

Ing. Juan Manuel  
Hernández Martínez

Optimización de vigas de concreto postensado mediante  
el algoritmo de enjambre de partículas

2026



Universidad Autónoma de Querétaro  
Facultad de Ingeniería

**OPTIMIZACIÓN DE VIGAS DE CONCRETO POSTENSADO  
MEDIANTE EL ALGORITMO DE ENJAMBRE DE PARTÍCULAS**

Tesis

Que como parte de los requisitos para  
obtener el Grado de

Maestro en Ciencias (Estructuras)

Presenta:

Ing. Juan Manuel Hernández Martínez

Dirigido por:

Dr. Luis Francisco Pérez Moreno

Querétaro, Qro., junio de 2026

La presente obra está bajo la licencia:  
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

### Usted es libre de:

**Compartir** — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

### Bajo los siguientes términos:



**Atribución** — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



**NoComercial** — Usted no puede hacer uso del material con [propósitos comerciales](#).



**SinDerivadas** — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

**No hay restricciones adicionales** — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

### Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.



Universidad Autónoma de Querétaro  
Facultad de Ingeniería  
Maestría en Ciencias (Estructuras)

Optimización de vigas de concreto postensado mediante el  
algoritmo de enjambre de partículas

Tesis

Que como parte de los requisitos para obtener el Grado de  
Maestro en Ciencias (Estructuras)

Presenta:

Ing. Juan Manuel Hernández Martínez

Dirigido por:

Dr. Luis Francisco Pérez Moreno

Dr. Luis Francisco Pérez Moreno  
Presidente

Dr. Miguel Ángel Pérez Lara y Hernández  
Secretario

Dr. Enrique Rico García  
Vocal

Dr. Iván Fermín Arjona Catzim  
Suplente

M. en C. Joaquín Noriega Jiménez  
Suplente

Centro Universitario, Querétaro, Qro.  
Junio de 2026  
México

## **Dedicatorias**

Dedico este trabajo, en primer lugar, a mi familia, por ser mi mayor fuente de fortaleza e inspiración. Por acompañarme en cada etapa de mi formación y por su apoyo incondicional en los momentos más difíciles y también en los mejores momentos, este logro también les pertenece.

A mi pareja, por su cariño, sus palabras de aliento y por recordarme siempre la importancia de no rendirme. Su presencia ha sido fundamental en este camino.

Dedico también este trabajo a mis profesores, quienes a lo largo de mi formación compartieron generosamente sus conocimientos, experiencia y vocación.

Finalmente, dedico esta tesis a todos aquellos que de una u otra manera estuvieron presentes durante este proceso.

## **Agradecimientos**

Expreso mi más sincero agradecimiento a mi asesor de tesis, Dr. Luis Francisco Pérez Moreno, por su guía, paciencia, compromiso y valiosas observaciones durante el desarrollo de este trabajo.

Agradezco también a los miembros de mi sínodo, por el tiempo dedicado a la revisión de este trabajo, así como por sus comentarios, observaciones y aportaciones que contribuyeron a mejorar la calidad de esta tesis.

A la Universidad Autónoma de Querétaro por brindarme el espacio, la formación y las oportunidades necesarias para desarrollarme académicamente. Me siento orgulloso de haber formado parte de esta institución.

También agradezco a CONACYT por el apoyo otorgado mediante la beca, la cual hizo posible la realización de mis estudios y el desarrollo de esta investigación.

## INDICE

<b>I. INTRODUCCIÓN .....</b>	<b>15</b>
PLANTEAMIENTO DEL PROBLEMA.....	17
JUSTIFICACIÓN.....	19
<b>II. ANTECEDENTES.....</b>	<b>20</b>
OPTIMIZACIÓN ESTRUCTURAL.....	20
<i>Minimización de daño estructural.....</i>	<i>20</i>
<i>Optimización de costos.....</i>	<i>21</i>
<i>Emisiones de CO<sub>2</sub>.....</i>	<i>22</i>
PRESFUERZO.....	24
<i>Aspectos clave del presfuerzo.....</i>	<i>25</i>
<i>Comportamiento sísmico de estructuras presforzadas.....</i>	<i>26</i>
<i>Sostenibilidad: Reducciones de emisiones de CO<sub>2</sub>.....</i>	<i>27</i>
<b>III. FUNDAMENTACIÓN TEÓRICA. ....</b>	<b>29</b>
PRESFUERZO.....	29
<i>Postensado.....</i>	<i>30</i>
<i>Pérdidas.....</i>	<i>31</i>
<i>Etapas.....</i>	<i>31</i>
<i>Esfuerzos permisibles.....</i>	<i>33</i>
ESTADOS LÍMITE.....	34
<i>Estado límite de servicio.....</i>	<i>34</i>

<i>Estados límite últimos</i> .....	35
<i>Flexión</i> .....	35
<i>Cortante</i> .....	42
OPTIMIZACIÓN.....	46
<i>Conceptos Básicos</i> .....	46
<i>Métodos metaheurísticos</i> .....	47
<i>Algoritmo de enjambre de partículas</i> .....	47
<i>Fundamento matemático</i> .....	48
<i>Variantes del PSO</i> .....	49
<i>Factores Clave del PSO</i> .....	49
<i>Comparación con Otros Métodos</i> .....	50
<b>IV. HIPÓTESIS Y OBJETIVOS.....</b>	<b>51</b>
HIPÓTESIS .....	51
OBJETIVO GENERAL.....	51
OBJETIVOS ESPECÍFICOS .....	51
<b>V. METODOLOGÍA.....</b>	<b>52</b>
ETAPA 1. MODELACIÓN PARAMÉTRICA DE CASOS DE ESTUDIO Y ANÁLISIS	
ESTRUCTURAL .....	52
<i>Casos de estudio</i> .....	52
<i>Dimensionamiento y materiales</i> .....	53
<i>Cargas y combinaciones de carga</i> .....	55

<i>Viga de estudio</i> .....	57
<i>Consideraciones del presfuerzo</i> .....	58
ETAPA 2. DISEÑO ESTRUCTURAL SIN Y CON PRESFUERZO OPTIMIZADO.....	59
<i>Diseño estructural sin presfuerzo</i> .....	59
<i>Diseño estructural con presfuerzo</i> .....	60
<i>Optimización</i> .....	62
ETAPA 3. CUANTIFICACIÓN DE MATERIAL Y COMPARACIÓN DE RESULTADOS.....	76
<b>VI. RESULTADOS Y DISCUSIÓN.....</b>	<b>77</b>
EVOLUCIÓN DE LA FUNCIÓN OBJETIVO EN CASOS DE ESTUDIO.....	77
SECCIÓN TRANSVERSAL Y CANTIDAD DE ACERO OBTENIDOS.....	88
<b>VII. CONCLUSIONES Y RECOMENDACIONES. ....</b>	<b>92</b>
<b>VIII. BIBLIOGRAFÍA.....</b>	<b>94</b>
<b>ANEXOS .....</b>	<b>97</b>
ANEXO A. TABLAS DE RESULTADOS Y ESFUERZOS COMPLEMENTARIOS .....	98
ANEXO B. CODIGO FUENTE EN PYTHON.....	103
<i>Código para generación de casos de estudio</i> .....	103
<i>Código para análisis estructural sin presfuerzo</i> .....	104
<i>Código para generación de modelo inicial a usar en algoritmo de PSO</i> .....	113
<i>Código para extracción de resultados de cada análisis en PSO</i> .....	121

<i>Funciones adicionales utilizadas en algoritmo de PSO.....</i>	<i>125</i>
<i>Código para cuantificación de acero considerando presfuerzo.....</i>	<i>127</i>
<i>Código para cuantificación de acero sin presfuerzo.....</i>	<i>129</i>
<i>Código para cuantificación de concreto .....</i>	<i>130</i>
<i>Código para diseño de cortante sin presfuerzo.....</i>	<i>131</i>
<i>Código para diseño a flexión sin presfuerzo .....</i>	<i>134</i>
<i>Código para revisión de desplazamientos.....</i>	<i>135</i>
<i>Código para diseño por esfuerzos permisibles .....</i>	<i>136</i>
<i>Código para diseño de cortante con presfuerzo.....</i>	<i>140</i>
<i>Código para diseño a flexión con presfuerzo .....</i>	<i>144</i>
<i>Código para algoritmo general de análisis sin presfuerzo.....</i>	<i>146</i>
<i>Código para algoritmo general de análisis con presfuerzo y PSO .....</i>	<i>148</i>

## Índice de tablas

<b>Tabla 1. Investigaciones realizadas sobre optimización de sistemas estructurales .....</b>	<b>23</b>
Tabla 2. Investigaciones realizadas sobre estructuras presforzadas.....	24
Tabla 3. Esfuerzos de tensión y compresión límites en concreto, extraída de (NTC, 2023a).....	34
Tabla 4. Esfuerzos de tensión límites en el acero, extraída de (NTC, 2023a) .....	34
Tabla 5. Casos de estudio .....	53
Tabla 6. Peralte mínimo para vigas sin presfuerzo, extraída de (NTC, 2023a).....	54
Tabla 7. Estado de cargas, para losa de entrepiso .....	55
Tabla 8. Estado de cargas, para losa de azotea.....	55
Tabla 9. Cargas vivas según uso de edificio.....	56
Tabla 10 (a). Factores de carga para combinaciones de carga, extraído de (NTC, 2023b).....	56
Tabla 11. Geometría de vigas de espacio de soluciones.....	62
Tabla 12. Máxima cantidad de torones de acuerdo al ancho de trabe.....	63
Tabla 13. Pesos para penalización por desplazamientos .....	65
Tabla 14. Pesos para penalización por cortante .....	66
Tabla 15. Pesos para penalización por flexión.....	66
Tabla 16. Pesos para penalización por esfuerzos permisibles. ....	67
Tabla 17. Procesos para análisis sin presfuerzo.....	72

Tabla 18. Procesos para análisis con presfuerzo optimizado con PSO .....	73
Tabla 19. Sección transversal de viga para ambos diseños .....	88
Tabla 20. Resultados de acero longitudinal, transversal y torones para ambos diseños.....	90
Tabla A 1. Resultados principales para análisis sin presfuerzo .....	98
Tabla A 2. Resultados técnicos para análisis sin presfuerzo .....	99
Tabla A 3. Resultados principales para análisis presforzado con PSO. ....	100
Tabla A 4. Resultados técnicos para análisis presforzado con PSO .....	101
Tabla A 5. Análisis de esfuerzos permisibles para caso 5 .....	102

## Índice de gráficas

Gráfica 1. Métodos de optimización más usados en ingeniería estructural. ....	50
Gráfica 2. Caso 1; Hospital, 3 niveles, 9 m .....	77
Gráfica 3. Caso 2; Hospital, 3 niveles, 11 m .....	78
Gráfica 4. Caso 3; Hospital, 3 niveles, 13 m .....	78
Gráfica 5. Caso 4; Hospital, 5 niveles, 9 m .....	79
Gráfica 6. Caso 5; Hospital, 5 niveles, 11 m .....	79
Gráfica 7. Caso 6; Hospital, 5 niveles, 13 m .....	80
Gráfica 8. Caso 7; Hospital, 7 niveles, 9 m .....	80
Gráfica 9. Caso 8; Hospital, 7 niveles, 11 m .....	81
Gráfica 10. Caso 9; Hospital, 7 niveles, 13 m .....	81
Gráfica 11. Caso 10; Estacionamiento, 3 niveles, 9 m.....	82
Gráfica 12. Caso 11; Estacionamiento, 3 niveles, 11 m.....	82
Gráfica 13. Caso 12; Estacionamiento, 3 niveles, 13 m.....	83
Gráfica 14. Caso 13; Estacionamiento, 5 niveles, 9 m.....	83
Gráfica 15. Caso 14; Estacionamiento, 5 niveles, 11 m.....	84
Gráfica 16. Caso 15; Estacionamiento, 5 niveles, 13 m.....	84
Gráfica 17. Caso 16; Estacionamiento, 7 niveles, 9 m.....	85
Gráfica 18. Caso 17; Estacionamiento, 7 niveles, 11 m.....	85
Gráfica 19. Caso 18; Estacionamiento, 7 niveles, 13 m.....	86

Gráfica 20. Comparativa de volumen de concreto para diseño sin presfuerzo vs diseño con presfuerzo y PSO .....	89
Gráfica 21. Porcentaje de ahorro de concreto .....	89
Gráfica 22. Comparativa de cantidad de acero para viga de estudio con diseño convencional vs diseño presforzado con PSO.....	91
Gráfica 23. Porcentaje de ahorro de acero .....	91

## Índice de figuras

Figura 1. Diagramas de momento con diferentes excentricidades .....	29
Figura 2. Trayectoria de los cables .....	31
Figura 3. Esfuerzos flectores en una viga en sección simple y compuesta .....	33
Figura 4. Resistencia a flexión de elemento en sección simple. ....	36
Figura 5. Secciones a revisar por cortante.....	43
Figura 6. Contribución de torones al cortante resistente.....	45
Figura 7. Algoritmos bioinspirados.....	47
Figura 8. Viga de estudio .....	58
Figura 9. Trazo y aplicación de postensado mediante herramienta integrada de SAP 2000.....	58
Figura 10. Metodología general de este trabajo de investigación .....	70
Figura 11. Diagrama de flujo de algoritmo de PSO.....	71

## RESUMEN

El diseño estructural de vigas de concreto postensado se ha convertido en una técnica altamente utilizada en edificios con grandes claros, debido a sus ventajas para controlar deformaciones y minimizar uso de materiales al manejar secciones estructurales más esbeltas. Sin embargo, los métodos convencionales de diseño cuentan con enfoques deterministas que no se dan a la tarea de explorar un mayor número de soluciones que puedan resultar más eficientes que la obtenida. Cabe mencionar, que, a pesar de lo mencionado anteriormente, el diseño convencional de una viga incluyendo postensado no siempre resulta en el mayor ahorro de material, es por ello que en la presente investigación se buscará optimizar el diseño estructural de vigas de concreto postensado mediante la aplicación del algoritmo de Optimización de Enjambre de Partículas (PSO), encontrando las mejores configuraciones estructurales para las vigas de concreto en función de las restricciones normativas. Se llevará a cabo el modelado paramétrico de un edificio prototipo con claros mayores a 9 metros para evaluar el comportamiento estructural con y sin presfuerzo de las vigas que conforman un eje de crujeas representativo. A través del PSO, se optimizarán variables como la sección transversal, acero de refuerzo y cantidad de presfuerzo comparando los resultados con métodos tradicionales de diseño, esperando una reducción de al menos el 20% del material mencionando, sin comprometer la seguridad estructural.

**Palabras clave:** Optimización, Algoritmo de Enjambre de partículas, Concreto postensado, Reducción de material, Diseño estructural.

## ABSTRACT

The structural design of post-tensioned concrete beams has become highly used in buildings with large spans, due to its advantages in controlling deformations and minimizing material usage by handling slender structural sections. However, conventional design methods rely on deterministic approaches that do not explore a larger number of solutions that may be more efficient than the one obtained.

It is worth mentioning that, in spite of the above, the conventional design of a beam, including post-tensioning, does not always result in the greatest material savings, which is why this research will seek to optimize the structural design of beams by applying the Particle Swarm Optimization Algorithm (PSO), finding the best structural configurations for concrete beams according to the normative constraints.

The parametric modeling of a prototype building with spans greater than 9 meters will be done in order to evaluate the structural behavior with and without prestressed reinforcement of the beams forming a representative bay axis. Through the PSO algorithm, variables such as cross section, reinforcing steel and amount of prestressing will be optimized, by comparing the results with traditional design methods, expecting a reduction of at least 20% of the mentioned material, without compromising structural safety.

