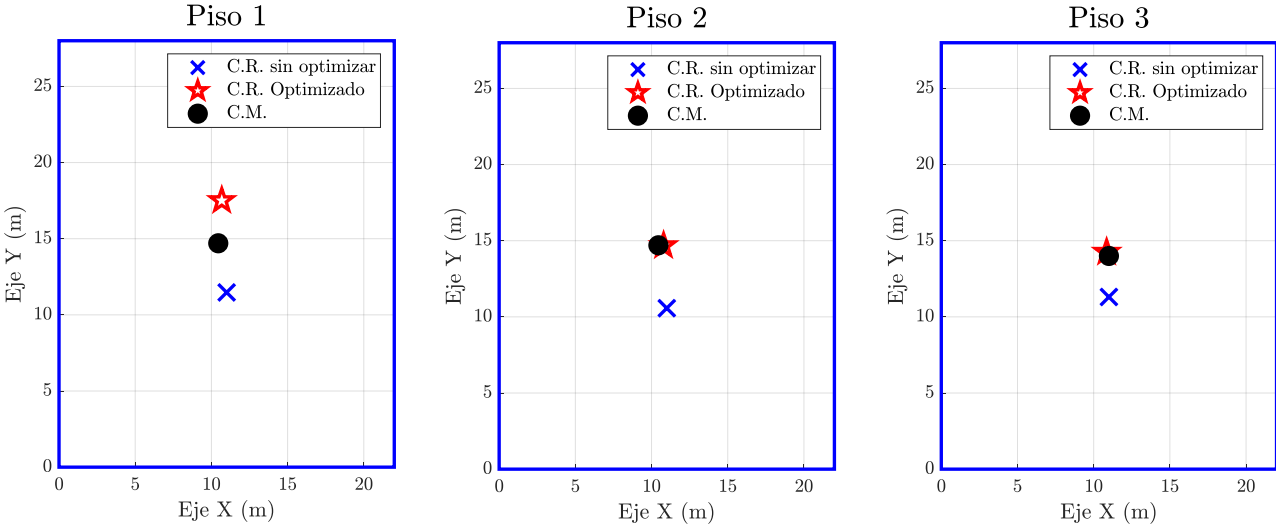
En el modelo sin optimizar, el Modo 2 combina una participación traslacional importante en X ( $U_x \approx 43\%$ ) con una participación rotacional relevante ( $R_z \approx 33\%$ ). Además, el Modo 3 mantiene una contribución rotacional aún mayor ( $R_z \approx 43\%$ ), con traslación también significativa ( $U_x \approx 34\%$ ). Esto indica que la estructura presenta una respuesta torsional apreciable asociada a los modos de menor orden, es decir, torsión participando desde los modos dominantes.

En el modelo optimizado, el comportamiento cambia de forma clara: el Modo 2 pasa a ser traslacional en X ( $U_x \approx 70\%$ ) con rotación prácticamente nula ( $R_z \approx 0\%$ ), mientras que la rotación se concentra en el Modo 3 ( $R_z \approx 69\%$ ), con traslación casi despreciable ( $U_x \approx 0\%$ ). En otras palabras, la optimización logra separar la traslación dominante (modo bajo) de la torsión (modo posterior), reduciendo el acoplamiento torsional en los modos fundamentales.

Entonces, tanto para el Caso 1 como para el Caso 2 se observa el mismo comportamiento: una reorganización modal posterior a la optimización, en la que los modos fundamentales concentran la mayor participación de masa traslacional en las direcciones principales, mientras que la participación torsional se traslada a un modo de orden superior, con participación traslacional despreciable. Este desacoplamiento entre traslación y torsión es indicativo de una respuesta global más controlada, ya que reduce la probabilidad de que la torsión gobierne la dinámica en el rango modal que más contribuye a los desplazamientos y derivas, lo que resulta consistente con una mejora del desempeño torsional de las estructuras optimizadas.

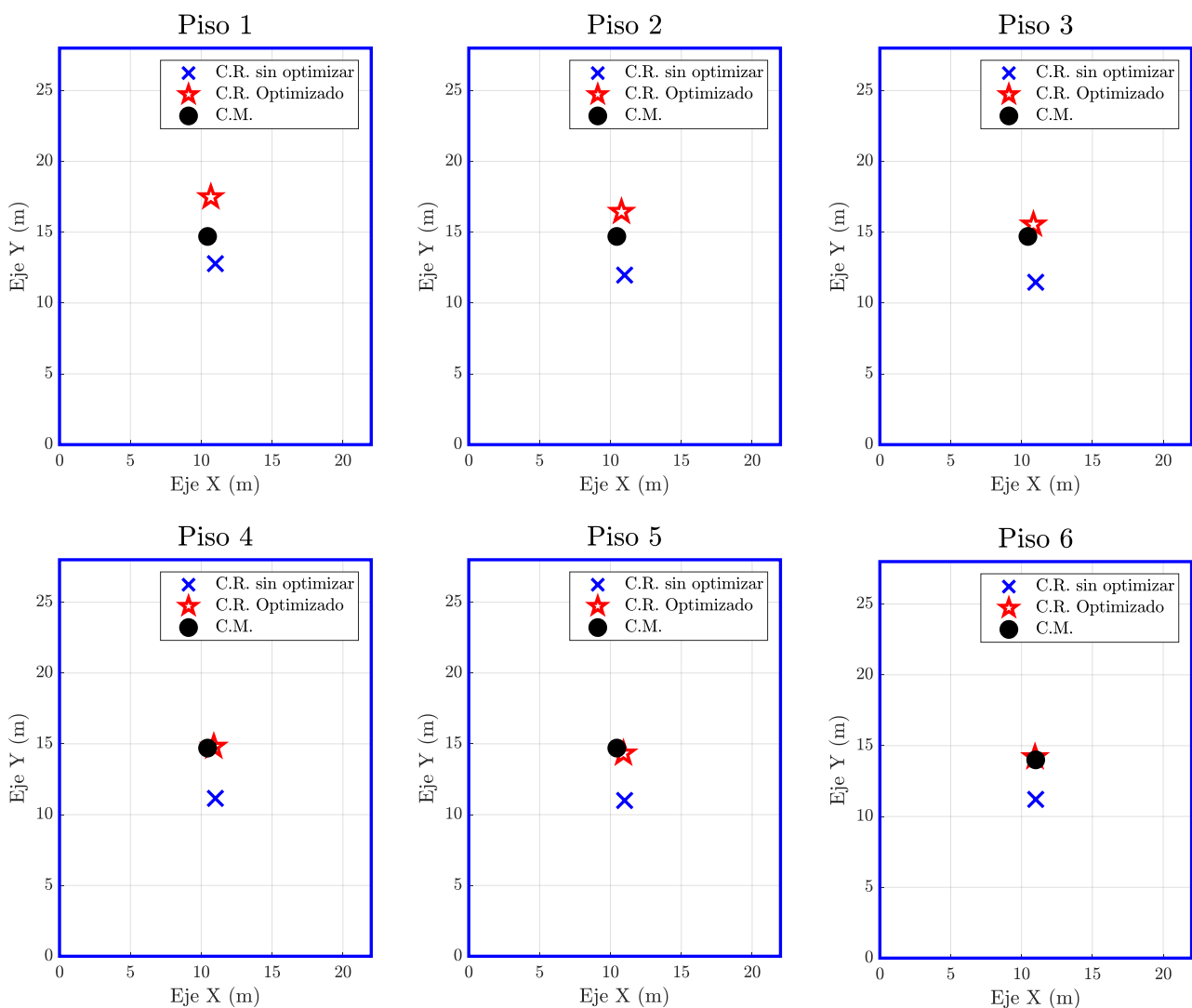
**3.1.7 Comparación de excentricidades estáticas**

Las Figuras 3.9 y 3.10 presentan la ubicación del Centro de Rigidez (C.R.) y del Centro de Masa (C.M.) en cada piso, para los modelos sin optimizar y optimizado, en los casos 1 (3 pisos) y 2 (6 pisos), respectivamente. La excentricidad estática se interpreta como la separación entre ambos centros en planta; por tanto, una menor distancia C.R.–C.M. implica una distribución más equilibrada de rigideces y una menor propensión a generar respuesta torsional ante acciones laterales.



**Figura 3.9 Comparación de las excentricidades antes y después de la optimización, caso 1.**

En la Figura 3.9 se observa que, en el modelo sin optimizar, el C.R. se encuentra desplazado respecto al C.M., con una separación más evidente en el Piso 1, llegando a una excentricidad máxima del 15% en la dirección Y (crítica), lo que es coherente con un comportamiento torsional más pronunciado en ese nivel. Tras la optimización, el C.R. se desplaza hacia el C.M. en los tres pisos, reduciendo la excentricidad estática máxima a un 10% y logrando una alineación más cercana entre ambos centros.



**Figura 3.10 Comparación de las excentricidades antes y después de la optimización, caso 2.**

En la Figura 3.10 se evidencia una tendencia similar: el modelo sin optimizar mantiene una separación consistente entre el C.R. y el C.M. a lo largo de los pisos, con una excentricidad estática máxima del 13% en la dirección Y (crítica), lo cual favorece la aparición de torsión. Luego de la optimización, el C.R. se aproxima al C.M. en todos los

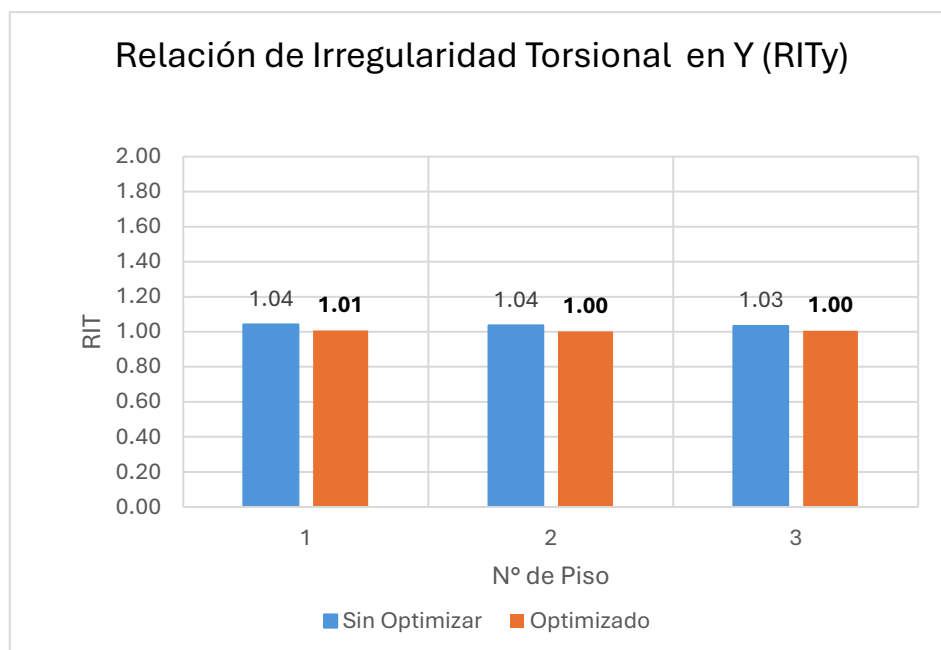
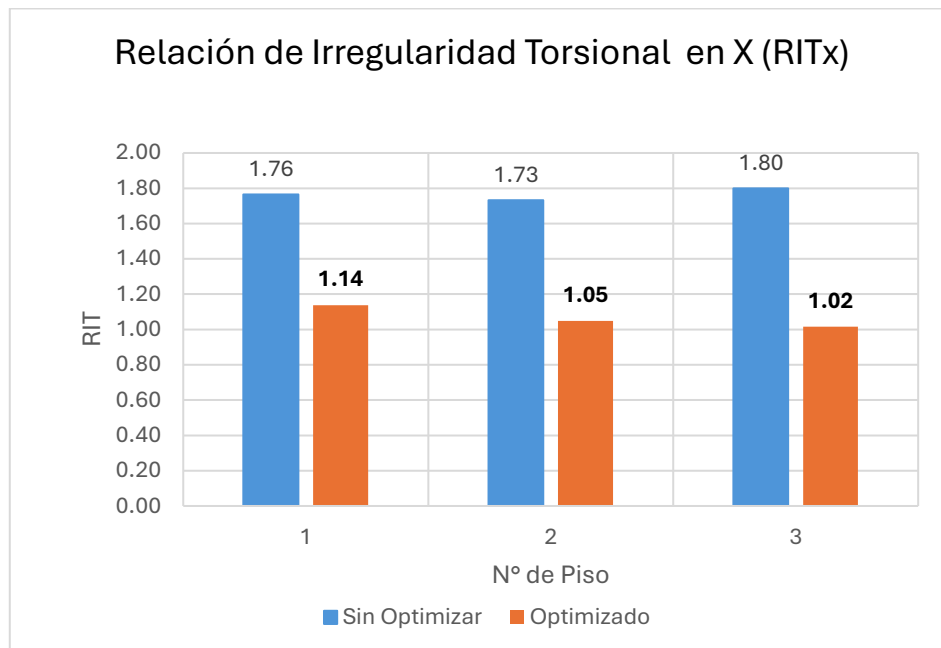
niveles, mostrando una reducción de la excentricidad estática máxima al 10%. Además, la mejora se aprecia de forma relativamente uniforme en pisos superiores, lo que sugiere que el ajuste de rigideces no se concentró únicamente en un nivel, sino que produjo un reequilibrio global del sistema resistente.

En ambos casos, la optimización desplaza el C.R. en dirección al C.M., disminuyendo la excentricidad estática por piso. Este resultado es consistente con el objetivo del procedimiento de diseño: redistribuir rigideces de manera estratégica para reducir la respuesta torsional y, con ello, mejorar el desempeño dinámico esperado de la estructura.

### **3.1.8 Comparación de la Relación de Irregularidad Torsional (RIT)**

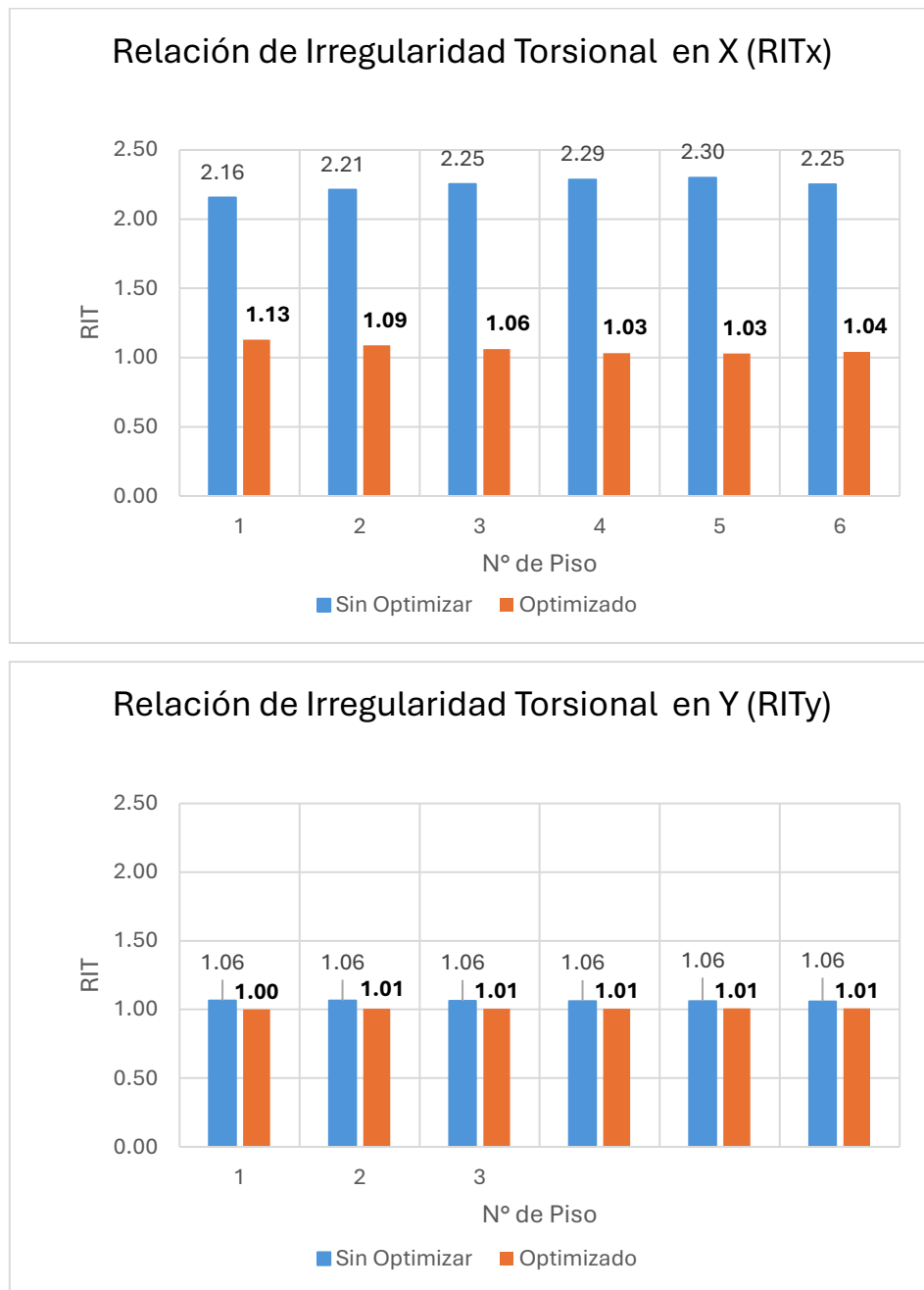
La Relación de Irregularidad Torsional (RIT) permite identificar si la respuesta de un piso presenta torsión significativa. En esta investigación se adopta el criterio normativo de que una estructura se considera irregular por torsión cuando el RIT máximo es mayor que 1.2 (Ministerio de Desarrollo Urbano y Vivienda, 2014b). Esta clasificación tiene implicaciones directas en el diseño sísmico: el espectro se penaliza mediante el factor de irregularidad torsional ( $\approx 0.9$ ), y se amplifican las fuerzas internas en los elementos al incrementarse la excentricidad accidental mediante el factor  $A_x$ , aumentando la demanda principalmente en los elementos ubicados hacia los extremos del diafragma.

En las Figuras 3.11 y 3.12 se muestra los valores de la Relación de Irregularidad Torsional en direcciones X y Y (RIT<sub>x</sub> y RIT<sub>y</sub>) para las estructuras de estudio (caso 1 y caso 2, respectivamente).



**Figura 3.11 Comparación de RIT antes y después de la optimización, caso 1.**

Con relación al caso 1, en el modelo sin optimizar, la torsión se manifiesta principalmente en X, con  $RIT_x \approx 1.73-1.80$  en los tres niveles, superando 1.2 y clasificando la estructura como irregular por torsión. En Y, los valores se mantienen cercanos a 1 ( $RIT_y \approx 1.03-1.04$ ), indicando baja torsión. Tras la optimización,  $RIT_x$  disminuye a  $\approx 1.02-1.14$ , acercándose a un comportamiento regular, mientras que  $RIT_y$  permanece  $\sim 1.00-1.01$ .



**Figura 3.12 Comparación de RIT antes y después de la optimización, caso 2.**

Para el caso 2, el modelo sin optimizar presenta  $RIT_x \approx 2.16-2.30$  en toda la altura, evidenciando irregularidad torsional marcada. Luego de la optimización,  $RIT_x$  se reduce a  $\approx 1.03-1.13$  en todos los pisos, quedando por debajo del umbral de 1.2. En Y, la RIT es cercana a 1 tanto antes como después ( $\approx 1.06 \rightarrow 1.00-1.01$ ).

En ambos casos, la optimización reduce de forma marcada la RIT en la dirección crítica (X), acercándola a valores próximos a 1. En particular, el caso 2 muestra el cambio más relevante, ya que pasa de una condición claramente irregular en toda la altura a un estado regular según el criterio  $RIT_{max} \leq 1.2$ , lo que permite evitar la penalización espectral (0.9) y la amplificación de demanda asociada al factor  $A_x$ . Esta mejora es consistente con la redistribución de rigidez lograda mediante la modificación estratégica de secciones, y se refleja en la evolución de fuerzas internas y cuantías requeridas en los elementos.

### **3.2 Análisis de Resultados**

Los resultados muestran una mejora significativa del comportamiento dinámico en ambos casos de estudio. La reducción de las excentricidades en todos los pisos se reflejó en valores de RIT menores a 1.2, límite que clasifica a la estructura como irregular o no por torsión. Como consecuencia, según la norma vigente, ya no es requisito aplicar penalizaciones asociadas a la irregularidad torsional, como la reducción del espectro elástico mediante el factor de irregularidad en planta correspondiente y la amplificación de la excentricidad accidental a través de  $A_x$ , lo que tiende a incrementar las fuerzas de diseño.

Desde el punto de vista modal, el comportamiento inicial, que se caracterizaba por modos fundamentales con participación torsional relevante (en el segundo modo), cambió hacia una respuesta traslacional en los primeros modos, con altos porcentajes de masa participativa en  $U_x$  y  $U_y$  (dirección X y Y). A su vez, la participación torsional se desplazó hacia el modo superior, lo que es favorable debido a que la respuesta sísmica suele estar dominada por los modos de menor período. Este cambio confirma que la rigidización estratégica de columnas mediante optimización no solo reduce la excentricidad estática, sino que también “modifica” los modos de vibración de las estructuras reordenando la contribución de la torsión dentro de la respuesta dinámica.

El mecanismo que explica esta mejora se asocia a la rigidización estratégica de columnas, seleccionadas por el algoritmo de optimización para acercar el centro de

rigidez C.R. al centro de masa C.M. sin requerir un incremento uniforme de rigidez en toda la estructura. Esta modificación de las rigideces reduce el giro de los diafragmas y reduce la concentración de esfuerzos en los pórticos más alejados del C.R., sino que reparte las fuerzas de corte y momento en función de la rigidez de los elementos.

En la respuesta interna, la disminución de la torsión se manifiesta como una redistribución de esfuerzos en función de la rigidez de los elementos y una reducción de concentraciones locales de cortantes y momentos en elementos de los marcos alejados del C.R. De forma coherente con este comportamiento, las cuantías totales longitudinales y transversales requeridas en columnas disminuyen moderadamente, mientras que en las vigas se observa una reducción generalizada de cuantías a flexión positiva y negativa. En conjunto, estos cambios sugieren que el control de la torsión no solo mejora el comportamiento dinámico, sino que también puede implicar diseños más eficientes al reducir demandas asociadas a efectos torsionales, cumpliendo así con el objetivo general de la presente investigación.

Finalmente, se destaca que la formulación de la función objetivo buscó evitar una rigidización excesiva: las derivas se incorporaron para controlar la respuesta sin forzar valores cercanos a cero, favoreciendo que la estructura se rigidice únicamente en la magnitud necesaria para cumplir simultáneamente con excentricidades, participación torsional y uniformidad geométrica. No obstante, los resultados dependen de los pesos adoptados, límites de diseño y supuestos de modelación, por lo que su calibración es un aspecto clave para las aplicaciones futuras del algoritmo.

### **3.2.1 Implicaciones económicas de la optimización**

Como complemento al análisis de comportamiento estructural, se realizó una comparación del costo de materiales asociado a los elementos principales de los sistemas aporticados, considerando únicamente los rubros de hormigón y acero de refuerzo en columnas y vigas. Esta evaluación se incorporó con el objetivo de identificar si la modificación de rigidez obtenida mediante el procedimiento de optimización, además

de reducir los efectos torsionales, podría implicar cambios relevantes en el consumo de materiales.

Para el Caso 1, los resultados muestran que la estructura optimizada presenta una reducción del 19% en el costo combinado de hormigón y acero de refuerzo (columnas y vigas) respecto a la estructura inicial. Este comportamiento se explica porque la optimización no incrementa de forma uniforme la rigidez global; por el contrario, aumenta selectivamente las secciones en aquellos elementos que contribuyen de forma más efectiva a disminuir las excentricidades y la respuesta torsional, mientras que en otros elementos se reduce la demanda de refuerzo y/o se evita un sobredimensionamiento innecesario.

Estos resultados sugieren que el control de la torsión mediante ajustes de rigidez no solo mejora el comportamiento dinámico, sino que también puede optimizar el uso de materiales y, por tanto, reducir costos, al menos para el caso 1.

Sin embargo, para el Caso 2 la reducción del costo combinado (hormigón + acero) fue del orden del 5%. En este caso, la optimización tendió a aumentar significativamente el volumen de hormigón en columnas, mientras que el ahorro se concentró principalmente en las vigas. Este comportamiento sugiere que, a medida que aumenta la altura, mitigar la torsión únicamente mediante la rigidización de columnas en un sistema aperturado puede perder eficiencia económica, ya que requiere incrementos importantes de rigidez para controlar la respuesta torsional. Por esta razón, resulta pertinente evaluar alternativas estructurales más eficientes para el control lateral, como sistemas duales pórtico-muro, donde los muros aportan rigidez y control de derivas y torsión con menor penalización en material.

No obstante, esta conclusión se interpreta como una tendencia observada en los modelos de estudio y bajo las restricciones adoptadas, por lo que su generalización requiere evaluar otras tipologías, alturas y configuraciones de muros.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

El trabajo de investigación desarrollado se basa en la modificación localizada de rigideces de columnas mediante la implementación de un algoritmo de optimización (PSO). Esta herramienta ha demostrado ser efectiva para mitigar la respuesta torsional en edificaciones tipo pórtico de hormigón armado con irregularidad en planta tipo 1.

### 4.1 Conclusiones

- Se concluye que la incorporación de elementos como escaleras o rampas en los modelos estructurales de análisis es determinante, ya que su presencia altera la distribución de rigidez y la ubicación del centro de rigidez, pudiendo inducir o intensificar la respuesta torsional de la edificación.
- El procedimiento de optimización basado en el ajuste estratégico de rigideces mediante la modificación localizada de secciones de columnas demostró ser efectivo para “transformar” los modos de vibración fundamentales de torsionales a traslacionales. Esta modificación redujo la respuesta torsional inducida por la irregularidad en planta mejorando el comportamiento dinámico de las edificaciones analizadas.
- La efectividad del algoritmo resultó sensible a los coeficientes de importancia asignados a cada término. El término asociado al control de excentricidades fue el de mayor importancia dentro de la función objetivo, lo que permitió concluir que la excentricidad gobierna el comportamiento torsional de las edificaciones analizadas.
- La incorporación del término de variabilidad de secciones dentro de la función objetivo fue fundamental para garantizar una transición suave entre secciones de un mismo eje. Asimismo, la inclusión del término asociado a las derivas permitió

controlar el incremento excesivo de rigidez global, asegurando que el PSO no rigidizara más de lo necesario.

- En la edificación de baja altura, la optimización permitió una reducción significativa del costo combinado de hormigón y acero. En contraste, en la edificación de mayor altura, los momentos torsores acumulados requirieron incrementos considerables de secciones de columnas tal que el ahorro de acero en vigas no compensó el volumen de hormigón adicional, resultando en un beneficio económico menor. Para estructuras de mayor altura, la optimización debería complementarse con la integración de otros sistemas, como muros de corte, que permitan controlar la torsión sin penalizar tanto el volumen de las columnas.
- A pesar de las diferencias observadas en la eficiencia económica entre ambos casos de estudio, el algoritmo resultó igualmente efectivo para desacoplar los modos de vibración y reducir el comportamiento predominante torsional, demostrando que la metodología es robusta y aplicable a edificaciones de distinta altura.

A continuación, se describen las fortalezas y debilidades del proceso de optimización:

**Fortalezas:** El PSO plantea un enfoque integral del proceso de optimización al combinar excentricidades, modos/masa participativa, variabilidad geométrica entre secciones y restricciones de deriva.

Por otra parte, la investigación abarca dos casos de estudio con distinta altura, lo que sugiere la generalidad del procedimiento de optimización.

**Debilidades:** La calidad de los resultados depende de los pesos/importancia de cada término de la función objetivo, además de los parámetros del optimizador (por ejemplo, pesos de ajuste entre partículas del enjambre, población, inercia, número de iteraciones y tolerancias), por lo que la correcta calibración del PSO es indispensable.

Otra de las limitaciones del estudio son los supuestos de modelamiento de las estructuras, como apoyo con base empotrada, diafragma rígido, secciones uniformes en toda la altura de la columna, análisis determinista, etc.

## 4.2 Recomendaciones

Con base en los resultados obtenidos, se proponen las siguientes recomendaciones para trabajos futuros para ampliar el alcance del procedimiento de optimización y aumentar su versatilidad de aplicación en la práctica profesional y en investigación:

- Generalización a otras tipologías e irregularidades combinadas

Se sugiere extender el procedimiento a tipologías con muros estructurales o sistemas duales, así como a irregularidades combinadas (torsión con otras irregularidades en planta y en altura). Esto permitiría evaluar el rango de aplicabilidad del método y establecer recomendaciones de uso para configuraciones típicas de diseño.

- Optimización multiobjetivo orientada a costos

Si bien en esta investigación se verificó que es posible mejorar el comportamiento dinámico y reducir el efecto torsional, una extensión del presente trabajo es reformular la función objetivo añadiendo restricciones de volumen de materiales asociadas a los costos de construcción.