**Key words:** Optimization, Particle Swarm Algorithm, Post-tensioned concrete, Material reduction, Structural design.

## I. Introducción

La optimización en ingeniería estructural es el proceso en el que se busca el diseño más eficiente y económico para una estructura. Esto implica encontrar el equilibrio entre utilizar la menor cantidad de material posible y garantizar que la estructura sea lo suficientemente segura para resistir todas las fuerzas a las que estará expuesta durante su periodo de funcionamiento, también conocido como vida útil (Asghshahr, 2021).

Se pretende la búsqueda de una solución que no solo cumpla con las normas de seguridad, sino que también permita ahorrar tiempo, dinero y materiales, asegurando que cada componente estructural esté calculado para maximizar su rendimiento y minimizar costos.

La optimización y el uso de técnicas constructivas como el concreto postensado, elementos pretensados y la adición de fibras de carbono están estrechamente relacionados, ya que se busca maximizar el desempeño de una estructura mientras se minimizan los recursos empleados. Al combinar estos elementos, se logra diseñar estructuras más ligeras, con mayor capacidad para resistir cargas, y con un uso más eficiente de los materiales, lo cual se traduce en un ahorro económico y un mejor aprovechamiento del espacio (Quintanilla Huayta, 2016).

Mientras que la optimización se enfoca en encontrar el equilibrio ideal entre seguridad, costos y eficiencia en el uso de materiales, el concreto postensado es una técnica utilizada en la construcción para hacer que el concreto sea más resistente a las tensiones y fuerzas a las que está sometido, sin necesidad de aumentar significativamente el tamaño de los elementos estructurales (Tuncer et al., 2025)

Esta técnica funciona colocando cables de acero dentro de la estructura y tensándolos después de que el concreto se ha endurecido, lo cual ayuda a que el

elemento estructural resista mejor las fuerzas que se le aplican. Sin embargo, es complicado poder determinar de manera precisa la fuerza inicial que está actuando gracias al tensado, así como la obtención de las pérdidas de presfuerzo que se tienen, lo cual es un desafío matemático y normativo para este tipo de vigas (Park et al., 2026). Esta técnica se usa mucho en puentes, edificios altos y grandes losas, ya que permite construir elementos más delgados y livianos, pero igualmente resistentes.

En el caso de las vigas en marcos estructurales, el postensado permite que estos elementos puedan resistir mayores esfuerzos sin necesidad de incrementar su tamaño. Esto permite cubrir mayores claros sin necesidad de columnas adicionales, lo cual hace que los edificios o estructuras sean más espaciosos y eficientes en cuanto al uso de materiales.

Para esta investigación se empleó la herramienta computacional del Interfaz de Programación de Aplicaciones (API) del software de diseño estructural SAP 2000 con el lenguaje de programación de Python, orientándolo hacia la optimización de trabes concreto postensado, de tal modo que se automatizaron diversos procesos para analizar varios casos de estudio.

La optimización se realizó mediante el algoritmo metaheurístico de Enjambre de partículas (PSO), el cual es un método originalmente propuesto por Kennedy & Eberhart (2002), que se inspira en el comportamiento social de conjuntos de especies como parvadas o cardúmenes. Este método es altamente empleado en problemas de esta categoría por su simplicidad de implementación, bajo costo computacional y capacidad de convergencia hacia soluciones cercanas al óptimo global (Velázquez & Mejía, 2016).

En este trabajo de investigación se logró reducir el peso de concreto y de acero al menos entre el 15 y 30% para las vigas de estudio mediante la metodología propuesta en comparación a los diseños estructurales tradicionales que no

contemplan el efecto del postensado y no están sujetos a un proceso de optimización mediante algoritmos metaheurísticos.

Se presentan los trabajos realizados por diversos autores relacionados con el tema de este trabajo de investigación en el segundo capítulo, además de mostrar una tabla comparativa sobre su metodología aplicada, objetivo de investigación y resultados de forma breve, para posteriormente hacer la distinción de este trabajo con respecto a los expuestos en dicho capítulo. Luego, en el tercer capítulo, se describe la fundamentación teórica que sienta las bases de este trabajo, en cuanto a los dos temas principales que son el presfuerzo y la optimización.

El capítulo cuarto define la hipótesis mientras que el quinto muestra los objetivos, tanto generales, como particulares que definen el orden o secuencia que se llevó a cabo durante este trabajo. Después, en el sexto capítulo de metodología, se muestra el desglose de actividades, consideraciones y herramientas utilizadas para poder ir cumpliendo cada uno de los objetivos necesarios para la culminación de este trabajo para que, de esta manera, en el séptimo y octavo capítulo, respectivamente, se muestren los resultados obtenidos y las conclusiones pertinentes.

### **Planteamiento del problema**

En la práctica profesional actual, los métodos de diseño tradicionales suelen requerir numerosas iteraciones manuales, las cuales dependen de la experiencia que tenga el ingeniero para proponer la cantidad de acero de refuerzo, la distribución del postensado, o las secciones transversales de los elementos. Esto puede traducirse en diseños conservadores que aumentan el requerimiento de materiales, además de largas jornadas de trabajo para la etapa de diseño.

Todo esto ocasiona que haya un coste mayor para la obra por causa de un excesivo uso de recursos, tomando en cuenta desde el encofrado, concreto, acero, hasta la mano de obra requerida para llevar a cabo el proyecto, mismos elementos

que afectan directamente el impacto ambiental por las emisiones de CO<sub>2</sub> que los materiales o la misma maquinaria involucran.

Es por ello que la investigación moderna busca acabar con la dependencia empírica guiando este campo de análisis a modelos computacionales que sean capaces de predecir con gran exactitud el comportamiento de los miembros estructurales (Wen et al., 2026).

Cabe mencionar, que actualmente se tienen ciertas limitantes arquitectónicas en lo que a grandes claros refiere, debido a la imposibilidad de darle solución a estas propuestas mediante concreto con refuerzo simple, siendo el postensado la propuesta ideal para comenzar a considerar este tipo de estructuras como viables.

En lugar de llevar a cabo las prácticas actuales se deberían utilizar procedimientos que además de pensar en la integridad estructural, no dejen de lado factores importantes como el uso de recursos y el impacto que esto conlleva al medio ambiente o el coste de la obra, siendo la implementación de algoritmos de optimización una manera eficiente para combatir estas problemáticas con el fin obtener una mejor propuesta desde la concepción del proyecto estructural de un edificio hasta su construcción.

## **Justificación.**

Actualmente, después del agua, el concreto es el material de construcción más utilizado en el mundo, debido a su resistencia, durabilidad y costo accesible. La industria cementera ha tenido un constante crecimiento de tal manera que en el año 2020 se tuvo una demanda global de 4.1 mil toneladas de cemento, esperando que esta cifra siga en aumento por el gran desarrollo que están teniendo tanto los países emergentes como los ya desarrollados. Por otro lado, según IEA (2022), el sector del concreto es responsable del 7% de las emisiones globales de CO<sub>2</sub>, por lo que cualquier reducción en el uso de grandes volúmenes de este puede tener un impacto positivo en términos medioambientales.

En este contexto, resulta fundamental aplicar estudios enfocados en la optimización de vigas de concreto postensado, pues esto puede ayudar a obtener diseños más económicos, seguros y sostenibles, especialmente para infraestructuras a gran escala.

Otro punto importante es la integración de metodologías de optimización modernas, como lo son los algoritmos metaheurísticos, ya que estos ayudan a maximizar los beneficios del postensado, ayudando a optimizar tanto la cantidad de acero y concreto, como el comportamiento estructural en comparación de técnicas convencionales de diseño de elementos de concreto reforzado.

Los resultados de este trabajo sientan las bases para buscar diseños más eficientes, que además de priorizar la economía del proyecto, no comprometan la seguridad estructural del edificio pretendiendo que se tenga una aplicación práctica en cualquier diseño para edificios de concreto con grandes claros, tales como hospitales, escuelas, centros comerciales, oficinas, departamentos, entre otros, para los cuales, la aplicación del concreto postensado beneficie tanto en costos como en tiempos de construcción.

## **II. Antecedentes**

En este trabajo es imprescindible contar con la referencia de investigaciones en cuanto a los dos temas pilares de esta tesis, los cuales son la optimización estructural y el presfuerzo, a continuación, se muestran algunos de los trabajos ya realizados que engloban los campos ya mencionados.

### **Optimización estructural**

Según (Quintanilla Huayta, 2016), en el ámbito de la optimización en estructuras, se han trabajado diferentes enfoques a lo largo del tiempo, de los cuales se pueden identificar algunas muy importantes, tales como el control de daño estructural, la minimización de costos y la sostenibilidad, los cuales serán descritos a continuación.

#### **Minimización de daño estructural**

El primer enfoque mencionado está dirigido a minimizar el daño estructural que se tendrá durante eventos sísmicos y busca reducir la concentración local de esfuerzos en algunos elementos, procurando que los efectos sísmicos se distribuyan de una manera uniforme a nivel global en la estructura.

En la investigación de (Dong et al., 2024), se formula la función objetivo dependiendo de la respuesta sísmica estructural, utilizando la variable de desplazamiento que hay en entrepisos, IDR (*Interstorey Drift Ratio*), esta misma variable la relaciona con las rotaciones en rótulas plásticas, la demanda de ductilidad por piso, el índice de daño local o global, y el coeficiente de variación de la respuesta sísmica.

Dentro de las investigaciones realizadas en este enfoque destaca la de (Hajirasouliha et al., 2012), donde se implementó el diseño sísmico de estructuras de concreto basado en desempeño, en el cual, se realizó la distribución de refuerzo longitudinal mediante un análisis dinámico no lineal y modelos de elementos finitos

de los elementos más fuertes a los más débiles, esto con el fin de uniformizar las deformaciones y reducir el daño global de la estructura bajo efectos sísmicos.

En esta investigación se evaluaron los daños estructurales y las rótulas plásticas en los extremos de las secciones mediante el índice de daño de Park y Ang modificado, a medida que se iba haciendo la redistribución de acero. Finalmente se logró reducir hasta un 30% el daño global de la estructura en comparación con un diseño tradicional y se logró optimizar el refuerzo longitudinal teniendo una reducción de hasta el 50% en algunos casos.

### **Optimización de costos**

Por otro lado, se considera a la minimización de costos como uno de los enfoques más populares dentro de la optimización estructural, teniendo como principal objetivo la reducción de los costos iniciales de construcción, al minimizar la cantidad de material utilizado, como lo son los volúmenes de concreto y el peso del refuerzo de acero (Dong et al., 2024).

En diversas investigaciones también se analizan factores tales como los costos indirectos de obra, la cantidad de encofrado, la mano de obra, la maquinaria, entre otros, a diferencia de los métodos que solo se centran en disminuir las secciones transversales de los elementos estructurales y su acero de refuerzo (Nouri et al., 2020).

El uso de algoritmos genéticos por (Asghshahr, 2021) para efectuar la optimización de marcos de concreto reforzado tuvo como objetivo el uso eficiente de materiales sin descuidar la seguridad estructural, hizo un enfoque basado en confiabilidad por medio de un análisis probabilístico, donde propuso el uso de algoritmos genéticos para encontrar la configuración óptima de refuerzos tomando en cuenta el efecto sísmico. Su investigación obtuvo que al combinar estos métodos se puede reducir la cantidad de acero en comparación de los métodos tradicionales de diseño, manteniendo un nivel adecuado de confiabilidad estructural.

## **Emisiones de CO<sub>2</sub>**

La optimización del diseño de concreto reforzado puede no solo reducir costos, sino también minimizar los impactos ambientales asociados con la construcción y operación de las estructuras. Este enfoque multiobjetivo es crucial para asegurar tanto la eficiencia económica como la sostenibilidad a lo largo del ciclo de vida de una estructura (Nouri et al., 2020).

En un estudio realizado por (Mergos, 2018), se buscó desarrollar diseños sísmicos óptimos de marcos de concreto reforzado para minimizar las emisiones de CO<sub>2</sub> embebidas desde su fabricación y compararlos con diseños optimizados basados en el costo de construcción, apoyándose con el uso de algoritmos genéticos para el diseño de marcos de concreto reforzado. Evaluó distintos niveles de ductilidad, y aceleración sísmica de diseño, optimizando tanto el costo como las emisiones de CO<sub>2</sub> mediante la modificación de las secciones transversales de los elementos estructurales y propiedades de los aceros de refuerzo.

Los resultados obtenidos destacaron una reducción del 60% en las emisiones de CO<sub>2</sub> para regiones de sismicidad alta, con estructuras de ductilidad alta y media en comparación con estructuras de ductilidad baja. Además, se observó que las emisiones de CO<sub>2</sub> aumentan muy poco al usar clases de concreto de mayor resistencia, sin embargo, al usar un concreto más resistente, el refuerzo de acero requerido es menor, lo cual ayuda a disminuir las emisiones de CO<sub>2</sub>. Finalmente se concluye que el diseño sísmico que busca minimizar las emisiones de CO<sub>2</sub> puede reducir significativamente el impacto ambiental, en regiones de alta sismicidad.

Por otro lado, el estudio realizado por (Nouri et al., 2020) muestra otra metodología para el diseño de marcos de concreto reforzado evaluando el costo total del ciclo de vida de la estructura y los impactos ambientales de esta. Hace mención de la falta de consideración de los códigos de diseño actuales respecto al impacto ambiental y los costos de los daños durante la vida útil de la estructura. En su trabajo utilizó tres funciones objetivo, minimizar costos iniciales y de daños

esperados, minimizar costos de ciclo total de vida y equilibrar costos y el puntaje ambiental. Se obtuvo que al dar prioridad a la primera función objetivo mencionada se redujo un 9% el costo total de ciclo de vida, y que con las otras funciones objetivo el ciclo de vida baja un 4% y la puntuación de impacto ambiental baja un 5%. Concluyendo que un factor con gran influencia en la disminución de impactos ambientales es el refuerzo.

**Tabla 1. Investigaciones realizadas sobre optimización de sistemas estructurales**

<b>Referencia</b>	<b>Metodología</b>	<b>Objetivos</b>	<b>Resultados</b>
(Nouri et al., 2020)	Métodos de respuesta simplificada	Reducción de costos y huella ambiental durante el ciclo de vida para marcos de concreto reforzado bajo cargas sísmicas	Se logró reducir hasta un 60% las emisiones de CO <sub>2</sub> en diseños de alta ductilidad en comparación con los de baja ductilidad.
(Deb, 2011)	PSO	Reducción de peso estructural y control de derivas en edificios de acero con marcos resistentes a momento bajo cargas sísmicas	Se redujo el peso estructural hasta 6.65 toneladas usando el método PSO, y se redujo el valor de deriva a 0.0063 con PSO, frente a 0.0071 con NSGA-II.
(Asghshahr, 2021)	Algoritmos Genéticos (GA)	Optimización basada en confiabilidad estructural aplicado a marcos de concreto reforzado con diferentes configuraciones y condiciones de carga	Mejora en el índice de confiabilidad de hasta un 25%, de acuerdo a las condiciones de carga y configuración estructural.
(Mergos, 2018)	Algoritmos Genéticos (GA)	Minimizar emisiones de CO <sub>2</sub> para marcos de concreto reforzado de edificios de baja a mediana altura	Reducción de hasta 13% en emisiones de CO <sub>2</sub> sin comprometer desempeño estructural

Al analizar todos los estudios anteriores, además de los mencionados en la Tabla 1, se observa que en ninguno se aborda el tema del postensado en vigas de concreto con fines de optimización estructural. Este es un tema de gran utilidad pues permite reducir las secciones transversales de los elementos que componen una

estructura, así como ahorrar en la cantidad de volúmenes de material empleados, contribuyendo a la concepción de proyectos arquitectónicos más ambiciosos que busquen librar grandes claros haciendo uso de estructuras de concreto reforzado, todo esto gracias a la introducción de fuerzas internas que ayudan a contrarrestar las cargas aplicadas.