- Incorporación de la Interacción Suelo-Estructura (ISE)

La respuesta de una edificación es sensible a la rigidez del suelo de apoyo y a la flexibilidad de la cimentación, por lo que el supuesto de base empotrada puede subestimar o sobreestimar reacciones, derivas y rotaciones en ciertos escenarios, especialmente en suelos blandos. Se recomienda extender el procedimiento para considerar la ISE mediante resortes traslacionales y rotacionales, evaluando cómo la flexibilidad basal afecta la distribución de demandas.

- Incorporación de modelos sustitutos

Para reducir el costo computacional asociado a ejecutar un análisis estructural en cada iteración del PSO, se propone entrenar modelos sustitutos (por ejemplo, redes neuronales) que aprendan la relación entre las variables de diseño y las respuestas estructurales (excentricidades, derivas, participación modal, etc.). De este modo, el optimizador consultaría al modelo sustituto durante la búsqueda global y reservaría los

análisis estructurales completos para etapas de verificación y refinamiento. Esto permitirá reducir drásticamente tiempos de optimización en estructuras más complejas.

- Desarrollo de una herramienta automatizada e integración como extensión con software comercial

Se recomienda implementar el optimizador como una herramienta automatizada de MATLAB conectada a un software de análisis estructural mediante su API (por ejemplo, SAP2000 o ETABS). El complemento podría leer el modelo, modificar secciones según las variables del optimizador, ejecutar análisis, extraer automáticamente respuestas (RIT, derivas, participación modal, fuerzas internas) y reportar la solución y sus verificaciones normativas. Esta contribución convertiría el procedimiento en un recurso fácilmente aplicable en el campo profesional, disminuyendo la dependencia de procesos manuales.

# BIBLIOGRAFÍA

- Arnold, C., & Reitherman, R. (1982). *Building Configuration and Seismic Design The Architecture of Earthquake Resistance " INFORMATION RESOURCES NATIONAL SCIENCE FOUNDATION.*  
<https://books.google.com.ec/books?id=G9FRAAAAMAAJ>
- Asamblea General ONU. (2015). *Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible.*
- Awad, R. Rochel. (2012). *Análisis y diseño sísmico de edificios.* Universidad EAFIT.
- Balaji Rao, A., Reddy, P. S., & Meghana Shaly, Ch. (2022). Mode shape modification of irregular design of buildings. *Materials Today: Proceedings*, 62, 1790–1795.  
<https://doi.org/10.1016/j.matpr.2021.12.374>
- Ballesteros-Salazar, K. S., Caizaguano-Montero, D. G., Haro-Báez, A. G., & Toulkeridis, T. (2022). Case Study of the Application of an Innovative Guide for the Seismic Vulnerability Evaluation of Schools Located in Sangolquí, Interandean Valley in Ecuador. *Buildings*, 12(9), 1471. <https://doi.org/10.3390/buildings12091471>
- Beauval, C., Marinière, J., Yepes, H., Audin, L., Nocquet, J. -M., Alvarado, A., Baize, S., Aguilar, J., Singaicho, J. -C., & Jomard, H. (2018). A New Seismic Hazard Model for Ecuador. *Bulletin of the Seismological Society of America*, 108(3A), 1443–1464.  
<https://doi.org/10.1785/0120170259>
- Cimellaro, G. Paolo., & Marasco, Sebastiano. (2018). *Introduction to dynamics of structures and earthquake engineering.* Springer.
- De la Llera, J. C., & Chopra, A. K. (1994). Accidental torsion in buildings due to base rotational excitation. *Earthquake Engineering & Structural Dynamics*, 23(9), 1003–1021. <https://doi.org/10.1002/eqe.4290230906>
- De-la-Colina, J., & Almeida, C. (2004). Probabilistic Study on Accidental Torsion of Low-Rise Buildings. *Earthquake Spectra*, 20(1), 25–41.  
<https://doi.org/10.1193/1.1646391>
- Garcia Reyes. (1998). *Dinámica Estructural Aplicada al Diseño Sísmico.* Universidad de los Andes.

- Gokdemir, H., Ozbasaran, H., Dogan, M., Unluoglu, E., & Albayrak, U. (2013). Effects of torsional irregularity to structures during earthquakes. *Engineering Failure Analysis*, 35, 713–717. <https://doi.org/10.1016/j.engfailanal.2013.06.028>
- Guevara, L., Alonso, J., & Fortoul, E. (1992). Floor-plan Shape Influence on the Response of Earthquakes. *Earthquake Engineering. 10th World Conference*.
- Kennedy, J., & Eberhart, R. (s/f). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Makarios, T. (2008). Practical calculation of the torsional stiffness radius of multistorey tall buildings. *The Structural Design of Tall and Special Buildings*, 17(1), 39–65. <https://doi.org/10.1002/tal.316>
- MathWorks*. (s/f).
- Ministerio de Desarrollo Urbano y Vivienda. (2014a). *Norma Ecuatoriana de la Construcción. Peligro Sísmico. Diseño Sismo Resistente*. Dirección de Comunicación Social, Miduvi.
- Ministerio de Desarrollo Urbano y Vivienda. (2014b). *Norma Ecuatoriana de la Construcción. Peligro Sísmico. Diseño Sismo Resistente*. Dirección de Comunicación Social, Miduvi.
- Naveen E, S., Abraham, N. M., & S D, A. K. (2019). Analysis of Irregular Structures under Earthquake Loads. *Procedia Structural Integrity*, 14, 806–819. <https://doi.org/10.1016/j.prostr.2019.07.059>
- Özmen, G., Girgin, K., & Durgun, Y. (2014). Torsional irregularity in multi-story structures. *International Journal of Advanced Structural Engineering*, 6(4), 121–131. <https://doi.org/10.1007/s40091-014-0070-5>
- Paulay, T., & Priestly, M. J. N. (1992). *Seismic Design of Reinforced Concrete and Masonry Buildings*. Wiley. <https://doi.org/10.1002/9780470172841>
- Pujar, P., & Amaresh. (2017). SEISMIC ANALYSIS OF PLAN IRREGULAR MULTI-STORIED BUILDING WITH AND WITHOUT SHEAR WALLS. *International Research Journal of Engineering and Technology (IRJET)*, 04(08).
- Sim, C., Villalobos, E., Smith, J. P., Rojas, P., Pujol, S., Puranam, A. Y., & Laughery, L. A. (2017). Performance of Low-rise Reinforced Concrete Buildings in the 2016 Ecuador Earthquake. *Purdue Publications*.

- Stathopoulos, K. G., & Anagnostopoulos, S. A. (2005). Inelastic torsion of multistorey buildings under earthquake excitations. *Earthquake Engineering & Structural Dynamics*, 34(12), 1449–1465. <https://doi.org/10.1002/eqe.486>
- Valmundsson, E. V., & Nau, J. M. (1997). Seismic Response of Building Frames with Vertical Structural Irregularities. *Journal of Structural Engineering*, 123(1), 30–41. [https://doi.org/10.1061/\(ASCE\)0733-9445\(1997\)123:1\(30\)](https://doi.org/10.1061/(ASCE)0733-9445(1997)123:1(30))
- Villalobos, E., Sim, C., Smith-Pardo, J. P., Rojas, P., Pujol, S., & Kreger, M. E. (2018). The 16 April 2016 Ecuador Earthquake Damage Assessment Survey. *Earthquake Spectra*, 34(3), 1201–1217. <https://doi.org/10.1193/060217EQS106M>

# ANEXOS

```
% Optimización por Enjambre de Partículas PSO
% Programado por Maria Gracia Villao
% Escuela Superior Politécnica del Litoral ESPOL
% April 2025

clc, clear all, close all
tic
global tt y_f fm mp dt cont AA BB CC DD IC a2 b2 c2 d2 v0 cp kp lb ub opti vidObj1
vidObj2 fig1 fig2
cont = 1;

% Parameter initial guess

bc1A = 0.45;      % Initial step [m]
bc2A = 0.45;      % Initial step [m]
bc3A = 0.45;      % Initial step [m]
bc4A = 0.45;      % Initial step [m]
bc5A = 0.45;      % Initial step [m]
bc1B = 0.45;      % Initial step [m]
bc2B = 0.45;      % Initial step [m]
bc3B = 0.45;      % Initial step [m]
bc4B = 0.45;      % Initial step [m]
bc5B = 0.45;      % Initial step [m]
bc1C = 0.45;      % Initial step [m]
bc2C = 0.45;      % Initial step [m]
bc3C = 0.45;      % Initial step [m]
bc4C = 0.45;      % Initial step [m]
bc5C = 0.45;      % Initial step [m]
bc1D = 0.45;      % Initial step [m]
bc2D = 0.45;      % Initial step [m]
bc3D = 0.45;      % Initial step [m]
bc4D = 0.45;      % Initial step [m]
bc5D = 0.45;      % Initial step [m]
bc1E = 0.45;      % Initial step [m]
bc2E = 0.45;      % Initial step [m]
bc3E = 0.45;      % Initial step [m]
bc4E = 0.45;      % Initial step [m]
bc5E = 0.45;      % Initial step [m]

hc1A = 0.45;      % Initial step [m]
hc2A = 0.45;      % Initial step [m]
hc3A = 0.45;      % Initial step [m]
hc4A = 0.45;      % Initial step [m]
hc5A = 0.45;      % Initial step [m]
hc1B = 0.45;      % Initial step [m]
hc2B = 0.45;      % Initial step [m]
hc3B = 0.45;      % Initial step [m]
hc4B = 0.45;      % Initial step [m]
hc5B = 0.45;      % Initial step [m]
hc1C = 0.45;      % Initial step [m]
hc2C = 0.45;      % Initial step [m]
hc3C = 0.45;      % Initial step [m]
hc4C = 0.45;      % Initial step [m]
```



```

beq = [];
nlcon = [];

%% GENETIC ALGORITHM
% options = optimoptions(@ga,...
%   'Display','iter',...
%   'PlotFcn',{@gaplotbestf,@gaplotgenealogy},...
%   'FitnessLimit',.32);
%   'PopulationSize',50,...
%   'SelectionFcn',@selectionstochunif,...
%   'InitialPopulationRange',[lb;ub],...
%   'CrossoverFraction',.8,...
%   'MutationFcn',{@mutationadaptfeasible},...
%   'EliteCount',ceil(0.05*50));
% [p, Fval,output] = ga(@cost_fcn,length(p0),A,b,Aeq,beq,lb,ub,nlcon,options);
%,options);

%% PARTICLE SWARM

% CONFIG PSO – rápido y estable
rng(1,'twister'); % Reproducible

% (1) Opciones del refinamiento local (polish) con fmincon
hybOpts = optimoptions('fmincon', ...
    'Algorithm','interior-point', ...
    'Display','iter', ...
    'MaxIterations',10, ...
    'OptimalityTolerance',1e-6, ...
    'StepTolerance',1e-10);
% (2) Opciones del PSO
options = optimoptions('particleswarm', ...
    'OutputFcn',@log_plot,...
    'Display','iter', ... % 'none' si quieres silencioso
    'SwarmSize', 30, ... % 30-60 suele ir bien (<30 vars)
    'MaxIterations', 5000, ...
    'MaxStallIterations', 90, ... % detén antes si no mejora
    'FunctionTolerance', 1e-4, ... % tolerancia más laxa = más rápido
    'ObjectiveLimit', -Inf, ... % pon un objetivo si sabes un umbral
    'InertiaRange', [0.2 0.7], ... % inercia lineal baja→alta
    'SelfAdjustmentWeight',1.50,...
    'SocialAdjustmentWeight',2.00,...
    'HybridFcn', {@fmincon, hybOpts}); % "asienta" la mejor solución
nVars = numel(lb);
% (3) Llamada al PSO
[p, Fval,output] = particleswarm(@cost_fcn_ejemplo,nVars,lb,ub,options); %,options);

% options = optimoptions(@particleswarm,...
%   'Display','iter',...
%   'PlotFcn',{@pswplotbestf},...
%   'InertiaRange',[0.3 0.9],...
%   'SelfAdjustmentWeight',1.50,...
%   'SocialAdjustmentWeight',2.00,...

```

```

% 'MaxIterations',300,...
% 'MaxStallIterations', 300, ...
% 'SwarmSize',30,...
% 'FunctionTolerance',1e-8,...
% 'ObjectiveLimit',-Inf);
%
% [p, Fval,output] = particleswarm(@cost_fcn_ejemplo,length(p0),lb,ub,options);
%,options);
% 'UseParallel', true, ...           % evalúa partículas en paralelo

p = 0.05 * round(p/0.05);

%% show final objective
disp(['Final SSE: ' num2str(cost_fcn_ejemplo(p))])

% Optimized parameter values

bc1A = p(1);
bc2A = p(2);
bc3A = p(3);
bc4A = p(4);
bc5A = p(5);
bc1B = p(6);
bc2B = p(7);
bc3B = p(8);
bc4B = p(9);
bc5B = p(10);
bc1C = p(11);
bc2C = p(12);
bc3C = p(13);
bc4C = p(14);
bc5C = p(15);
bc1D = p(16);
bc2D = p(17);
bc3D = p(18);
bc4D = p(19);
bc5D = p(20);
bc1E = p(21);
bc2E = p(22);
bc3E = p(23);
bc4E = p(24);
bc5E = p(25);

hc1A = p(26);
hc2A = p(27);
hc3A = p(28);
hc4A = p(29);
hc5A = p(30);
hc1B = p(31);
hc2B = p(32);
hc3B = p(33);
hc4B = p(34);
hc5B = p(35);
hc1C = p(36);

```

```

hc2C = p(37);
hc3C = p(38);
hc4C = p(39);
hc5C = p(40);
hc1D = p(41);
hc2D = p(42);
hc3D = p(43);
hc4D = p(44);
hc5D = p(45);
hc1E = p(46);
hc2E = p(47);
hc3E = p(48);
hc4E = p(49);
hc5E = p(50);

```

```

% bc1A = p(1);
% bc2A = p(2);
% bc3A = p(3);
% bc4A = p(4);
% bc5A = p(5);
% hc1A = p(6);
% hc2A = p(7);
% hc3A = p(8);
% hc4A = p(9);
% hc5A = p(10);

```

```

disp(['DimX Columna Eje 1A: ' num2str(bc1A)])
disp(['DimX Columna Eje 2A: ' num2str(bc2A)])
disp(['DimX Columna Eje 3A: ' num2str(bc3A)])
disp(['DimX Columna Eje 4A: ' num2str(bc4A)])
disp(['DimX Columna Eje 5A: ' num2str(bc5A)])
disp(['DimX Columna Eje 1B: ' num2str(bc1B)])
disp(['DimX Columna Eje 2B: ' num2str(bc2B)])
disp(['DimX Columna Eje 3B: ' num2str(bc3B)])
disp(['DimX Columna Eje 4B: ' num2str(bc4B)])
disp(['DimX Columna Eje 5B: ' num2str(bc5B)])
disp(['DimX Columna Eje 1C: ' num2str(bc1C)])
disp(['DimX Columna Eje 2C: ' num2str(bc2C)])
disp(['DimX Columna Eje 3C: ' num2str(bc3C)])
disp(['DimX Columna Eje 4C: ' num2str(bc4C)])
disp(['DimX Columna Eje 5C: ' num2str(bc5C)])
disp(['DimX Columna Eje 1D: ' num2str(bc1D)])
disp(['DimX Columna Eje 2D: ' num2str(bc2D)])
disp(['DimX Columna Eje 3D: ' num2str(bc3D)])
disp(['DimX Columna Eje 4D: ' num2str(bc4D)])
disp(['DimX Columna Eje 5D: ' num2str(bc5D)])
disp(['DimX Columna Eje 1E: ' num2str(bc1E)])
disp(['DimX Columna Eje 2E: ' num2str(bc2E)])
disp(['DimX Columna Eje 3E: ' num2str(bc3E)])
disp(['DimX Columna Eje 4E: ' num2str(bc4E)])
disp(['DimX Columna Eje 5E: ' num2str(bc5E)])
disp(['_____'])
disp(['DimY Columna Eje 1A: ' num2str(hc1A)])
disp(['DimY Columna Eje 2A: ' num2str(hc2A)])
disp(['DimY Columna Eje 3A: ' num2str(hc3A)])
disp(['DimY Columna Eje 4A: ' num2str(hc4A)])

```

```

disp(['DimY Columna Eje 5A: ' num2str(hc5A)])
disp(['DimY Columna Eje 1B: ' num2str(hc1B)])
disp(['DimY Columna Eje 2B: ' num2str(hc2B)])
disp(['DimY Columna Eje 3B: ' num2str(hc3B)])
disp(['DimY Columna Eje 4B: ' num2str(hc4B)])
disp(['DimY Columna Eje 5B: ' num2str(hc5B)])
disp(['DimY Columna Eje 1C: ' num2str(hc1C)])
disp(['DimY Columna Eje 2C: ' num2str(hc2C)])
disp(['DimY Columna Eje 3C: ' num2str(hc3C)])
disp(['DimY Columna Eje 4C: ' num2str(hc4C)])
disp(['DimY Columna Eje 5C: ' num2str(hc5C)])
disp(['DimY Columna Eje 1D: ' num2str(hc1D)])
disp(['DimY Columna Eje 2D: ' num2str(hc2D)])
disp(['DimY Columna Eje 3D: ' num2str(hc3D)])
disp(['DimY Columna Eje 4D: ' num2str(hc4D)])
disp(['DimY Columna Eje 5D: ' num2str(hc5D)])
disp(['DimY Columna Eje 1E: ' num2str(hc1E)])
disp(['DimY Columna Eje 2E: ' num2str(hc2E)])
disp(['DimY Columna Eje 3E: ' num2str(hc3E)])
disp(['DimY Columna Eje 4E: ' num2str(hc4E)])
disp(['DimY Columna Eje 5E: ' num2str(hc5E)])

toc

% close(fig1);
%
% close(fig2);

close(vidObj1);
%
% close(vidObj2);

load train
sound(y,2*Fs)

fig1=figure('Name','Dimensiones optimización','NumberTitle','off');
fig1.Position = [20 50 1500 700];

%clf(1)

subplot(5,5,1)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc1A)],opti.bc1A,'.')
axis([0 cont lb(1) ub(1)])
ylabel('DimX Col 1A, [m]')

subplot(5,5,2)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc2A)],opti.bc2A,'.')
axis([0 cont lb(2) ub(2)])
ylabel('DimX Col 2A, [m]')

subplot(5,5,3)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on

```

```

plot([1:length(opti.bc3A)],opti.bc3A, '.')
axis([0 cont lb(3) ub(3)])
ylabel('DimX Col 3A, [m]')

subplot(5,5,4)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc4A)],opti.bc4A, '.')
axis([0 cont lb(4) ub(4)])
ylabel('DimX Col 4A, [m]')

subplot(5,5,5)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc5A)],opti.bc5A, '.')
axis([0 cont lb(5) ub(5)])
ylabel('DimX Col 5A, [m]')

subplot(5,5,6)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc1B)],opti.bc1B, '.')
axis([0 cont lb(6) ub(6)])
ylabel('DimX Col 1B, [m]')

subplot(5,5,7)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc2B)],opti.bc2B, '.')
axis([0 cont lb(7) ub(7)])
ylabel('DimX Col 2B, [m]')

subplot(5,5,8)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc3B)],opti.bc3B, '.')
axis([0 cont lb(8) ub(8)])
ylabel('DimX Col 3B, [m]')

subplot(5,5,9)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc4B)],opti.bc4B, '.')
axis([0 cont lb(9) ub(9)])
ylabel('DimX Col 4B, [m]')

subplot(5,5,10)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc5B)],opti.bc5B, '.')
axis([0 cont lb(10) ub(10)])
ylabel('DimX Col 5B, [m]')

subplot(5,5,11)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc1C)],opti.bc1C, '.')
axis([0 cont lb(11) ub(11)])
ylabel('DimX Col 1C, [m]')

subplot(5,5,12)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc2C)],opti.bc2C, '.')

```

```

axis([0 cont lb(12) ub(12)])
ylabel('DimX Col 2C, [m]')

subplot(5,5,13)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc3C)],opti.bc3C,'.')
axis([0 cont lb(13) ub(13)])
ylabel('DimX Col 3C, [m]')

subplot(5,5,14)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc4C)],opti.bc4C,'.')
axis([0 cont lb(14) ub(14)])
ylabel('DimX Col 4C, [m]')

subplot(5,5,15)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc5C)],opti.bc5C,'.')
axis([0 cont lb(15) ub(15)])
ylabel('DimX Col 5C, [m]')

subplot(5,5,16)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc1D)],opti.bc1D,'.')
axis([0 cont lb(16) ub(16)])
ylabel('DimX Col 1D, [m]')

subplot(5,5,17)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc2D)],opti.bc2D,'.')
axis([0 cont lb(17) ub(17)])
ylabel('DimX Col 2D, [m]')

subplot(5,5,18)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc3D)],opti.bc3D,'.')
axis([0 cont lb(18) ub(18)])
ylabel('DimX Col 3D, [m]')

subplot(5,5,19)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc4D)],opti.bc4D,'.')
axis([0 cont lb(19) ub(19)])
ylabel('DimX Col 4D, [m]')

subplot(5,5,20)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc5D)],opti.bc5D,'.')
axis([0 cont lb(20) ub(20)])
ylabel('DimX Col 5D, [m]')

subplot(5,5,21)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc1E)],opti.bc1E,'.')
axis([0 cont lb(21) ub(21)])
ylabel('DimX Col 1E, [m]')

```

```

subplot(5,5,22)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc2E)],opti.bc2E,'.')
axis([0 cont lb(22) ub(22)])
ylabel('DimX Col 2E, [m]')

subplot(5,5,23)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc3E)],opti.bc3E,'.')
axis([0 cont lb(23) ub(23)])
ylabel('DimX Col 3E, [m]')

subplot(5,5,24)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc4E)],opti.bc4E,'.')
axis([0 cont lb(24) ub(24)])
ylabel('DimX Col 4E, [m]')

subplot(5,5,25)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.bc5E)],opti.bc5E,'.')
axis([0 cont lb(25) ub(25)])
ylabel('DimX Col 5E, [m]')

exportgraphics(fig1, 'grafico_optimizacionX_prueba5.pdf', 'ContentType','vector',
'Resolution',400);

%% =====

fig3=figure('Name','Dimensiones optimización en Y','NumberTitle','off');
fig3.Position = [20 50 1500 700];

%clf(1)

subplot(5,5,1)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc1A)],opti.hc1A,'.')
axis([0 cont lb(26) ub(26)])
ylabel('DimY Col 1A, [m]')

subplot(5,5,2)
%plot([0 cont],0.30*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc2A)],opti.hc2A,'.')
axis([0 cont lb(27) ub(27)])
ylabel('DimY Col 2A, [m]')

subplot(5,5,3)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc3A)],opti.hc3A,'.')
axis([0 cont lb(28) ub(28)])
ylabel('DimY Col 3A, [m]')

subplot(5,5,4)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc4A)],opti.hc4A,'.')

```

```

axis([0 cont lb(29) ub(29)])
ylabel('DimY Col 4A, [m]')

subplot(5,5,5)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc5A)],opti.hc5A,'.')
axis([0 cont lb(30) ub(30)])
ylabel('DimY Col 5A, [m]')

subplot(5,5,6)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc1B)],opti.hc1B,'.')
axis([0 cont lb(31) ub(31)])
ylabel('DimY Col 1B, [m]')

subplot(5,5,7)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc2B)],opti.hc2B,'.')
axis([0 cont lb(32) ub(32)])
ylabel('DimY Col 2B, [m]')

subplot(5,5,8)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc3B)],opti.hc3B,'.')
axis([0 cont lb(33) ub(33)])
ylabel('DimX Col 3B, [m]')

subplot(5,5,9)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc4B)],opti.hc4B,'.')
axis([0 cont lb(34) ub(34)])
ylabel('DimY Col 4B, [m]')

subplot(5,5,10)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc5B)],opti.hc5B,'.')
axis([0 cont lb(35) ub(35)])
ylabel('DimY Col 5B, [m]')

subplot(5,5,11)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc1C)],opti.hc1C,'.')
axis([0 cont lb(36) ub(36)])
ylabel('DimY Col 1C, [m]')

subplot(5,5,12)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc2C)],opti.hc2C,'.')
axis([0 cont lb(37) ub(37)])
ylabel('DimY Col 2C, [m]')

subplot(5,5,13)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc3C)],opti.hc3C,'.')
axis([0 cont lb(38) ub(38)])

```

```

ylabel('DimY Col 3C, [m]')

subplot(5,5,14)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc4C)],opti.hc4C,'.')
axis([0 cont lb(39) ub(39)])
ylabel('DimY Col 4C, [m]')

subplot(5,5,15)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc5C)],opti.hc5C,'.')
axis([0 cont lb(40) ub(40)])
ylabel('DimY Col 5C, [m]')

subplot(5,5,16)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc1D)],opti.hc1D,'.')
axis([0 cont lb(41) ub(41)])
ylabel('DimY Col 1D, [m]')

subplot(5,5,17)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc2D)],opti.hc2D,'.')
axis([0 cont lb(42) ub(42)])
ylabel('DimY Col 2D, [m]')

subplot(5,5,18)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc3D)],opti.hc3D,'.')
axis([0 cont lb(43) ub(43)])
ylabel('DimY Col 3D, [m]')

subplot(5,5,19)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc4D)],opti.hc4D,'.')
axis([0 cont lb(44) ub(44)])
ylabel('DimY Col 4D, [m]')

subplot(5,5,20)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc5D)],opti.hc5D,'.')
axis([0 cont lb(45) ub(45)])
ylabel('DimY Col 5D, [m]')

subplot(5,5,21)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc1E)],opti.hc1E,'.')
axis([0 cont lb(46) ub(46)])
ylabel('DimY Col 1E, [m]')

subplot(5,5,22)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc2E)],opti.hc2E,'.')
axis([0 cont lb(47) ub(47)])
ylabel('DimY Col 2E, [m]')

```

```

subplot(5,5,23)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc3E)],opti.hc3E,'.')
axis([0 cont lb(48) ub(48)])
ylabel('DimY Col 3E, [m]')

subplot(5,5,24)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc4E)],opti.hc4E,'.')
axis([0 cont lb(49) ub(49)])
ylabel('DimY Col 4E, [m]')

subplot(5,5,25)
%plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
plot([1:length(opti.hc5E)],opti.hc5E,'.')
axis([0 cont lb(50) ub(50)])
ylabel('DimY Col 5E, [m]')

exportgraphics(fig3, 'grafico_optimizacionY_prueba5.pdf', 'ContentType','vector',
'Resolution',400);

%% ===== Funciones =====
function stop = log_plot(optimValues, state)
    persistent history fig2 ax2
    stop = false;

    switch state
        case 'init'
            history = [];
            fig2 = figure('Name','Costo PSO', ...
                'NumberTitle','off', ...
                'HandleVisibility','on');
            ax2 = axes(fig2);

        case 'iter'
            history(end+1) = optimValues.bestfval;
            if mod(numel(history),20)==0
                plot(ax2, history,'LineWidth',1.5);
                grid(ax2,'on');
                xlabel(ax2,'Iteración');
                ylabel(ax2,'Costo');
                title(ax2,'Evolución del mejor valor de la función costo');
                drawnow limitrate
            end

        case 'done'
            % último dibujo (sin clf)
            plot(ax2, history,'LineWidth',1.5);
            grid(ax2,'on');
            title(ax2,'Evolución de la función objetivo (costo)');

            % exporta aquí
            exportgraphics(fig2,'grafico_costo_prueba5.pdf', ...
                'ContentType','vector','Resolution',400);

            assignin('base','history',history);

```

end  
end

```

% Función Objetivo
% Programado por Maria Gracia Villao
% Escuela Superior Politécnica del Litoral ESPOL
% April 2025
function obj = cost_fcn_ejemplo(p)

p5 = p;
obj = mifuncionobjoriginal(p);

end

function obj = mifuncionobjoriginal(p)

    global y_f tt cont lb ub opti vidObj1 vidObj2 fig1 fig2

    % simulate model
    [IITx, IITy, Ex, Ey, driftX_max, driftY_max, PMtorsional] =
OptimizacionColumnasEdificio1SoloDiagonal(p);

    %Penalización de dispersion de dimensiones de columnas
    CVmax = 0.15;    % objetivo permitido

% Normalizaciones de referencia (normas / límites)
    Ey_lim = 2.8; Ex_lim = 2.2;
    drift_lim = 2.0;    % 2.0%
    PMT_ref = 0.1;    % objetivo
    CV_ref = 0.15;    % CV máximo permitido

    Pcv1 = penalty_CV_group(p, [1 2 3 4 5], CVmax);    % grupo 1
    Pcv2 = penalty_CV_group(p, [6 7 8 9 10], CVmax);    % grupo 2
    Pcv3 = penalty_CV_group(p, [11 12 13 14 15], CVmax);    % grupo 3
    Pcv4 = penalty_CV_group(p, [16 17 18 19 20], CVmax);    % grupo 4
    Pcv5 = penalty_CV_group(p, [21 22 23 24 25], CVmax);    % grupo 5
    Pcv6 = penalty_CV_group(p, [26 27 28 29 30], CVmax);    % grupo 6
    Pcv7 = penalty_CV_group(p, [31 32 33 34 35], CVmax);    % grupo 7
    Pcv8 = penalty_CV_group(p, [36 37 38 39 40], CVmax);    % grupo 8
    Pcv9 = penalty_CV_group(p, [41 42 43 44 45], CVmax);    % grupo 9
    Pcv10 = penalty_CV_group(p, [46 47 48 49 50], CVmax);    % grupo 10

    Pcv_total = Pcv1 + Pcv2 + Pcv3 + Pcv4 + Pcv5 + Pcv6 + Pcv7 + Pcv8 + Pcv9 + Pcv10;

% Pesos de términos de penalización
    w_RIT = 0;
    w_Ecc = 0.55;
    w_drift = 0.25;
    w_PM = 0.10;
    w_CV = 0.10;

% Objetivo "Hinge"
    pen_Ey = max(0, abs(Ey)./Ey_lim - 1).^2;
    pen_Ex = max(0, abs(Ex)./Ex_lim - 1).^2;

```

```

pen_IITx = max(0, max(IITx) - 1).^2;
pen_IITy = max(0, max(IITy) - 1).^2;

pen_PM = (PMTorsional./PMT_ref).^2;

pen_driftX = (driftX_max./drift_lim - 1).^2;
pen_driftY = (driftY_max./drift_lim - 1).^2;

pen_CV = (Pcv_total./CV_ref).^2;

obj = sum(pen_Ey.*w_Ecc + pen_Ex.*w_Ecc + pen_IITx.*w_RIT + pen_IITy.*w_RIT +
pen_PM.*w_PM + pen_driftX.*w_drift + pen_driftY.*w_drift + pen_CV.*w_CV);

opti.bc1A(cont) = p(1);
opti.bc2A(cont) = p(2);
opti.bc3A(cont) = p(3);
opti.bc4A(cont) = p(4);
opti.bc5A(cont) = p(5);
opti.bc1B(cont) = p(6);
opti.bc2B(cont) = p(7);
opti.bc3B(cont) = p(8);
opti.bc4B(cont) = p(9);
opti.bc5B(cont) = p(10);
opti.bc1C(cont) = p(11);
opti.bc2C(cont) = p(12);
opti.bc3C(cont) = p(13);
opti.bc4C(cont) = p(14);
opti.bc5C(cont) = p(15);
opti.bc1D(cont) = p(16);
opti.bc2D(cont) = p(17);
opti.bc3D(cont) = p(18);
opti.bc4D(cont) = p(19);
opti.bc5D(cont) = p(20);
opti.bc1E(cont) = p(21);
opti.bc2E(cont) = p(22);
opti.bc3E(cont) = p(23);
opti.bc4E(cont) = p(24);
opti.bc5E(cont) = p(25);

opti.hc1A(cont) = p(26);
opti.hc2A(cont) = p(27);
opti.hc3A(cont) = p(28);
opti.hc4A(cont) = p(29);
opti.hc5A(cont) = p(30);
opti.hc1B(cont) = p(31);
opti.hc2B(cont) = p(32);
opti.hc3B(cont) = p(33);
opti.hc4B(cont) = p(34);
opti.hc5B(cont) = p(35);
opti.hc1C(cont) = p(36);
opti.hc2C(cont) = p(37);
opti.hc3C(cont) = p(38);
opti.hc4C(cont) = p(39);
opti.hc5C(cont) = p(40);
opti.hc1D(cont) = p(41);

```

```

opti.hc2D(cont) = p(42);
opti.hc3D(cont) = p(43);
opti.hc4D(cont) = p(44);
opti.hc5D(cont) = p(45);
opti.hc1E(cont) = p(46);
opti.hc2E(cont) = p(47);
opti.hc3E(cont) = p(48);
opti.hc4E(cont) = p(49);
opti.hc5E(cont) = p(50);

% figure(1)
% fig1=figure(1);
% fig1.Position = [20 50 800 400];
% clf(1)

% subplot(2,3,1)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1A,'.')
% axis([0 cont lb(1) ub(1)])
% ylabel('DimX Col 1A, [m]')

% subplot(2,3,2)
% plot([0 cont],0.30*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2A,'.')
% axis([0 cont lb(2) ub(2)])
% ylabel('DimX Col 2A, [m]')
%
% subplot(2,3,3)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3A,'.')
% axis([0 cont lb(3) ub(3)])
% ylabel('DimX Col 3A, [m]')
%
% subplot(2,3,4)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4A,'.')
% axis([0 cont lb(4) ub(4)])
% ylabel('DimX Col 4A, [m]')
%
% subplot(2,3,5)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5A,'.')
% axis([0 cont lb(5) ub(5)])
% ylabel('DimX Col 5A, [m]')

% subplot(5,5,1)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1A,'.')
% axis([0 cont lb(1) ub(1)])
% ylabel('DimX Col 1A, [m]')

% subplot(5,5,2)
% plot([0 cont],0.30*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2A,'.')
% axis([0 cont lb(2) ub(2)])

```

```

% ylabel('DimX Col 2A, [m]')
%
% subplot(1,1,1)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3A,'.')
% axis([0 cont lb(3) ub(3)])
% ylabel('Ancho Col 3A, [m]')

% subplot(5,5,4)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4A,'.')
% axis([0 cont lb(4) ub(4)])
% ylabel('DimX Col 4A, [m]')
%
% subplot(5,5,5)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5A,'.')
% axis([0 cont lb(5) ub(5)])
% ylabel('DimX Col 5A, [m]')
%
%
% subplot(5,5,6)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1B,'.')
% axis([0 cont lb(6) ub(6)])
% ylabel('DimX Col 1B, [m]')
%
% subplot(5,5,7)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2B,'.')
% axis([0 cont lb(7) ub(7)])
% ylabel('DimX Col 2B, [m]')
%
% subplot(5,5,8)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3B,'.')
% axis([0 cont lb(8) ub(8)])
% ylabel('DimX Col 3B, [m]')
%
% subplot(5,5,9)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4B,'.')
% axis([0 cont lb(9) ub(9)])
% ylabel('DimX Col 4B, [m]')
%
% subplot(5,5,10)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5B,'.')
% axis([0 cont lb(10) ub(10)])
% ylabel('DimX Col 5B, [m]')
%
%
% subplot(5,5,11)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1C,'.')
% axis([0 cont lb(11) ub(11)])
% ylabel('DimX Col 1C, [m]')

```

```

%
% subplot(5,5,12)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2C,'.')
% axis([0 cont lb(12) ub(12)])
% ylabel('DimX Col 2C, [m]')
%
% subplot(5,5,13)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3C,'.')
% axis([0 cont lb(13) ub(13)])
% ylabel('DimX Col 3C, [m]')
%
% subplot(5,5,14)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4C,'.')
% axis([0 cont lb(14) ub(14)])
% ylabel('DimX Col 4C, [m]')
%
% subplot(5,5,15)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5C,'.')
% axis([0 cont lb(15) ub(15)])
% ylabel('DimX Col 5C, [m]')
%
%
% subplot(5,5,16)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1D,'.')
% axis([0 cont lb(16) ub(16)])
% ylabel('DimX Col 1D, [m]')
%
% subplot(5,5,17)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2D,'.')
% axis([0 cont lb(17) ub(17)])
% ylabel('DimX Col 2D, [m]')
%
% subplot(5,5,18)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3D,'.')
% axis([0 cont lb(18) ub(18)])
% ylabel('DimX Col 3D, [m]')
%
% subplot(5,5,19)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4D,'.')
% axis([0 cont lb(19) ub(19)])
% ylabel('DimX Col 4D, [m]')
%
% subplot(5,5,20)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5D,'.')
% axis([0 cont lb(20) ub(20)])
% ylabel('DimX Col 5D, [m]')
%
% subplot(5,5,21)

```

```

% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc1E,'.')
% axis([0 cont lb(21) ub(21)])
% ylabel('DimX Col 1E, [m]')
%
% subplot(5,5,22)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc2E,'.')
% axis([0 cont lb(22) ub(22)])
% ylabel('DimX Col 2E, [m]')
%
% subplot(5,5,23)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc3E,'.')
% axis([0 cont lb(23) ub(23)])
% ylabel('DimX Col 3E, [m]')
%
% subplot(5,5,24)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc4E,'.')
% axis([0 cont lb(24) ub(24)])
% ylabel('DimX Col 4E, [m]')
%
% subplot(5,5,25)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.bc5E,'.')
% axis([0 cont lb(25) ub(25)])
% ylabel('DimX Col 5E, [m]')

% frame1 = getframe(fig1);           %Captura el frame de la figura
% writeVideo(vidObj1, frame1);      %Escribe el frame en el video
% % close(gcf);                       %Cierra la figura si deseas

% cont = cont +1;

%%%%%%%%%%

% figure(2)
% fig2=figure(2);
% fig2.Position = [20 40 1500 700];
% clf(2)
%
% subplot(2,3,1)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1A,'.')
% axis([0 cont lb(6) ub(6)])
% ylabel('DimY Col 1A, [m]')
%
% subplot(2,3,2)
% plot([0 cont],0.30*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2A,'.')
% axis([0 cont lb(7) ub(7)])
% ylabel('DimY Col 2A, [m]')

```

```

%
% subplot(2,3,3)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3A,'.')
% axis([0 cont lb(8) ub(8)])
% ylabel('DimY Col 3A, [m]')
%
% subplot(2,3,4)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4A,'.')
% axis([0 cont lb(9) ub(9)])
% ylabel('DimY Col 4A, [m]')
%
% subplot(2,3,5)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5A,'.')
% axis([0 cont lb(10) ub(10)])
% ylabel('DimY Col 5A, [m]')

% subplot(5,5,2)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1A,'.')
% axis([0 cont lb(26) ub(26)])
% ylabel('DimY Col 1A, [m]')
%
% subplot(5,5,2)
% plot([0 cont],0.30*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2A,'.')
% axis([0 cont lb(27) ub(27)])
% ylabel('DimY Col 2A, [m]')
%
% subplot(5,5,3,2)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3A,'.')
% axis([0 cont lb(28) ub(28)])
% ylabel('Largo Col 3A, [m]')
%
% subplot(5,5,4)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4A,'.')
% axis([0 cont lb(29) ub(29)])
% ylabel('DimY Col 4A, [m]')
%
% subplot(5,5,5)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5A,'.')
% axis([0 cont lb(30) ub(30)])
% ylabel('DimY Col 5A, [m]')

% subplot(5,5,6)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1B,'.')

```

```

% axis([0 cont lb(31) ub(31)])
% ylabel('DimY Col 1B, [m]')
%
% subplot(5,5,7)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2B,'.')
% axis([0 cont lb(32) ub(32)])
% ylabel('DimY Col 2B, [m]')
%
% subplot(5,5,8)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3B,'.')
% axis([0 cont lb(33) ub(33)])
% ylabel('DimY Col 3B, [m]')
%
% subplot(5,5,9)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4B,'.')
% axis([0 cont lb(34) ub(34)])
% ylabel('DimY Col 4B, [m]')
%
% subplot(5,5,10)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5B,'.')
% axis([0 cont lb(35) ub(35)])
% ylabel('DimY Col 5B, [m]')
%
%
% subplot(5,5,11)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1C,'.')
% axis([0 cont lb(36) ub(36)])
% ylabel('DimY Col 1C, [m]')
%
% subplot(5,5,12)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2C,'.')
% axis([0 cont lb(37) ub(37)])
% ylabel('DimY Col 2C, [m]')
%
% subplot(5,5,13)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3C,'.')
% axis([0 cont lb(38) ub(38)])
% ylabel('DimY Col 3C, [m]')
%
% subplot(5,5,14)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4C,'.')
% axis([0 cont lb(39) ub(39)])
% ylabel('DimY Col 4C, [m]')
%
% subplot(5,5,15)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5C,'.')
% axis([0 cont lb(40) ub(40)])
% ylabel('DimY Col 5C, [m]')

```

```

%
%
% subplot(5,5,16)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1D,'.')
% axis([0 cont lb(41) ub(41)])
% ylabel('DimY Col 1D, [m]')
%
% subplot(5,5,17)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2D,'.')
% axis([0 cont lb(42) ub(42)])
% ylabel('DimY Col 2D, [m]')
%
% subplot(5,5,18)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3D,'.')
% axis([0 cont lb(43) ub(43)])
% ylabel('DimY Col 3D, [m]')
%
% subplot(5,5,19)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4D,'.')
% axis([0 cont lb(44) ub(44)])
% ylabel('DimY Col 4D, [m]')
%
% subplot(5,5,20)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5D,'.')
% axis([0 cont lb(45) ub(45)])
% ylabel('DimY Col 5D, [m]')
%
% subplot(5,5,21)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc1E,'.')
% axis([0 cont lb(46) ub(46)])
% ylabel('DimY Col 1E, [m]')
%
% subplot(5,5,22)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc2E,'.')
% axis([0 cont lb(47) ub(47)])
% ylabel('DimY Col 2E, [m]')
%
% subplot(5,5,23)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc3E,'.')
% axis([0 cont lb(48) ub(48)])
% ylabel('DimY Col 3E, [m]')
%
% subplot(5,5,24)
% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc4E,'.')
% axis([0 cont lb(49) ub(49)])
% ylabel('DimY Col 4E, [m]')
%
% subplot(5,5,25)

```

```

% plot([0 cont],0.3*[1 1],'-','linewidth',1.5), hold on, grid on
% plot([1:cont],opti.hc5E,'.')
% axis([0 cont lb(50) ub(50)])
% ylabel('DimY Col 5E, [m]')

cont = cont +1;

% frame1 = getframe(fig1);           %Captura el frame de la figura
% writeVideo(vidObj1, frame1);       %Escribe el frame en el video
% close(gcf);                         %Cierra la figura si deseas

% frame2 = getframe(fig2);           %Captura el frame de la figura
% writeVideo(vidObj2, frame2);       %Escribe el frame en el video
% %close(gcf);                       %Cierra la figura si deseas

% %%%%%%%%%%%
% currFrame = getframe(gcf);
% writeVideo(vidObj,currFrame);
% %%%%%%%%%%%

end

function penalty = penalty_CV_group(x, idx_group, CVmax)
    % Extrae las variables del grupo
    g = x(idx_group);

    % Calcula el CV
    mu = mean(g);
    sigma = std(g);
    CV = sigma / mu;

    % Penalización tipo bisagra
    penalty = max(0, CV - CVmax)^2;
end

```

```
% ANÁLISIS MODAL Y OPTIMIZACIÓN EN RESPUESTA TORSIONAL PÓRTICO 3D
% Programado por Maria Gracia Villao
% Escuela Superior Politécnica del Litoral
% Trabajo de Titulación Maestría en Estructuras civiles Sismorresistentes
```

```
function [IITx, IITy, Ex, Ey, driftX_max, driftY_max, PMtorsional] =
OptimizacionColumnasEdificio1SoloDiagonal(p)
```

```
format bank
```

```
%Variables
```

```
%Dimensiones en X de Columnas
```

```
bc1A = p(1);
bc2A = p(2);
bc3A = p(3);
bc4A = p(4);
bc5A = p(5);
bc1B = p(6);
bc2B = p(7);
bc3B = p(8);
bc4B = p(9);
bc5B = p(10);
bc1C = p(11);
bc2C = p(12);
bc3C = p(13);
bc4C = p(14);
bc5C = p(15);
bc1D = p(16);
bc2D = p(17);
bc3D = p(18);
bc4D = p(19);
bc5D = p(20);
bc1E = p(21);
bc2E = p(22);
bc3E = p(23);
bc4E = p(24);
bc5E = p(25);
```

```
%Dimensiones en Y de Columnas
```

```
hc1A = p(26);
hc2A = p(27);
hc3A = p(28);
hc4A = p(29);
hc5A = p(30);
hc1B = p(31);
hc2B = p(32);
hc3B = p(33);
hc4B = p(34);
hc5B = p(35);
hc1C = p(36);
hc2C = p(37);
hc3C = p(38);
hc4C = p(39);
hc5C = p(40);
hc1D = p(41);
hc2D = p(42);
```

```

hc3D = p(43);
hc4D = p(44);
hc5D = p(45);
hc1E = p(46);
hc2E = p(47);
hc3E = p(48);
hc4E = p(49);
hc5E = p(50);

```

### %% Definición de Materiales y Configuración Estructural

```

E = 24855577549.348;           %Módulo Elasticidad [Pa]
carga.pis = 1000;             %Masa de piso 1 y 2 por m2 [kg/m2]
carga.cub = 940;             %Masa piso de cubierta por m2 [kg/m2]
razon.amort = 0.05;          %Razon amortiguamiento de estructura
% Cant.pisos = 3;
zeta = razon.amort*ones(1,9); %matriz con razones de amortiguamiento
[1xGDL]
Port.x = 5;
Port.y = 5;

```

### %% Pisos

```

a1 = 26.25;                   %Longitud de piso 1 [m]
b1 = 22;                      %Ancho de piso 1 [m]
Area.pis = a1*b1;            %Área de piso 1 [m2]
%Cubierta
a2 = 28;                      %Longitud de cubierta [m]
b2 = 22;                      %Ancho de cubierta [m]
Area.cub = a2*b2;           %Área de cubierta [m2]

```

### %% Características de PORTICO A en sentido "x"

#### %% Columnas 1A, 2A, 3A, 4A, 5A

```
Lc = 3;                       %Longitud de columna [m]
```

#### %% Columna 1A

```

%bc1A ;                      %Ancho columna 1A [m]
%hc1A ;                      %Largo columna 1A [m]
Ac1A = bc1A*hc1A;           %Area de columna 1A [m^2]
Ic1A = 0.8*(hc1A*bc1A^3/12); %Inercia agrietada columna 1A [m^4]

```

#### %% Columna 2A

```

%bc2A ;                      %Ancho columna 2A [m]
%hc2A ;                      %Largo columna 2A [m]
Ac2A = bc2A*hc2A;           %Area de columna 2A [m^2]
Ic2A = 0.8*(hc2A*bc2A^3/12); %Inercia agrietada columna 2A [m^4]

```

#### %% Columna 3A

```

%bc3A ;                      %Ancho columna 3A [m]
%hc3A ;                      %Largo columna 3A [m]
Ac3A = bc3A*hc3A;           %Area de columna 3A [m^2]
Ic3A = 0.8*(hc3A*bc3A^3/12); %Inercia agrietada columna 3A [m^4]

```

#### %% Columna 4A

```
%bc4A ;                      %Ancho columna 4A [m]
```

```

%hc4A ; %Largo columna 4A [m]
Ac4A = bc4A*hc4A; %Area de columna 4A [m^2]
Ic4A = 0.8*(hc4A*bc4A^3/12); %Inercia agrietada columna 4A [m^4]

%Columna 5A
%bc5A ; %Ancho columna 5A [m]
%hc5A ; %Largo columna 5A [m]
Ac5A = bc5A*hc5A; %Area de columna 5A [m^2]
Ic5A = 0.8*(hc5A*bc5A^3/12); %Inercia agrietada columna 5A [m^4]

%Viga
Lv1 = 5.5; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*(bv1*hv1^3/12); %Inercia agrietada viga [m^4]

%% Coordenadas de los nodos de portico en "x"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [Lv1 0 0];
nod(6,:) = [Lv1 0 Lc];
nod(7,:) = [Lv1 0 2*Lc];
nod(8,:) = [Lv1 0 3*Lc];

nod(9,:) = [2*Lv1 0 0];
nod(10,:) = [2*Lv1 0 Lc];
nod(11,:) = [2*Lv1 0 2*Lc];
nod(12,:) = [2*Lv1 0 3*Lc];

nod(13,:) = [3*Lv1 0 0];
nod(14,:) = [3*Lv1 0 Lc];
nod(15,:) = [3*Lv1 0 2*Lc];
nod(16,:) = [3*Lv1 0 3*Lc];

nod(17,:) = [4*Lv1 0 0];
nod(18,:) = [4*Lv1 0 Lc];
nod(19,:) = [4*Lv1 0 2*Lc];
nod(20,:) = [4*Lv1 0 3*Lc];

con = [1 2
2 3
3 4
5 6
6 7
7 8
9 10
10 11
11 12
13 14
14 15
15 16]

```

```

17 18
18 19
19 20
2 6
6 10
10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20];

```

```

%% Matriz de transformacion para cada elemento

```

```

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:,[1 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:,[1 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:,[1 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:,[1 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[1 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:,[1 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[1 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[1 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[1 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[1 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[1 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[1 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[1 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[1 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[1 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[1 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[1 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[1 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[1 3]), con(19,:));

```

```

% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[1 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[1 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[1 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[1 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[1 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[1 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[1 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[1 3]), con(27,:));

%% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac1A,E,Ic1A,L1); % Elemento 1
k2 = K_f(Ac1A,E,Ic1A,L2); % Elemento 2
k3 = K_f(Ac1A,E,Ic1A,L3); % Elemento 3
k4 = K_f(Ac2A,E,Ic2A,L4); % Elemento 4
k5 = K_f(Ac2A,E,Ic2A,L5); % Elemento 5
k6 = K_f(Ac2A,E,Ic2A,L6); % Elemento 6
k7 = K_f(Ac3A,E,Ic3A,L7); % Elemento 7
k8 = K_f(Ac3A,E,Ic3A,L8); % Elemento 8
k9 = K_f(Ac3A,E,Ic3A,L9); % Elemento 9
k10 = K_f(Ac4A,E,Ic4A,L10); % Elemento 10
k11 = K_f(Ac4A,E,Ic4A,L11); % Elemento 11
k12 = K_f(Ac4A,E,Ic4A,L12); % Elemento 12
k13 = K_f(Ac5A,E,Ic5A,L13); % Elemento 13
k14 = K_f(Ac5A,E,Ic5A,L14); % Elemento 14
k15 = K_f(Ac5A,E,Ic5A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6

```

```

K7 = bet_7'*k7*bet_7;    % Elemento 7
K8 = bet_8'*k8*bet_8;    % Elemento 8
K9 = bet_9'*k9*bet_9;    % Elemento 9
K10 = bet_10'*k10*bet_10;    % Elemento 10
K11 = bet_11'*k11*bet_11;    % Elemento 11
K12 = bet_12'*k12*bet_12;    % Elemento 12
K13 = bet_13'*k13*bet_13;    % Elemento 13
K14 = bet_14'*k14*bet_14;    % Elemento 14
K15 = bet_15'*k15*bet_15;    % Elemento 15
K16 = bet_16'*k16*bet_16;    % Elemento 16
K17 = bet_17'*k17*bet_17;    % Elemento 17
K18 = bet_18'*k18*bet_18;    % Elemento 18
K19 = bet_19'*k19*bet_19;    % Elemento 19
K20 = bet_20'*k20*bet_20;    % Elemento 20
K21 = bet_21'*k21*bet_21;    % Elemento 21
K22 = bet_22'*k22*bet_22;    % Elemento 22
K23 = bet_23'*k23*bet_23;    % Elemento 23
K24 = bet_24'*k24*bet_24;    % Elemento 24
K25 = bet_25'*k25*bet_25;    % Elemento 25
K26 = bet_26'*k26*bet_26;    % Elemento 26
K27 = bet_27'*k27*bet_27;    % Elemento 27

```

```

%% Ensamblar matriz global KK para un pórtico

```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

```

```

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

%% Eliminar GDL conocidos o cero

% 'Matriz de rigidez condensada:'
% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.xxA = K_con;

%% Caracteristicas de PORTICO B en sentido "x"

%Columnas 1B, 2B, 3B, 4B, 5B
Lc = 3; %Altura de columna [m]

%Columna 1B
% bc1B = 0.45 ; %Ancho columna 1B [m]
% hc1B = 0.45; %Largo columna 1B [m]
Ac1B = bc1B*hc1B; %Area de columna 1B [m^2]
Ic1B = 0.8*hc1B*bc1B^3/12; %Inercia columna 1B [m^4]

%Columna 2B
% bc2B = 0.45; %Ancho columna 2B [m]
% hc2B = 0.45; %Largo columna 2B [m]
Ac2B = bc2B*hc2B; %Area de columna 2B [m^2]
Ic2B = 0.8*hc2B*bc2B^3/12; %Inercia columna 2B [m^4]

```

```

%Columna 3B
% bc3B = 0.45; %Ancho columna 3B [m]
% hc3B = 0.45; %Largo columna 3B [m]
Ac3B = bc3B*hc3B; %Area de columna 3B [m^2]
Ic3B = 0.8*hc3B*bc3B^3/12; %Inercia columna 3B [m^4]

%Columna 4B
% bc4B = 0.45; %Ancho columna 4B [m]
% hc4B = 0.45; %Largo columna 4B [m]
Ac4B = bc4B*hc4B; %Area de columna 4B [m^2]
Ic4B = 0.8*hc4B*bc4B^3/12; %Inercia columna 4B [m^4]

%Columna 5B
% bc5B = 0.45; %Ancho columna 5B [m]
% hc5B = 0.45; %Largo columna 5B [m]
Ac5B = bc5B*hc5B; %Area de columna 5B [m^2]
Ic5B = 0.8*hc5B*bc5B^3/12; %Inercia columna 5B [m^4]

%Viga
Lv = 5.5; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*bv1*hv1^3/12; %Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "x"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [Lv 0 0];
nod(6,:) = [Lv 0 Lc];
nod(7,:) = [Lv 0 2*Lc];
nod(8,:) = [Lv 0 3*Lc];

nod(9,:) = [2*Lv 0 0];
nod(10,:) = [2*Lv 0 Lc];
nod(11,:) = [2*Lv 0 2*Lc];
nod(12,:) = [2*Lv 0 3*Lc];

nod(13,:) = [3*Lv 0 0];
nod(14,:) = [3*Lv 0 Lc];
nod(15,:) = [3*Lv 0 2*Lc];
nod(16,:) = [3*Lv 0 3*Lc];

nod(17,:) = [4*Lv 0 0];
nod(18,:) = [4*Lv 0 Lc];
nod(19,:) = [4*Lv 0 2*Lc];
nod(20,:) = [4*Lv 0 3*Lc];

con = [1 2
       2 3
       3 4
       5 6
       6 7

```

```

7 8
9 10
10 11
11 12
13 14
14 15
15 16
17 18
18 19
19 20
2 6
6 10
10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20];

```

```
%% Matriz de transformacion para cada elemento
```

```

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:, [1 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:, [1 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:, [1 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:, [1 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:, [1 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:, [1 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:, [1 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:, [1 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:, [1 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:, [1 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:, [1 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:, [1 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:, [1 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:, [1 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:, [1 3]), con(15,:));

% Elemento 16 (nodo 2-6)

```

```

[bet_16, L16] = mat_CS(nod(:,[1 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[1 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[1 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[1 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[1 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[1 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[1 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[1 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[1 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[1 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[1 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[1 3]), con(27,:));

%% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac1B,E,Ic1B,L1); % Elemento 1
k2 = K_f(Ac1B,E,Ic1B,L2); % Elemento 2
k3 = K_f(Ac1B,E,Ic1B,L3); % Elemento 3
k4 = K_f(Ac2B,E,Ic2B,L4); % Elemento 4
k5 = K_f(Ac2B,E,Ic2B,L5); % Elemento 5
k6 = K_f(Ac2B,E,Ic2B,L6); % Elemento 6
k7 = K_f(Ac3B,E,Ic3B,L7); % Elemento 7
k8 = K_f(Ac3B,E,Ic3B,L8); % Elemento 8
k9 = K_f(Ac3B,E,Ic3B,L9); % Elemento 9
k10 = K_f(Ac4B,E,Ic4B,L10); % Elemento 10
k11 = K_f(Ac4B,E,Ic4B,L11); % Elemento 11
k12 = K_f(Ac4B,E,Ic4B,L12); % Elemento 12
k13 = K_f(Ac5B,E,Ic5B,L13); % Elemento 13
k14 = K_f(Ac5B,E,Ic5B,L14); % Elemento 14
k15 = K_f(Ac5B,E,Ic5B,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%%

```

```
% Matriz global  $K_g^{\left(e\right)}$  de cada elemento
```

```
K1 = bet_1'*k1*bet_1;    % Elemento 1
K2 = bet_2'*k2*bet_2;    % Elemento 2
K3 = bet_3'*k3*bet_3;    % Elemento 3
K4 = bet_4'*k4*bet_4;    % Elemento 4
K5 = bet_5'*k5*bet_5;    % Elemento 5
K6 = bet_6'*k6*bet_6;    % Elemento 6
K7 = bet_7'*k7*bet_7;    % Elemento 7
K8 = bet_8'*k8*bet_8;    % Elemento 8
K9 = bet_9'*k9*bet_9;    % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20
K21 = bet_21'*k21*bet_21; % Elemento 21
K22 = bet_22'*k22*bet_22; % Elemento 22
K23 = bet_23'*k23*bet_23; % Elemento 23
K24 = bet_24'*k24*bet_24; % Elemento 24
K25 = bet_25'*k25*bet_25; % Elemento 25
K26 = bet_26'*k26*bet_26; % Elemento 26
K27 = bet_27'*k27*bet_27; % Elemento 27
```