## Presfuerzo

Por otro lado, es necesario remarcar que se han realizado diversas investigaciones abordando las bondades que la técnica del presfuerzo nos proporciona, desde su aplicación para optimizar estructuras y hacer más eficiente el uso de materiales, hasta estudios dirigidos a su diseño que se adentran al modelado numérico en softwares especializados y a la influencia de la configuración de tendones y nivel de presfuerzo aplicado, es por ello que a continuación se resumen algunos de los trabajos relacionados a este campo de estudio en la Tabla 2, para posteriormente dar un contexto más detallado de lo que cada uno refiere.

**Tabla 2. Investigaciones realizadas sobre estructuras presforzadas.**

<b>Referencia</b>	<b>Metodología</b>	<b>Objetivos</b>	<b>Resultados</b>
(Kravanec et al., 2021)	Técnica de corte y modelado numérico 2D en ATENA.	Evaluar pérdidas de presfuerzo en puente postensado con 60 años de servicio.	Pérdidas de presfuerzo significativas por relajación de acero  Mediciones in situ con modelos numéricos para predecir las pérdidas con precisión
(Guo et al., 2017)	Celdas de carga, sensores de vibración (VWVG) y sensores elasto-magnéticos (EM). Monitoreo durante un año.	Desarrollar modelo mejorado para predecir pérdidas de presfuerzo a largo plazo en vigas postensadas.	Distribución de tendones, nivel de tensión inicial y grouting afectan las pérdidas modelo de predicción de pérdidas con margen de error del 10%

**Tabla 2. Continuación**

<b>Referencia</b>	<b>Metodología</b>	<b>Objetivos</b>	<b>Resultados</b>
(T. H. Wang et al., 2013)	Medición de frecuencias naturales usando impacto de martillo y acelerómetros.  Uso de un modelo matemático de Rayleigh.	Investigar la variación de la rigidez a flexión en vigas postensadas con tendones curvos y rectos bajo presfuerzo.	La frecuencia natural disminuye con el aumento del presfuerzo en vigas con tendones curvos, pero no cambia en vigas con tendones rectos.
(Quintanilla Huayta, 2016)	Ensayos experimentales sobre el comportamiento sísmico de vigas de concreto postensado con tendones adheridos y no adheridos.	Evaluar cómo el tipo de tendón influye en la resistencia sísmica de vigas de concreto postensado.	Las vigas con tendones adheridos tienen mayor resistencia sísmica y control de deformaciones que las de tendones no adheridos.  Conclusión: en zonas sísmicas, las vigas con tendones adheridos son más eficientes para resistir fuerzas sísmicas.
(Mergos, 2018)	Optimización con algoritmos genéticos.	Minimizar las emisiones de CO <sub>2</sub> de estructuras de concreto reforzado.	Reducción del 60% en las emisiones de CO <sub>2</sub> para estructuras de alta ductilidad en comparación con estructuras de baja ductilidad.

### **Aspectos clave del presfuerzo**

Algo a tomar en cuenta en el presfuerzo son las pérdidas, las cuales pueden ocurrir con el paso del tiempo debido a diversos factores. En este ámbito destaca la investigación realizada por (Kraľovanec et al., 2021), pues dirigió su investigación a la evaluación de las pérdidas de presfuerzo para un puente postensado en Eslovaquia, el cual, ya llevaba varias décadas en funcionamiento. Se utilizó la técnica de corte, la cual es un método no destructivo que sirve para medir las

pérdidas de presfuerzo y de esta forma tener un mejor entendimiento de cómo se comportan las vigas en su vida útil.

Finalmente obtuvo que se tenían significativamente altas pérdidas de presfuerzo en las vigas del puente, el cual, contaba con al menos 60 años de vida útil, sin embargo, el autor destacó que al tomar mediciones en sitio y combinarlas con modelación numérica, es posible predecir con mayor precisión las pérdidas de presfuerzo con el paso del tiempo, lo cual afirma su metodología como una herramienta de gran utilidad en el ámbito de la rehabilitación de estructuras existentes.

Las pérdidas de presfuerzo no solo requieren medirse después de los años, sino que también requieren un monitoreo con el paso constante del tiempo, es aquí donde (Guo et al., 2017) llevó a cabo un estudio experimental que buscó generar un modelo predictivo de las pérdidas en vigas de concreto postensado. La técnica utilizada consistió en un monitoreo que combinó celdas de carga, sensores de vibración y elasto-magnéticos, que le permitieron obtener una visión integral de los efectos ocasionados por el presfuerzo, tales como pérdidas por relajación de acero y contracción de concreto.

Este estudio concluyó que algunos de los factores que más influyen en la magnitud de las pérdidas de presfuerzo, son la distribución de los torones, el presfuerzo inicial y el grouting de los ductos para los tendones, de tal modo que gracias a los datos que recopiló, logró desarrollar un modelo mejorado para predecir las pérdidas cuyo margen de error es inferior al 10%, superando los modelos existentes.

### **Comportamiento sísmico de estructuras presforzadas**

En cuanto a la rigidez de las vigas de concreto postensado, (T. H. Wang et al., 2013) investigaron la forma en que el presfuerzo afecta a la rigidez a flexión de las vigas de concreto, llevando a cabo una medición de las frecuencias naturales de

varias vigas postensadas antes y después de la aplicación del presfuerzo, mediante un sistema de impacto de martillo y acelerómetros con los cuales registraron las vibraciones. Todo esto con el fin de determinar el grado de influencia de la configuración de los torones en cuanto su trayectoria recta o curva en la rigidez a flexión y el comportamiento dinámico de las vigas de concreto postensado.

Se obtuvo que las vigas cuyos tendones cuentan con trayectoria curva contaron con una disminución en su frecuencia natural una vez que se aplicó el presfuerzo, lo cual se infiere como una rigidez mayor en comparativa con las de torones en trayectoria recta que no presentaron cambios significativos en su frecuencia al aplicárseles el presfuerzo, dando lugar a resultados de gran valor para el análisis sísmico de estructuras de concreto postensado

Una vez adentrados en el tema sísmico de vigas postensadas, es importante mencionar a (Quintanilla Huayta, 2016), quien realizó un estudio experimental sobre el comportamiento sísmico de vigas de concreto postensado con torones adheridos y no adheridos, todo esto, con el fin medir la influencia de estos en la capacidad portante de las vigas ante fuerzas sísmicas.

Los resultados obtenidos mostraron una mayor resistencia sísmica para las vigas con torones adheridos, las cuales presentaron un mejor control de deformaciones que su contraparte. Esto es, debido la mejor transferencia de esfuerzos a través de toda la viga que se tienen al utilizar torones adheridos, concluyendo que es más recomendable el uso de vigas con torones adheridos en zonas de alta sismicidad.

### **Sostenibilidad: Reducciones de emisiones de CO<sub>2</sub>**

En otro contexto, el presfuerzo también está relacionado con la sostenibilidad en lo que a construcción refiere. Según (Mergos, 2018), el concreto postensado además de mejorar el desempeño estructural, ayuda a la reducción de emisiones

de CO<sub>2</sub> pues la reducción de cantidades de concreto y acero en los elementos estructurales conlleva a la disminución del impacto ambiental en la construcción.

Dicho estudio mostró una reducción de hasta el 60% de emisiones de CO<sub>2</sub> para estructuras presforzadas en zonas de alta demanda sísmica. Todo esto es crucial no solo para buscar la economía en la construcción, sino también para la construcción de estructuras que sean más amigables con el medio ambiente, lo cual, es algo que cada vez se busca con gran ambición en la construcción moderna.

El motivo de este trabajo es implementar un algoritmo metaheurístico, que ayude a explorar una gran cantidad de combinaciones variando la sección transversal de las trabes y la configuración del armado de acero con y sin presfuerzo, lo cual se comparó con un diseño tradicional sin considerar acero de presfuerzo realizado en el software comercial SAP 2000 mediante la API con el lenguaje de programación Python. Finalmente, se compararon estos resultados y se comprobó si realmente el acero postensado, aplicado con la metodología del presente trabajo, resulta en un sistema más conveniente considerando los volúmenes de concreto y aceros empleados y el tiempo dedicado al diseño.

### III. Fundamentación teórica.

Los dos pilares que sustentan este trabajo de investigación son la optimización y el presfuerzo, ambos aplicados a las vigas de concreto reforzado, por lo cual, en este apartado se introduce el marco teórico referente a ello.

#### Presfuerzo

El concreto con presfuerzo, tanto pretensado como postensado, consiste en la creación esfuerzos permanentes en un elemento estructural con el fin de aumentar su resistencia y su respuesta en cargas de servicio. Para esto, se colocan estratégicamente cables de acero de alta resistencia, los cuales se tensan con cierta fuerza antes o después del fraguado del concreto, posteriormente, se destensan los cables, liberando de esta manera esa tensión y ocasionando esfuerzos tanto de tensión como de compresión en distintas fibras del elemento de concreto (Quintanilla Huayta, 2016). En esta técnica influye en gran manera la excentricidad con la que se colocan los cables de presfuerzo con respecto al eje neutro, en la Figura 1 se aprecia como varían los diagramas de momento para tres situaciones diferentes (Hajirasouliha et al., 2012). En la viga A se coloca el acero en el eje neutro, para la viga B se coloca debajo del eje neutro y en la viga C lleva una trayectoria curva o catenaria, lo cual es el caso que se estudiará en este trabajo.

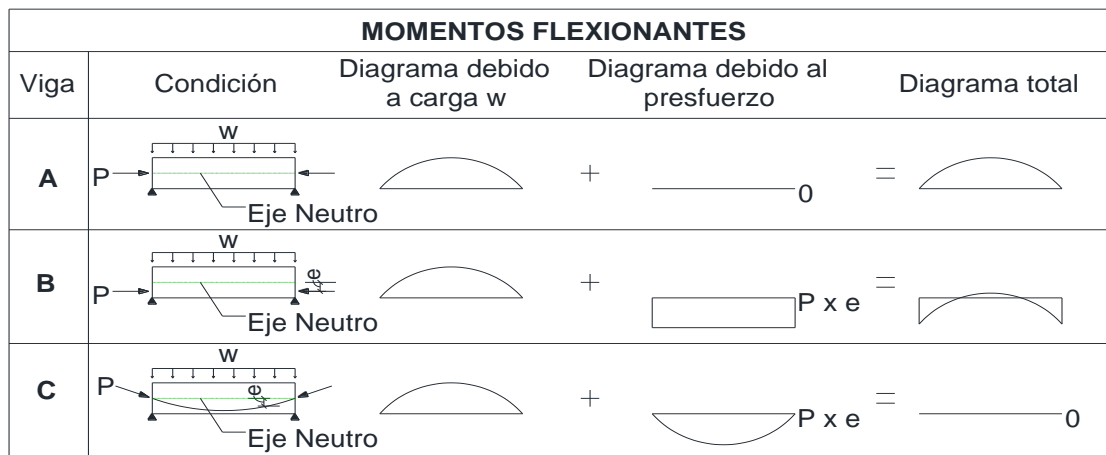


Figura 1. Diagramas de momento con diferentes excentricidades

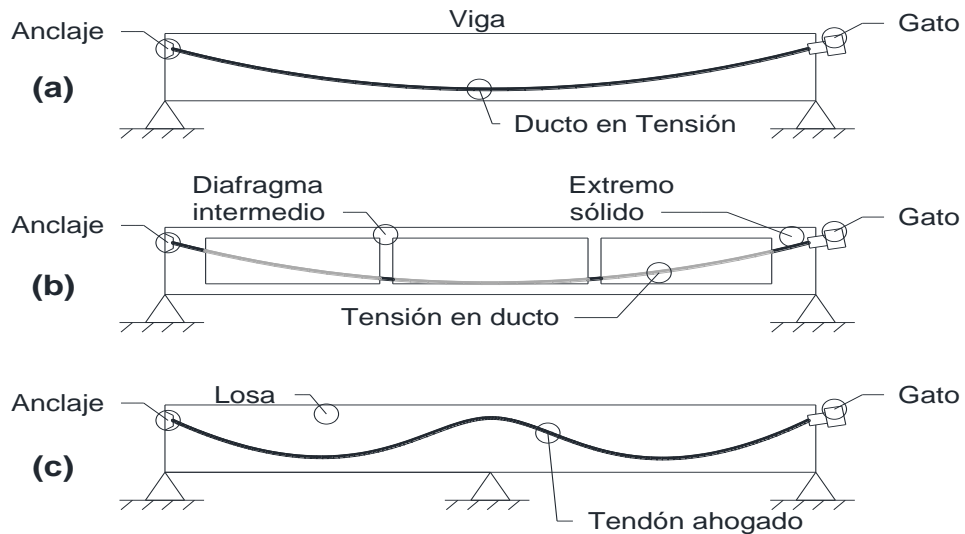
En este proceso, los elementos postensados permiten contrarrestar fuerzas de carga mediante un sistema de tendones de acero tensados y anclados tras el fraguado del concreto, lo cual incrementa su resistencia y eficiencia estructural. Los elementos postensados ofrecen ventajas controlando la deflexión y agrietamiento, permitiendo un uso óptimo de materiales de alta resistencia, y un diseño más esbelto y eficiente (Quintanilla Huayta, 2016). Las trayectorias de los torones pueden ser curvas para elementos hiperestáticos, de este modo se logra un diseño más eficiente que conlleva a estructuras más livianas y de mayor durabilidad, aptas para puentes, edificios de grandes claros, y techumbres industriales. Esto permite variar la excentricidad de los tendones, de modo que se adaptan las cargas y los esfuerzos para mejorar la respuesta estructural ante cargas gravitacionales y dinámicas (Alcocer, 2012).

### **Postensado**

La técnica del postensado consiste en tensar los torones de acero y anclarlos en los extremos del elemento una vez que el concreto fraguó y alcanzó la resistencia adecuada para soportar las cargas inducidas por el presfuerzo. El concreto utilizado es de alta resistencia, con una resistencia a la compresión ( $f'_c$ ) entre 350-500 kg/cm<sup>2</sup> y el acero de presfuerzo, como torones y cables, en este caso llevan forma de espiral o hélice torcido, alcanzando una resistencia a la ruptura ( $F_{sr}$ ) de 19,000 kg/cm<sup>2</sup> para grado 270K o 270,000 lb/in<sup>2</sup>, además de un módulo elástico que oscila entre 1'900,000 y 1'960,000 kg/cm<sup>2</sup>. Los torones se tensan hasta un 80% de su esfuerzo a la ruptura. Las características descritas de los materiales son esenciales para lograr la adherencia y resistencia requeridas en estructuras de gran importancia. Este apartado, además de lo subsecuente que se refiere a la teoría de diseño de elementos con presfuerzo, está basado en (ANIPPAC, 2018) y (NTC, 2023a), donde se explica de manera detallada todo lo relacionado a este campo de estudio.

Las trayectorias del presfuerzo pueden ser curvas, lo que permite diseñar con mayor eficiencia elementos hiperestáticos y evitar esfuerzos en los extremos del

elemento como se observa en (c) de la Figura 2, en (a) y (b) se observa una viga con apoyos en sus extremos cuya trayectoria solo es la curva que comprende dicho tramo.



**Figura 2. Trayectoria de los cables**

### **Pérdidas**

Algo a tomar en cuenta son las pérdidas de esfuerzo que se tienen durante este proceso, estas pueden ser inmediatas o diferidas, y se nombran de acuerdo al momento en el que estas ocurren. Algunos ejemplos de pérdidas inmediatas son el deslizamiento del anclaje, por fricción, desviación de torones, acortamiento elástico, relajación instantánea, mientras que las diferidas son la contracción por secado del concreto, el flujo plástico y la relajación diferida. Todas estas ocasionan una reducción de la fuerza efectiva la cual puede ser hasta del 30% diseñando de manera conservadora (Quintanilla Huayta, 2016).