```
%% Ensamblar matriz global KK para un pórtico
```

```
dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
```

```
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;
```

```
KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;
```

```
KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;
```

```
%% Eliminar GDL conocidos o cero
```

```
% 'Matriz de rigidez condensada:'
```

```
    % gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
```

```
    % gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
```

```
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
```

```
Kred = KK(gdl,gdl);
```

```
% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
```

```
Kred(1,:) = Kred(1,:)+Kred(7,:)+Kred(13,:)+Kred(19,:)+Kred(25,:);
```

```
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
```

```
Kred(3,:) = Kred(3,:)+Kred(9,:)+Kred(15,:)+Kred(21,:)+Kred(27,:);
```

```
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
```

```
Kred(5,:) = Kred(5,:)+Kred(11,:)+Kred(17,:)+Kred(23,:)+Kred(29,:);
```

```
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);
```

```
% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
```

```
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);
```

```
% Acomodar los GDL horizontales en las 1ras filas/columnas
```

```
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);
```

```
% Condensar grados de libertad rotacionales
```

```
kpp = Kn(1:3,1:3);
```

```
kpo = Kn(1:3,4:end);
```

```
kop = kpo.';
```

```
koo = Kn(4:end,4:end);
```

```
K_con = kpp-kpo*(koo\kop);
```

```
Kc.xxv = K_con;
```

```
%% Caracteristicas de portico C en sentido "x"
```

```
%Columnas 1C, 2C, 3C, 4C, 5C
```

```
Lc = 3; %Longitud de columna [m]
```

```
%Columna 1C
```

```
% bc1C = 0.45;
```

```
%Ancho columna 1C [m]
```

```
% hc1C = 0.45;
```

```
%Largo columna 1C [m]
```

```

Ac1C = bc1C*hc1C; %Area de columna 1C [m^2]
Ic1C = 0.8*hc1C*bc1C^3/12; %Inercia columna 1C [m^4]

%Columna 2C
% bc2C = 0.45; %Ancho columna 2C [m]
% hc2C = 0.45; %Largo columna 2C [m]
Ac2C = bc2C*hc2C; %Area de columna 2C [m^2]
Ic2C = 0.8*hc2C*bc2C^3/12; %Inercia columna 2C [m^4]

%Columna 3C
% bc3C = 0.45; %Ancho columna 3C [m]
% hc3C = 0.45; %Largo columna 3C [m]
Ac3C = bc3C*hc3C; %Area de columna 3C [m^2]
Ic3C = 0.8*hc3C*bc3C^3/12; %Inercia columna 3C [m^4]

%Columna 4C
% bc4C = 0.45; %Ancho columna 4C [m]
% hc4C = 0.45; %Largo columna 4C [m]
Ac4C = bc4C*hc4C; %Area de columna 4C [m^2]
Ic4C = 0.8*hc4C*bc4C^3/12; %Inercia columna 4C [m^4]

%Columna 5C
% bc5C = 0.45; %Ancho columna 5C [m]
% hc5C = 0.45; %Largo columna 5C [m]
Ac5C = bc5C*hc5C; %Area de columna 5C [m^2]
Ic5C = 0.8*hc5C*bc5C^3/12; %Inercia columna 5C [m^4]

%Viga
Lv = 5.5; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*bv1*hv1^3/12; %Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "x"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [Lv 0 0];
nod(6,:) = [Lv 0 Lc];
nod(7,:) = [Lv 0 2*Lc];
nod(8,:) = [Lv 0 3*Lc];

nod(9,:) = [2*Lv 0 0];
nod(10,:) = [2*Lv 0 Lc];
nod(11,:) = [2*Lv 0 2*Lc];
nod(12,:) = [2*Lv 0 3*Lc];

nod(13,:) = [3*Lv 0 0];
nod(14,:) = [3*Lv 0 Lc];
nod(15,:) = [3*Lv 0 2*Lc];
nod(16,:) = [3*Lv 0 3*Lc];

nod(17,:) = [4*Lv 0 0];

```

```

nod(18,:) = [4*Lvx 0 Lc];
nod(19,:) = [4*Lvx 0 2*Lc];
nod(20,:) = [4*Lvx 0 3*Lc];

```

```

con = [1 2
      2 3
      3 4
      5 6
      6 7
      7 8
      9 10
      10 11
      11 12
      13 14
      14 15
      15 16
      17 18
      18 19
      19 20
      2 6
      6 10
      10 14
      14 18
      3 7
      7 11
      11 15
      15 19
      4 8
      8 12
      12 16
      16 20];

```

```

%% Matriz de transformacion para cada elemento

```

```

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:, [1 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:, [1 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:, [1 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:, [1 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:, [1 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:, [1 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:, [1 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:, [1 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:, [1 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:, [1 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:, [1 3]), con(11,:));
% Elemento 12 (nodo 15-16)

```

```

[bet_12, L12] = mat_CS(nod(:,[1 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[1 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[1 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[1 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[1 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[1 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[1 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[1 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[1 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[1 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[1 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[1 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[1 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[1 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[1 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[1 3]), con(27,:));

%% Matriz local  $K_L^{\left\{e\right\}}$  de cada elemento

k1 = K_f(Ac1C,E,Ic1C,L1); % Elemento 1
k2 = K_f(Ac1C,E,Ic1C,L2); % Elemento 2
k3 = K_f(Ac1C,E,Ic1C,L3); % Elemento 3
k4 = K_f(Ac2C,E,Ic2C,L4); % Elemento 4
k5 = K_f(Ac2C,E,Ic2C,L5); % Elemento 5
k6 = K_f(Ac2C,E,Ic2C,L6); % Elemento 6
k7 = K_f(Ac3C,E,Ic3C,L7); % Elemento 7
k8 = K_f(Ac3C,E,Ic3C,L8); % Elemento 8
k9 = K_f(Ac3C,E,Ic3C,L9); % Elemento 9
k10 = K_f(Ac4C,E,Ic4C,L10); % Elemento 10
k11 = K_f(Ac4C,E,Ic4C,L11); % Elemento 11
k12 = K_f(Ac4C,E,Ic4C,L12); % Elemento 12
k13 = K_f(Ac5C,E,Ic5C,L13); % Elemento 13
k14 = K_f(Ac5C,E,Ic5C,L14); % Elemento 14
k15 = K_f(Ac5C,E,Ic5C,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21

```

```

k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20
K21 = bet_21'*k21*bet_21; % Elemento 21
K22 = bet_22'*k22*bet_22; % Elemento 22
K23 = bet_23'*k23*bet_23; % Elemento 23
K24 = bet_24'*k24*bet_24; % Elemento 24
K25 = bet_25'*k25*bet_25; % Elemento 25
K26 = bet_26'*k26*bet_26; % Elemento 26
K27 = bet_27'*k27*bet_27; % Elemento 27

%% Ensamblar matriz global KK para un pórtico

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

```

```

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

```

```

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

```

```

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

```

```

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

```

```
%% Eliminar GDL conocidos o cero
```

```
% 'Matriz de rigidez condensada:'
```

```

% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

```

```
% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
```

```

Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

```

```
% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
```

```

Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

```

```
% Acomodar los GDL horizontales en las 1ras filas/columnas
```

```

Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

```

```
% Condensar grados de libertad rotacionales
```

```

kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

```

```

K_con = kpp-kpo*(koo\kop);
Kc.xx C = K_con;

```

```
%% Características del portico D en sentido "x"
```

```

%Columnas 1D, 2D, 3D, 4D, 5D
Lc = 3; %Longitud de columna [m]

%Columna 1D
% bc1D = 0.45; %Ancho columna 1D [m]
% hc1D = 0.45; %Largo columna 1D [m]
Ac1D = bc1D*hc1D; %Area de columna 1D [m^2]
Ic1D = 0.8*hc1D*bc1D^3/12; %Inercia columna 1D [m^4]

%Columna 2D
% bc2D = 0.45; %Ancho columna 2D [m]
% hc2D = 0.45; %Largo columna 2D [m]
Ac2D = bc2D*hc2D; %Area de columna 2D [m^2]
Ic2D = 0.8*hc2D*bc2D^3/12; %Inercia columna 2D [m^4]

%Columna 3D
% bc3D = 0.45; %Ancho columna 3D [m]
% hc3D = 0.45; %Largo columna 3D [m]
Ac3D = bc3D*hc3D; %Area de columna 3D [m^2]
Ic3D = 0.8*hc3D*bc3D^3/12; %Inercia columna 3D [m^4]

%Columna 4D
% bc4D = 0.45; %Ancho columna 4D [m]
% hc4D = 0.45; %Largo columna 4D [m]
Ac4D = bc4D*hc4D; %Area de columna 4D [m^2]
Ic4D = 0.8*hc4D*bc4D^3/12; %Inercia columna 4D [m^4]

%Columna 5D
% bc5D = 0.45; %Ancho columna 5D [m]
% hc5D = 0.45; %Largo columna 5D [m]
Ac5D = bc5D*hc5D; %Area de columna 5D [m^2]
Ic5D = 0.8*hc5D*bc5D^3/12; %Inercia columna 5D [m^4]

%Viga
Lv = 5.5; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*bv1*hv1^3/12; %Inercia viga [m^4]

%DIAGONAL
hv2 = 0.25; %Peralte viga tipo 1 [m]
bv2 = 0.25; %Ancho viga tipo 1 [m]
Avd = bv2*hv2; %Area de diagonal
Ivd = 0.5*bv2*hv2^3/12; %Inercia diagonal [m^4]

%% Coordenadas de los nodos de portico en "x"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [Lv 0 0];
nod(6,:) = [Lv 0 Lc];
nod(7,:) = [Lv 0 2*Lc];

```

```

nod(8,:) = [Lvx 0 3*Lc];

nod(9,:) = [2*Lvx 0 0];
nod(10,:) = [2*Lvx 0 Lc];
nod(11,:) = [2*Lvx 0 2*Lc];
nod(12,:) = [2*Lvx 0 3*Lc];

nod(13,:) = [3*Lvx 0 0];
nod(14,:) = [3*Lvx 0 Lc];
nod(15,:) = [3*Lvx 0 2*Lc];
nod(16,:) = [3*Lvx 0 3*Lc];

nod(17,:) = [4*Lvx 0 0];
nod(18,:) = [4*Lvx 0 Lc];
nod(19,:) = [4*Lvx 0 2*Lc];
nod(20,:) = [4*Lvx 0 3*Lc];

con = [1 2
       2 3
       3 4
       5 6
       6 7
       7 8
       9 10
       10 11
       11 12
       13 14
       14 15
       15 16
       17 18
       18 19
       19 20
       2 6
       6 10
       10 14
       14 18
       3 7
       7 11
       11 15
       15 19
       4 8
       8 12
       12 16
       16 20
       13 18
       15 18];

%%
% Matriz de transformacion para cada elemento

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:, [1 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:, [1 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:, [1 3]), con(3,:));
% Elemento 4 (nodo 5-6)

```

```

[bet_4, L4] = mat_CS(nod(:,[1 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[1 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:,[1 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[1 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[1 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[1 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[1 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[1 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[1 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[1 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[1 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[1 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[1 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[1 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[1 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[1 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[1 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[1 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[1 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[1 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[1 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[1 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[1 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[1 3]), con(27,:));

%Escalera
% Elemento 28 (nodo 14-17)
[bet_28, L28] = mat_CS(nod(:,[1 3]), con(28,:));
% Elemento 29 (nodo 14-19)
[bet_29, L29] = mat_CS(nod(:,[1 3]), con(29,:));

```

```

%%
% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac1D,E,Ic1D,L1); % Elemento 1
k2 = K_f(Ac1D,E,Ic1D,L2); % Elemento 2
k3 = K_f(Ac1D,E,Ic1D,L3); % Elemento 3
k4 = K_f(Ac2D,E,Ic2D,L4); % Elemento 4
k5 = K_f(Ac2D,E,Ic2D,L5); % Elemento 5
k6 = K_f(Ac2D,E,Ic2D,L6); % Elemento 6
k7 = K_f(Ac3D,E,Ic3D,L7); % Elemento 7
k8 = K_f(Ac3D,E,Ic3D,L8); % Elemento 8
k9 = K_f(Ac3D,E,Ic3D,L9); % Elemento 9
k10 = K_f(Ac4D,E,Ic4D,L10); % Elemento 10
k11 = K_f(Ac4D,E,Ic4D,L11); % Elemento 11
k12 = K_f(Ac4D,E,Ic4D,L12); % Elemento 12
k13 = K_f(Ac5D,E,Ic5D,L13); % Elemento 13
k14 = K_f(Ac5D,E,Ic5D,L14); % Elemento 14
k15 = K_f(Ac5D,E,Ic5D,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%Escalera (diagonal)
k28 = K_f(Avd,E,Ivd,L28); % Elemento 28
k29 = K_f(Avd,E,Ivd,L29); % Elemento 29

%%
% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17

```

```

K18 = bet_18'*k18*bet_18;    % Elemento 18
K19 = bet_19'*k19*bet_19;    % Elemento 19
K20 = bet_20'*k20*bet_20;    % Elemento 20
K21 = bet_21'*k21*bet_21;    % Elemento 21
K22 = bet_22'*k22*bet_22;    % Elemento 22
K23 = bet_23'*k23*bet_23;    % Elemento 23
K24 = bet_24'*k24*bet_24;    % Elemento 24
K25 = bet_25'*k25*bet_25;    % Elemento 25
K26 = bet_26'*k26*bet_26;    % Elemento 26
K27 = bet_27'*k27*bet_27;    % Elemento 27

```

```
%Escalera
```

```

K28 = bet_28'*k28*bet_28;    % Elemento 28
K29 = bet_29'*k29*bet_29;    % Elemento 29

```

```
%%
```

```
% Ensamblar matriz global KK para un pórtico
```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6)      = K1;
KK(4:9,4:9)      = KK(4:9,4:9) + K2;
KK(7:12,7:12)    = KK(7:12,7:12) + K3;

KK(13:18,13:18)  = KK(13:18,13:18) + K4;
KK(16:21,16:21)  = KK(16:21,16:21) + K5;
KK(19:24,19:24)  = KK(19:24,19:24) + K6;

KK(25:30,25:30)  = KK(25:30,25:30) + K7;
KK(28:33,28:33)  = KK(28:33,28:33) + K8;
KK(31:36,31:36)  = KK(31:36,31:36) + K9;

KK(37:42,37:42)  = KK(37:42,37:42) + K10;
KK(40:45,40:45)  = KK(40:45,40:45) + K11;
KK(43:48,43:48)  = KK(43:48,43:48) + K12;

KK(49:54,49:54)  = KK(49:54,49:54) + K13;
KK(52:57,52:57)  = KK(52:57,52:57) + K14;
KK(55:60,55:60)  = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

```

```

%Escalera
KK([37:39 52:54],[37:39 52:54]) = KK([37:39 52:54],[37:39 52:54]) + K28;
KK([43:45 52:54],[43:45 52:54]) = KK([43:45 52:54],[43:45 52:54]) + K29;

%%
% Eliminar GDL conocidos o cero

% 'Matriz de rigidez condensada:'
% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.xxD = K_con;

%% Caracteristicas del portico E en sentido "x"

%Columnas 1E, 2E, 3E, 4E, 5E
Lc = 3; %Longitud de columna [m]

%Columna 1E
% bc1E = 0.45; %Ancho columna 1E [m]
% hc1E = 0.45; %Largo columna 1E [m]
Ac1E = bc1E*hc1E; %Area de columna 1E [m^2]
Ic1E = 0.8*hc1E*bc1E^3/12; %Inercia columna 1E [m^4]

%Columna 2E
% bc2E = 0.45; %Ancho columna 2E [m]
% hc2E = 0.45; %Largo columna 2E [m]
Ac2E = bc2E*hc2E; %Area de columna 2E [m^2]

```

```

Ic2E = 0.8*hc2E*bc2E^3/12;           %Inercia columna 2E [m^4]

%Columna 3E
% bc3E = 0.45;                       %Ancho columna 3E [m]
% hc3E = 0.45;                       %Largo columna 3E [m]
Ac3E = bc3E*hc3E;                   %Area de columna 3E [m^2]
Ic3E = 0.8*hc3E*bc3E^3/12;         %Inercia columna 3E [m^4]

%Columna 4E
% bc4E = 0.45;                       %Ancho columna 4E [m]
% hc4E = 0.45;                       %Largo columna 4E [m]
Ac4E = bc4E*hc4E;                   %Area de columna 4E [m^2]
Ic4E = 0.8*hc4E*bc4E^3/12;         %Inercia columna 4E [m^4]

%Columna 5E
% bc5E = 0.45;                       %Ancho columna 5E [m]
% hc5E = 0.45;                       %Largo columna 5E [m]
Ac5E = bc5E*hc5E;                   %Area de columna 5E [m^2]
Ic5E = 0.8*hc5E*bc5E^3/12;         %Inercia columna 5E [m^4]

%Viga
Lv1 = 5.5;                          %Longitud de viga [m]
hv1 = 0.45;                          %Peralte viga tipo 1 [m]
bv1 = 0.25;                          %Ancho viga tipo 1 [m]
Av = bv1*hv1;                        %Area de viga
Iv = 0.5*bv1*hv1^3/12;               %Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "x"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [Lv1 0 0];
nod(6,:) = [Lv1 0 Lc];
nod(7,:) = [Lv1 0 2*Lc];
nod(8,:) = [Lv1 0 3*Lc];

nod(9,:) = [2*Lv1 0 0];
nod(10,:) = [2*Lv1 0 Lc];
nod(11,:) = [2*Lv1 0 2*Lc];
nod(12,:) = [2*Lv1 0 3*Lc];

nod(13,:) = [3*Lv1 0 0];
nod(14,:) = [3*Lv1 0 Lc];
nod(15,:) = [3*Lv1 0 2*Lc];
nod(16,:) = [3*Lv1 0 3*Lc];

nod(17,:) = [4*Lv1 0 0];
nod(18,:) = [4*Lv1 0 Lc];
nod(19,:) = [4*Lv1 0 2*Lc];
nod(20,:) = [4*Lv1 0 3*Lc];

con = [1 2
       2 3]

```

```

3 4
5 6
6 7
7 8
9 10
10 11
11 12
13 14
14 15
15 16
17 18
18 19
19 20
2 6
6 10
10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20
14 17
14 19
16 19];

```

```

%% Matriz de transformacion para cada elemento

```

```

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:, [1 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:, [1 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:, [1 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:, [1 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:, [1 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:, [1 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:, [1 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:, [1 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:, [1 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:, [1 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:, [1 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:, [1 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:, [1 3]), con(13,:));

```

```

% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[1 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[1 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[1 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[1 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[1 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[1 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[1 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[1 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[1 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[1 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[1 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[1 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[1 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[1 3]), con(27,:));

```

```

%% Matriz local  $K_L^{\left\{e\right\}}$  de cada elemento

```

```

k1 = K_f(Ac1E,E,Ic1E,L1); % Elemento 1
k2 = K_f(Ac1E,E,Ic1E,L2); % Elemento 2
k3 = K_f(Ac1E,E,Ic1E,L3); % Elemento 3
k4 = K_f(Ac2E,E,Ic2E,L4); % Elemento 4
k5 = K_f(Ac2E,E,Ic2E,L5); % Elemento 5
k6 = K_f(Ac2E,E,Ic2E,L6); % Elemento 6
k7 = K_f(Ac3E,E,Ic3E,L7); % Elemento 7
k8 = K_f(Ac3E,E,Ic3E,L8); % Elemento 8
k9 = K_f(Ac3E,E,Ic3E,L9); % Elemento 9
k10 = K_f(Ac4E,E,Ic4E,L10); % Elemento 10
k11 = K_f(Ac4E,E,Ic4E,L11); % Elemento 11
k12 = K_f(Ac4E,E,Ic4E,L12); % Elemento 12
k13 = K_f(Ac5E,E,Ic5E,L13); % Elemento 13
k14 = K_f(Ac5E,E,Ic5E,L14); % Elemento 14
k15 = K_f(Ac5E,E,Ic5E,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22

```

```

k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

```

```

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

```

```

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20
K21 = bet_21'*k21*bet_21; % Elemento 21
K22 = bet_22'*k22*bet_22; % Elemento 22
K23 = bet_23'*k23*bet_23; % Elemento 23
K24 = bet_24'*k24*bet_24; % Elemento 24
K25 = bet_25'*k25*bet_25; % Elemento 25
K26 = bet_26'*k26*bet_26; % Elemento 26
K27 = bet_27'*k27*bet_27; % Elemento 27

```

```

%% Ensamblar matriz global KK para un pórtico

```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;

```

```
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;
```

```
KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;
```

```
KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;
```

```
KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;
```

```
KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;
```

```
%% Eliminar GDL conocidos o cero
```

```
% 'Matriz de rigidez condensada:'
```

```
    % gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
    % gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
```

```
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);
```

```
% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
```

```
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);
```

```
% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
% u46, u52, u55, u58
```

```
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);
```

```
% Acomodar los GDL horizontales en las 1ras filas/columnas
```

```
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);
```

```
% Condensar grados de libertad rotacionales
```

```
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);
```

```

K_con = kpp-kpo*(koo\kop);
Kc.xxE = K_con;

%% Caracteristicas del PORTICO 1 sentido "y"

%Columnas 1A, 1B, 1C, 1D, 1E
Lc = 3; %Longitud de columna [m]

%Columna 1A
%bc1A ; %Ancho columna 1A [m]
%hc1A ; %Largo columna 1A [m]
Ac1A = bc1A*hc1A; %Area de columna 1A [m^2]
Ic1A = 0.8*bc1A*hc1A^3/12; %Inercia columna 1A [m^4]

%Columna 1B
% bc1B = 0.45; %Ancho columna 1B [m]
% hc1B = 0.45; %Largo columna 1B [m]
Ac1B = bc1B*hc1B; %Area de columna 1B [m^2]
Ic1B = 0.8*bc1B*hc1B^3/12; %Inercia columna 1B [m^4]

%Columna 1C
% bc1C = 0.45; %Ancho columna 1C [m]
% hc1C = 0.45; %Largo columna 1C [m]
Ac1C = bc1C*hc1C; %Area de columna 1C [m^2]
Ic1C = 0.8*bc1C*hc1C^3/12; %Inercia columna 1C [m^4]

%Columna 1D
% bc1D = 0.45; %Ancho columna 1D [m]
% hc1D = 0.45; %Largo columna 1D [m]
Ac1D = bc1D*hc1D; %Area de columna 1D [m^2]
Ic1D = 0.8*bc1D*hc1D^3/12; %Inercia columna 1D [m^4]

%Columna 1E
% bc1E = 0.45; %Ancho columna 1E [m]
% hc1E = 0.45; %Largo columna 1E [m]
Ac1E = bc1E*hc1E; %Area de columna 1E [m^2]
Ic1E = 0.8*bc1E*hc1E^3/12; %Inercia columna 1E [m^4]

%Viga
Lvy = 7; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*bv1*hv1^3/12; %Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "y"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [0 Lvy 0];
nod(6,:) = [0 Lvy Lc];
nod(7,:) = [0 Lvy 2*Lc];
nod(8,:) = [0 Lvy 3*Lc];

```

```

nod(9,:) = [0 2*Lvy 0];
nod(10,:) = [0 2*Lvy Lc];
nod(11,:) = [0 2*Lvy 2*Lc];
nod(12,:) = [0 2*Lvy 3*Lc];

nod(13,:) = [0 3*Lvy 0];
nod(14,:) = [0 3*Lvy Lc];
nod(15,:) = [0 3*Lvy 2*Lc];
nod(16,:) = [0 3*Lvy 3*Lc];

nod(17,:) = [0 4*Lvy 0];
nod(18,:) = [0 4*Lvy Lc];
nod(19,:) = [0 4*Lvy 2*Lc];
nod(20,:) = [0 4*Lvy 3*Lc];

con = [1 2
       2 3
       3 4
       5 6
       6 7
       7 8
       9 10
       10 11
       11 12
       13 14
       14 15
       15 16
       17 18
       18 19
       19 20
       2 6
       6 10
       10 14
       14 18
       3 7
       7 11
       11 15
       15 19
       4 8
       8 12
       12 16
       16 20];

%% Matriz de transformacion para cada elemento

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:,[2 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:,[2 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:,[2 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:,[2 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[2 3]), con(5,:));
% Elemento 6 (nodo 7-8)

```

```

[bet_6, L6] = mat_CS(nod(:,[2 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[2 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[2 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[2 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[2 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[2 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[2 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[2 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[2 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[2 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[2 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[2 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[2 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[2 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[2 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[2 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[2 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[2 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[2 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[2 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[2 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[2 3]), con(27,:));

%% Matriz local  $K_L^{\left(e\right)}$  de cada elemento

k1 = K_f(Ac1E,E,Ic1E,L1); % Elemento 1
k2 = K_f(Ac1E,E,Ic1E,L2); % Elemento 2
k3 = K_f(Ac1E,E,Ic1E,L3); % Elemento 3
k4 = K_f(Ac1D,E,Ic1D,L4); % Elemento 4
k5 = K_f(Ac1D,E,Ic1D,L5); % Elemento 5
k6 = K_f(Ac1D,E,Ic1D,L6); % Elemento 6
k7 = K_f(Ac1C,E,Ic1C,L7); % Elemento 7
k8 = K_f(Ac1C,E,Ic1C,L8); % Elemento 8
k9 = K_f(Ac1C,E,Ic1C,L9); % Elemento 9
k10 = K_f(Ac1B,E,Ic1B,L10); % Elemento 10

```

```

k11 = K_f(Ac1B,E,Ic1B,L11); % Elemento 11
k12 = K_f(Ac1B,E,Ic1B,L12); % Elemento 12
k13 = K_f(Ac1A,E,Ic1A,L13); % Elemento 13
k14 = K_f(Ac1A,E,Ic1A,L14); % Elemento 14
k15 = K_f(Ac1A,E,Ic1A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20
K21 = bet_21'*k21*bet_21; % Elemento 21
K22 = bet_22'*k22*bet_22; % Elemento 22
K23 = bet_23'*k23*bet_23; % Elemento 23
K24 = bet_24'*k24*bet_24; % Elemento 24
K25 = bet_25'*k25*bet_25; % Elemento 25
K26 = bet_26'*k26*bet_26; % Elemento 26
K27 = bet_27'*k27*bet_27; % Elemento 27

%% Ensamblar matriz global KK para un pórtico

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

```

```

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

```

```

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

```

```

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

```

```

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

```

```

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

```

```

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

```

```

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

```

```
%%
```

```
% Eliminar GDL conocidos o cero
```

```
% 'Matriz de rigidez condensada:'
```

```
    % gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
```

```
    % gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
```

```
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55  
57 58 60];
```

```
Kred = KK(gdl,gdl);
```

```
% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y  
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
```

```
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
```

```
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
```

```
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
```

```
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
```

```
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
```

```
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);
```

```
% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
```

```
% u46, u52, u55, u58
```

```
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26  
28 30]);
```

```
% Acomodar los GDL horizontales en las 1ras filas/columnas
```

```

Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.yy1 = K_con;

%% Caracteristicas del PORTICO 2 sentido "y"

%Columnas 2A, 2B, 2C, 2D, 2E
Lc = 3; %Longitud de columna [m]

%Columna 2A
%bc2A ; %Ancho columna 2A [m]
%hc2A ; %Largo columna 2A [m]
Ac2A = bc2A*hc2A; %Area de columna 2A [m^2]
Ic2A = 0.8*bc2A*hc2A^3/12; %Inercia columna 2A [m^4]

%Columna 2B
% bc2B = 0.45; %Ancho columna 2B [m]
% hc2B = 0.45; %Largo columna 2B [m]
Ac2B = bc2B*hc2B; %Area de columna 2B [m^2]
Ic2B = 0.8*bc2B*hc2B^3/12; %Inercia columna 2B [m^4]

%Columna 2C
% bc2C = 0.45; %Ancho columna 2C [m]
% hc2C = 0.45; %Largo columna 2C [m]
Ac2C = bc2C*hc2C; %Area de columna 2C [m^2]
Ic2C = 0.8*bc2C*hc2C^3/12; %Inercia columna 2C [m^4]

%Columna 2D
% bc2D = 0.45; %Ancho columna 2D [m]
% hc2D = 0.45; %Largo columna 2D [m]
Ac2D = bc2D*hc2D; %Area de columna 2D [m^2]
Ic2D = 0.8*bc2D*hc2D^3/12; %Inercia columna 2D [m^4]

%Columna 2E
% bc2E = 0.45; %Ancho columna 2E [m]
% hc2E = 0.45; %Largo columna 2E [m]
Ac2E = bc2E*hc2E; %Area de columna 2E [m^2]
Ic2E = 0.8*bc2E*hc2E^3/12; %Inercia columna 2E [m^4]

%Viga
Lvy = 7; %Longitud de viga [m]
hv1 = 0.45; %Peralte viga tipo 1 [m]
bv1 = 0.25; %Ancho viga tipo 1 [m]
Av = bv1*hv1; %Area de viga
Iv = 0.5*bv1*hv1^3/12; %Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "y"

```

```

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [0 Lvy 0];
nod(6,:) = [0 Lvy Lc];
nod(7,:) = [0 Lvy 2*Lc];
nod(8,:) = [0 Lvy 3*Lc];

nod(9,:) = [0 2*Lvy 0];
nod(10,:) = [0 2*Lvy Lc];
nod(11,:) = [0 2*Lvy 2*Lc];
nod(12,:) = [0 2*Lvy 3*Lc];

nod(13,:) = [0 3*Lvy 0];
nod(14,:) = [0 3*Lvy Lc];
nod(15,:) = [0 3*Lvy 2*Lc];
nod(16,:) = [0 3*Lvy 3*Lc];

nod(17,:) = [0 4*Lvy 0];
nod(18,:) = [0 4*Lvy Lc];
nod(19,:) = [0 4*Lvy 2*Lc];
nod(20,:) = [0 4*Lvy 3*Lc];

con = [1 2
2 3
3 4
5 6
6 7
7 8
9 10
10 11
11 12
13 14
14 15
15 16
17 18
18 19
19 20
2 6
6 10
10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20];

%% Matriz de transformacion para cada elemento

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:,[2 3]), con(1,:));

```

```

% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:,[2 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:,[2 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:,[2 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[2 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:,[2 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[2 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[2 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[2 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[2 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[2 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[2 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[2 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[2 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[2 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[2 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[2 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[2 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[2 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[2 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[2 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[2 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[2 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[2 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[2 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[2 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[2 3]), con(27,:));

% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac2E,E,Ic2E,L1); % Elemento 1

```

```

k2 = K_f(Ac2E,E,Ic2E,L2); % Elemento 2
k3 = K_f(Ac2E,E,Ic2E,L3); % Elemento 3
k4 = K_f(Ac2D,E,Ic2D,L4); % Elemento 4
k5 = K_f(Ac2D,E,Ic2D,L5); % Elemento 5
k6 = K_f(Ac2D,E,Ic2D,L6); % Elemento 6
k7 = K_f(Ac2C,E,Ic2C,L7); % Elemento 7
k8 = K_f(Ac2C,E,Ic2C,L8); % Elemento 8
k9 = K_f(Ac2C,E,Ic2C,L9); % Elemento 9
k10 = K_f(Ac2B,E,Ic2B,L10); % Elemento 10
k11 = K_f(Ac2B,E,Ic2B,L11); % Elemento 11
k12 = K_f(Ac2B,E,Ic2B,L12); % Elemento 12
k13 = K_f(Ac2A,E,Ic2A,L13); % Elemento 13
k14 = K_f(Ac2A,E,Ic2A,L14); % Elemento 14
k15 = K_f(Ac2A,E,Ic2A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20
K21 = bet_21'*k21*bet_21; % Elemento 21
K22 = bet_22'*k22*bet_22; % Elemento 22
K23 = bet_23'*k23*bet_23; % Elemento 23
K24 = bet_24'*k24*bet_24; % Elemento 24
K25 = bet_25'*k25*bet_25; % Elemento 25
K26 = bet_26'*k26*bet_26; % Elemento 26
K27 = bet_27'*k27*bet_27; % Elemento 27

```

```
%% Ensamblar matriz global KK para un pórtico
```

```
dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

%% Eliminar GDL conocidos o cero

% 'Matriz de rigidez condensada:'
% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1,:)+Kred(7,:)+Kred(13,:)+Kred(19,:)+Kred(25,:);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,:)+Kred(9,:)+Kred(15,:)+Kred(21,:)+Kred(27,:);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,:)+Kred(11,:)+Kred(17,:)+Kred(23,:)+Kred(29,:);
```

```

Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
% u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.yy2 = K_con;

%% Caracteristicas de PORTICO 3 en sentido "y"

%Columnas 3A, 3B, 3C, 3D, 3E
Lc = 3; %Longitud de columna [m]

%Columna 3A
%bc3A ; %Ancho columna 3A [m]
%hc3A ; %Largo columna 3A [m]
Ac3A = bc3A*hc3A; %Area de columna 3A [m^2]
Ic3A = 0.8*bc3A*hc3A^3/12; %Inercia columna 3A [m^4]

%Columna 3B
% bc3B = 0.45; %Ancho columna 3B [m]
% hc3B = 0.45; %Largo columna 3B [m]
Ac3B = bc3B*hc3B; %Area de columna 3B [m^2]
Ic3B = 0.8*bc3B*hc3B^3/12; %Inercia columna 3B [m^4]

%Columna 3C
% bc3C = 0.45; %Ancho columna 3C [m]
% hc3C = 0.45; %Largo columna 3C [m]
Ac3C = bc3C*hc3C; %Area de columna 3C [m^2]
Ic3C = 0.8*bc3C*hc3C^3/12; %Inercia columna 3C [m^4]

%Columna 3D
% bc3D = 0.45; %Ancho columna 3D [m]
% hc3D = 0.45; %Largo columna 3D [m]
Ac3D = bc3D*hc3D; %Area de columna 3D [m^2]
Ic3D = 0.8*bc3D*hc3D^3/12; %Inercia columna 3D [m^4]

%Columna 3E
% bc3E = 0.45; %Ancho columna 3E [m]
% hc3E = 0.45; %Largo columna 3E [m]
Ac3E = bc3E*hc3E; %Area de columna 3E [m^2]
Ic3E = 0.8*bc3E*hc3E^3/12; %Inercia columna 3E [m^4]

%Viga
Lvy = 7; %Longitud de viga [m]

```

```

hv1 = 0.45;                                %Peralte viga tipo 1 [m]
bv1 = 0.25;                                %Ancho viga tipo 1 [m]
Av = bv1*hv1;                              %Area de viga
Iv = 0.5*bv1*hv1^3/12;                    %Inercia viga [m^4]

%% Coordinadas de los nodos de portico en "y"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [0 Lvy 0];
nod(6,:) = [0 Lvy Lc];
nod(7,:) = [0 Lvy 2*Lc];
nod(8,:) = [0 Lvy 3*Lc];

nod(9,:) = [0 2*Lvy 0];
nod(10,:) = [0 2*Lvy Lc];
nod(11,:) = [0 2*Lvy 2*Lc];
nod(12,:) = [0 2*Lvy 3*Lc];

nod(13,:) = [0 3*Lvy 0];
nod(14,:) = [0 3*Lvy Lc];
nod(15,:) = [0 3*Lvy 2*Lc];
nod(16,:) = [0 3*Lvy 3*Lc];

nod(17,:) = [0 4*Lvy 0];
nod(18,:) = [0 4*Lvy Lc];
nod(19,:) = [0 4*Lvy 2*Lc];
nod(20,:) = [0 4*Lvy 3*Lc];

con = [1 2
       2 3
       3 4
       5 6
       6 7
       7 8
       9 10
       10 11
       11 12
       13 14
       14 15
       15 16
       17 18
       18 19
       19 20
       2 6
       6 10
       10 14
       14 18
       3 7
       7 11
       11 15
       15 19
       4 8
       8 12

```

```

12 16
16 20];

%% Matriz de transformacion para cada elemento

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:,[2 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:,[2 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:,[2 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:,[2 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[2 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:,[2 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[2 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[2 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[2 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[2 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[2 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[2 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[2 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[2 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[2 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[2 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[2 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[2 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:,[2 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[2 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[2 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[2 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[2 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[2 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[2 3]), con(25,:));
% Elemento 26 (nodo 12-16)

```

```

[bet_26, L26] = mat_CS(nod(:,[2 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[2 3]), con(27,:));

%% Matriz local  $K_L^{\left(e\right)}$  de cada elemento

k1 = K_f(Ac3E,E,Ic3E,L1); % Elemento 1
k2 = K_f(Ac3E,E,Ic3E,L2); % Elemento 2
k3 = K_f(Ac3E,E,Ic3E,L3); % Elemento 3
k4 = K_f(Ac3D,E,Ic3D,L4); % Elemento 4
k5 = K_f(Ac3D,E,Ic3D,L5); % Elemento 5
k6 = K_f(Ac3D,E,Ic3D,L6); % Elemento 6
k7 = K_f(Ac3C,E,Ic3C,L7); % Elemento 7
k8 = K_f(Ac3C,E,Ic3C,L8); % Elemento 8
k9 = K_f(Ac3C,E,Ic3C,L9); % Elemento 9
k10 = K_f(Ac3B,E,Ic3B,L10); % Elemento 10
k11 = K_f(Ac3B,E,Ic3B,L11); % Elemento 11
k12 = K_f(Ac3B,E,Ic3B,L12); % Elemento 12
k13 = K_f(Ac3A,E,Ic3A,L13); % Elemento 13
k14 = K_f(Ac3A,E,Ic3A,L14); % Elemento 14
k15 = K_f(Ac3A,E,Ic3A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global  $K_g^{\left(e\right)}$  de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12
K13 = bet_13'*k13*bet_13; % Elemento 13
K14 = bet_14'*k14*bet_14; % Elemento 14
K15 = bet_15'*k15*bet_15; % Elemento 15
K16 = bet_16'*k16*bet_16; % Elemento 16
K17 = bet_17'*k17*bet_17; % Elemento 17
K18 = bet_18'*k18*bet_18; % Elemento 18
K19 = bet_19'*k19*bet_19; % Elemento 19
K20 = bet_20'*k20*bet_20; % Elemento 20

```

```

K21 = bet_21'*k21*bet_21;    % Elemento 21
K22 = bet_22'*k22*bet_22;    % Elemento 22
K23 = bet_23'*k23*bet_23;    % Elemento 23
K24 = bet_24'*k24*bet_24;    % Elemento 24
K25 = bet_25'*k25*bet_25;    % Elemento 25
K26 = bet_26'*k26*bet_26;    % Elemento 26
K27 = bet_27'*k27*bet_27;    % Elemento 27

```

```
%% Ensamblar matriz global KK para un pórtico
```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6)    = K1;
KK(4:9,4:9)    = KK(4:9,4:9) + K2;
KK(7:12,7:12)  = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

```

```

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

```

```

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

```

```

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

```

```
%% Eliminar GDL conocidos o cero
```

```
% 'Matriz de rigidez condensada:'
```

```

% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

```

```

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
% u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.yy3 = K_con;

%% Caracteristicas de PORTICO 4 en sentido "y"

%Columnas 4A, 4B, 4C, 4D, 4E
Lc = 3; %Longitud de columna [m]

%Columna 4A
%bc4A ; %Ancho columna 4A [m]
%hc4A ; %Largo columna 4A [m]
Ac4A = bc4A*hc4A; %Area de columna 4A [m^2]
Ic4A = 0.8*bc4A*hc4A^3/12; %Inercia columna 4A [m^4]

%Columna 4B
% bc4B = 0.45; %Ancho columna 4B [m]
% hc4B = 0.45; %Largo columna 4B [m]
Ac4B = bc4B*hc4B; %Area de columna 4B [m^2]
Ic4B = 0.8*bc4B*hc4B^3/12; %Inercia columna 4B [m^4]

%Columna 4C
% bc4C = 0.45; %Ancho columna 4C [m]
% hc4C = 0.45; %Largo columna 4C [m]
Ac4C = bc4C*hc4C; %Area de columna 4C [m^2]
Ic4C = 0.8*bc4C*hc4C^3/12; %Inercia columna 4C [m^4]

%Columna 4D
% bc4D = 0.45; %Ancho columna 4D [m]
% hc4D = 0.45; %Largo columna 4D [m]
Ac4D = bc4D*hc4D; %Area de columna 4D [m^2]
Ic4D = 0.8*bc4D*hc4D^3/12; %Inercia columna 4D [m^4]

```

```

%Columna 4E
% bc4E = 0.45;
% hc4E = 0.45;
Ac4E = bc4E*hc4E;
Ic4E = 0.8*bc4E*hc4E^3/12;

%Ancho columna 4E [m]
%Largo columna 4E [m]
%Area de columna 4E [m^2]
%Inercia columna 4E [m^4]

%Viga
Lvy = 7;
hv1 = 0.45;
bv1 = 0.25;
Av = bv1*hv1;
Iv = 0.5*bv1*hv1^3/12;

%Longitud de viga [m]
%Peralte viga tipo 1 [m]
%Ancho viga tipo 1 [m]
%Area de viga
%Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "y"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [0 Lvy 0];
nod(6,:) = [0 Lvy Lc];
nod(7,:) = [0 Lvy 2*Lc];
nod(8,:) = [0 Lvy 3*Lc];

nod(9,:) = [0 2*Lvy 0];
nod(10,:) = [0 2*Lvy Lc];
nod(11,:) = [0 2*Lvy 2*Lc];
nod(12,:) = [0 2*Lvy 3*Lc];

nod(13,:) = [0 3*Lvy 0];
nod(14,:) = [0 3*Lvy Lc];
nod(15,:) = [0 3*Lvy 2*Lc];
nod(16,:) = [0 3*Lvy 3*Lc];

nod(17,:) = [0 4*Lvy 0];
nod(18,:) = [0 4*Lvy Lc];
nod(19,:) = [0 4*Lvy 2*Lc];
nod(20,:) = [0 4*Lvy 3*Lc];

con = [1 2
2 3
3 4
5 6
6 7
7 8
9 10
10 11
11 12
13 14
14 15
15 16
17 18
18 19
19 20
2 6
6 10

```

```

10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20];

%% Matriz de transformacion para cada elemento

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:, [2 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:, [2 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:, [2 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:, [2 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:, [2 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:, [2 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:, [2 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:, [2 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:, [2 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:, [2 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:, [2 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:, [2 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:, [2 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:, [2 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:, [2 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:, [2 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:, [2 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:, [2 3]), con(18,:));
% Elemento 19 (nodo 14-18)
[bet_19, L19] = mat_CS(nod(:, [2 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:, [2 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:, [2 3]), con(21,:));
% Elemento 22 (nodo 11-15)

```

```

[bet_22, L22] = mat_CS(nod(:,[2 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[2 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[2 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[2 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[2 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[2 3]), con(27,:));

%% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac4E,E,Ic4E,L1); % Elemento 1
k2 = K_f(Ac4E,E,Ic4E,L2); % Elemento 2
k3 = K_f(Ac4E,E,Ic4E,L3); % Elemento 3
k4 = K_f(Ac4D,E,Ic4D,L4); % Elemento 4
k5 = K_f(Ac4D,E,Ic4D,L5); % Elemento 5
k6 = K_f(Ac4D,E,Ic4D,L6); % Elemento 6
k7 = K_f(Ac4C,E,Ic4C,L7); % Elemento 7
k8 = K_f(Ac4C,E,Ic4C,L8); % Elemento 8
k9 = K_f(Ac4C,E,Ic4C,L9); % Elemento 9
k10 = K_f(Ac4B,E,Ic4B,L10); % Elemento 10
k11 = K_f(Ac4B,E,Ic4B,L11); % Elemento 11
k12 = K_f(Ac4B,E,Ic4B,L12); % Elemento 12
k13 = K_f(Ac4A,E,Ic4A,L13); % Elemento 13
k14 = K_f(Ac4A,E,Ic4A,L14); % Elemento 14
k15 = K_f(Ac4A,E,Ic4A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6
K7 = bet_7'*k7*bet_7; % Elemento 7
K8 = bet_8'*k8*bet_8; % Elemento 8
K9 = bet_9'*k9*bet_9; % Elemento 9
K10 = bet_10'*k10*bet_10; % Elemento 10
K11 = bet_11'*k11*bet_11; % Elemento 11
K12 = bet_12'*k12*bet_12; % Elemento 12

```

```

K13 = bet_13'*k13*bet_13;    % Elemento 13
K14 = bet_14'*k14*bet_14;    % Elemento 14
K15 = bet_15'*k15*bet_15;    % Elemento 15
K16 = bet_16'*k16*bet_16;    % Elemento 16
K17 = bet_17'*k17*bet_17;    % Elemento 17
K18 = bet_18'*k18*bet_18;    % Elemento 18
K19 = bet_19'*k19*bet_19;    % Elemento 19
K20 = bet_20'*k20*bet_20;    % Elemento 20
K21 = bet_21'*k21*bet_21;    % Elemento 21
K22 = bet_22'*k22*bet_22;    % Elemento 22
K23 = bet_23'*k23*bet_23;    % Elemento 23
K24 = bet_24'*k24*bet_24;    % Elemento 24
K25 = bet_25'*k25*bet_25;    % Elemento 25
K26 = bet_26'*k26*bet_26;    % Elemento 26
K27 = bet_27'*k27*bet_27;    % Elemento 27

```

```

%% Ensamblar matriz global KK para un pórtico

```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6)      = K1;
KK(4:9,4:9)      = KK(4:9,4:9) + K2;
KK(7:12,7:12)    = KK(7:12,7:12) + K3;

KK(13:18,13:18)  = KK(13:18,13:18) + K4;
KK(16:21,16:21)  = KK(16:21,16:21) + K5;
KK(19:24,19:24)  = KK(19:24,19:24) + K6;

KK(25:30,25:30)  = KK(25:30,25:30) + K7;
KK(28:33,28:33)  = KK(28:33,28:33) + K8;
KK(31:36,31:36)  = KK(31:36,31:36) + K9;

KK(37:42,37:42)  = KK(37:42,37:42) + K10;
KK(40:45,40:45)  = KK(40:45,40:45) + K11;
KK(43:48,43:48)  = KK(43:48,43:48) + K12;

KK(49:54,49:54)  = KK(49:54,49:54) + K13;
KK(52:57,52:57)  = KK(52:57,52:57) + K14;
KK(55:60,55:60)  = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

```

```

%% Eliminar GDL conocidos o cero

```

```

% 'Matriz de rigidez condensada:'
% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1,)+Kred(7,)+Kred(13,)+Kred(19,)+Kred(25,);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3,)+Kred(9,)+Kred(15,)+Kred(21,)+Kred(27,);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5,)+Kred(11,)+Kred(17,)+Kred(23,)+Kred(29,);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.yy4 = K_con;

%% Caracteristicas de PORTICO 5 en sentido "y"

%Columnas 5A, 5B, 5C, 5D, 5E
Lc = 3; %Longitud de columna [m]

%Columna 5A
%bc5A ; %Ancho columna 5A [m]
%hc5A ; %Largo columna 5A [m]
Ac5A = bc5A*hc5A; %Area de columna 5A [m^2]
Ic5A = 0.8*bc5A*hc5A^3/12; %Inercia columna 5A [m^4]

%Columna 5B
% bc5B = 0.45; %Ancho columna 5B [m]
% hc5B = 0.45; %Largo columna 5B [m]
Ac5B = bc5B*hc5B; %Area de columna 5B [m^2]
Ic5B = 0.8*bc5B*hc5B^3/12; %Inercia columna 5B [m^4]

%Columna 5C
% bc5C = 0.45; %Ancho columna 5C [m]
% hc5C = 0.45; %Largo columna 5C [m]
Ac5C = bc5C*hc5C; %Area de columna 5C [m^2]
Ic5C = 0.8*bc5C*hc5C^3/12; %Inercia columna 5C [m^4]

```

```

%Columna 5D
% bc5D = 0.45;
% hc5D = 0.45;
Ac5D = bc5D*hc5D;
Ic5D = 0.8*bc5D*hc5D^3/12;

%Ancho columna 5D [m]
%Largo columna 5D [m]
%Area de columna 5D [m^2]
%Inercia columna 5D [m^4]

%Columna 5E
% bc5E = 0.45;
% hc5E = 0.45;
Ac5E = bc5E*hc5E;
Ic5E = 0.8*bc5E*hc5E^3/12;

%Ancho columna 5E [m]
%Largo columna 5E [m]
%Area de columna 5E [m^2]
%Inercia columna 5E [m^4]

%Viga
Lvy = 7;
hv1 = 0.45;
bv1 = 0.25;
Av = bv1*hv1;
Iv = 0.5*bv1*hv1^3/12;

%Longitud de viga [m]
%Peralte viga tipo 1 [m]
%Ancho viga tipo 1 [m]
%Area de viga
%Inercia viga [m^4]

%% Coordenadas de los nodos de portico en "y"

nod(1,:) = [0 0 0];
nod(2,:) = [0 0 Lc];
nod(3,:) = [0 0 2*Lc];
nod(4,:) = [0 0 3*Lc];

nod(5,:) = [0 Lvy 0];
nod(6,:) = [0 Lvy Lc];
nod(7,:) = [0 Lvy 2*Lc];
nod(8,:) = [0 Lvy 3*Lc];

nod(9,:) = [0 2*Lvy 0];
nod(10,:) = [0 2*Lvy Lc];
nod(11,:) = [0 2*Lvy 2*Lc];
nod(12,:) = [0 2*Lvy 3*Lc];

nod(13,:) = [0 3*Lvy 0];
nod(14,:) = [0 3*Lvy Lc];
nod(15,:) = [0 3*Lvy 2*Lc];
nod(16,:) = [0 3*Lvy 3*Lc];

nod(17,:) = [0 4*Lvy 0];
nod(18,:) = [0 4*Lvy Lc];
nod(19,:) = [0 4*Lvy 2*Lc];
nod(20,:) = [0 4*Lvy 3*Lc];

con = [1 2
2 3
3 4
5 6
6 7
7 8
9 10
10 11
11 12
13 14
14 15]

```

```

15 16
17 18
18 19
19 20
2 6
6 10
10 14
14 18
3 7
7 11
11 15
15 19
4 8
8 12
12 16
16 20];

```

```
%% Matriz de transformacion para cada elemento
```

```

% Elemento 1 (nodo 1-2)
[bet_1, L1] = mat_CS(nod(:,[2 3]), con(1,:));
% Elemento 2 (nodo 2-3)
[bet_2, L2] = mat_CS(nod(:,[2 3]), con(2,:));
% Elemento 3 (nodo 3-4)
[bet_3, L3] = mat_CS(nod(:,[2 3]), con(3,:));
% Elemento 4 (nodo 5-6)
[bet_4, L4] = mat_CS(nod(:,[2 3]), con(4,:));
% Elemento 5 (nodo 6-7)
[bet_5, L5] = mat_CS(nod(:,[2 3]), con(5,:));
% Elemento 6 (nodo 7-8)
[bet_6, L6] = mat_CS(nod(:,[2 3]), con(6,:));
% Elemento 7 (nodo 9-10)
[bet_7, L7] = mat_CS(nod(:,[2 3]), con(7,:));
% Elemento 8 (nodo 10-11)
[bet_8, L8] = mat_CS(nod(:,[2 3]), con(8,:));
% Elemento 9 (nodo 11-12)
[bet_9, L9] = mat_CS(nod(:,[2 3]), con(9,:));
% Elemento 10 (nodo 13-14)
[bet_10, L10] = mat_CS(nod(:,[2 3]), con(10,:));
% Elemento 11 (nodo 14-15)
[bet_11, L11] = mat_CS(nod(:,[2 3]), con(11,:));
% Elemento 12 (nodo 15-16)
[bet_12, L12] = mat_CS(nod(:,[2 3]), con(12,:));
% Elemento 13 (nodo 17-18)
[bet_13, L13] = mat_CS(nod(:,[2 3]), con(13,:));
% Elemento 14 (nodo 18-19)
[bet_14, L14] = mat_CS(nod(:,[2 3]), con(14,:));
% Elemento 15 (nodo 19-20)
[bet_15, L15] = mat_CS(nod(:,[2 3]), con(15,:));

% Elemento 16 (nodo 2-6)
[bet_16, L16] = mat_CS(nod(:,[2 3]), con(16,:));
% Elemento 17 (nodo 6-10)
[bet_17, L17] = mat_CS(nod(:,[2 3]), con(17,:));
% Elemento 18 (nodo 10-14)
[bet_18, L18] = mat_CS(nod(:,[2 3]), con(18,:));
% Elemento 19 (nodo 14-18)

```

```

[bet_19, L19] = mat_CS(nod(:,[2 3]), con(19,:));
% Elemento 20 (nodo 3-7)
[bet_20, L20] = mat_CS(nod(:,[2 3]), con(20,:));
% Elemento 21 (nodo 7-11)
[bet_21, L21] = mat_CS(nod(:,[2 3]), con(21,:));
% Elemento 22 (nodo 11-15)
[bet_22, L22] = mat_CS(nod(:,[2 3]), con(22,:));
% Elemento 23 (nodo 15-19)
[bet_23, L23] = mat_CS(nod(:,[2 3]), con(23,:));
% Elemento 24 (nodo 4-8)
[bet_24, L24] = mat_CS(nod(:,[2 3]), con(24,:));
% Elemento 25 (nodo 8-12)
[bet_25, L25] = mat_CS(nod(:,[2 3]), con(25,:));
% Elemento 26 (nodo 12-16)
[bet_26, L26] = mat_CS(nod(:,[2 3]), con(26,:));
% Elemento 27 (nodo 16-20)
[bet_27, L27] = mat_CS(nod(:,[2 3]), con(27,:));

%% Matriz local $K_L^{\left(e\right)}$ de cada elemento

k1 = K_f(Ac5E,E,Ic5E,L1); % Elemento 1
k2 = K_f(Ac5E,E,Ic5E,L2); % Elemento 2
k3 = K_f(Ac5E,E,Ic5E,L3); % Elemento 3
k4 = K_f(Ac5D,E,Ic5D,L4); % Elemento 4
k5 = K_f(Ac5D,E,Ic5D,L5); % Elemento 5
k6 = K_f(Ac5D,E,Ic5D,L6); % Elemento 6
k7 = K_f(Ac5C,E,Ic5C,L7); % Elemento 7
k8 = K_f(Ac5C,E,Ic5C,L8); % Elemento 8
k9 = K_f(Ac5C,E,Ic5C,L9); % Elemento 9
k10 = K_f(Ac5B,E,Ic5B,L10); % Elemento 10
k11 = K_f(Ac5B,E,Ic5B,L11); % Elemento 11
k12 = K_f(Ac5B,E,Ic5B,L12); % Elemento 12
k13 = K_f(Ac5A,E,Ic5A,L13); % Elemento 13
k14 = K_f(Ac5A,E,Ic5A,L14); % Elemento 14
k15 = K_f(Ac5A,E,Ic5A,L15); % Elemento 15

k16 = K_f(Av,E,Iv,L16); % Elemento 16
k17 = K_f(Av,E,Iv,L17); % Elemento 17
k18 = K_f(Av,E,Iv,L18); % Elemento 18
k19 = K_f(Av,E,Iv,L19); % Elemento 19
k20 = K_f(Av,E,Iv,L20); % Elemento 20
k21 = K_f(Av,E,Iv,L21); % Elemento 21
k22 = K_f(Av,E,Iv,L22); % Elemento 22
k23 = K_f(Av,E,Iv,L23); % Elemento 23
k24 = K_f(Av,E,Iv,L24); % Elemento 24
k25 = K_f(Av,E,Iv,L25); % Elemento 25
k26 = K_f(Av,E,Iv,L26); % Elemento 26
k27 = K_f(Av,E,Iv,L27); % Elemento 27

%% Matriz global $K_g^{\left(e\right)}$ de cada elemento

K1 = bet_1'*k1*bet_1; % Elemento 1
K2 = bet_2'*k2*bet_2; % Elemento 2
K3 = bet_3'*k3*bet_3; % Elemento 3
K4 = bet_4'*k4*bet_4; % Elemento 4
K5 = bet_5'*k5*bet_5; % Elemento 5
K6 = bet_6'*k6*bet_6; % Elemento 6

```

```

K7 = bet_7'*k7*bet_7;    % Elemento 7
K8 = bet_8'*k8*bet_8;    % Elemento 8
K9 = bet_9'*k9*bet_9;    % Elemento 9
K10 = bet_10'*k10*bet_10;    % Elemento 10
K11 = bet_11'*k11*bet_11;    % Elemento 11
K12 = bet_12'*k12*bet_12;    % Elemento 12
K13 = bet_13'*k13*bet_13;    % Elemento 13
K14 = bet_14'*k14*bet_14;    % Elemento 14
K15 = bet_15'*k15*bet_15;    % Elemento 15
K16 = bet_16'*k16*bet_16;    % Elemento 16
K17 = bet_17'*k17*bet_17;    % Elemento 17
K18 = bet_18'*k18*bet_18;    % Elemento 18
K19 = bet_19'*k19*bet_19;    % Elemento 19
K20 = bet_20'*k20*bet_20;    % Elemento 20
K21 = bet_21'*k21*bet_21;    % Elemento 21
K22 = bet_22'*k22*bet_22;    % Elemento 22
K23 = bet_23'*k23*bet_23;    % Elemento 23
K24 = bet_24'*k24*bet_24;    % Elemento 24
K25 = bet_25'*k25*bet_25;    % Elemento 25
K26 = bet_26'*k26*bet_26;    % Elemento 26
K27 = bet_27'*k27*bet_27;    % Elemento 27

```

```

%% Ensamblar matriz global KK para un pórtico

```

```

dof = 3*size(nod,1);
KK = zeros(dof);
KK(1:6,1:6) = K1;
KK(4:9,4:9) = KK(4:9,4:9) + K2;
KK(7:12,7:12) = KK(7:12,7:12) + K3;

KK(13:18,13:18) = KK(13:18,13:18) + K4;
KK(16:21,16:21) = KK(16:21,16:21) + K5;
KK(19:24,19:24) = KK(19:24,19:24) + K6;

KK(25:30,25:30) = KK(25:30,25:30) + K7;
KK(28:33,28:33) = KK(28:33,28:33) + K8;
KK(31:36,31:36) = KK(31:36,31:36) + K9;

KK(37:42,37:42) = KK(37:42,37:42) + K10;
KK(40:45,40:45) = KK(40:45,40:45) + K11;
KK(43:48,43:48) = KK(43:48,43:48) + K12;

KK(49:54,49:54) = KK(49:54,49:54) + K13;
KK(52:57,52:57) = KK(52:57,52:57) + K14;
KK(55:60,55:60) = KK(55:60,55:60) + K15;