### **Etapas**

Para la revisión del estado de esfuerzos en el que se encuentran los elementos presforzados, se toman en cuenta distintas etapas de su vida útil, esto debido a las condiciones que se presentan en cada una de ellas; las etapas consideradas son las siguientes:

**Etapa de Transferencia:** Se da cuando los alambres se cortan en elementos pretensados o en postensados cuando se libera la presión de los gatos, de modo que se transfiere la fuerza de presfuerzo al concreto, esto se debe realizar una vez que el concreto alcanzó un 80% de su resistencia final. Durante esta etapa ocurren las pérdidas inmediatas de presfuerzo y se presenta la contra flecha máxima ya que el presfuerzo solo es contrarrestado por el peso propio del elemento.

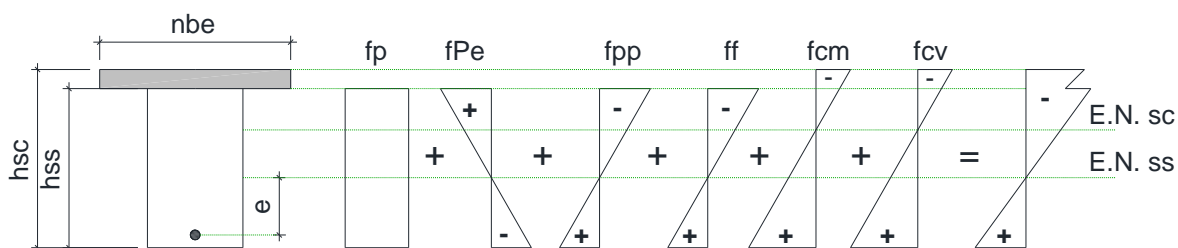
**Etapa Intermedia:** Ocurre durante el transporte y montaje del elemento, para esta etapa es importante es la disposición de apoyos temporales y dispositivos de montaje previendo la condición estática en el diseño.

**Etapa Final:** En esta etapa se evalúan todas las combinaciones de carga aplicadas, se revisan las condiciones de servicio asegurando que los esfuerzos, deformaciones y grietas estén dentro de los límites permitidos y se considera también la resistencia última del elemento. Para esta etapa, elementos presforzados estarán trabajando en conjunto como sección compuesta con el firme colado en sitio, incrementando su inercia y capacidad resistente de la sección.

Es necesario revisar los esfuerzos actuantes en el elemento para cada una de las etapas antes descritas. Para esto, se muestran en la Figura 3 y Ecuaciones (1) y (2), los esfuerzos,  $f$ , que deben calcularse considerando las diferentes acciones a las que está sometido el elemento estructural, y, además, sus propiedades geométricas.

$$f = -f_p \pm f_{pe} \pm f_{pp} \pm f_f \pm f_{cm} \pm f_{cv} \quad (1)$$

$$f = \frac{P}{A_{ss}} \pm \frac{P_e}{I_{ss}} y \pm \frac{M_{pp}}{I_{ss}} y \pm \frac{M_f}{I_{ss}} y \pm \frac{M_{cm}}{I_{sc}} y \pm \frac{M_{cv}}{I_{sc}} y \quad (2)$$



**Figura 3. Esfuerzos flectores en una viga en sección simple y compuesta**

$P$  = fuerza de presfuerzo efectiva

$e$  = excentricidad del presfuerzo

$M_{pp}$  = momento por peso propio

$M_f$  = momento debido al firme

$M_{cm}$  = momento debido a la sobrecarga muerta

$M_{cv}$  = momento debido a la carga viva

$A$  = área de la sección

$I$  = momento de inercia de la sección

$y$  = distancia a la fibra donde se calculan los esfuerzos

ss = Subíndice que refiere a la sección simple

sc = Subíndice que refiere a la sección compuesta

$f_p$  = Esfuerzo por fuerza de presfuerzo efectiva

$f_{Pe}$  = Esfuerzo por presfuerzo con excentricidad

$f_{pp}$  = Esfuerzo por peso propio

$f_f$  = Esfuerzo debido al firme

$f_{cm}$  = Esfuerzo debido a la sobrecarga muerta

$f_{cv}$  = Esfuerzo debido a la carga viva

$h_{ss}$  = Altura de sección simple

$h_{sc}$  = Altura de sección compuesta

$n_{be}$  = Ancho de patín en sección compuesta

E.N. = Eje neutro

### Esfuerzos permisibles

Para el concreto se debe tener cuidado de no exceder los esfuerzos de tensión y compresión especificados en la Tabla 3, siendo  $f'_{ci}$  la resistencia a compresión del concreto al momento de la transferencia del presfuerzo, por otro lado, para el acero de presfuerzo, también se respetan los límites establecidos por la Tabla 4 donde  $f_{py}$  se refiere al esfuerzo de fluencia y  $f_{sr}$  el esfuerzo resistente. También hay que considerar que el esfuerzo efectivo final debe ser al menos la mitad del esfuerzo resistente.

**Tabla 3. Esfuerzos de tensión y compresión límites en concreto, extraída de (NTC, 2023a)**

Inmediatamente después de la transferencia	Bajo cargas de servicio
Fibra extrema en compresión en $0.60 f'_{ci}$	Fibra extrema en compresión: Debido al presfuerzo más las cargas sostenidas $0.45 f'_{ci}$ Debido al presfuerzo más la carga total $0.60 f'_{ci}$
Fibra extrema en tensión $0.80\sqrt{f'_{ci}}$	
Fibra extrema en tensión en extremos de miembros simplemente apoyados $1.60\sqrt{f'_{ci}}$	Fibra extrema en tensión $1.60\sqrt{f'_{ci}}$

**Tabla 4. Esfuerzos de tensión límites en el acero, extraída de (NTC, 2023a)**

Condición	Con respecto a la fuerza de fluencia $f_{py}$	Con respecto a la fuerza de ruptura $f_{sr}$
Debida a la fuerza del gato	$0.94 f_{py}$	$0.80 f_{sr}$  $0.74 f_{sr}$
Inmediatamente después de la transferencia de presfuerzo	$0.82 f_{py}$	
Tendones de postensado, en anclajes y acopladores, inmediatamente después del anclaje de los tendones	-	$0.70 f_{sr}$

## Estados límite

### Estado límite de servicio

Para el caso de elementos con presfuerzo, el efecto del presfuerzo ocasiona una contra flecha, lo cual va presentarse en mayor medida durante la etapa de transferencia, a esta deformación se le conoce como deformación inicial, por otro

lado, una vez aplicadas las cargas en la condición de servicio, el elemento tiende a deformarse hacia abajo, en este punto tendremos la deformación final.

Para calcular las deflexiones se siguen utilizando los métodos de la teoría elástica, en este caso se debe considerar la deformación causada por el presfuerzo, para esto, es posible basarse en la curvatura con el principio de área de momentos. Además, se siguen manejando los límites establecidos en (NTC, 2023b) del claro dividido entre 240, más cinco milímetros, o el claro dividido entre 480, más tres milímetros para elementos cuya deformación afecte a elementos no estructurales.

### **Estados límite últimos**

Se toman en cuenta los estados límite últimos de flexión y esfuerzo cortante para el diseño de vigas de concreto postensado. A continuación, se describe la teoría que involucra dichos estados límite.

### **Flexión**

Para el diseño a flexión de elementos postensados se sigue la misma teoría elástica que los elementos sin presfuerzo, se toma en cuenta el comportamiento descrito en la Figura 4, basándose en las siguientes hipótesis:

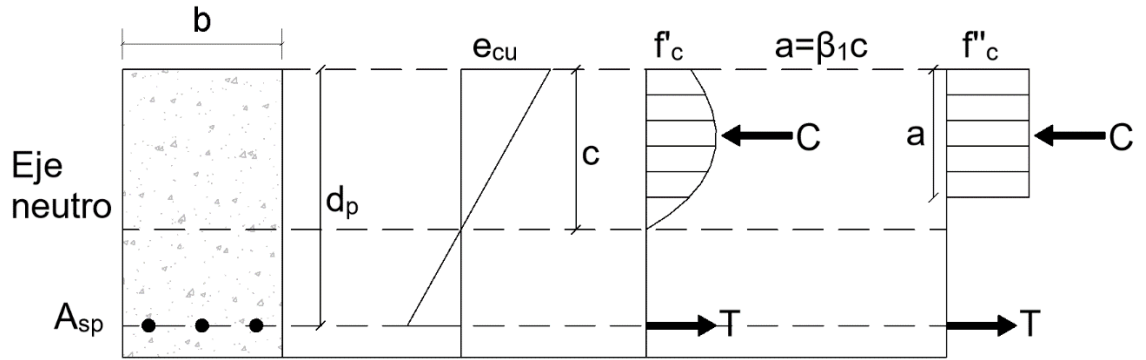
- Se tiene una distribución plana de deformaciones unitarias longitudinales en cada sección transversal del elemento.
- Existe una perfecta adherencia perfecta entre el concreto y aceros de presfuerzo y refuerzo.
- La resistencia del concreto a la tensión es despreciable.
- La deformación unitaria del concreto a compresión al alcanzar la resistencia última es 0.003.
- El bloque de esfuerzos de compresión en el concreto cuando al alcanza la resistencia última se distribuye de manera uniforme, con una profundidad  $a = \beta_1 c$ , siendo  $c$  es la distancia al eje neutro, y  $f_c''$  el ancho de dicho bloque.

Se determina  $\beta_1$  con respecto a las Ecuaciones (3) a (5).

$$\beta_1 = 0.85 \text{ si } f_c^* \leq 300 \frac{kg}{cm^2} \quad (3)$$

$$\beta_1 = 1.05 - \frac{f_c'}{1400} \geq 0.65 \text{ si } f_c' > 300 \frac{kg}{cm^2} \quad (4)$$

$$f_c^* = 0.8f_c' \quad (5)$$



**Figura 4. Resistencia a flexión de elemento en sección simple.**

El Índice de Presfuerzo es una herramienta utilizada para simplificar cálculos en elementos presforzados, evaluando el aporte del acero de presfuerzo y de refuerzo en la resistencia del elemento. Este índice se define como la relación entre el momento resistente debido al presfuerzo y el momento total de resistencia, considerando también el aporte del acero de refuerzo.

$$I_p = \frac{M_{R_p}}{M_{R_r} + M_{R_p}} = \frac{A_{sp}f_{sp}}{A_{sp}f_{sp} + A_{sr}f_y} \quad (6)$$

$M_{R_p}$  y  $M_{R_r}$  son los momentos resistentes suministrados por el acero presforzado y por el acero de refuerzo, respectivamente. Para conocer la resistencia del elemento es posible obtener el esfuerzo  $f_{sp}$  mediante la Ecuación (7), esto considerando que no existe acero de presfuerzo en la zona de compresión.

$$f_{sp} = f_{sr} [1 - 0.5(q_p + q - q')] \quad (7)$$

Siendo  $f_{sr}$  es el esfuerzo resistente del acero de presfuerzo y  $q_p$ ,  $q$  y  $q'$  los aceros de presfuerzo y refuerzo en tensión y compresión, respectivamente, obtenidos por las Ecuaciones (8) a (10).

$$q_p = \frac{p_p f_{sr}}{f_c''} \quad (8)$$

$$q = \frac{p f_y}{f_c''} \quad (9)$$

$$q' = \frac{p' f_y}{f_c''} \quad (10)$$

Las cuantías de refuerzo correspondientes se obtienen mediante las Ecuaciones (11) a (13).

$$p_p = \frac{A_{sp}}{bd_{sp}} \quad (11)$$

$$p = \frac{A_s}{bd} \quad (12)$$

$$p' = \frac{A'_s}{bd} \quad (13)$$

Si hay acero de compresión,  $q_p + q - q'$  no debe ser menor que 0.17. Si no se considera el acero de compresión para los cálculos, dicha cantidad donde  $q' = 0$ , sí puede tomar valores más pequeños que 0.17.

Para la revisión de elementos a flexión, la resistencia se da por un par interno compuesto por la fuerza de compresión  $C$ , obtenida según la Ecuación (14), la cual proporciona el concreto y dependiendo el caso, también el acero de compresión, y por la fuerza de tensión  $T$ , la cual se compone del aporte de la fuerza de tensión del acero de presfuerzo  $T_{sp}$  y de refuerzo  $T_s$ , según la Ecuación (15).

$$C = af_c''b \quad (14)$$

$$T = T_{sp} + T_s \quad (15)$$

siendo  $a$  el peralte del bloque de compresión y  $b$  el ancho del elemento a revisión. La fuerza de tensión se obtiene por las áreas de acero  $A_{sp}$  y  $A_s$  correspondientes al presfuerzo y refuerzo, respectivamente, donde  $f_y$  es el esfuerzo de fluencia del acero de refuerzo y  $f_{sp}$  el del acero de presfuerzo, según las Ecuaciones (16) y (17).

$$T_{sp} = A_{sp}f_{sp} \quad (16)$$

$$T_s = A_s f_s \quad (17)$$

De esta manera es posible encontrar el momento resistente de acuerdo a la Ecuación (18), donde  $F_R = 0.9$ , es el factor de reducción para flexión y  $z_{sp}$  y  $z_s$  los brazos de palanca de cada fuerza de tensión a la fuerza de compresión, obtenidos según las Ecuaciones (19) y (20).

$$M_R = F_R(T_{sp}z_{sp} + T_s z_s) \quad (18)$$

$$z_{sp} = d_{sp} - a/2 \quad (19)$$

$$z_s = d_s - a/2 \quad (20)$$

Normalmente el acero  $A_{sp}$  se propone en su mayoría para que el elemento estructural se comporte de manera adecuada en su etapa de servicio, por lo que el acero  $A_s$  solamente sirve en caso que se desee aumentar la resistencia del miembro a flexión. Finalmente, por equilibrio de fuerzas y sustituyendo elementos obtendremos las Ecuaciones (21) y (22), para finalmente despejar la altura del bloque de compresión  $a$  y calcularlo mediante la Ecuación (23), donde  $c$  es la altura existente entre el eje neutro del elemento y su fibra extrema a compresión.

$$T = C \quad (21)$$

$$A_s f_y + A_{sp} f_{sp} = a f_c'' b \quad (22)$$

$$a = \frac{A_{sp} f_{sp} + A_s f_y}{b f_c''} \quad (23)$$

Ahora es posible obtener el momento resistente  $M_R$  con la Ecuación (18) para compararlo con el momento último  $M_U$  actuante durante la etapa de transferencia garantizando que el primero sea mayor o igual al segundo.