KK([4:6 16:18],[4:6 16:18]) = KK([4:6 16:18],[4:6 16:18]) + K16;
KK([16:18 28:30],[16:18 28:30]) = KK([16:18 28:30],[16:18 28:30]) + K17;
KK([28:30 40:42],[28:30 40:42]) = KK([28:30 40:42],[28:30 40:42]) + K18;
KK([40:42 52:54],[40:42 52:54]) = KK([40:42 52:54],[40:42 52:54]) + K19;

KK([7:9 19:21],[7:9 19:21]) = KK([7:9 19:21],[7:9 19:21]) + K20;
KK([19:21 31:33],[19:21 31:33]) = KK([19:21 31:33],[19:21 31:33]) + K21;
KK([31:33 43:45],[31:33 43:45]) = KK([31:33 43:45],[31:33 43:45]) + K22;
KK([43:45 55:57],[43:45 55:57]) = KK([43:45 55:57],[43:45 55:57]) + K23;

```

```

KK([10:12 22:24],[10:12 22:24]) = KK([10:12 22:24],[10:12 22:24]) + K24;
KK([22:24 34:36],[22:24 34:36]) = KK([22:24 34:36],[22:24 34:36]) + K25;
KK([34:36 46:48],[34:36 46:48]) = KK([34:36 46:48],[34:36 46:48]) + K26;
KK([46:48 58:60],[46:48 58:60]) = KK([46:48 58:60],[46:48 58:60]) + K27;

%% Eliminar GDL conocidos o cero

% 'Matriz de rigidez condensada:'
% gdl.vert = [5 8 11 17 20 23 29 32 35 41 44 47 53 56 59]
% gdl.apoyo = [1:3 13:15 25:27 37:39 49:51]
gdl = [4 6 7 9 10 12 16 18 19 21 22 24 28 30 31 33 34 36 40 42 43 45 46 48 52 54 55
57 58 60];
Kred = KK(gdl,gdl);

% Sumar los GDL horizontales (u4+u16+u28+u40+u52, u7+u19+u31+u43+u55 y
u10+u22+u34+u46+u58) pero en sus nuevas posiciones
Kred(1,:) = Kred(1:)+Kred(7:)+Kred(13:)+Kred(19:)+Kred(25:);
Kred(:,1) = Kred(:,1)+Kred(:,7)+Kred(:,13)+Kred(:,19)+Kred(:,25);
Kred(3,:) = Kred(3:)+Kred(9:)+Kred(15:)+Kred(21:)+Kred(27:);
Kred(:,3) = Kred(:,3)+Kred(:,9)+Kred(:,15)+Kred(:,21)+Kred(:,27);
Kred(5,:) = Kred(5:)+Kred(11:)+Kred(17:)+Kred(23:)+Kred(29:);
Kred(:,5) = Kred(:,5)+Kred(:,11)+Kred(:,17)+Kred(:,23)+Kred(:,29);

% Eliminar los GDLs horizontales u16, u19, u22, u28, u31, u34, u40, u43,
u46, u52, u55, u58
Krn = Kred([1:6 8 10 12 14 16 18 20 22 24 26 28 30],[1:6 8 10 12 14 16 18 20 22 24 26
28 30]);

% Acomodar los GDL horizontales en las 1ras filas/columnas
Kn = Krn([1 3 5 2 4 6:18],[1 3 5 2 4 6:18]);

% Condensar grados de libertad rotacionales
kpp = Kn(1:3,1:3);
kpo = Kn(1:3,4:end);
kop = kpo.';
koo = Kn(4:end,4:end);

K_con = kpp-kpo*(koo\kop);
Kc.yy5 = K_con;

%% Transformación de coordenadas de pórticos a toda la estructura y
% Calculo de Centro de Masa

% Definir 2 puntos (xa,ya) y (xb,yb) para cada pórtico en cada piso
% Eje A (pórtico 81-85-89-93-97)
% Eje B (pórtico 61-65-69-73-77)
% Eje C (pórtico 41-45-49-53-57)
% Eje D (pórtico 21-25-29-33-37)
% Eje E (pórtico 1-5-9-13-17)
% Eje 1 (pórtico 1-21-41-61-81)
% Eje 2 (pórtico 5-25-45-65-85)
% Eje 3 (pórtico 9-29-49-69-89)
% Eje 4 (pórtico 13-33-53-73-93)
% Eje 5 (pórtico 17-37-57-77-97)

% Piso 1
M.p1 = carga.pis * Lvx * Lvy; %Masa de Paño de losa 1 (1-5-25-21)

```

M.p2 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 2 (21-25-45-41)
M.p3 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 3 (41-45-65-61)
M.p4 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 4 (61-65-85-81)
M.p5 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 5 (5-9-29-25)
M.p6 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 6 (25-29-49-45)
M.p7 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 7 (45-49-69-65)
M.p8 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 8 (65-69-89-85)
M.p9 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 9 (9-13-33-29)
M.p10 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 10 (29-33-53-49)
M.p11 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 11 (49-53-73-69)
M.p12 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 12 (69-73-93-89)
M.p13 = 0;	%Masa de Paño de losa 13 (13-17-37-33)
M.p14 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 14 (33-37-57-53)
M.p15 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 15 (53-57-77-73)
M.p16 = carga.pis * Lvx * Lvy;	%Masa de Paño de losa 16 (73-77-97-93)

dx.p1 = Lvx/2;  
dx.p2 = Lvx/2;  
dx.p3 = Lvx/2;  
dx.p4 = Lvx/2;  
dx.p5 = Lvx + (Lvx/2);  
dx.p6 = Lvx + (Lvx/2);  
dx.p7 = Lvx + (Lvx/2);  
dx.p8 = Lvx + (Lvx/2);  
dx.p9 = 2\*Lvx + (Lvx/2);  
dx.p10 = 2\*Lvx + (Lvx/2);  
dx.p11 = 2\*Lvx + (Lvx/2);  
dx.p12 = 2\*Lvx + (Lvx/2);  
dx.p13 = 3\*Lvx + (Lvx/2);  
dx.p14 = 3\*Lvx + (Lvx/2);  
dx.p15 = 3\*Lvx + (Lvx/2);  
dx.p16 = 3\*Lvx + (Lvx/2);

dy.p1 = Lvy/2;  
dy.p2 = Lvy + (Lvy/2);  
dy.p3 = 2\*Lvy + (Lvy/2);  
dy.p4 = 3\*Lvy + (Lvy/2);  
dy.p5 = Lvy/2;  
dy.p6 = Lvy + (Lvy/2);  
dy.p7 = 2\*Lvy + (Lvy/2);  
dy.p8 = 3\*Lvy + (Lvy/2);  
dy.p9 = Lvy/2;  
dy.p10 = Lvy + (Lvy/2);  
dy.p11 = 2\*Lvy + (Lvy/2);  
dy.p12 = 3\*Lvy + (Lvy/2);  
dy.p13 = Lvy/2;  
dy.p14 = Lvy + (Lvy/2);  
dy.p15 = 2\*Lvy + (Lvy/2);  
dy.p16 = 3\*Lvy + (Lvy/2);

CMx.p1 = (M.p1\*dx.p1 + M.p2\*dx.p2 + M.p3\*dx.p3 + M.p4\*dx.p4 + M.p5\*dx.p5 + M.p6\*dx.p6 + M.p7\*dx.p7 + M.p8\*dx.p8 + M.p9\*dx.p9 + M.p10\*dx.p10 + M.p11\*dx.p11 + M.p12\*dx.p12 + M.p13\*dx.p13 + M.p14\*dx.p14 + M.p15\*dx.p15 + M.p16\*dx.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M.p14+M.p15+M.p16);  
CMy.p1 = (M.p1\*dy.p1 + M.p2\*dy.p2 + M.p3\*dy.p3 + M.p4\*dy.p4 + M.p5\*dy.p5 + M.p6\*dy.p6 + M.p7\*dy.p7 + M.p8\*dy.p8 + M.p9\*dy.p9 + M.p10\*dy.p10 + M.p11\*dy.p11 + M.p12\*dy.p12 +

```

M.p13*dy.p13 + M.p14*dy.p14 + M.p15*dy.p15 +
M.p16*dy.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M
.p14+M.p15+M.p16);
CM.p1 = [CMx.p1 CMy.p1];           %Centro de masa de piso 1

% coor = [xa ya xb yb]
coor = [...
0 4*Lvy 4*Lvx 4*Lvy           %Portico A
0 3*Lvy 4*Lvx 3*Lvy           %Portico B
0 2*Lvy 4*Lvx 2*Lvy           %Portico C
0 1*Lvy 4*Lvx 1*Lvy           %Portico D
0 0 4*Lvx 0                   %Portico E
0 0 0 4*Lvy                   %Portico 1
1*Lvx 0 1*Lvx 4*Lvy           %Portico 2
2*Lvx 0 2*Lvx 4*Lvy           %Portico 3
3*Lvx 0 3*Lvx 4*Lvy           %Portico 4
4*Lvx 0 4*Lvx 4*Lvy           %Portico 5
];
% Se calcula d
d = sqrt((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
C = (coor(:,3)-coor(:,1))./d;
S = (coor(:,4)-coor(:,2))./d;
% Definir vector desde c/pórtico hasta el CM de piso 1
r1 = (CM.p1(2)-coor(:,2)).*C-(CM.p1(1)-coor(:,1)).*S;
% Matriz de transformación PISO 1
mt.p1 = [C S r1]';

% Piso 2
M.p1 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 1
(1-5-25-21)
M.p2 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 2
(21-25-45-41)
M.p3 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 3
(41-45-65-61)
M.p4 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 4 (61-65-85-81)
M.p5 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 5 (5-9-29-25)
M.p6 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 6 (25-29-49-45)
M.p7 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 7 (45-49-69-65)
M.p8 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 8 (65-69-89-85)
M.p9 = carga.pis * Lvx * Lvy;           %Masa de Paño de losa 9 (9-13-33-29)
M.p10 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 10 (29-33-53-49)
M.p11 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 11 (49-53-73-69)
M.p12 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 12 (69-73-93-89)
M.p13 = 0;                               %Masa de Paño de losa 13 (13-17-37-33)
M.p14 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 14 (33-37-57-53)
M.p15 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 15 (53-57-77-73)
M.p16 = carga.pis * Lvx * Lvy;          %Masa de Paño de losa 16 (73-77-97-93)

dx.p1 = Lvx/2;
dx.p2 = Lvx/2;
dx.p3 = Lvx/2;
dx.p4 = Lvx/2;
dx.p5 = Lvx + (Lvx/2);
dx.p6 = Lvx + (Lvx/2);
dx.p7 = Lvx + (Lvx/2);
dx.p8 = Lvx + (Lvx/2);

```

```

dx.p9 = 2*Lvx + (Lvx/2);
dx.p10 = 2*Lvx + (Lvx/2);
dx.p11 = 2*Lvx + (Lvx/2);
dx.p12 = 2*Lvx + (Lvx/2);
dx.p13 = 3*Lvx + (Lvx/2);
dx.p14 = 3*Lvx + (Lvx/2);
dx.p15 = 3*Lvx + (Lvx/2);
dx.p16 = 3*Lvx + (Lvx/2);

dy.p1 = Lvy/2;
dy.p2 = Lvy + (Lvy/2);
dy.p3 = 2*Lvy + (Lvy/2);
dy.p4 = 3*Lvy + (Lvy/2);
dy.p5 = Lvy/2;
dy.p6 = Lvy + (Lvy/2);
dy.p7 = 2*Lvy + (Lvy/2);
dy.p8 = 3*Lvy + (Lvy/2);
dy.p9 = Lvy/2;
dy.p10 = Lvy + (Lvy/2);
dy.p11 = 2*Lvy + (Lvy/2);
dy.p12 = 3*Lvy + (Lvy/2);
dy.p13 = Lvy/2;
dy.p14 = Lvy + (Lvy/2);
dy.p15 = 2*Lvy + (Lvy/2);
dy.p16 = 3*Lvy + (Lvy/2);

CMx.p2 = (M.p1*dx.p1 + M.p2*dx.p2 + M.p3*dx.p3 + M.p4*dx.p4 + M.p5*dx.p5 + M.p6*dx.p6
+ M.p7*dx.p7 + M.p8*dx.p8 + M.p9*dx.p9 + M.p10*dx.p10 + M.p11*dx.p11 + M.p12*dx.p12 +
M.p13*dx.p13 + M.p14*dx.p14 + M.p15*dx.p15 +
M.p16*dx.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M
.p14+M.p15+M.p16);
CMY.p2 = (M.p1*dy.p1 + M.p2*dy.p2 + M.p3*dy.p3 + M.p4*dy.p4 + M.p5*dy.p5 + M.p6*dy.p6
+ M.p7*dy.p7 + M.p8*dy.p8 + M.p9*dy.p9 + M.p10*dy.p10 + M.p11*dy.p11 + M.p12*dy.p12 +
M.p13*dy.p13 + M.p14*dy.p14 + M.p15*dy.p15 +
M.p16*dy.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M
.p14+M.p15+M.p16);
CM.p2 = [CMx.p2 CMY.p2];           %Centro de masa de piso 2

% coor = [xa ya xb yb]
coor = [...
0 4*Lvy 4*Lvx 4*Lvy           %Portico A
0 3*Lvy 4*Lvx 3*Lvy           %Portico B
0 2*Lvy 4*Lvx 2*Lvy           %Portico C
0 1*Lvy 4*Lvx 1*Lvy           %Portico D
0 0 4*Lvx 0                   %Portico E
0 0 0 4*Lvy                   %Portico 1
1*Lvx 0 1*Lvx 4*Lvy           %Portico 2
2*Lvx 0 2*Lvx 4*Lvy           %Portico 3
3*Lvx 0 3*Lvx 4*Lvy           %Portico 4
4*Lvx 0 4*Lvx 4*Lvy           %Portico 5
];
% Se calcula d
d = sqrt(((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
C = (coor(:,3)-coor(:,1))./d;
S = (coor(:,4)-coor(:,2))./d;
% Definir vector desde c/pórtico hasta el CM de piso 2
r2 = (CM.p2(2)-coor(:,2)).*C-(CM.p2(1)-coor(:,1)).*S;

```

```
% Matriz de transformación PISO 2
```

```
mt.p2 = [C S r2]';
```

```
% Piso 3
```

```
M.p1 = carga.cub * Lvx * Lvy;
```

```
M.p2 = carga.cub * Lvx * Lvy;
```

```
M.p3 = carga.cub * Lvx * Lvy;
```

```
M.p4 = carga.cub * Lvx * Lvy;
```

```
M.p5 = carga.cub * Lvx * Lvy;
```

```
M.p6 = carga.cub * Lvx * Lvy;
```

```
M.p7 = carga.cub * Lvx * Lvy;
```

```
M.p8 = carga.cub * Lvx * Lvy;
```

```
M.p9 = carga.cub * Lvx * Lvy;
```

```
M.p10 = carga.cub * Lvx * Lvy;
```

```
M.p11 = carga.cub * Lvx * Lvy;
```

```
M.p12 = carga.cub * Lvx * Lvy;
```

```
M.p13 = carga.cub * Lvx * Lvy;
```

```
M.p14 = carga.cub * Lvx * Lvy;
```

```
M.p15 = carga.cub * Lvx * Lvy;
```

```
M.p16 = carga.cub * Lvx * Lvy;
```

```
%Masa de Paño de losa 1 (1-5-25-21)
```

```
%Masa de Paño de losa 2 (21-25-45-41)
```

```
%Masa de Paño de losa 3 (41-45-65-61)
```

```
%Masa de Paño de losa 4 (61-65-85-81)
```

```
%Masa de Paño de losa 5 (5-9-29-25)
```

```
%Masa de Paño de losa 6 (25-29-49-45)
```

```
%Masa de Paño de losa 7 (45-49-69-65)
```

```
%Masa de Paño de losa 8 (65-69-89-85)
```

```
%Masa de Paño de losa 9 (9-13-33-29)
```

```
%Masa de Paño de losa 10 (29-33-53-49)
```

```
%Masa de Paño de losa 11 (49-53-73-69)
```

```
%Masa de Paño de losa 12 (69-73-93-89)
```

```
%Masa de Paño de losa 13 (13-17-37-33)
```

```
%Masa de Paño de losa 14 (33-37-57-53)
```

```
%Masa de Paño de losa 15 (53-57-77-73)
```

```
%Masa de Paño de losa 16 (73-77-97-93)
```

```
dx.p1 = Lvx/2;
```

```
dx.p2 = Lvx/2;
```

```
dx.p3 = Lvx/2;
```

```
dx.p4 = Lvx/2;
```

```
dx.p5 = Lvx + (Lvx/2);
```

```
dx.p6 = Lvx + (Lvx/2);
```

```
dx.p7 = Lvx + (Lvx/2);
```

```
dx.p8 = Lvx + (Lvx/2);
```

```
dx.p9 = 2*Lvx + (Lvx/2);
```

```
dx.p10 = 2*Lvx + (Lvx/2);
```

```
dx.p11 = 2*Lvx + (Lvx/2);
```

```
dx.p12 = 2*Lvx + (Lvx/2);
```

```
dx.p13 = 3*Lvx + (Lvx/2);
```

```
dx.p14 = 3*Lvx + (Lvx/2);
```

```
dx.p15 = 3*Lvx + (Lvx/2);
```

```
dx.p16 = 3*Lvx + (Lvx/2);
```

```
dy.p1 = Lvy/2;
```

```
dy.p2 = Lvy + (Lvy/2);
```

```
dy.p3 = 2*Lvy + (Lvy/2);
```

```
dy.p4 = 3*Lvy + (Lvy/2);
```

```
dy.p5 = Lvy/2;
```

```
dy.p6 = Lvy + (Lvy/2);
```

```
dy.p7 = 2*Lvy + (Lvy/2);
```

```
dy.p8 = 3*Lvy + (Lvy/2);
```

```
dy.p9 = Lvy/2;
```

```
dy.p10 = Lvy + (Lvy/2);
```

```
dy.p11 = 2*Lvy + (Lvy/2);
```

```
dy.p12 = 3*Lvy + (Lvy/2);
```

```
dy.p13 = Lvy/2;
```

```
dy.p14 = Lvy + (Lvy/2);
```

```
dy.p15 = 2*Lvy + (Lvy/2);
```

```
dy.p16 = 3*Lvy + (Lvy/2);
```

```
CMx.p3 = (M.p1*dx.p1 + M.p2*dx.p2 + M.p3*dx.p3 + M.p4*dx.p4 + M.p5*dx.p5 + M.p6*dx.p6  
+ M.p7*dx.p7 + M.p8*dx.p8 + M.p9*dx.p9 + M.p10*dx.p10 + M.p11*dx.p11 + M.p12*dx.p12 +
```

```

M.p13*dx.p13 + M.p14*dx.p14 + M.p15*dx.p15 +
M.p16*dx.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M
.p14+M.p15+M.p16);
CM.y.p3 = (M.p1*dy.p1 + M.p2*dy.p2 + M.p3*dy.p3 + M.p4*dy.p4 + M.p5*dy.p5 + M.p6*dy.p6
+ M.p7*dy.p7 + M.p8*dy.p8 + M.p9*dy.p9 + M.p10*dy.p10 + M.p11*dy.p11 + M.p12*dy.p12 +
M.p13*dy.p13 + M.p14*dy.p14 + M.p15*dy.p15 +
M.p16*dy.p16)/(M.p1+M.p2+M.p3+M.p4+M.p5+M.p6+M.p7+M.p8+M.p9+M.p10+M.p11+M.p12+M.p13+M
.p14+M.p15+M.p16);

CM.p3 = [CMx.p3 CM.y.p3];           %Centro de masa de piso 3

% coor = [xa ya xb yb]
coor = [...
0 4*Lvy 4*Lvx 4*Lvy           %Portico A
0 3*Lvy 4*Lvx 3*Lvy           %Portico B
0 2*Lvy 4*Lvx 2*Lvy           %Portico C
0 1*Lvy 4*Lvx 1*Lvy           %Portico D
0 0 4*Lvx 0                   %Portico E
0 0 0 4*Lvy                   %Portico 1
1*Lvx 0 1*Lvx 4*Lvy           %Portico 2
2*Lvx 0 2*Lvx 4*Lvy           %Portico 3
3*Lvx 0 3*Lvx 4*Lvy           %Portico 4
4*Lvx 0 4*Lvx 4*Lvy           %Portico 5
];
% Se calcula d
d = sqrt((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
C = (coor(:,3)-coor(:,1))./d;
S = (coor(:,4)-coor(:,2))./d;
% Definir vector desde c/pórtico hasta el CM de piso 3
r3 = (CM.p3(2)-coor(:,2)).*C-(CM.p3(1)-coor(:,1)).*S;
% Matriz de transformación PISO 3
mt.p3 = [C S r3]';

%Vectores de Centro de masa por piso
XCM = [CM.p1(1) CM.p2(1) CM.p3(1)]'
YCM = [CM.p1(2) CM.p2(2) CM.p3(2)]'

% Las matrices de transformación para cada pórtico quedan:
T.eA = zeros(9,3);   T.eA(1:3,1) = mt.p1(:,1);   T.eA(4:6,2) = mt.p2(:,1);
T.eA(7:9,3) = mt.p3(:,1);
T.eB = zeros(9,3);   T.eB(1:3,1) = mt.p1(:,2);   T.eB(4:6,2) = mt.p2(:,2);
T.eB(7:9,3) = mt.p3(:,2);
T.eC = zeros(9,3);   T.eC(1:3,1) = mt.p1(:,3);   T.eC(4:6,2) = mt.p2(:,3);
T.eC(7:9,3) = mt.p3(:,3);
T.eD = zeros(9,3);   T.eD(1:3,1) = mt.p1(:,4);   T.eD(4:6,2) = mt.p2(:,4);
T.eD(7:9,3) = mt.p3(:,4);
T.eE = zeros(9,3);   T.eE(1:3,1) = mt.p1(:,5);   T.eE(4:6,2) = mt.p2(:,5);
T.eE(7:9,3) = mt.p3(:,5);
T.e1 = zeros(9,3);   T.e1(1:3,1) = mt.p1(:,6);   T.e1(4:6,2) = mt.p2(:,6);
T.e1(7:9,3) = mt.p3(:,6);
T.e2 = zeros(9,3);   T.e2(1:3,1) = mt.p1(:,7);   T.e2(4:6,2) = mt.p2(:,7);
T.e2(7:9,3) = mt.p3(:,7);
T.e3 = zeros(9,3);   T.e3(1:3,1) = mt.p1(:,8);   T.e3(4:6,2) = mt.p2(:,8);
T.e3(7:9,3) = mt.p3(:,8);
T.e4 = zeros(9,3);   T.e4(1:3,1) = mt.p1(:,9);   T.e4(4:6,2) = mt.p2(:,9);
T.e4(7:9,3) = mt.p3(:,9);

```

```
T.e5 = zeros(9,3); T.e5(1:3,1) = mt.p1(:,10); T.e5(4:6,2) = mt.p2(:,10);
T.e5(7:9,3) = mt.p3(:,10);
```

```
% Matriz de rigidez global para cada pórtico
```

```
Kg.eA = T.eA*Kc.xxA*T.eA'; % PÓRTICO A
Kg.eB = T.eB*Kc.xxB*T.eB'; % PÓRTICO B
Kg.eC = T.eC*Kc.xxC*T.eC'; % PÓRTICO C
Kg.eD = T.eD*Kc.xxD*T.eD'; % PÓRTICO D
Kg.eE = T.eE*Kc.xxE*T.eE'; % PÓRTICO E
Kg.e1 = T.e1*Kc.yy1*T.e1'; % PÓRTICO 1
Kg.e2 = T.e2*Kc.yy2*T.e2'; % PÓRTICO 2
Kg.e3 = T.e3*Kc.yy3*T.e3'; % PÓRTICO 3
Kg.e4 = T.e4*Kc.yy4*T.e4'; % PÓRTICO 4
Kg.e5 = T.e5*Kc.yy5*T.e5'; % PÓRTICO 5
```

```
% Matriz de rigidez de toda la estructura para los 9 GDLs.
```

```
KKg = Kg.eA + Kg.eB + Kg.eC + Kg.eD + Kg.eE + Kg.e1 + Kg.e2 + Kg.e3 + Kg.e4 + Kg.e5;
issymmetric(round(KKg,2));
```

```
%% Cálculo de Centro de Rigidez. Método 1
```

```
format bank
YCR_final = zeros(3,1);
j=1;
for i=1:3:7
    Fx = zeros(9,1);
    Fx(i) = 10000000;
    Rzx = KKg\Fx;

    Mzx = zeros(9,1);
    Mzx(i+2) = 10000000;
    Rzz = KKg\Mzx;
    YCR_metodo1 = Rzx./Rzz;

    YCR_final(j) = YCM(j) + YCR_metodo1(i+2);
    j=j+1;
end
YCR_final

XCR_final = zeros(3,1);
j=1;
for i=2:3:8
    Fy = zeros(9,1);
    Fy(i) = 10000000;
    Rzy = KKg\Fy;

    Mzy = zeros(9,1);
    Mzy(i+1) = 10000000;
    Rzz = KKg\Mzy;
    XCR_metodo1 = -Rzy./Rzz;

    XCR_final(j) = XCM(j) + XCR_metodo1(i+1);
    j=j+1;
end
```

```
XCR_final
```

```
%% Transformación de coordenadas de pórticos a toda la estructura con respecto al CR
```

```
% Definir 2 puntos (xa,ya) y (xb,yb) para cada pórtico en cada piso
```

```
% Eje A (pórtico 81-85-89-93-97)
```

```
% Eje B (pórtico 61-65-69-73-77)
```

```
% Eje C (pórtico 41-45-49-53-57)
```

```
% Eje D (pórtico 21-25-29-33-37)
```

```
% Eje E (pórtico 1-5-9-13-17)
```

```
% Eje 1 (pórtico 1-21-41-61-81)
```

```
% Eje 2 (pórtico 5-25-45-65-85)
```

```
% Eje 3 (pórtico 9-29-49-69-89)
```

```
% Eje 4 (pórtico 13-33-53-73-93)
```

```
% Eje 5 (pórtico 17-37-57-77-97)
```

```
%Piso 1
```

```
CR.p1 = [XCR_final(1) YCR_final(1)];
```

```
%Centro de rigidez de piso 1 POR
```

```
METODO 1
```

```
% coor = [xa ya xb yb]
```

```
coor = [...
```

```
0 4*Lvy 4*Lvx 4*Lvy %Portico A
```

```
0 3*Lvy 4*Lvx 3*Lvy %Portico B
```

```
0 2*Lvy 4*Lvx 2*Lvy %Portico C
```

```
0 1*Lvy 4*Lvx 1*Lvy %Portico D
```

```
0 0 4*Lvx 0 %Portico E
```

```
0 0 0 4*Lvy %Portico 1
```

```
1*Lvx 0 1*Lvx 4*Lvy %Portico 2
```

```
2*Lvx 0 2*Lvx 4*Lvy %Portico 3
```

```
3*Lvx 0 3*Lvx 4*Lvy %Portico 4
```

```
4*Lvx 0 4*Lvx 4*Lvy %Portico 5
```

```
];
```

```
% Se calcula d
```

```
d = sqrt((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
```

```
C = (coor(:,3)-coor(:,1))./d;
```

```
S = (coor(:,4)-coor(:,2))./d;
```

```
% Definir vector desde c/pórtico hasta el CM de piso 1
```

```
r1 = (CR.p1(2)-coor(:,2)).*C-(CR.p1(1)-coor(:,1)).*S;
```

```
% Matriz de transformación PISO 1
```

```
mt.p1 = [C S r1]';
```

```
% Piso 2
```

```
CR.p2 = [XCR_final(2) YCR_final(2)];
```

```
%Centro de rigidez de piso 2 POR
```

```
METODO 1
```

```
% coor = [xa ya xb yb]
```

```
coor = [...
```

```
0 4*Lvy 4*Lvx 4*Lvy %Portico A
```

```
0 3*Lvy 4*Lvx 3*Lvy %Portico B
```

```
0 2*Lvy 4*Lvx 2*Lvy %Portico C
```

```
0 1*Lvy 4*Lvx 1*Lvy %Portico D
```

```
0 0 4*Lvx 0 %Portico E
```

```
0 0 0 4*Lvy %Portico 1
```

```
1*Lvx 0 1*Lvx 4*Lvy %Portico 2
```

```
2*Lvx 0 2*Lvx 4*Lvy %Portico 3
```

```
3*Lvx 0 3*Lvx 4*Lvy %Portico 4
```

```

4*Lvx 0 4*Lvx 4*Lvy          %Portico 5
];
% Se calcula d
d = sqrt((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
C = (coor(:,3)-coor(:,1))./d;
S = (coor(:,4)-coor(:,2))./d;
% Definir vector desde c/pórtico hasta el CM de piso 2
r2 = (CR.p2(2)-coor(:,2)).*C-(CR.p2(1)-coor(:,1)).*S;
% Matriz de transformación PISO 2
mt.p2 = [C S r2]';

% Piso 3
CR.p3 = [XCR_final(3) YCR_final(3)];          %Centro de rigidez de piso 3 POR METODO
1