Posteriormente, se hace el análisis para la sección compuesta, para determinar el momento resistente en elementos presforzados con sección compuesta, es fundamental considerar la resistencia del concreto del firme, ya que ahí es donde se localiza la fuerza de compresión principal. Este esfuerzo,  $f_c'$ , debe ser considerado incluso para el cálculo del esfuerzo del acero de presfuerzo,  $f_{sp}$ . Si la altura “ $a$ ” excede el espesor del firme, el análisis debe incorporar dos fuerzas de compresión, una previamente conocida,  $C_1$  y otra desconocida,  $C_2$  para evaluar con precisión la capacidad resistente del elemento, las cuales se calcularán según las Ecuaciones (24) y (25) respectivamente.

$$C_1 = b t_{firme} f_c'' \quad (24)$$

$$C_2 = (a - t_{firme}) b f_c'' \quad (25)$$

Para calcular el peralte del bloque de compresiones primero se determina la fuerza de tensión  $T_1$ , la cual corresponde a  $C_1$ , de modo que dicha relación queda expresada según la Ecuación (26).

$$T_1 = C_1 \quad (26)$$

$$A_{sp1} f_{sp} = b t_{firme} f_c'' \quad (27)$$

Al sustituir los elementos necesarios de la Ecuación (26), llegamos a la Ecuación (27), donde tenemos que el acero  $A_{sp1}$  es el encargado de equilibrar la fuerza  $C_1$  que está actuando en el concreto del firme, por lo tanto, el acero restante  $A_{sp2}$  y  $A_s$ , conforman la fuerza de tensión  $T_2$  que se encarga de equilibrar la fuerza de compresión  $C_2$  situada dentro del patín de la trabe según las Ecuaciones (28) y (29).

$$T_2 = C_2 \quad (28)$$

$$A_{sp2}f_{sp} + A_s f_y = (a - t_{firme}) b f_c''_{trabe} \quad (29)$$

Finalmente se despeja la altura del bloque de compresiones  $a$  y se obtiene la Ecuación (30).

$$a = t_{firme} + \frac{A_{sp2}f_{sp} + A_s f_y}{b f_c''_{trabe}} \quad (30)$$

Finalmente, se calcula el  $M_R$  con la Ecuación (31) y se compara con el  $M_U$  correspondiente.

$$M_R = F_R \left[ T_1 \left( d_{sp} - \frac{t_{firme}}{2} \right) + T_2 \left( d_{sp} - t_{firme} - \left( \frac{a - t_{firme}}{2} \right) \right) \right] \quad (31)$$

### **Acero mínimo**

Para asegurar la resistencia última a flexión en un elemento, es esencial que esta se logre después de que el elemento se haya agrietado. Esto implica proporcionar suficiente prefuerzo o refuerzo a tensión, de manera que el momento resistente supere al momento en el cual se producen las primeras grietas como se describe en la Ecuación (32).

$$M_R \geq (1.5 - 0.3I_p)M_{agr} \quad (32)$$

De acuerdo al índice de presfuerzo,  $I_p$ , los valores dentro del paréntesis de la Ecuación (32) oscilan entre 1.5 para elementos que no poseen presfuerzo y 1.2 para aquellos que están completamente presforzados. Por otro lado, para evaluar  $M_{agr}$  se toma en cuenta el módulo de ruptura no reducido  $f_r$  de acuerdo a la Ecuación (33).

$$f_r = 2\sqrt{f'_c} \quad (33)$$

La suma de esfuerzos en la fibra en tensión corresponde a la Ecuación (34), para finalmente despejar el  $M_{agr}$  y obtener la Ecuación (35).

$$\frac{M_{agr}}{I} y_i - \frac{Pe}{I} y_i - \frac{P}{A} = 2\sqrt{f'_c} \quad (34)$$

$$M_{agr} = \frac{I}{y_i} \left[ \frac{Pe}{I} y_i + \frac{P}{A} + 2\sqrt{f'_c} \right] \quad (35)$$

En secciones compuestas, el momento de agrietamiento se calcula como la suma de dos componentes según la Ecuación (36), un momento  $M_1$ , generado por el peso propio del elemento y del firme actuando sobre la sección simple de acuerdo a la Ecuación (37), y un momento  $M_2$ , que es el responsable del agrietamiento en la sección, el cual se obtiene según la Ecuación (38).

$$M_{agr} = M_1 + M_2 \quad (36)$$

$$M_1 = M_{popo} + M_{firme} \quad (37)$$

$$M_2 = \frac{I_{sc}}{y_{isc}} \left[ \frac{Pe}{I} y_i + \frac{P}{A} + 2\sqrt{f'_c} - \frac{M_1}{I} y_i \right] \quad (38)$$

### **Acero máximo**

Se debe asegurar que el elemento falle de manera dúctil. Para ello, es fundamental verificar que la deformación en los aceros sea al menos un 33% superior a la deformación correspondiente a la fluencia como se muestra en la Ecuación (39). En este contexto, el valor de  $\varepsilon_{sp}$  debe considerar la deformación inicial del presfuerzo.

$$\varepsilon_{sp} \geq 1.33\varepsilon_{sy} \quad (39)$$

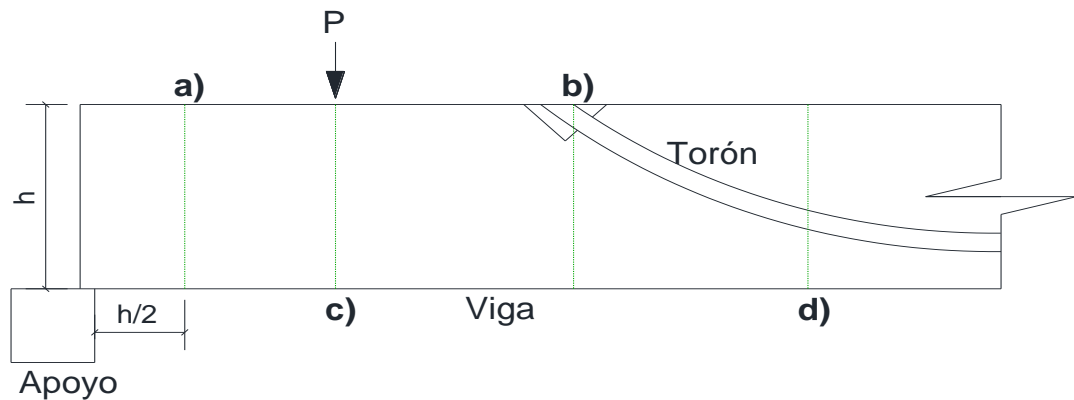
### **Cortante**

A diferencia de los elementos sin presfuerzo, cuando se tiene un elemento con desvío de torones, lo cual es el caso de estudio de este trabajo, se debe tomar en cuenta la componente vertical del presfuerzo, misma que actúa en dirección contraria al cortante actuante en la viga, de esta manera la fuerza cortante neta será lo mostrado en la Ecuación (40).

$$V_{neta} = V_{cargas} - V_{py} \quad (40)$$

Para revisar el esfuerzo cortante en elementos presforzados, se debe poner un mayor enfoque en las zonas mostradas en la Figura 5, las cuales son:

- A una distancia  $h/2$  del paño del apoyo
- En donde hay desvío de torones, y al finalizar los cables
- En donde exista una carga concentrada
- En distintos puntos de la longitud total de la trabe, tales como  $L/4$ ,  $L/8$ , con el fin de que el diagrama de cortante se ajuste a la separación de los estribos.



**Figura 5. Secciones a revisar por cortante**

El cortante resistente para elementos con un índice de presfuerzo mayor a 0.4 y con tendones bien adheridos, la fuerza  $V_{CR}$  se obtiene con la Ecuación (41).

$$V_{CR} = F_R b d (0.15 \sqrt{f_c^*} + 50 \frac{V d_p}{M}) \quad (41)$$

Donde:

$$F_R = 0.75$$

$M$  = Momento flexionante

$V$  = Cortante

$d$  = Peralte efectivo que va desde la fibra extrema a compresión, al centroide de los tendones de presfuerzo que se encuentran a tensión, este no debe ser menor que el 80% del peralte total.

$d_p$  = Distancia de la fibra extrema a compresión al centroide de los tendones de presfuerzo, considerando también los que pueden existir fuera de la zona de tensión, este será menor o igual a  $d$ .

Lo obtenido por  $\frac{Vd_p}{M}$  en la Ecuación (41), no se considera mayor a la unidad. Además, se procura que la resistencia  $V_{CR}$  se encuentre dentro de los límites propuestos en la Ecuación (42).

$$0.5F_Rbd\sqrt{f_c^*} \leq V_{CR} \leq 1.3F_Rbd\sqrt{f_c^*} \quad (42)$$

El peralte efectivo se calcula haciendo uso de la Ecuación (43)

$$d = \frac{A_{sp}f_{sp}d_p + A_s f_y d_s}{A_{sp}f_{sp} + A_s f_y} \quad (43)$$

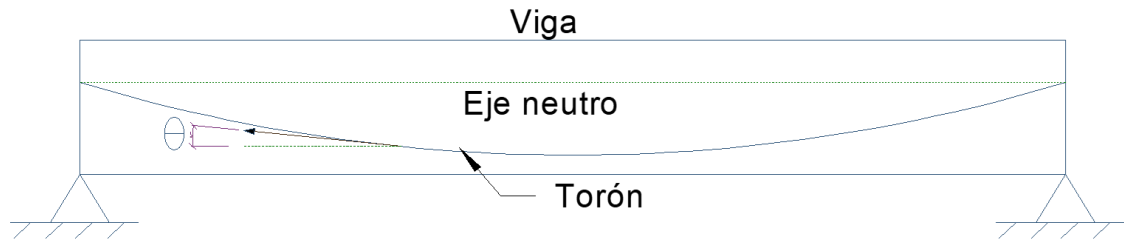
Esta metodología puede resultar en cierto punto conservadora, por lo cual existe un procedimiento para obtener el valor de  $V_{CR}$  donde este es el menor de  $V_{ci}$  o  $V_{cw}$ , los cuales se calculan para un agrietamiento por flexión cortante y agrietamiento por cortante, respectivamente. Se utiliza la Ecuación (44) para la obtención de  $V_{ci}$ , además este no debe ser menor que  $0.45bdf_c^*$ .

$$V_{ci} = F_R 0.16bd\sqrt{f_c^*} + V_{pp} + \frac{V_{sc}M_{ag}}{M_{sc}} \leq 0.45bdf_c^* \quad (44)$$

Siendo  $V_{pp}$  el cortante obtenido por el peso propio del elemento y de la sección compuesta, y  $V_{sc}$  y  $M_{sc}$  el cortante y momentos obtenidos por las sobrecargas viva y muerta, sin considerar el peso propio del elemento ni del firme.

Por otro lado, para obtener el cortante resistente del concreto por agrietamiento del alma se utiliza la Ecuación (45) donde  $V_p$  es la componente vertical de la fuerza de presfuerzo efectiva sin factorizar y considerando la longitud de adherencia, por otro lado  $f_{cc}$  será el esfuerzo de compresión ocasionado por la fuerza de presfuerzo efectiva que actuará en el centroide del concreto.

$$V_{cw} = F_Rbd(0.93\sqrt{f_c^*} + 0.02f_{cc}) + V_p \quad (45)$$



**Figura 6. Contribución de torones al cortante resistente**

De acuerdo a la Figura 6 se puede deducir que el valor de  $V_p$  se puede obtener considerando la excentricidad  $e$  que es la distancia el eje centroidal del presfuerzo al eje neutro de la sección,  $\theta$ , que es el ángulo de inclinación del cable de presfuerzo, por último, se considera la fuerza de presfuerzo  $P$ , todo esto expresado en la Ecuación (46).

$$V_p = Pe \sin \theta \quad (46)$$

Cabe mencionar que el cortante actuante  $V_u$ , no debe ser igual a lo mostrado en la Ecuación (47), aquí el peralte efectivo  $d$ , comprendido hacia el eje centroidal del acero de presfuerzo de tensión no se limita al 80% de la altura total, como se hizo en ecuaciones anteriores.

$$V_U = F_R 2.5bd\sqrt{f_c^*} \quad (47)$$

Todo esto, para obtener el aporte del concreto a la resistencia a cortante considerando el presfuerzo, para verificar el aporte del acero, se realiza el análisis como lo dicta (NTC, 2023a) en lo que refiere a las estructuras de concreto en el apartado de estado límite de cortante.

## **Optimización**

La optimización es una disciplina matemática enfocada en encontrar la mejor solución posible para un problema específico en el cual, se deben respetar ciertas restricciones o condiciones de acuerdo con (A. Kaveh, 2017; Yang et al., 2016). Consiste en determinar el conjunto de valores que minimicen o maximicen una función objetivo, la cual se representa mediante ciertas variables de decisión.

Es considerada un campo de la inteligencia artificial y las ciencias de la computación que busca encontrar soluciones óptimas o cercanas a óptimas para problemas complejos utilizando métodos matemáticos y algoritmos especializados. Este proceso es fundamental en la ingeniería moderna, donde se deben considerar múltiples variables y restricciones para obtener resultados eficientes y sostenibles, además de ser utilizada en una variedad de campos como la economía, la ingeniería, la ciencia de datos, la logística, y muchos más, debido a su capacidad para mejorar procesos y resultados mediante el análisis y la búsqueda de soluciones óptimas.

### **Conceptos Básicos**

- **Función objetivo:** La función objetivo es una expresión matemática que arroje como resultado el valor que se pretende maximizar o minimizar. Gracias a esta, podremos definir que tan bien se adaptan las diferentes soluciones a lo que buscamos en el problema de optimización.
- **Variables de Decisión:** Estas pueden ajustarse para alcanzar el objetivo. El núcleo de la optimización consiste en la selección de valores óptimos para estas variables.
- **Restricciones:** Estas definen los límites en los que se deben operar las variables de decisión. Pueden ser igualdades o desigualdades que ayuden a expresar las condiciones a cumplir.

## Métodos metaheurísticos

Entre los enfoques de optimización más utilizados destacan los algoritmos metaheurísticos, que son métodos basados en la simulación de procesos naturales o sociales. Las metaheurísticas son estrategias generales para diseñar o mejorar procedimientos heurísticos que se utilizan para resolver problemas de optimización y búsqueda. Estos métodos no garantizan encontrar la solución globalmente óptima, pero son capaces de proporcionar soluciones aceptables en problemas de alta complejidad (Yang et al., 2016). Dentro de las clasificaciones que encontramos en los algoritmos metaheurísticos, están los algoritmos bioinspirados, los cuales imitan comportamientos de los seres vivos. Algunos ejemplos comunes de estos son la Optimización por Colonia de Hormigas (ACO), el algoritmo de Optimización de Enjambre de Partículas (PSO), además de los mostrados en la Figura 7.

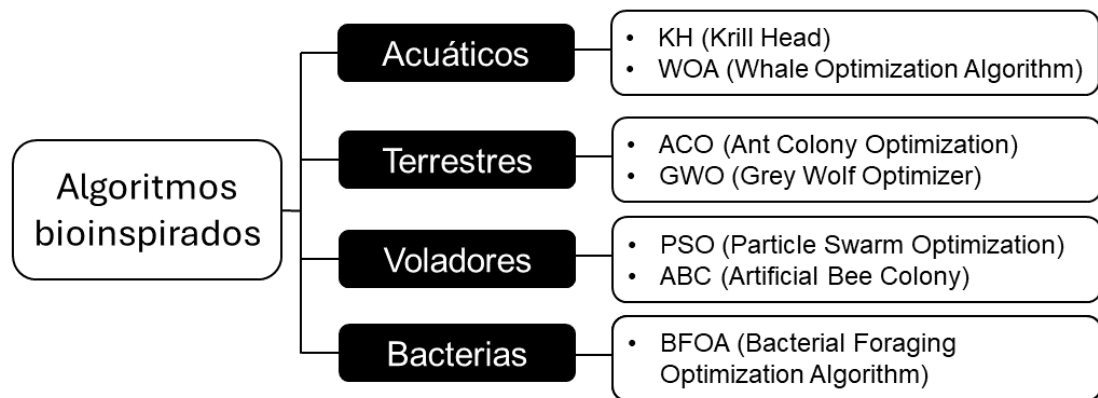


Figura 7. Algoritmos bioinspirados

### Algoritmo de enjambre de partículas.

El PSO fue desarrollado por James Kennedy y Russell Eberhart en 1995. Este método está inspirado en los comportamientos colectivos observados en la naturaleza, como el movimiento coordinado de bandadas de aves o cardúmenes de peces. Se enfoca en la colaboración y la comunicación entre partículas dentro de un espacio de búsqueda para encontrar soluciones óptimas (Freitas et al., 2020). A diferencia de otros algoritmos como los genéticos, el PSO no utiliza operadores

como mutación o cruza, sino que aprovecha la inteligencia colectiva para dirigir la búsqueda.

### Fundamento matemático

En esencia, el PSO se compone de partículas que representan soluciones. Cada partícula se guía por su mejor posición conocida ( $p_{best}$ ) y la mejor posición global ( $g_{best}$ ), que representan un equilibrio entre exploración y explotación del espacio de búsqueda. Todo esto se realiza combinando dos componentes:

- Factor cognitivo: Representa la tendencia de una partícula a regresar a su mejor posición individual ( $p_{best}$ ).
- Factor social: Representa la influencia del mejor desempeño global ( $g_{best}$ ) en la partícula.

El PSO se basa en ecuaciones que determinan la actualización de la velocidad y posición de las partículas. Estas ecuaciones incluyen términos aleatorios para fomentar la exploración del espacio de búsqueda y evitar convergencias prematuras en óptimos locales. Para actualizar la velocidad se utiliza la ecuación:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (p_{best_i} - x_i) + c_2 r_2 (g_{best_i} - x_i) \quad (48)$$

Donde  $\omega$  es el factor de inercia que controla el balance entre exploración y explotación,  $c_1$  y  $c_2$  son coeficientes cognitivos y social respectivamente, y  $r_1$ ,  $r_2$  son valores aleatorios entre 0 y 1 que fomentan la diversidad en la búsqueda y  $p_{best}$  y  $g_{best}$  las mejores posiciones alcanzadas.. Por otro lado, la posición se actualiza según la siguiente ecuación.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (49)$$

Donde  $x_i$  es la posición,  $v_i$  la velocidad y  $k$  el número de iteración (Jain et al., 2018).

### **Variantes del PSO**

Debido a que se observó que el algoritmo presentaba problemas en cuanto a una convergencia prematura y baja exploración del espacio de soluciones, se desarrollaron distintas variantes de este:

- PSO con inercia adaptativa (DI-PSO): Se mejora la convergencia pues modifica de forma dinámica los parámetros del algoritmo, evitando el estancamiento en óptimos locales (Annicchiarico & Cerrolaza, 2019).
- PSO Multiobjetivo (MOPSO): Es útil para optimizar múltiples objetivos de manera simultánea, generando un conjunto de soluciones de Pareto (Zhu et al., 2017)
- Híbridos PSO-GA (Algoritmos genéticos): Combina el PSO con Algoritmos Genéticos para aumentar la diversidad de soluciones (Ayyash & Hejazi, 2023).

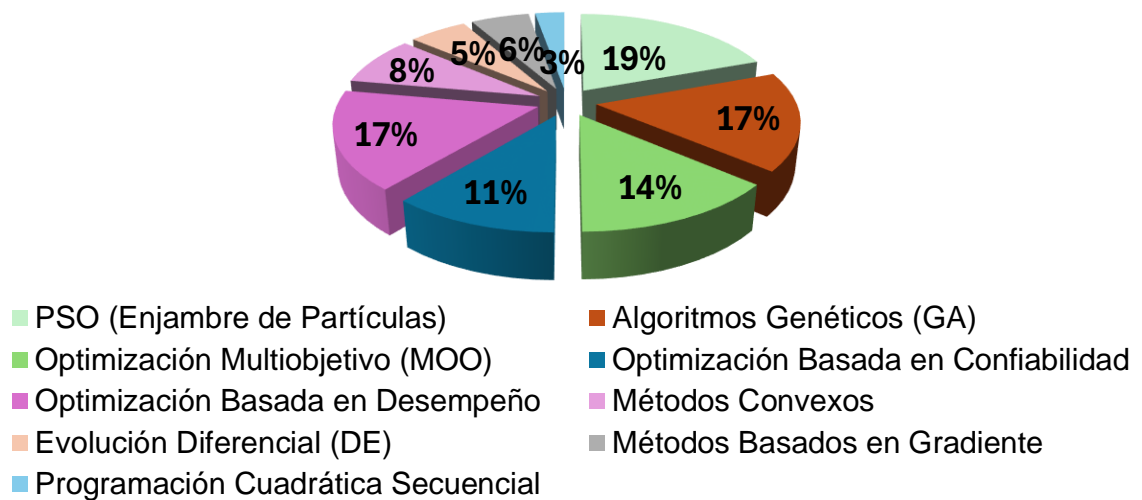
### **Factores Clave del PSO**

- Inercia y Constricción: La inercia determina la estabilidad del movimiento de las partículas, mientras que los factores de constricción equilibran la búsqueda global y local. Una inercia alta favorece la exploración, mientras que una baja facilita la explotación de áreas específicas.
- Topologías de Comunicación: El modelo  $g_{best}$  permite que todas las partículas compartan información, acelerando la convergencia, aunque aumenta el riesgo de estancamiento en óptimos locales.

## Comparación con Otros Métodos

El PSO comparte características con los Algoritmos Genéticos (GA) en el uso de poblaciones, pero se diferencia por su simplicidad y en la forma de actualizar soluciones. Los GA dependen de operadores como la mutación y la cruce, mientras que el PSO utiliza únicamente la velocidad y posición para explorar el espacio de búsqueda. El PSO se compara de forma favorable ante otros métodos debido al bajo costo computacional que este requiere, su capacidad de adaptarse a problemas estructurales de gran escala, y la estabilidad que presenta ante problemas no lineales (Reyes-Sierra & Coello Coello, 2006).

En el contexto del diseño de traveses de concreto postensado, el PSO proporciona un enfoque eficiente para optimizar parámetros como el volumen de materiales, geometría de los elementos y tiempos de diseño. Al ser un método metaheurístico, permite considerar múltiples restricciones y objetivos adaptándose a las complejidades del problema. Como se puede observar en la Gráfica 1, el PSO es el método más utilizado de acuerdo a los trabajos investigados para este documento, seguido por los GA y métodos de optimización multiobjetivo, es por esto, que en este trabajo se optará por mezclar el PSO con la optimización multiobjetivo.



Gráfica 1. Métodos de optimización más usados en ingeniería estructural.

## **IV. Hipótesis y objetivos**

### **Hipótesis**

Mediante la aplicación del algoritmo de Optimización de Enjambre de Partículas (PSO) en el diseño estructural de vigas de concreto postensado para edificios de claros mayores a 9 metros se reducen en al menos un 20% los volúmenes de material empleados, debido a que se determina la combinación cercana al óptimo global de la sección transversal, cantidad de acero de refuerzo convencional y de presfuerzo, cumpliendo con las restricciones normativas de estados límite de resistencia y servicio durante la vida útil de la estructura.

### **Objetivo general**

Realizar el análisis y diseño estructural de vigas de concreto postensado en edificios con claros mayores a 9 metros con base en el algoritmo de PSO donde se minimice el consumo de materiales, garantizando que se cumplan los estados límite de resistencia y servicio según las normativas vigentes.

### **Objetivos específicos**

- Automatización en Python mediante el API (Interfaz de Programación de Aplicaciones) de SAP 2000 para análisis estructural de casos de estudio.
- Diseño estructural sin presfuerzo y diseño con presfuerzo optimizado con algoritmo de PSO, ambos automatizados en Python para cada caso de estudio.
- Comparación de volúmenes de material en ambos diseños para cada caso de estudio

## **V. Metodología.**

En este capítulo se describe la metodología seguida para evaluar el impacto del postensado y el algoritmo de Optimización de Enjambre de Partículas (PSO) en el diseño de vigas de concreto reforzado para edificios con grandes claros.

La investigación es de tipo aplicado y cuantitativo, basada en la modelación numérica y la optimización estructural. Se hace uso del software SAP2000, combinado con el lenguaje de programación Python mediante su API (Interfaz de Programación de Aplicaciones).

La metodología se estructura en tres etapas principales:

- Modelado paramétrico y análisis estructural de casos de estudio mediante la API de SAP2000 desde Python.
- Diseño estructural en Python sin y con presfuerzo, implementando el problema de optimización.
- Comparación de resultados, cuantificando los volúmenes de concreto y acero para el diseño convencional y el diseño optimizado.

### **Etapas 1. Modelación paramétrica de casos de estudio y análisis estructural**

#### **Casos de estudio**

Como caso de estudio se consideran edificios tipo con un sistema estructural de marcos de concreto reforzado, de tres, cinco y siete niveles, con uso hospitalario y de estacionamiento, cuatro crujías en cada dirección, y claros de 9, 11 y 13 metros. Al combinar estos factores, se obtiene un total de 18 casos de estudio como se muestra en la Tabla 5.

**Tabla 5. Casos de estudio**

<b>Caso</b>	<b>Niveles</b>	<b>Uso de edificación</b>	<b>Claro</b>
1	3	Hospital	9 m
2	3	Hospital	11 m
3	3	Hospital	13 m
4	5	Hospital	9 m
5	5	Hospital	11 m
6	5	Hospital	13 m
7	7	Hospital	9 m
8	7	Hospital	11 m
9	7	Hospital	13 m
10	3	Estacionamiento	9 m
11	3	Estacionamiento	11 m
12	3	Estacionamiento	13 m
13	5	Estacionamiento	9 m
14	5	Estacionamiento	11 m
15	5	Estacionamiento	13 m
16	7	Estacionamiento	9 m
17	7	Estacionamiento	11 m
18	7	Estacionamiento	13 m

### **Dimensionamiento y materiales**

Una vez definida la geometría de los edificios de diseño, se programó paramétricamente el análisis estructural de los edificios en Python utilizando la API del software de diseño SAP 2000, en la que se especificó el material de concreto con un  $f'_c$  de 400 kg/cm<sup>2</sup>, el acero de refuerzo convencional con un  $f_y$  de 4,200 kg/cm<sup>2</sup> y los torones postensados con una resistencia a la ruptura  $f_{sr}$  de 19,000 kg/cm<sup>2</sup>.

Las secciones geométricas de las vigas de concreto se basan en un predimensionamiento sugerido por la Tabla 6, donde  $L$  es el claro máximo considerado, y el ancho se consideró como el múltiplo de 5 cm inmediato superior de la mitad del peralte; para la losa reticular, la Ecuación 50 fue el criterio mínimo de peralte, redondeándolo al múltiplo de 5 cm inmediato superior. Las separaciones

entre las nervaduras en ambas direcciones se tomaron de 75 cm en ambas direcciones y con un ancho de 15 cm y una capa de compresión de 5 cm.

$$h = \frac{L(0.8 + \frac{f_y}{1400})}{36 + 9\beta} \geq 9 \text{ cm} \quad (50)$$

Donde  $\beta$  es el cociente del claro en la dirección larga y corta de la losa.

Por otro lado, para las columnas, se obtuvo una descarga axial última para una columna central, según el número de niveles y áreas tributarias mediante la combinación de diseño gravitacional, y se dimensionó según la Ecuación 51.

$$b = \frac{P_u}{0.3f'_c} \quad (51)$$

Donde  $P_u$ , es la carga axial última. Al igual que las vigas, el ancho de columnas se redondeó al múltiplo de 5 cm inmediato superior, manteniendo medidas coherentes a la práctica profesional.

Estas propiedades de los materiales son coherentes con las recomendaciones de (NTC, 2023a) y (ANIPPAC, 2018), y se utilizan tanto en los diseños convencionales como en los diseños postensados optimizados por PSO.

**Tabla 6. Peralte mínimo para vigas sin presfuerzo, extraída de (NTC, 2023a)**

<b>Condición de apoyo</b>	<b>Altura mínima</b>
Simplemente apoyada	L / 16
Un extremo continuo	L / 18.5
Dos extremos continuos	L / 21
Voladizo	L / 8

Al crear de forma paramétrica mediante la API del software el modelo, la geometría, propiedades de los materiales, secciones de los elementos y las cargas pueden gestionarse directamente desde el código de programación, permitiendo reproducir y modificar el modelo de forma.

## Cargas y combinaciones de carga

En cuanto a las acciones de carga, se consideran:

- **Cargas muertas**, incluyendo el peso propio de vigas, losas, recubrimientos, acabados, instalaciones y cargas adicionales de concreto y mortero, de acuerdo con la Tabla 7 y Tabla 8
- **Cargas vivas**, en Tabla 9, según los valores establecidos para hospitales y estacionamientos en (NTC, 2023b).
- **Acciones accidentales**, incluyendo un análisis modal espectral con el espectro elástico de diseño correspondiente a la ciudad de Querétaro, mostrado en la Grafica 1, incorporado en las combinaciones de carga de diseño.

Tabla 7. Estado de cargas, para losa de entrepiso

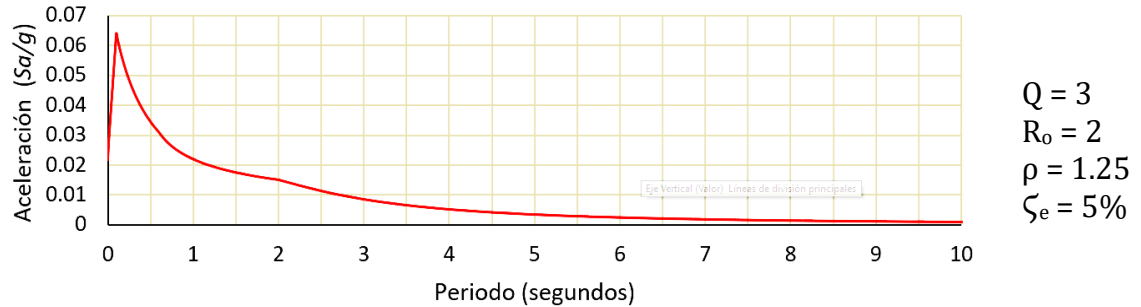
Material	Espesor (m)	Peso volumétrico (kg/m <sup>3</sup> )	Peso (kg/m <sup>2</sup> )
Loseta cerámica	0.01	1840	18.4
Adhesivo	0.015	1500	22.5
Plafón	-	-	20
Instalaciones	-	-	20
Carga adicional por concreto			20
Carga adicional por mortero			20
Carga muerta sin sistema de piso			120.9

Tabla 8. Estado de cargas, para losa de azotea

Material	Espesor (m)	Peso volumétrico (kg/m <sup>3</sup> )	Peso (kg/m <sup>2</sup> )
Impermeabilizante	0.002	2500	5
Lechada	0.005	2100	10.5
Mortero	0.01	1900	19
Relleno	0.1	1300	130
Plafón	-	-	0
Instalaciones	-	-	20
Carga adicional por concreto			20
Carga adicional por mortero			20
Carga muerta sin sistema de piso			224.5

**Tabla 9. Cargas vivas según uso de edificio.**

Destino de piso	W (kg/m <sup>2</sup> )	W <sub>a</sub> (kg/m <sup>2</sup> )	W <sub>m</sub> (kg/m <sup>2</sup> )
Estacionamiento	40	100	250
Hospital	80	100	190



**Grafica 1. Espectro elástico para acción sísmica**

Las combinaciones de carga consideradas, se muestran en la Tabla 10 (a) y la Tabla 10 (b), para el diseño por LRFD (Diseño por factores de carga y resistencia) y por ASD (Diseño por Esfuerzos Permisibles).

#### Combinaciones de carga para diseño por LRFD

Para el diseño estructural por LRFD, se generó una envolvente, que engloba las combinaciones de diseño descritas en la Tabla 10 (a), con esto, se realizaron las revisiones por flexión y cortante. Por otro lado, para los desplazamientos en la viga de estudio, se utiliza la combinación de estado límite de servicio (ELS).

**Tabla 10 (a). Factores de carga para combinaciones de carga, extraído de (NTC, 2023b)**

Combo	P <sub>p</sub>	C <sub>m</sub>	W	W <sub>a</sub>	W <sub>m</sub>	Sismo X	Sismo Y
ELU	1.3	1.3	-	-	1.5	-	-
ELS	1	1	1	-	-	-	-
ACC X	1.1	1.1	-	1.1	-	1.1	0.33
ACC Y	1.1	1.1	-	1.1	-	0.33	1.1

P<sub>p</sub>: Peso propio

C<sub>m</sub>: Sobrecarga muerta

W: Carga viva media

ELU: Estado límite último

ELS: Estado límite de servicio

W<sub>a</sub>: Carga viva instantánea  
W<sub>m</sub>: Carga viva máxima

ACC X: Combinación accidental en dirección X  
ACC Y: Combinación accidental en dirección Y

### Combinaciones de carga para diseño por ASD

En el caso del diseño por ASD, los casos de carga se mantienen con un factor de carga unitario, sin embargo, las resistencias de los elementos se reducen de acuerdo a lo mostrado en la Tabla 3 y Tabla 4 del apartado de Fundamentación teórica.