% coor = [xa ya xb yb]
coor = [...
0 4*Lvy 4*Lvx 4*Lvy          %Portico A
0 3*Lvy 4*Lvx 3*Lvy          %Portico B
0 2*Lvy 4*Lvx 2*Lvy          %Portico C
0 1*Lvy 4*Lvx 1*Lvy          %Portico D
0 0 4*Lvx 0                  %Portico E
0 0 0 4*Lvy                  %Portico 1
1*Lvx 0 1*Lvx 4*Lvy          %Portico 2
2*Lvx 0 2*Lvx 4*Lvy          %Portico 3
3*Lvx 0 3*Lvx 4*Lvy          %Portico 4
4*Lvx 0 4*Lvx 4*Lvy          %Portico 5
];
% Se calcula d
d = sqrt((coor(:,3)-coor(:,1)).^2+((coor(:,4)-coor(:,2)).^2));
C = (coor(:,3)-coor(:,1))./d;
S = (coor(:,4)-coor(:,2))./d;
% Definir vector desde c/pórtico hasta el CM de piso 3
r3 = (CR.p3(2)-coor(:,2)).*C-(CR.p3(1)-coor(:,1)).*S;
% Matriz de transformación PISO 3
mt.p3 = [C S r3]';

% Las matrices de transformación para cada pórtico quedan:
TCR.eA = zeros(9,3);   TCR.eA(1:3,1) = mt.p1(:,1);   TCR.eA(4:6,2) = mt.p2(:,1);
TCR.eA(7:9,3) = mt.p3(:,1);
TCR.eB = zeros(9,3);   TCR.eB(1:3,1) = mt.p1(:,2);   TCR.eB(4:6,2) = mt.p2(:,2);
TCR.eB(7:9,3) = mt.p3(:,2);
TCR.eC = zeros(9,3);   TCR.eC(1:3,1) = mt.p1(:,3);   TCR.eC(4:6,2) = mt.p2(:,3);
TCR.eC(7:9,3) = mt.p3(:,3);
TCR.eD = zeros(9,3);   TCR.eD(1:3,1) = mt.p1(:,4);   TCR.eD(4:6,2) = mt.p2(:,4);
TCR.eD(7:9,3) = mt.p3(:,4);
TCR.eE = zeros(9,3);   TCR.eE(1:3,1) = mt.p1(:,5);   TCR.eE(4:6,2) = mt.p2(:,5);
TCR.eE(7:9,3) = mt.p3(:,5);
TCR.e1 = zeros(9,3);   TCR.e1(1:3,1) = mt.p1(:,6);   TCR.e1(4:6,2) = mt.p2(:,6);
TCR.e1(7:9,3) = mt.p3(:,6);
TCR.e2 = zeros(9,3);   TCR.e2(1:3,1) = mt.p1(:,7);   TCR.e2(4:6,2) = mt.p2(:,7);
TCR.e2(7:9,3) = mt.p3(:,7);
TCR.e3 = zeros(9,3);   TCR.e3(1:3,1) = mt.p1(:,8);   TCR.e3(4:6,2) = mt.p2(:,8);
TCR.e3(7:9,3) = mt.p3(:,8);
TCR.e4 = zeros(9,3);   TCR.e4(1:3,1) = mt.p1(:,9);   TCR.e4(4:6,2) = mt.p2(:,9);
TCR.e4(7:9,3) = mt.p3(:,9);

```

```

TCR.e5 = zeros(9,3);   TCR.e5(1:3,1) = mt.p1(:,10);   TCR.e5(4:6,2) = mt.p2(:,10);
TCR.e5(7:9,3) = mt.p3(:,10);

% Matriz de rigidez global para cada pórtico
KgCR.eA = TCR.eA*Kc.xxA*TCR.eA';   % PÓRTICO A
KgCR.eB = TCR.eB*Kc.xxB*TCR.eB';   % PÓRTICO B
KgCR.eC = TCR.eC*Kc.xxC*TCR.eC';   % PÓRTICO C
KgCR.eD = TCR.eD*Kc.xxD*TCR.eD';   % PÓRTICO D
KgCR.eE = TCR.eE*Kc.xxE*TCR.eE';   % PÓRTICO E
KgCR.e1 = TCR.e1*Kc.yy1*TCR.e1';   % PÓRTICO 1
KgCR.e2 = TCR.e2*Kc.yy2*TCR.e2';   % PÓRTICO 2
KgCR.e3 = TCR.e3*Kc.yy3*TCR.e3';   % PÓRTICO 3
KgCR.e4 = TCR.e4*Kc.yy4*TCR.e4';   % PÓRTICO 4
KgCR.e5 = TCR.e5*Kc.yy5*TCR.e5';   % PÓRTICO 5

% Matriz de rigidez de toda la estructura para los 9 GDLs.
KKgCR = KgCR.eA + KgCR.eB + KgCR.eC + KgCR.eD + KgCR.eE + KgCR.e1 + KgCR.e2 + KgCR.e3
+ KgCR.e4 + KgCR.e5;
issymmetric(round(KKgCR,2));

%Respuesta de rotaciones debido a aplicacion de Fuerza en CR
format long
Rzx_p1 = KKgCR\((10000000*[1 0 0 0 0 0 0 0 0]'));
Rzx_p2 = KKgCR\((10000000*[0 0 0 1 0 0 0 0 0]'));
Rzx_p3 = KKgCR\((10000000*[0 0 0 0 0 0 1 0 0]'));

Rzx_rotacion = [Rzx_p1(3) Rzx_p2(6) Rzx_p3(9)]';

format long
Rzy_p1 = KKgCR\((10000000*[0 1 0 0 0 0 0 0 0]'));
Rzy_p2 = KKgCR\((10000000*[0 0 0 0 1 0 0 0 0]'));
Rzy_p3 = KKgCR\((10000000*[0 0 0 0 0 0 0 1 0]'));

Rzy_rotacion = [Rzy_p1(3) Rzy_p2(6) Rzy_p3(9)]';

%% Matriz de masa
% Masas traslacionales

format bank
% Piso 1
m.tx1 = carga.pis* Area.pis;
m.ty1 = m.tx1;
% Piso 2
m.tx2 = carga.pis* Area.pis;
m.ty2 = m.tx2;
% Piso 3
m.tx3 = carga.cub* Area.cub;
m.ty3 = m.tx3;

% Masas rotacionales

%Piso 1
%Inercias con respecto al eje X
Ix.pano1 = ((LvX*Lvy^3/12)+(LvX*Lvy*(dy.p1-CM.p1(2))^2));

```

```

Ix.pano2 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p2-CM.p1(2))^2));
Ix.pano3 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p3-CM.p1(2))^2));
Ix.pano4 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p4-CM.p1(2))^2));
Ix.pano5 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p5-CM.p1(2))^2));
Ix.pano6 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p6-CM.p1(2))^2));
Ix.pano7 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p7-CM.p1(2))^2));
Ix.pano8 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p8-CM.p1(2))^2));
Ix.pano9 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p9-CM.p1(2))^2));
Ix.pano10 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p10-CM.p1(2))^2));
Ix.pano11 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p11-CM.p1(2))^2));
Ix.pano12 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p12-CM.p1(2))^2));
Ix.pano14 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p14-CM.p1(2))^2));
Ix.pano15 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p15-CM.p1(2))^2));
Ix.pano16 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p16-CM.p1(2))^2));
Inerciay.piso1 =
Ix.pano1+Ix.pano2+Ix.pano3+Ix.pano4+Ix.pano5+Ix.pano6+Ix.pano7+Ix.pano8+Ix.pano9+Ix.p
ano10+Ix.pano11+Ix.pano12+Ix.pano14+Ix.pano15+Ix.pano16;
%Inercias con respecto al eje Y
Iy.pano1 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p1-CM.p1(1))^2));
Iy.pano2 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p2-CM.p1(1))^2));
Iy.pano3 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p3-CM.p1(1))^2));
Iy.pano4 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p4-CM.p1(1))^2));
Iy.pano5 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p5-CM.p1(1))^2));
Iy.pano6 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p6-CM.p1(1))^2));
Iy.pano7 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p7-CM.p1(1))^2));
Iy.pano8 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p8-CM.p1(1))^2));
Iy.pano9 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p9-CM.p1(1))^2));
Iy.pano10 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p10-CM.p1(1))^2));
Iy.pano11 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p11-CM.p1(1))^2));
Iy.pano12 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p12-CM.p1(1))^2));
Iy.pano14 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p14-CM.p1(1))^2));
Iy.pano15 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p15-CM.p1(1))^2));
Iy.pano16 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p16-CM.p1(1))^2));
Inerciay.piso1 =
Iy.pano1+Iy.pano2+Iy.pano3+Iy.pano4+Iy.pano5+Iy.pano6+Iy.pano7+Iy.pano8+Iy.pano9+Iy.p
ano10+Iy.pano11+Iy.pano12+Iy.pano14+Iy.pano15+Iy.pano16;
m.z1 = carga.pis * (Inerciay.piso1 + Inerciay.piso1);

```

```
%Piso 2
```

```
%Inercias con respecto al eje X
```

```

Ix.pano1 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p1-CM.p2(2))^2));
Ix.pano2 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p2-CM.p2(2))^2));
Ix.pano3 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p3-CM.p2(2))^2));
Ix.pano4 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p4-CM.p2(2))^2));
Ix.pano5 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p5-CM.p2(2))^2));
Ix.pano6 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p6-CM.p2(2))^2));
Ix.pano7 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p7-CM.p2(2))^2));
Ix.pano8 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p8-CM.p2(2))^2));
Ix.pano9 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p9-CM.p2(2))^2));
Ix.pano10 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p10-CM.p2(2))^2));
Ix.pano11 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p11-CM.p2(2))^2));
Ix.pano12 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p12-CM.p2(2))^2));
Ix.pano14 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p14-CM.p2(2))^2));
Ix.pano15 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p15-CM.p2(2))^2));
Ix.pano16 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p16-CM.p2(2))^2));

```

```

Inerciay.piso2 =
Ix.pano1+Ix.pano2+Ix.pano3+Ix.pano4+Ix.pano5+Ix.pano6+Ix.pano7+Ix.pano8+Ix.pano9+Ix.p
ano10+Ix.pano11+Ix.pano12+Ix.pano14+Ix.pano15+Ix.pano16;
%Inercias con respecto al eje Y
Iy.pano1 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p1-CM.p2(1))^2));
Iy.pano2 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p2-CM.p2(1))^2));
Iy.pano3 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p3-CM.p2(1))^2));
Iy.pano4 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p4-CM.p2(1))^2));
Iy.pano5 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p5-CM.p2(1))^2));
Iy.pano6 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p6-CM.p2(1))^2));
Iy.pano7 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p7-CM.p2(1))^2));
Iy.pano8 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p8-CM.p2(1))^2));
Iy.pano9 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p9-CM.p2(1))^2));
Iy.pano10 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p10-CM.p2(1))^2));
Iy.pano11 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p11-CM.p2(1))^2));
Iy.pano12 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p12-CM.p2(1))^2));
Iy.pano14 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p14-CM.p2(1))^2));
Iy.pano15 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p15-CM.p2(1))^2));
Iy.pano16 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p16-CM.p2(1))^2));
Inerciay.piso2 =
Iy.pano1+Iy.pano2+Iy.pano3+Iy.pano4+Iy.pano5+Iy.pano6+Iy.pano7+Iy.pano8+Iy.pano9+Iy.p
ano10+Iy.pano11+Iy.pano12+Iy.pano14+Iy.pano15+Iy.pano16;
m.z2 = carga.pis * (Inerciay.piso2 + Inerciay.piso2);

```

```
%Piso 3
```

```
%Inercias con respecto al eje X
```

```

Ix.pano1 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p1-CM.p3(2))^2));
Ix.pano2 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p2-CM.p3(2))^2));
Ix.pano3 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p3-CM.p3(2))^2));
Ix.pano4 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p4-CM.p3(2))^2));
Ix.pano5 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p5-CM.p3(2))^2));
Ix.pano6 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p6-CM.p3(2))^2));
Ix.pano7 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p7-CM.p3(2))^2));
Ix.pano8 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p8-CM.p3(2))^2));
Ix.pano9 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p9-CM.p3(2))^2));
Ix.pano10 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p10-CM.p3(2))^2));
Ix.pano11 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p11-CM.p3(2))^2));
Ix.pano12 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p12-CM.p3(2))^2));
Ix.pano13 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p13-CM.p3(2))^2));
Ix.pano14 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p14-CM.p3(2))^2));
Ix.pano15 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p15-CM.p3(2))^2));
Ix.pano16 = ((Lvx*Lvy^3/12)+(Lvx*Lvy*(dy.p16-CM.p3(2))^2));
Inerciay.piso3 =
Ix.pano1+Ix.pano2+Ix.pano3+Ix.pano4+Ix.pano5+Ix.pano6+Ix.pano7+Ix.pano8+Ix.pano9+Ix.p
ano10+Ix.pano11+Ix.pano12+Ix.pano13+Ix.pano14+Ix.pano15+Ix.pano16;

```

```
%Inercias con respecto al eje Y
```

```

Iy.pano1 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p1-CM.p3(1))^2));
Iy.pano2 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p2-CM.p3(1))^2));
Iy.pano3 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p3-CM.p3(1))^2));
Iy.pano4 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p4-CM.p3(1))^2));
Iy.pano5 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p5-CM.p3(1))^2));
Iy.pano6 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p6-CM.p3(1))^2));
Iy.pano7 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p7-CM.p3(1))^2));
Iy.pano8 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p8-CM.p3(1))^2));
Iy.pano9 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p9-CM.p3(1))^2));
Iy.pano10 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p10-CM.p3(1))^2));
Iy.pano11 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p11-CM.p3(1))^2));

```

```

Iy.pano12 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p12-CM.p3(1))^2));
Iy.pano13 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p13-CM.p3(1))^2));
Iy.pano14 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p14-CM.p3(1))^2));
Iy.pano15 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p15-CM.p3(1))^2));
Iy.pano16 = ((Lvy*Lvx^3/12)+(Lvx*Lvy*(dx.p16-CM.p3(1))^2));
Inerciay.piso3 =
Iy.pano1+Iy.pano2+Iy.pano3+Iy.pano4+Iy.pano5+Iy.pano6+Iy.pano7+Iy.pano8+Iy.pano9+Iy.p
ano10+Iy.pano11+Iy.pano12+Iy.pano13+Iy.pano14+Iy.pano15+Iy.pano16;
m.z3 = carga.cub * (Inerciay.piso3 + Inerciay.piso3);

```

```
%% Matriz de masa queda
```

```

MM = diag([m.tx1 m.ty1 m.z1 m.tx2 m.ty2 m.z2 m.tx3 m.ty3 m.z3]);
MasaTotal = sum([m.tx1 m.ty1 m.z1 m.tx2 m.ty2 m.z2 m.tx3 m.ty3 m.z3]);

```

```
%% Análisis modal
```

```

[fi, w_cuad] = eig(MM\Kkg);
[omega, ind]= sort(sqrt(diag(w_cuad)));
omega = diag(round(100*omega)/100); % Frecuencias de vibración [rad/seg]
frecs = diag(omega)/2/pi; % Frecuencias de vibración [Hz]
periodos = 1./frecs % Periodos de vibración [seg]
fi = fi(:,ind'); % Modes

```

```
% Por Espacio de estados
```

```

[AA,BB,CC,DD,GAM] = rigid_diap3pisos(MM,Kkg,zeta);
sys_str = ss(AA,BB,CC,DD);
damp(sys_str);

```

```
%% Factor de contribución modal
```

```

% disp('=====');
% disp('Contribución modal');
iota = [1 0 0 1 0 0 1 0 0; 0 1 0 0 1 0 0 1 0; 0 0 1 0 0 1 0 0 1]';
Mn = fi'*MM*fi; % Probar que es M es diagonal
Ln = fi'*MM*iota;
Gn = Ln./diag(Mn); % Factor de participación modal (Gamma)
mef = Ln.^2./diag(Mn); % Masa que contribuye en cada modo [kg]
M_tot = iota'*MM*iota; % Masa total de la estructura [kg]
% disp('Porcentaje de participación de masa [%]:')
PM = 100*mef/M_tot % Participación modal [%]

```

```
%% Análisis modal espectral ESPECTRO NEC-15 (RSA)
```

```

% disp('=====');
% disp('Obtención del espectro de desplazamiento segun la NEC-15');

```

```

TT = round(periodos,2);
z = 0.4;
Fa = 1.2;
Fd = 1.19;
Fs = 1.28;
R = 5;

```

```

[Sa] = resp_spectrum_NEC15(TT,Fa,Fd,Fs,z,R);           % Ordenadas del espectro de
Pseudo-aceleraciones aceleraciones del espectro de diseno NEC 2015 [g]

Sd = zeros(1,length(Sa));                             % Ordenadas del espectro de
diseno de Desplazamientos Nec 2015 [m]
for i=1:length(Sa)
    Sd(i) = Sa(i)*9.81*(TT(i,1)^2)/(2*pi)^2;
end
Sd;

%%
% Determinar el desplazamiento de piso para cada modo cuando el sismo actua
% en direccion x, y direccion y

um1 = abs(Gn(1,:)).*fi(:,1)*Sd(1);   % Modo 1
um2 = abs(Gn(2,:)).*fi(:,2)*Sd(2);   % Modo 2
um3 = abs(Gn(3,:)).*fi(:,3)*Sd(3);   % Modo 3
um4 = abs(Gn(4,:)).*fi(:,4)*Sd(4);   % Modo 4
um5 = abs(Gn(5,:)).*fi(:,5)*Sd(5);   % Modo 5
um6 = abs(Gn(6,:)).*fi(:,6)*Sd(6);   % Modo 6
um7 = abs(Gn(7,:)).*fi(:,7)*Sd(7);   % Modo 7
um8 = abs(Gn(8,:)).*fi(:,8)*Sd(8);   % Modo 8
um9 = abs(Gn(9,:)).*fi(:,9)*Sd(9);   % Modo 9

um_sismoX = [um1(:,1) um2(:,1) um3(:,1) um4(:,1) um5(:,1) um6(:,1) um7(:,1) um8(:,1)
um9(:,1)];
um_sismoY = [um1(:,2) um2(:,2) um3(:,2) um4(:,2) um5(:,2) um6(:,2) um7(:,2) um8(:,2)
um9(:,2)];
um_sismoRZ = [um1(:,3) um2(:,3) um3(:,3) um4(:,3) um5(:,3) um6(:,3) um7(:,3) um8(:,3)
um9(:,3)];

%% Combinación modal de la respuesta (SRSS)

% Respuesta máxima en desplazamiento de la estructura por GDL
%disp('Obtención de las respuestas maximas de desplazamiento');
%disp('debido al sismo actuando en las direcciones de analisis [cm]');

RSA.u1_maxNEC15_SRSS =
[sqrt(um1(1,1)^2+um2(1,1)^2+um3(1,1)^2+um4(1,1)^2+um5(1,1)^2+um6(1,1)^2+um7(1,1)^2+um
8(1,1)^2+um9(1,1)^2)
sqrt(um1(1,2)^2+um2(1,2)^2+um3(1,2)^2+um4(1,2)^2+um5(1,2)^2+um6(1,2)^2+um7(1,2)^2+um8
(1,2)^2+um9(1,2)^2)
sqrt(um1(1,3)^2+um2(1,3)^2+um3(1,3)^2+um4(1,3)^2+um5(1,3)^2+um6(1,3)^2+um7(1,3)^2+um8
(1,3)^2+um9(1,3)^2)].*100;
RSA.u2_maxNEC15_SRSS =
[sqrt(um1(2,1)^2+um2(2,1)^2+um3(2,1)^2+um4(2,1)^2+um5(2,1)^2+um6(2,1)^2+um7(2,1)^2+um
8(2,1)^2+um9(2,1)^2)
sqrt(um1(2,2)^2+um2(2,2)^2+um3(2,2)^2+um4(2,2)^2+um5(2,2)^2+um6(2,2)^2+um7(2,2)^2+um8
(2,2)^2+um9(2,2)^2)
sqrt(um1(2,3)^2+um2(2,3)^2+um3(2,3)^2+um4(2,3)^2+um5(2,3)^2+um6(2,3)^2+um7(2,3)^2+um8
(2,3)^2+um9(2,3)^2)].*100;
RSA.u3_maxNEC15_SRSS =
[sqrt(um1(3,1)^2+um2(3,1)^2+um3(3,1)^2+um4(3,1)^2+um5(3,1)^2+um6(3,1)^2+um7(3,1)^2+um
8(3,1)^2+um9(3,1)^2)
sqrt(um1(3,2)^2+um2(3,2)^2+um3(3,2)^2+um4(3,2)^2+um5(3,2)^2+um6(3,2)^2+um7(3,2)^2+um8
(3,2)^2+um9(3,2)^2)

```

```

sqrt(um1(3,3)^2+um2(3,3)^2+um3(3,3)^2+um4(3,3)^2+um5(3,3)^2+um6(3,3)^2+um7(3,3)^2+um8
(3,3)^2+um9(3,3)^2)].*100;
RSA.u4_maxNEC15_SRSS =
[sqrt(um1(4,1)^2+um2(4,1)^2+um3(4,1)^2+um4(4,1)^2+um5(4,1)^2+um6(4,1)^2+um7(4,1)^2+um
8(4,1)^2+um9(4,1)^2)
sqrt(um1(4,2)^2+um2(4,2)^2+um3(4,2)^2+um4(4,2)^2+um5(4,2)^2+um6(4,2)^2+um7(4,2)^2+um8
(4,2)^2+um9(4,2)^2)
sqrt(um1(4,3)^2+um2(4,3)^2+um3(4,3)^2+um4(4,3)^2+um5(4,3)^2+um6(4,3)^2+um7(4,3)^2+um8
(4,3)^2+um9(4,3)^2)].*100;
RSA.u5_maxNEC15_SRSS =
[sqrt(um1(5,1)^2+um2(5,1)^2+um3(5,1)^2+um4(5,1)^2+um5(5,1)^2+um6(5,1)^2+um7(5,1)^2+um
8(5,1)^2+um9(5,1)^2)
sqrt(um1(5,2)^2+um2(5,2)^2+um3(5,2)^2+um4(5,2)^2+um5(5,2)^2+um6(5,2)^2+um7(5,2)^2+um8
(5,2)^2+um9(5,2)^2)
sqrt(um1(5,3)^2+um2(5,3)^2+um3(5,3)^2+um4(5,3)^2+um5(5,3)^2+um6(5,3)^2+um7(5,3)^2+um8
(5,3)^2+um9(5,3)^2)].*100;
RSA.u6_maxNEC15_SRSS =
[sqrt(um1(6,1)^2+um2(6,1)^2+um3(6,1)^2+um4(6,1)^2+um5(6,1)^2+um6(6,1)^2+um7(6,1)^2+um
8(6,1)^2+um9(6,1)^2)
sqrt(um1(6,2)^2+um2(6,2)^2+um3(6,2)^2+um4(6,2)^2+um5(6,2)^2+um6(6,2)^2+um7(6,2)^2+um8
(6,2)^2+um9(6,2)^2)
sqrt(um1(6,3)^2+um2(6,3)^2+um3(6,3)^2+um4(6,3)^2+um5(6,3)^2+um6(6,3)^2+um7(6,3)^2+um8
(6,3)^2+um9(6,3)^2)].*100;
RSA.u7_maxNEC15_SRSS =
[sqrt(um1(7,1)^2+um2(7,1)^2+um3(7,1)^2+um4(7,1)^2+um5(7,1)^2+um6(7,1)^2+um7(7,1)^2+um
8(7,1)^2+um9(7,1)^2)
sqrt(um1(7,2)^2+um2(7,2)^2+um3(7,2)^2+um4(7,2)^2+um5(7,2)^2+um6(7,2)^2+um7(7,2)^2+um8
(7,2)^2+um9(7,2)^2)
sqrt(um1(7,3)^2+um2(7,3)^2+um3(7,3)^2+um4(7,3)^2+um5(7,3)^2+um6(7,3)^2+um7(7,3)^2+um8
(7,3)^2+um9(7,3)^2)].*100;
RSA.u8_maxNEC15_SRSS =
[sqrt(um1(8,1)^2+um2(8,1)^2+um3(8,1)^2+um4(8,1)^2+um5(8,1)^2+um6(8,1)^2+um7(8,1)^2+um
8(8,1)^2+um9(8,1)^2)
sqrt(um1(8,2)^2+um2(8,2)^2+um3(8,2)^2+um4(8,2)^2+um5(8,2)^2+um6(8,2)^2+um7(8,2)^2+um8
(8,2)^2+um9(8,2)^2)
sqrt(um1(8,3)^2+um2(8,3)^2+um3(8,3)^2+um4(8,3)^2+um5(8,3)^2+um6(8,3)^2+um7(8,3)^2+um8
(8,3)^2+um9(8,3)^2)].*100;
RSA.u9_maxNEC15_SRSS =
[sqrt(um1(9,1)^2+um2(9,1)^2+um3(9,1)^2+um4(9,1)^2+um5(9,1)^2+um6(9,1)^2+um7(9,1)^2+um
8(9,1)^2+um9(9,1)^2)
sqrt(um1(9,2)^2+um2(9,2)^2+um3(9,2)^2+um4(9,2)^2+um5(9,2)^2+um6(9,2)^2+um7(9,2)^2+um8
(9,2)^2+um9(9,2)^2)
sqrt(um1(9,3)^2+um2(9,3)^2+um3(9,3)^2+um4(9,3)^2+um5(9,3)^2+um6(9,3)^2+um7(9,3)^2+um8
(9,3)^2+um9(9,3)^2)].*100;
RSA;

```

```

% Derivas máximas divididas entre la altura de piso
% disp('Obtención de las derivas maximas en las direcciones de analisis [%]')
% driftX_max = [RSA.u1_maxNEC15 RSA.u4_maxNEC15-RSA.u1_maxNEC15 RSA.u7_maxNEC15-
RSA.u4_maxNEC15]'*100/Lc/100 %derivas en x piso 1, 2 y 3
% driftY_max = [RSA.u2_maxNEC15 RSA.u5_maxNEC15-RSA.u2_maxNEC15 RSA.u8_maxNEC15-
RSA.u5_maxNEC15]'*100/Lc/100 %derivas en y piso 1, 2 y 3

```

```

%% Combinación modal de la respuesta (CQC)
frecs;
beta = zeros(length(frecs),length(frecs));

```

```

for j=1:1:length(frecs)
for i=1:1:length(frecs)
    beta(i,j) = frecs(i)/frecs(j);
end
end
beta;
etha=zeta(1);

```

```

rho = zeros(length(frecs),length(frecs));
for j=1:1:length(frecs)
for i=1:1:length(frecs)
    rho(i,j) = (8*etha^2*(1+beta(i,j))*(beta(i,j))^(3/2))/((1-
(beta(i,j))^2)^2+(4*etha^2*beta(i,j)*(1+(beta(i,j))^2)^2));
end
end
rho;

```

format bank

```

RSA.u1_maxNEC15_CQC = [sqrt(um_sismoX(1,:)*rho*um_sismoX(1,:))
sqrt(um_sismoY(1,:)*rho*um_sismoY(1,:)) sqrt(um_sismoRZ(1,:)*rho*um_sismoRZ(1,:))
].*100;
RSA.u2_maxNEC15_CQC = [sqrt(um_sismoX(2,:)*rho*um_sismoX(2,:))
sqrt(um_sismoY(2,:)*rho*um_sismoY(2,:)) sqrt(um_sismoRZ(2,:)*rho*um_sismoRZ(2,:))
].*100;
RSA.u3_maxNEC15_CQC = [sqrt(um_sismoX(3,:)*rho*um_sismoX(3,:))
sqrt(um_sismoY(3,:)*rho*um_sismoY(3,:)) sqrt(um_sismoRZ(3,:)*rho*um_sismoRZ(3,:))
];
RSA.u4_maxNEC15_CQC = [sqrt(um_sismoX(4,:)*rho*um_sismoX(4,:))
sqrt(um_sismoY(4,:)*rho*um_sismoY(4,:)) sqrt(um_sismoRZ(4,:)*rho*um_sismoRZ(4,:))
].*100;
RSA.u5_maxNEC15_CQC = [sqrt(um_sismoX(5,:)*rho*um_sismoX(5,:))
sqrt(um_sismoY(5,:)*rho*um_sismoY(5,:)) sqrt(um_sismoRZ(5,:)*rho*um_sismoRZ(5,:))
].*100;
RSA.u6_maxNEC15_CQC = [sqrt(um_sismoX(6,:)*rho*um_sismoX(6,:))
sqrt(um_sismoY(6,:)*rho*um_sismoY(6,:)) sqrt(um_sismoRZ(6,:)*rho*um_sismoRZ(6,:))
];
RSA.u7_maxNEC15_CQC = [sqrt(um_sismoX(7,:)*rho*um_sismoX(7,:))
sqrt(um_sismoY(7,:)*rho*um_sismoY(7,:)) sqrt(um_sismoRZ(7,:)*rho*um_sismoRZ(7,:))
].*100;
RSA.u8_maxNEC15_CQC = [sqrt(um_sismoX(8,:)*rho*um_sismoX(8,:))
sqrt(um_sismoY(8,:)*rho*um_sismoY(8,:)) sqrt(um_sismoRZ(8,:)*rho*um_sismoRZ(8,:))
].*100;
RSA.u9_maxNEC15_CQC = [sqrt(um_sismoX(9,:)*rho*um_sismoX(9,:))
sqrt(um_sismoY(9,:)*rho*um_sismoY(9,:)) sqrt(um_sismoRZ(9,:)*rho*um_sismoRZ(9,:))
];
RSA;

```

%% Obtencion de Relacion de Irregularidad torsional

% Sismo en direccion X

format bank