Al tratar con elementos presforzados, según (NTC, 2023a) es necesario catalogar los elementos como clase SA (Sin agrietamiento), T (transición entre agrietado y no agrietado) y A (Agrietado), según el nivel de presfuerzo que contengan, es por ello, que al estar trabajando con elementos presforzados clases SA y T, es de carácter obligatorio la revisión de estos mediante ASD, no puede evaluarse simplemente como un miembro con refuerzo ordinario.

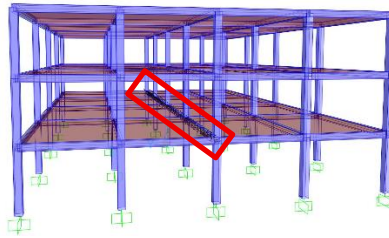
**Tabla 10 (b). Factores de carga para combinaciones para revisión por ASD**

Combo	P <sub>p</sub>	C <sub>m</sub>	W	W <sub>a</sub>	W <sub>m</sub>	Sismo X	Sismo Y
CS	1	1	-	-	-	-	-
CT	1	1	-	-	1	-	-

CS: Cargas sostenidas  
CT: Cargas totales

### Viga de estudio

Se tomó la viga continua del eje central de la planta baja en el lado largo del edificio, tal y como se muestra en la Figura 8, considerando que es una de las vigas más exigidas desde el punto de vista estructural para tenerla como punto de comparación en ambos diseños.

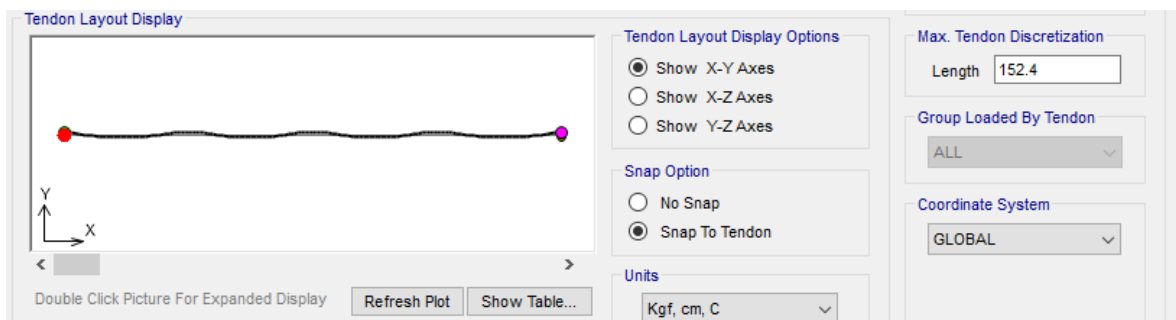


**Figura 8. Viga de estudio**

Se discretizó la viga de estudio en 10 partes iguales por claro para la generación de nodos internos que nos pudieran arrojar resultados sobre la deformación de la viga, esto para el análisis sin y con presfuerzo.

### **Consideraciones del presfuerzo**

Para realizar los modelos de análisis que llevan fuerzas de postensado, se utilizó la herramienta de SAP 2000 que permite ingresar las cargas del presfuerzo, mediante tendones dentro de vigas de concreto, como se ejemplifica en la Figura 9.



**Figura 9. Trazo y aplicación de postensado mediante herramienta integrada de SAP 2000**

Se dio de alta el material con propiedades de acero grado 270 para los torones y se consideró un esfuerzo de tensado de  $15,200 \text{ kg/cm}^2$  antes de pérdidas, también se dio de alta la sección de un tendón de media pulgada de diámetro con un área efectiva de  $0.987 \text{ cm}^2$ , la cual simplemente se multiplicó por la cantidad de torones que se consideraron en cada análisis, finalmente se colocaron las coordenadas del tendón en la viga de estudio, cumpliendo con un recubrimiento hacia el torón de al menos 7 cm tanto en lecho superior como inferior de la viga.

A efectos prácticos, se consideraron pérdidas inmediatas del 5% y pérdidas diferidas del 10% para los cálculos de tensiones y compresiones admisibles en las diferentes etapas de diseño y para obtener la deformación de la viga de estudio en el software.

## **Etapa 2. Diseño estructural sin y con presfuerzo optimizado.**

En esta etapa se llevó a cabo el diseño estructural para los modelos sin presfuerzo, y el diseño optimizado mediante PSO para los modelos con presfuerzo.

Se utilizaron dos tipos de modelos analíticos diferentes, uno para el análisis convencional, que no incluye el efecto del postensado, y otro en el que el postensado influye en los desplazamientos de la viga continua objeto de estudio.

Para llevar a cabo los diseños estructurales en Python, fue necesario obtener los esfuerzos y desplazamientos pertinentes del análisis de SAP2000 para las combinaciones de carga específicas.

### **Diseño estructural sin presfuerzo**

Una vez realizado el análisis estructural sin presfuerzo mediante la API, los valores de salida necesarios para las revisiones posteriores fueron los siguientes:

- **Momento último positivo y negativo** con envolvente de diseño
- **Cortante último absoluto** con envolvente de diseño
- **Desplazamiento máximo vertical** con combinación de servicio.

Una vez obtenidos los elementos mecánicos de la viga de estudio se procedieron a realizar los diseños por flexión, cortante y revisión de desplazamientos.

### **Diseño a flexión sin presfuerzo**

Con base en los momentos obtenidos, se implementan en Python las expresiones de LRFD según (NTC, 2023a). Para el momento máximo positivo y mínimo negativo se determina el acero longitudinal necesario en las caras superior e inferior de la viga, se revisa que este cumpla con el área de acero mínima y máxima por normativa. La propuesta de varillas se hace considerando que estas se distribuyen en una sola capa, en paquetes de máximo 2 varillas, todas del mismo diámetro, esto por simplicidad del código, procurando cumplir también con la separación mínima entre varillas estipulada en (NTC, 2023a).

### **Diseño a cortante sin presfuerzo**

Empleando el cortante máximo de diseño, se calculó el refuerzo transversal (diámetro y separación de estribos), verificando las contribuciones del concreto y del acero a la resistencia a cortante de acuerdo a las ecuaciones mostradas en el apartado de fundamentación teórica, siempre tomando en cuenta las separaciones mínima y máxima estipuladas en (NTC, 2023a), así como las áreas de acero mínima y máxima. Se consideraron estribos del número 3 como mínimo, y en caso de requerir una mayor área de acero, se propusieron estribos con varilla de media pulgada.

### **Revisión de desplazamientos sin presfuerzo**

Se revisaron las deformaciones de la viga en servicio y los límites de acuerdo con la normativa, lo cual, para el caso sin presfuerzo, no presenta problema alguno pues se está cumpliendo con el predimensionamiento propuesto en la Tabla 6.

### **Diseño estructural con presfuerzo**

Para llevar a cabo las revisiones correspondientes de elementos presforzados, se extrajeron de los modelos de análisis los siguientes datos:

- **Momento último positivo y negativo** con envolvente de diseño

- **Momento último positivo y negativo** con cargas sostenidas
- **Momento último positivo y negativo** con cargas totales
- **Momento último positivo y negativo** con peso propio
- **Cortante último absoluto** con envolvente de diseño
- **Desplazamiento máximo vertical** con combinación de servicio.

Para este caso, la acción de los torones fue incluida en la combinación de servicio, con tal de que estos influyeran en los desplazamientos obtenidos en la viga de estudio.

#### **Diseño a flexión con presfuerzo**

En el caso del diseño a flexión con presfuerzo, se toma en cuenta el aporte del acero presforzado de acuerdo a las ecuaciones mostradas en el apartado de marco teórico, se verifican los límites de acero mínimo y máximo, así como la separación entre varillas de acero pasivo, y activo para finalmente proponer la configuración de acero longitudinal, ya sea en varillas sencillas o formadas en paquetes de dos que cumpla con las solicitudes.

#### **Diseño a cortante con presfuerzo**

Para este caso, se toma en cuenta el aporte del concreto, los estribos, y el aporte de los torones de acuerdo a la posición en la viga que se esté evaluando, debido a la variación del ángulo  $\theta$  que tiene la trayectoria parabólica de los torones.

#### **Diseño por esfuerzos permisibles**

Para este análisis, se toman los momentos últimos para las combinaciones de cargas sostenidas, y cargas totales mostradas en la Tabla 10 (b), una vez obtenidos dichos momentos, mediante la Ecuación 2 se revisan los esfuerzos permisibles mostrados en las Tablas 3 y 4.

## Optimización

### Espacio de soluciones y variables de decisión

Para el diseño postensado se define un espacio de soluciones discretizado, formado a partir de las combinaciones de sección transversal de la viga y cantidad de torones de presfuerzo:

- Ancho y peralte de la viga: Los anchos de trabe se propusieron de mínimo 25 cm y máximo de 35 cm, siguiendo la recomendación de (NTC, 2023a) para vigas de concreto para estructuras de ductilidad baja. Por otro lado, los peraltes van desde 30 hasta 65 cm, dependiendo de los anchos que se tengan y el claro del caso de estudio, dichas combinaciones se observan en la Tabla 11.

**Tabla 11. Geometría de vigas de espacio de soluciones**

<b>Claros</b>	<b>Anchos (cm)</b>	<b>Altos (cm)</b>
9 metros	25-30	30-65
	32	32-65
	34	34-65
	35	35-65
11 metros	25-35	35-65
13 metros	25-35	40-65

Todas las medidas consideradas fueron múltiplos de 2 y 5 centímetros, procurando que el ancho no excediera al peralte para que la inercia del eje principal de la sea igual o mayor al del eje secundario, además, los peraltes partieron de 30, 35 y 40 centímetros para los claros de 9, 11 y 13 metros respectivamente, para que las vigas siempre fueran al menos 5 cm mayores que el peralte obtenido de la losa reticular en el predimensionamiento de esta.

- Número de torones de postensado: Se consideraron entre 6 a 10 torones como máximo, actuando hasta en 2 filas, de acuerdo al ancho de la trabe, respetando recubrimientos y separaciones mínimas entre sí y respecto al acero convencional tal cual se describen en la Tabla 12.

**Tabla 12. Máxima cantidad de torones de acuerdo al ancho de trabe**

<b>Anchos (cm)</b>	<b>Torones</b>
25	6
26	6
28	8
30	8
32	8
34	8
35	10

Cada combinación ( $b$ ,  $h$ ,  $n$  torones) define una solución candidata. Estas combinaciones se almacenan en una base de datos que constituye el espacio de búsqueda del problema de optimización.

Las variables de decisión del problema son, por tanto:

- Ancho  $b$  de la sección.
- Peralte  $h$  de la sección.
- Número de torones.

#### **Función objetivo**

El propósito de la optimización es minimizar el consumo de materiales en la viga postensada, considerando tanto el concreto como el acero convencional y de presfuerzo. Para ello se plantea una función objetivo agregada según la Ecuación 50.

$$F = \alpha P_c + (1 - \alpha)P_a + \text{penalizaciones} \quad (51)$$

donde:

- $F$  es el valor de la función objetivo asociada a cada solución candidata.
- $P_c$  es el peso del concreto de la viga.

- $P_a$  es el peso total de acero de refuerzo convencional y torones de postensado.
- $\alpha$  es un factor de ponderación entre 0 y 1 que asigna mayor importancia relativa a la reducción del concreto frente a la del acero.
- Las penalizaciones son obtenidas de acuerdo al apartado correspondiente.

De esta manera, el problema se formula como la minimización de  $F$ , manteniendo el cumplimiento de las restricciones normativas.

Dado que el predimensionamiento para el análisis convencional que se muestra en la Tabla 6 puede ser un poco conservador, el valor alfa de 0.6 inclina ligeramente la balanza hacia el ahorro de concreto en lugar del ahorro de acero, lo cual permite abordar el problema de optimización desde un frente adecuado.

#### **Restricciones del problema**

Las soluciones generadas deben cumplir con un conjunto de restricciones de diseño basadas en las (NTC, 2023a) y (ANIPPAC, 2018), las cuales son descritas a continuación.

- Esfuerzos permisibles en el concreto en fibras superior e inferior, tanto en estado de transferencia como en estado de servicio.
- Capacidad a flexión última de la sección, considerando los aceros a tensión, tanto el convencional como el de presfuerzo.
- Capacidad a cortante, incluyendo las contribuciones del concreto, de los estribos y de la componente vertical del presfuerzo.
- Deformaciones máximas en servicio, respetando los límites establecidos para vigas que soportan elementos no estructurales.
- Condiciones de detallado de refuerzo, como cuantías de acero, recubrimientos, diámetros y separaciones entre varillas y torones.

Las combinaciones que violan alguna de estas restricciones son penalizadas en la evaluación de la función objetivo, de modo que el algoritmo favorezca soluciones factibles.

### **Penalizaciones**

Para las penalizaciones de cada revisión, se utilizó una ecuación de la forma cuadrática, donde primero se obtuvo la ratio de acuerdo a la revisión y de acuerdo a la Ecuación 51, se obtuvo la penalización, misma que aun fue afectada por un respectivo peso para cada revisión.

$$penalizacion = (ratio - 1)^2 \quad (51)$$

Si la relación obtenida es menor que la unidad, significa que la sección de la solución es correcta, por lo que no hay penalización por esa revisión. Esto es muy utilizado en problemas de optimización, en el que existe una penalización cuadrática, que es cero si se cumple la restricción, suave si hay una pequeña infracción y muy punitiva para infracciones graves.

### **Penalización por desplazamientos**

Para el caso de la revisión por desplazamientos, basta con comparar el desplazamiento máximo obtenido en la viga, contra el desplazamiento permisible, si el desplazamiento permisible es mayor al actuante, no es necesario aplicar ninguna penalización, en caso contrario, se considera el peso de la Tabla 13, y se lleva a cabo la ecuación de penalización para la revisión del desplazamiento

**Tabla 13. Pesos para penalización por desplazamientos**

<b>Revisión</b>	<b>Peso</b>
Desplazamientos	100

### Penalización por cortante

En el caso de la revisión por cortante, se identifica inicialmente la condición del cortante último que está actuando en la viga, mismo que se revisa de acuerdo a (NTC, 2023a), posteriormente, si ninguna de las configuraciones posible de estribos cumple con el cortante actuante, se debe realizar la segunda penalización prevista en la Tabla 14, y finalmente se revisa el cumplimiento del área de acero del estribo, que cumpla con el área mínima estipulada en (NTC, 2023a).

Tabla 14. Pesos para penalización por cortante

Revisión	Peso
Cortante (Revisión de cortante ultimo $V_u$ )	100
Cortante (Revisión de configuración de estribos disponibles)	100
Cortante (Revisión de área de estribo mínima $A_v$ )	100

### Penalización por flexión

En el caso de la revisión por flexión, se penalizan dos cosas, en primer lugar, se evalúa el momento máximo resistente de la sección transversal en cuestión y este se compara con el momento último actuante para obtener la ratio de esa revisión, posteriormente, en el caso que el acero necesario no cumpla con la separación mínima entre varillas estipulada en (NTC, 2023a), de igual manera se penaliza la solución. Los pesos considerados para las penalizaciones por flexión se describen en la Tabla 15.

Tabla 15. Pesos para penalización por flexión

Revisión	Peso	Descripción técnica
Momento máximo resistente	100	El acero requerido es mayor al $A_{smax}$
Separación entre varillas	100	Ninguna combinación de varillas satisface las separaciones mínimas entre varillas.