```

RSA_NEC_CQC_SISMOX = [RSA.u1_maxNEC15_CQC(1)./100 RSA.u2_maxNEC15_CQC(1)./100
RSA.u3_maxNEC15_CQC(1) RSA.u4_maxNEC15_CQC(1)./100 RSA.u5_maxNEC15_CQC(1)./100

```

```
RSA.u6_maxNEC15_CQC(1) RSA.u7_maxNEC15_CQC(1)./100 RSA.u8_maxNEC15_CQC(1)./100
RSA.u9_maxNEC15_CQC(1)']';
```

```
ueA_geo = [RSA_NEC_CQC_SISMOX(1)+((4*Lvy)-YCM(1))*sin(RSA_NEC_CQC_SISMOX(3))
RSA_NEC_CQC_SISMOX(4)+((4*Lvy)-YCM(2))*sin(RSA_NEC_CQC_SISMOX(6))
RSA_NEC_CQC_SISMOX(7)+(4*Lvy-YCM(3))*sin(RSA_NEC_CQC_SISMOX(9))]'*.100
ueE_geo = [RSA_NEC_CQC_SISMOX(1)+(0-YCM(1))*sin(RSA_NEC_CQC_SISMOX(3))
RSA_NEC_CQC_SISMOX(4)+(0-YCM(2))*sin(RSA_NEC_CQC_SISMOX(6)) RSA_NEC_CQC_SISMOX(7)+(0-
YCM(3))*sin(RSA_NEC_CQC_SISMOX(9))]'*.100
```

```
uRel.eA_geo = [ueA_geo(1) ueA_geo(2)-ueA_geo(1) ueA_geo(3)-ueA_geo(2)]';
uRel.eE_geo = [ueE_geo(1) ueE_geo(2)-ueE_geo(1) ueE_geo(3)-ueE_geo(2)]';
```

```
IITxMax_geo = [max(uRel.eA_geo(1),uRel.eE_geo(1))/((uRel.eA_geo(1)+uRel.eE_geo(1))/2)
max(uRel.eA_geo(2),uRel.eE_geo(2))/((uRel.eA_geo(2)+uRel.eE_geo(2))/2)
max(uRel.eA_geo(3),uRel.eE_geo(3))/((uRel.eA_geo(3)+uRel.eE_geo(3))/2)];
IITxMin_geo = [min(uRel.eA_geo(1),uRel.eE_geo(1))/((uRel.eA_geo(1)+uRel.eE_geo(1))/2)
min(uRel.eA_geo(2),uRel.eE_geo(2))/((uRel.eA_geo(2)+uRel.eE_geo(2))/2)
min(uRel.eA_geo(3),uRel.eE_geo(3))/((uRel.eA_geo(3)+uRel.eE_geo(3))/2)];
IITx = max(IITxMax_geo, IITxMin_geo);
```

**% Sismo en direccion Y**

```
RSA_NEC_CQC_SISMOY = [RSA.u1_maxNEC15_CQC(2)./100 RSA.u2_maxNEC15_CQC(2)./100
RSA.u3_maxNEC15_CQC(2) RSA.u4_maxNEC15_CQC(2)./100 RSA.u5_maxNEC15_CQC(2)./100
RSA.u6_maxNEC15_CQC(2) RSA.u7_maxNEC15_CQC(2)./100 RSA.u8_maxNEC15_CQC(2)./100
RSA.u9_maxNEC15_CQC(2)']';
```

```
ue1_geo = [RSA_NEC_CQC_SISMOY(2)+(0-XCR_final(1))*sin(RSA_NEC_CQC_SISMOY(3))
RSA_NEC_CQC_SISMOY(5)+(0-XCR_final(2))*sin(RSA_NEC_CQC_SISMOY(6))
RSA_NEC_CQC_SISMOY(8)+(0-XCR_final(3))*sin(RSA_NEC_CQC_SISMOY(9))]'*.100
ue5_geo = [RSA_NEC_CQC_SISMOY(2)+((4*Lvx)-XCR_final(1))*sin(RSA_NEC_CQC_SISMOY(3))
RSA_NEC_CQC_SISMOY(5)+((4*Lvx)-XCR_final(2))*sin(RSA_NEC_CQC_SISMOY(6))
RSA_NEC_CQC_SISMOY(8)+((4*Lvx)-XCR_final(3))*sin(RSA_NEC_CQC_SISMOY(9))]'*.100
```

```
uRel.e1_geo = [ue1_geo(1) ue1_geo(2)-ue1_geo(1) ue1_geo(3)-ue1_geo(2)]';
uRel.e5_geo = [ue5_geo(1) ue5_geo(2)-ue5_geo(1) ue5_geo(3)-ue5_geo(2)]';
```

```
IITyMax_geo = [max(uRel.e1_geo(1),uRel.e5_geo(1))/((uRel.e1_geo(1)+uRel.e5_geo(1))/2)
max(uRel.e1_geo(2),uRel.e5_geo(2))/((uRel.e1_geo(2)+uRel.e5_geo(2))/2)
max(uRel.e1_geo(3),uRel.e5_geo(3))/((uRel.e1_geo(3)+uRel.e5_geo(3))/2)];
IITyMin_geo = [min(uRel.e1_geo(1),uRel.e5_geo(1))/((uRel.e1_geo(1)+uRel.e5_geo(1))/2)
min(uRel.e1_geo(2),uRel.e5_geo(2))/((uRel.e1_geo(2)+uRel.e5_geo(2))/2)
min(uRel.e1_geo(3),uRel.e5_geo(3))/((uRel.e1_geo(3)+uRel.e5_geo(3))/2)];
IITy = max(IITyMax_geo, IITyMin_geo);
```

**%Derivas máximas divididas entre la altura de piso**

```
% disp('=====');
% disp('Obtención de las derivas maximas en las direcciones de analisis [%]')
```

```
driftX_max = ([RSA.u1_maxNEC15_CQC(1) RSA.u4_maxNEC15_CQC(1)-RSA.u1_maxNEC15_CQC(1)
RSA.u7_maxNEC15_CQC(1)-RSA.u4_maxNEC15_CQC(1)]'.*100/Lc/100).*R*0.75 %derivs en
x piso 1, 2 y 3
```

```

driftY_max = ([RSA.u2_maxNEC15_CQC(2) RSA.u5_maxNEC15_CQC(2)-RSA.u2_maxNEC15_CQC(2)
RSA.u8_maxNEC15_CQC(2)-RSA.u5_maxNEC15_CQC(2)]'*100/Lc/100).*R*0.75 %derivas en
y piso 1, 2 y 3

driftPorticoX_max =
(max(abs(uRel.eA_geo*100/(Lc*100)),abs(uRel.eE_geo*100/(Lc*100)))).*R*0.75
driftPorticoY_max =
(max(abs(uRel.e1_geo*100/(Lc*100)),abs(uRel.e5_geo*100/(Lc*100)))).*R*0.75

%% Análisis modal espectral a partir de Acelerograma (RSA)
%
% disp('=====');
% disp('Obtención del espectro de desplazamiento a partir de acelerograma');
%
% acelerograma='elcentro.txt';
%
% dt = 0.02;
% zeta= 0.05;
% Tn_max = 2;
% dt_T = 0.01;
% uo = 0;
% vo = 0;
%
% [respuestasMaximas] = respSDOF(acelerograma,dt,dt_T,Tn_max,zeta,uo,vo);
%
% TT = round(periodos,2);
% SdRSA=zeros(1,length(TT)); %Ordenadas del espectro de respuesta de
Desplazamientos a partir de acelerograma
% for i=1:length(TT)
% TT(i);
% j=(TT(i)/dt_T)+1;
% j = round(j,0);
% SdRSA(i)=respuestasMaximas(j,1);
% end
% SdRSA;
%
%% Determinar el desplazamiento de piso para cada modo
%
%
% um11 = abs(Gn(1,:)).*fi(:,1)*SdRSA(1); % Modo 1
% um22 = abs(Gn(2,:)).*fi(:,2)*SdRSA(2); % Modo 2
% um33 = abs(Gn(3,:)).*fi(:,3)*SdRSA(3); % Modo 3
% um44 = abs(Gn(4,:)).*fi(:,4)*SdRSA(4); % Modo 4
% um55 = abs(Gn(5,:)).*fi(:,5)*SdRSA(5); % Modo 5
% um66 = abs(Gn(6,:)).*fi(:,6)*SdRSA(6); % Modo 6
% um77 = abs(Gn(7,:)).*fi(:,7)*SdRSA(7); % Modo 7
% um88 = abs(Gn(8,:)).*fi(:,8)*SdRSA(8); % Modo 8
% um99 = abs(Gn(9,:)).*fi(:,9)*SdRSA(9); % Modo 9
%
% %% Combinación modal de la respuesta (SRSS)
%
% % Respuesta máxima en desplazamiento
%
% RSA.u1_max =
[sqrt(um11(1,1)^2+um22(1,1)^2+um33(1,1)^2+um44(1,1)^2+um55(1,1)^2+um66(1,1)^2+um77(1,
1)^2+um88(1,1)^2+um99(1,1)^2)

```

```

sqrt(um11(1,2)^2+um22(1,2)^2+um33(1,2)^2+um44(1,2)^2+um55(1,2)^2+um66(1,2)^2+um77(1,2
)^2+um88(1,2)^2+um99(1,2)^2)
sqrt(um11(1,3)^2+um22(1,3)^2+um33(1,3)^2+um44(1,3)^2+um55(1,3)^2+um66(1,3)^2+um77(1,3
)^2+um88(1,3)^2+um99(1,3)^2)].*100;
% RSA.u2_max =
[sqrt(um11(2,1)^2+um22(2,1)^2+um33(2,1)^2+um44(2,1)^2+um55(2,1)^2+um66(2,1)^2+um77(2,
1)^2+um88(2,1)^2+um99(2,1)^2)
sqrt(um11(2,2)^2+um22(2,2)^2+um33(2,2)^2+um44(2,2)^2+um55(2,2)^2+um66(2,2)^2+um77(2,2
)^2+um88(2,2)^2+um99(2,2)^2)
sqrt(um11(2,3)^2+um22(2,3)^2+um33(2,3)^2+um44(2,3)^2+um55(2,3)^2+um66(2,3)^2+um77(2,3
)^2+um88(2,3)^2+um99(2,3)^2)].*100;
% RSA.u3_max =
[sqrt(um11(3,1)^2+um22(3,1)^2+um33(3,1)^2+um44(3,1)^2+um55(3,1)^2+um66(3,1)^2+um77(3,
1)^2+um88(3,1)^2+um99(3,1)^2)
sqrt(um11(3,2)^2+um22(3,2)^2+um33(3,2)^2+um44(3,2)^2+um55(3,2)^2+um66(3,2)^2+um77(3,2
)^2+um88(3,2)^2+um99(3,2)^2)
sqrt(um11(3,3)^2+um22(3,3)^2+um33(3,3)^2+um44(3,3)^2+um55(3,3)^2+um66(3,3)^2+um77(3,3
)^2+um88(3,3)^2+um99(3,3)^2)].*100;
% RSA.u4_max =
[sqrt(um11(4,1)^2+um22(4,1)^2+um33(4,1)^2+um44(4,1)^2+um55(4,1)^2+um66(4,1)^2+um77(4,
1)^2+um88(4,1)^2+um99(4,1)^2)
sqrt(um11(4,2)^2+um22(4,2)^2+um33(4,2)^2+um44(4,2)^2+um55(4,2)^2+um66(4,2)^2+um77(4,2
)^2+um88(4,2)^2+um99(4,2)^2)
sqrt(um11(4,3)^2+um22(4,3)^2+um33(4,3)^2+um44(4,3)^2+um55(4,3)^2+um66(4,3)^2+um77(4,3
)^2+um88(4,3)^2+um99(4,3)^2)].*100;
% RSA.u5_max =
[sqrt(um11(5,1)^2+um22(5,1)^2+um33(5,1)^2+um44(5,1)^2+um55(5,1)^2+um66(5,1)^2+um77(5,
1)^2+um88(5,1)^2+um99(5,1)^2)
sqrt(um11(5,2)^2+um22(5,2)^2+um33(5,2)^2+um44(5,2)^2+um55(5,2)^2+um66(5,2)^2+um77(5,2
)^2+um88(5,2)^2+um99(5,2)^2)
sqrt(um11(5,3)^2+um22(5,3)^2+um33(5,3)^2+um44(5,3)^2+um55(5,3)^2+um66(5,3)^2+um77(5,3
)^2+um88(5,3)^2+um99(5,3)^2)].*100;
% RSA.u6_max =
[sqrt(um11(6,1)^2+um22(6,1)^2+um33(6,1)^2+um44(6,1)^2+um55(6,1)^2+um66(6,1)^2+um77(6,
1)^2+um88(6,1)^2+um99(6,1)^2)
sqrt(um11(6,2)^2+um22(6,2)^2+um33(6,2)^2+um44(6,2)^2+um55(6,2)^2+um66(6,2)^2+um77(6,2
)^2+um88(6,2)^2+um99(6,2)^2)
sqrt(um11(6,3)^2+um22(6,3)^2+um33(6,3)^2+um44(6,3)^2+um55(6,3)^2+um66(6,3)^2+um77(6,3
)^2+um88(6,3)^2+um99(6,3)^2)].*100;
% RSA.u7_max =
[sqrt(um11(7,1)^2+um22(7,1)^2+um33(7,1)^2+um44(7,1)^2+um55(7,1)^2+um66(7,1)^2+um77(7,
1)^2+um88(7,1)^2+um99(7,1)^2)
sqrt(um11(7,2)^2+um22(7,2)^2+um33(7,2)^2+um44(7,2)^2+um55(7,2)^2+um66(7,2)^2+um77(7,2
)^2+um88(7,2)^2+um99(7,2)^2)
sqrt(um11(7,3)^2+um22(7,3)^2+um33(7,3)^2+um44(7,3)^2+um55(7,3)^2+um66(7,3)^2+um77(7,3
)^2+um88(7,3)^2+um99(7,3)^2)].*100;
% RSA.u8_max =
[sqrt(um11(8,1)^2+um22(8,1)^2+um33(8,1)^2+um44(8,1)^2+um55(8,1)^2+um66(8,1)^2+um77(8,
1)^2+um88(8,1)^2+um99(8,1)^2)
sqrt(um11(8,2)^2+um22(8,2)^2+um33(8,2)^2+um44(8,2)^2+um55(8,2)^2+um66(8,2)^2+um77(8,2
)^2+um88(8,2)^2+um99(8,2)^2)
sqrt(um11(8,3)^2+um22(8,3)^2+um33(8,3)^2+um44(8,3)^2+um55(8,3)^2+um66(8,3)^2+um77(8,3
)^2+um88(8,3)^2+um99(8,3)^2)].*100;
% RSA.u9_max =
[sqrt(um11(9,1)^2+um22(9,1)^2+um33(9,1)^2+um44(9,1)^2+um55(9,1)^2+um66(9,1)^2+um77(9,
1)^2+um88(9,1)^2+um99(9,1)^2)
sqrt(um11(9,2)^2+um22(9,2)^2+um33(9,2)^2+um44(9,2)^2+um55(9,2)^2+um66(9,2)^2+um77(9,2

```

```

)^2+um88(9,2)^2+um99(9,2)^2)
sqrt(um11(9,3)^2+um22(9,3)^2+um33(9,3)^2+um44(9,3)^2+um55(9,3)^2+um66(9,3)^2+um77(9,3
)^2+um88(9,3)^2+um99(9,3)^2)].*100;
% RSA;
%
% disp('Derivas de piso [%]');
% RSAdriftX_max = [RSA.u1_max(1) RSA.u4_max(1)-RSA.u1_max(1) RSA.u7_max(1)-
RSA.u4_max(1)]'.*100/(Lc*100);          %derivadas en x piso 1, 2 y 3
% RSAdriftY_max = [RSA.u2_max(2) RSA.u5_max(2)-RSA.u2_max(2) RSA.u8_max(2)-
RSA.u5_max(2)]'.*100/(Lc*100);          %derivadas en y piso 1, 2 y 3

% %% Respuesta en el tiempo
%
% disp('=====')
% disp('Respuesta en el tiempo sismo en X');
%
% [AA,BB,CC,DD,GAM] = rigid_diap1D(MM,KKg,zeta);
% sys_str = ss(AA,BB,CC,DD);
% load('elcentro.txt');
% dt = 0.02;
% dd = elcentro/(100);
% %dd = [dd dd dd];
% tt = [0:dt:(length(dd)-1)*dt]';
% X0 = zeros(size(sys_str.A,1),1);
% [yy,~,~,~] = lsim(sys_str,dd,tt,X0);
%
% RHA.u1_max(1) = max(abs(yy(:,1))).*100;
% RHA.u2_max(1) = max(abs(yy(:,2))).*100;
% RHA.u3_max(1) = max(abs(yy(:,3))).*100;
% RHA.u4_max(1) = max(abs(yy(:,4))).*100;
% RHA.u5_max(1) = max(abs(yy(:,5))).*100;
% RHA.u6_max(1) = max(abs(yy(:,6))).*100;
% RHA.u7_max(1) = max(abs(yy(:,7))).*100;
% RHA.u8_max(1) = max(abs(yy(:,8))).*100;
% RHA.u9_max(1) = max(abs(yy(:,9))).*100;
%
% disp('Respuesta en el tiempo sismo en Y');
%
% [AA,BB,CC,DD,GAM] = rigid_diap2D(MM,KKg,zeta);
% sys_str = ss(AA,BB,CC,DD);
% load('elcentro.txt');
% dt = 0.02;
% dd = elcentro/(100);
% %dd = [dd dd dd];
% tt = [0:dt:(length(dd)-1)*dt]';
% X0 = zeros(size(sys_str.A,1),1);
% [yy,~,~,~] = lsim(sys_str,dd,tt,X0);
%
% RHA.u1_max(2) = max(abs(yy(:,1))).*100;
% RHA.u2_max(2) = max(abs(yy(:,2))).*100;
% RHA.u3_max(2) = max(abs(yy(:,3))).*100;
% RHA.u4_max(2) = max(abs(yy(:,4))).*100;
% RHA.u5_max(2) = max(abs(yy(:,5))).*100;
% RHA.u6_max(2) = max(abs(yy(:,6))).*100;
% RHA.u7_max(2) = max(abs(yy(:,7))).*100;
% RHA.u8_max(2) = max(abs(yy(:,8))).*100;
% RHA.u9_max(2) = max(abs(yy(:,9))).*100;

```

```

% RHA;
%
%
% % Derivas máximas divididas entre la altura de piso
% % disp('Obtención de las derivas maximas en las direcciones de analisis [%]');
% % driftX_max = [RHA.u1_max(1) RHA.u4_max(1)-RHA.u1_max(1) RHA.u7_max(1)-
RHA.u4_max(1)]'.*100./(Lc*100)      %derivadas en x piso 1, 2 y 3
% % driftY_max = [RHA.u2_max(2) RHA.u5_max(2)-RHA.u2_max(2) RHA.u8_max(2)-
RHA.u5_max(2)]'.*100./(Lc*100)      %derivadas en y piso 1, 2 y 3
%
% % figure
% % subplot(2,1,1)
% % plot(tt,yy(:,1),'black-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 1, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % subplot(2,1,2)
% % plot(tt,yy(:,2),'magenta-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 2, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % figure
% % subplot(2,1,1)
% % plot(tt,yy(:,3),'blue-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 3, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % subplot(2,1,2)
% % plot(tt,yy(:,4),'red-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 4, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % figure
% % subplot(2,1,1)
% % plot(tt,yy(:,5),'cyan-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 5, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % subplot(2,1,2)
% % plot(tt,yy(:,6),'yellow-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 6, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % figure
% % subplot(2,1,1)
% % plot(tt,yy(:,7),'green-','LineWidth',0.2), grid on, grid minor
% % title('Respuesta de Desplazamiento en el Tiempo',['GDL = 7, \zeta =
',num2str(zeta)], 'FontSize',12)
% % xlabel('Tiempo [seg]'), ylabel('D [m]')
% %
% % subplot(2,1,2)

```

```

%% plot(tt,yy(:,8),'red-','LineWidth',0.2), grid on, grid minor
%% title('Respuesta de Desplazamiento en el Tiempo',['GDL = 8, \zeta =
',num2str(zeta)], 'FontSize',12)
%% xlabel('Tiempo [seg]'), ylabel('D [m]')
%%
%% figure
%% subplot(2,1,1)
%% plot(tt,yy(:,9),'magenta-','LineWidth',0.2), grid on, grid minor
%% title('Respuesta de Desplazamiento en el Tiempo',['GDL = 9, \zeta =
',num2str(zeta)], 'FontSize',12)
%% xlabel('Tiempo [seg]'), ylabel('D [m]')
%
%
%% Desplazamientos de cada portico
%
% format bank
%%Respuestas de desplazamientos por GDL en el tiempo, SISMO X
%
% [AAA,BBB,CCC,DDD,GAM] = rigid_diap1D(MM,KKg,zeta);
% sys_str = ss(AAA,BBB,CCC,DDD);
% load('elcentro.txt');
% dt = 0.02; %Tiempo de muestreo acelerograma
% dd = elcentro/(100); %Acelerograma en [m/seg^2]
% tt = (0:dt:(length(dd)-1)*dt)'; %Vector Tiempo
% X0 = zeros(size(sys_str.A,1),1); %condiciones iniciales
% [yyy,~,~,~] = lsim(sys_str,dd,tt,X0); %yyy = desplaz, aceler totales, aceler
rel, derivas
% %en el tiempo
%
%
%%Desplazamientos del difragma y de cada portico en cada piso en en tiempo en [m]
% U = [(yyy(:,1:9))]; %Desplazamientos en el tiempo de cada diafragma
%
% u.eA = T.eA' * U; %Desplazamientos en el tiempo del portico A en
cada piso
% u.eB = T.eB' * U; %Desplazamientos en el tiempo del portico B en
cada piso
% u.eC = T.eC' * U; %Desplazamientos en el tiempo del portico C en
cada piso
% u.eD = T.eD' * U; %Desplazamientos en el tiempo del portico D en
cada piso
% u.eE = T.eE' * U; %Desplazamientos en el tiempo del portico E en
cada piso
%
%
%%Desplazamientos maximos y minimos de cada portico en cada piso en [cm]
% ueAMax = [max(u.eA(1,:));max(u.eA(2,:));max(u.eA(3,:))].*100;
% ueEMax = [max(u.eE(1,:));max(u.eE(2,:));max(u.eE(3,:))].*100;
%
% ueAMin = [min(u.eA(1,:));min(u.eA(2,:));min(u.eA(3,:))].*100;
% ueEMin = [min(u.eE(1,:));min(u.eE(2,:));min(u.eE(3,:))].*100;
%
%
%%Desplazamientos Relativos Maximos y Minimos de cada portico en cada piso en en
tiempo en [cm]
%

```

```

% uRel.eAMax = [ueAMax(1,:);ueAMax(2,)-ueAMax(1,:); ueAMax(3,)-ueAMax(2,:)];
%Desplazamientos Rel en el tiempo del portico A en cada piso
% uRel.eEMax = [ueEMax(1,:); ueEMax(2,)-ueEMax(1,:); ueEMax(3,)-ueEMax(2,:)];
%Desplazamientos Rel en el tiempo del portico E en cada piso
%
% uRel.eAMin = [ueAMin(1,:);ueAMin(2,)-ueAMin(1,:); ueAMin(3,)-ueAMin(2,:)];
%Desplazamientos Rel en el tiempo del portico A en cada piso
% uRel.eEMin = [ueEMin(1,:); ueEMin(2,)-ueEMin(1,:); ueEMin(3,)-ueEMin(2,:)];
%Desplazamientos Rel en el tiempo del portico E en cada piso
%
%
% %Relación de Irregularidad Torsional en X
%
% IITxMax = [max(uRel.eAMax(1,:),uRel.eEMax(1,:))
./((uRel.eAMax(1,)+uRel.eEMax(1,))./2)
%      max(uRel.eAMax(2,:),uRel.eEMax(2,)) ./((uRel.eAMax(2,)+uRel.eEMax(2,))./2)
%      max(uRel.eAMax(3,:),uRel.eEMax(3,)) ./((uRel.eAMax(3,)+uRel.eEMax(3,))./2)];
% IITxMin = [min(uRel.eAMin(1,:),uRel.eEMin(1,:))
./((uRel.eAMin(1,)+uRel.eEMin(1,))./2)
%      min(uRel.eAMin(2,:),uRel.eEMin(2,)) ./((uRel.eAMin(2,)+uRel.eEMin(2,))./2)
%      min(uRel.eAMin(3,:),uRel.eEMin(3,)) ./((uRel.eAMin(3,)+uRel.eEMin(3,))./2)];
%
% IITx_2 = max(IITxMax, IITxMin);
%
%
% %Respuestas de desplazamientos por GDL en el tiempo, SISMO Y
%
% [AAA,BBB,CCC,DDD,GAM] = rigid_diap2D(MM,KKg,zeta);
% sys_str = ss(AAA,BBB,CCC,DDD);
% load('elcentro.txt');
% dt = 0.02; %Tiempo de muestreo acelerograma
% dd = elcentro/(100); %Acelerograma en [m/seg^2]
% tt = (0:dt:(length(dd)-1)*dt)'; %Vector Tiempo
% X0 = zeros(size(sys_str.A,1),1); %condiciones iniciales
% [yyy,~,~,~] = lsim(sys_str,dd,tt,X0); %yyy = desplaz, aceler totales, aceler rel,
derivadas
% %en el tiempo
% %Desplazamientos de cada portico en cada piso
% U = [(yyy(:,1:9))]; %Desplazamientos en el tiempo de cada diafragma
%
% u.e1 = T.e1' * U; %Desplazamientos en el tiempo del portico 1 en cada
piso
% u.e2 = T.e2' * U; %Desplazamientos en el tiempo del portico 2 en cada
piso
% u.e3 = T.e3' * U; %Desplazamientos en el tiempo del portico 3 en cada
piso
% u.e4 = T.e4' * U; %Desplazamientos en el tiempo del portico 4 en cada
piso
% u.e5 = T.e5' * U; %Desplazamientos en el tiempo del portico 5 en cada
piso
%
% %Desplazamientos maximos y minimos de cada portico en cada piso en [cm]
% ue1Max = [max(u.e1(1,:));max(u.e1(2,:));max(u.e1(3,))].*100;
% ue5Max = [max(u.e5(1,:));max(u.e5(2,:));max(u.e5(3,))].*100;
%
% ue1Min = [min(u.e1(1,:));min(u.e1(2,:));min(u.e1(3,))].*100;
% ue5Min = [min(u.e5(1,:));min(u.e5(2,:));min(u.e5(3,))].*100;

```

```

%
%
% %Desplazamientos Relativos Maximos y Minimos de cada portico en cada piso en en
tiempo en [cm]
%
% uRel.e1Max = [ue1Max(1,);ue1Max(2,)-ue1Max(1,); ue1Max(3,)-ue1Max(2,)];
%Desplazamientos Rel en el tiempo del portico A en cada piso
% uRel.e5Max = [ue5Max(1,); ue5Max(2,)-ue5Max(1,); ue5Max(3,)-ue5Max(2,)];
%Desplazamientos Rel en el tiempo del portico E en cada piso
%
% uRel.e1Min = [ue1Min(1,);ue1Min(2,)-ue1Min(1,); ue1Min(3,)-ue1Min(2,)];
%Desplazamientos Rel en el tiempo del portico A en cada piso
% uRel.e5Min = [ue5Min(1,); ue5Min(2,)-ue5Min(1,); ue5Min(3,)-ue5Min(2,)];
%Desplazamientos Rel en el tiempo del portico E en cada piso
%
% %Relación de Irregularidad Torsional en Y
%
% IITyMax = [max(uRel.e1Max(1,:),uRel.e5Max(1,:))
./((uRel.e1Max(1,)+uRel.e5Max(1,))./2)
%      max(uRel.e1Max(2,:),uRel.e5Max(2,:)) ./((uRel.e1Max(2,)+uRel.e5Max(2,))./2)
%      max(uRel.e1Max(3,:),uRel.e5Max(3,:))
./((uRel.e1Max(3,)+uRel.e5Max(3,))./2)];
% IITyMin = [min(uRel.e1Min(1,:),uRel.e5Min(1,:))
./((uRel.e1Min(1,)+uRel.e5Min(1,))./2)
%      min(uRel.e1Min(2,:),uRel.e5Min(2,:)) ./((uRel.e1Min(2,)+uRel.e5Min(2,))./2)
%      min(uRel.e1Min(3,:),uRel.e5Min(3,:)) ./((uRel.e1Min(3,)+uRel.e5Min(3,))./2)];
%
% IITy_2 = max(IITyMax, IITyMin);

%Excentricidad estatica

Ex = XCM - XCR_final

Ey = YCM - YCR_final

IITx

IITy

PMTorsional = [PM(1:2,3);0]

end

```