### Penalización por esfuerzos permisibles

Para la revisión por esfuerzos permisibles, el algoritmo castiga cada situación tal como se describe en la Tabla 16

**Tabla 16. Pesos para penalización por esfuerzos permisibles.**

<b>Revisión</b>	<b>Peso</b>	<b>Descripción técnica</b>
Compresión excesiva	100	Cuando esfuerzo de compresión supera lo permisible, busca evitar el aplastamiento del concreto
Tensión (Sin acero pasivo)	100	Cuando el elemento se encuentra en clase no agrietada, y supera el límite de tensión sin la ayuda del acero pasivo.
Tensión (Falla en fibra inferior)	100	Aplicado en clase en transición, donde la tensión actuante supera la resistente entre el concreto y el acero pasivo
Tensión (Falla en fibra superior)	100	Similar al anterior, pero es menor debido a la probabilidad de que esté comprometido el lecho superior.

Además, para la revisión de las tensiones, se identifica si el elemento que se está evaluando, entra como sección no agrietada o como sección en transición, de acuerdo a su cantidad de acero pasivo y activo, con base a esto se revisa la ratio de tensión entre la admisible y la actuante. Esto se hizo para cada revisión ASD y se obtuvo una penalización total como la sumatoria de cada una de las anteriores.

## Aplicación del algoritmo PSO

Una vez realizado el análisis estructural incorporando el postensado, para cada caso de estudio se implementa el algoritmo PSO en Python para explorar el espacio de soluciones definido y encontrar la configuración de viga postensada que minimiza la función objetivo.

Cada partícula del enjambre representa una solución candidata ( $b$ ,  $h$ ,  $n$  torones). En cada iteración:

- Se actualiza la velocidad y posición de cada partícula.
- Se selecciona, a partir de la base de datos, la combinación de sección y presfuerzo correspondiente a esa posición.
- Se llama, mediante la API de SAP2000, al modelo estructural postensado para obtener los esfuerzos y deformaciones de la viga con esa configuración.
- Se calcula la función objetivo  $F$  y se verifica el cumplimiento de las restricciones de diseño.
- Se actualizan las mejores soluciones individuales y globales del enjambre.
- Se realizan varias corridas independientes del algoritmo con el fin de observar la consistencia de los resultados y reducir la probabilidad de converger a óptimos locales.

Para realizar el diseño PSO, se consideraron 30 iteraciones y 15 partículas. Debido a la baja dimensionalidad del problema, es decir, al tener solamente 3 variables de decisión como lo son el ancho, alto de viga, y número de torones, (Kennedy & Eberhart, 2002) recomienda una población pequeña para cubrir el espacio de búsqueda, además del alto costo computacional que involucra el hacer cada análisis. Además, las 30 iteraciones fueron definidas considerando este número como un límite superior por seguridad, dentro del cual, se tiene el criterio de parada de 20 iteraciones que se menciona el próximo párrafo, de este modo, si

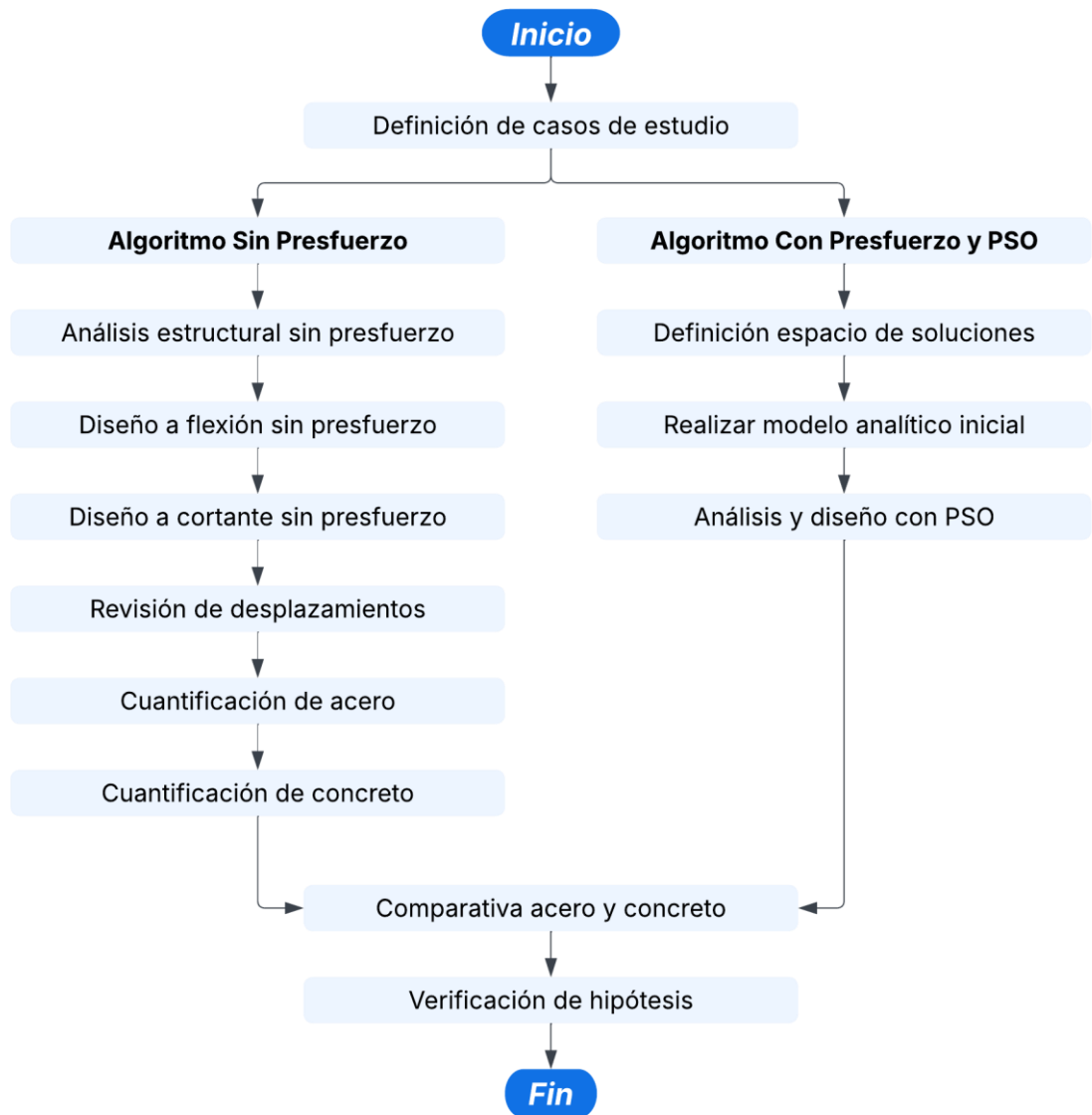
el resultado no mejora con el paso de las iteraciones, el algoritmo se detiene sin necesidad de realizar análisis innecesarios, esto es, gracias a pruebas preliminares que en las que se observó la convergencia del  $g_{best}$  alrededor de la iteración 20-25.

Por otro lado, para evitar tiempos excesivos de ejecución, se programó una memoria caché, que guarda los resultados de las soluciones evaluadas, de este modo, si se repite la misma partícula en otra iteración, ya no es necesario hacer todo el análisis completo y simplemente se usan los resultados previamente obtenidos, además, se programó un criterio de parada, el cual, detiene el algoritmo en caso que el  $g_{best}$  se repita durante 20 iteraciones consecutivas. Los valores de  $C_1$  y  $C_2$  se fijaron en 1.5 de acuerdo con el buen comportamiento de convergencia mostrado en (X. Wang & Qiu, 2013). Según los resultados obtenidos desde la creación del algoritmo PSO, se ha desarrollado el concepto de factor de inercia, con valores fijos entre 0,9 y 1,2 que dan buenos resultados, hasta trabajos actuales como el de (Meziane et al., 2019), que destaca el uso de la Ecuación 52, donde las soluciones comienzan desde la exploración y, a medida que las iteraciones llegan a su fin, las partículas se centran en la explotación, lo que garantiza una buena búsqueda en el espacio de soluciones inicialmente y, después de varias iteraciones, explota el área donde existe una mejor solución.

$$W = W_{max} - \frac{(W_{max} - W_{min})t}{Iteraciones - 1} \quad (52)$$

Donde  $W_{max}$  y  $W_{min}$  son 0,9 y 0,4 respectivamente,  $t$  es el número actual de iteraciones e  $Iteraciones$  es la cantidad total de iteraciones que se realizarán.

Con el propósito de ilustrar de manera general la metodología llevada a cabo en este trabajo, se presenta en la Figura 10, los procesos principales que se realizaron para cada análisis, el presforzado y optimizado con PSO, y el no presforzado.



**Figura 10. Metodología general de este trabajo de investigación**

En la Figura 11, se muestra el proceso del algoritmo de PSO aplicado para este trabajo de investigación, en donde se establece un criterio de parada del algoritmo, el cual dicta que cuando se repita 20 veces seguidas el valor de  $G_{best}$  el algoritmo se detendrá sin necesidad de alcanzar el número de iteraciones totales planteado previamente.

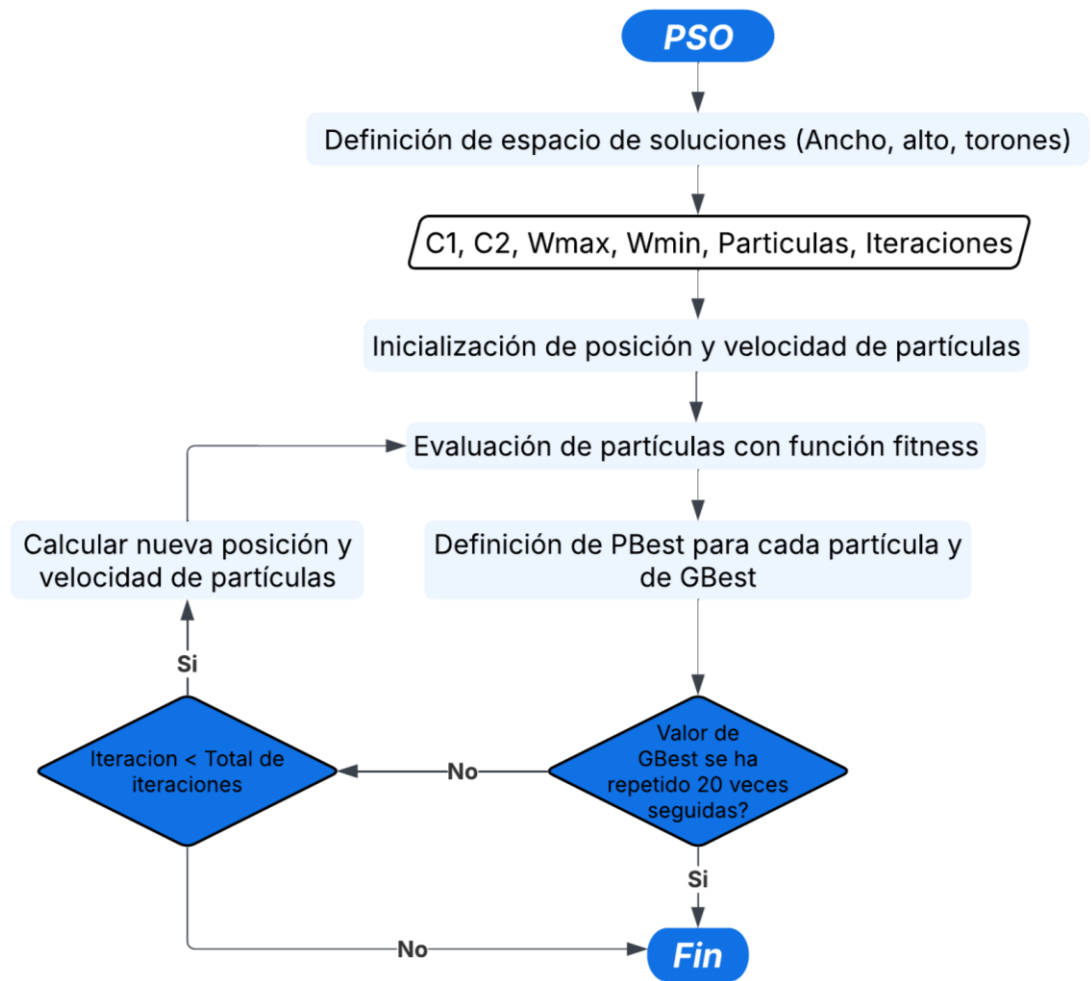


Figura 11. Diagrama de flujo de algoritmo de PSO

En la Tabla 17, se muestran los procesos llevados a cabo para el análisis sin presfuerzo, y en la Tabla 18, los procesos utilizados para el análisis con presfuerzo optimizado por PSO, además de las variables de entrada y salida de cada proceso y su función.

En el Anexo B, se muestran los códigos empleados con Python, desde los análisis estructurales con el API de SAP2000, hasta las revisiones por estados límite y la aplicación del algoritmo de optimización.

**Tabla 17. Procesos para análisis sin presfuerzo**

<b>Algoritmo Sin Presfuerzo</b>			
<b>Proceso</b>	<b>Objetivo</b>	<b>Datos de entrada</b>	<b>Datos de salida</b>
Casos de estudio	Definir lista con casos de estudio	<ul style="list-style-type: none"> <li>• Uso</li> <li>• Niveles</li> <li>• Claros</li> <li>• Crujías</li> </ul>	<ul style="list-style-type: none"> <li>• Lista de casos de estudio</li> </ul>
Análisis estructural sin presfuerzo	Realizar análisis estructural con API de SAP2000 en Python para obtención de geometría de viga y esfuerzos	<ul style="list-style-type: none"> <li>• Uso</li> <li>• Niveles</li> <li>• Claros</li> <li>• Crujías</li> </ul>	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Tabla resultados</li> <li>• <math>M_u</math></li> <li>• <math>V_u</math>,</li> <li>• Desp máx.</li> </ul>
Diseño a flexión sin presfuerzo	Obtener varillas de refuerzo por flexión	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• <math>M_u</math></li> </ul>	<ul style="list-style-type: none"> <li>• <math>A_s</math></li> <li>• <math>\emptyset</math> y # de varillas</li> <li>• <math>M_r</math></li> <li>• FS</li> <li>• Cuantía</li> </ul>
Diseño a cortante sin presfuerzo	Obtener estribos de refuerzo por cortante	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• <math>V_u</math></li> <li>• Cuantía</li> <li>• <math>\emptyset</math> de varillas</li> </ul>	<ul style="list-style-type: none"> <li>• <math>A_s</math></li> <li>• <math>\emptyset</math> y separación de estribos</li> <li>• <math>V_r</math></li> <li>• FS</li> </ul>
Revisión de desplazamientos	Confirmar si el desplazamiento vertical es correcto	<ul style="list-style-type: none"> <li>• Desp máx.</li> <li>• Claros</li> </ul>	<ul style="list-style-type: none"> <li>• Revisión</li> </ul>
Cuantificación de acero	Obtención del peso de acero de refuerzo	<ul style="list-style-type: none"> <li>• <math>\emptyset</math> y # de varillas</li> <li>• Crujías</li> <li>• Claros</li> <li>• <math>\emptyset</math> y sep. de estribos</li> <li>• B y h viga</li> </ul>	<ul style="list-style-type: none"> <li>• Peso acero</li> </ul>
Cuantificación de concreto	Obtención del peso de concreto	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Crujías</li> <li>• Claros</li> </ul>	<ul style="list-style-type: none"> <li>• Peso concreto</li> </ul>

**Tabla 18. Procesos para análisis con presfuerzo optimizado con PSO**

<b>Algoritmo Con Presfuerzo optimizado con PSO</b>			
<b>Proceso</b>	<b>Objetivo</b>	<b>Datos de entrada</b>	<b>Datos de salida</b>
Casos de estudio	Definir lista con casos de estudio	<ul style="list-style-type: none"> <li>• Uso</li> <li>• Niveles</li> <li>• Claros</li> <li>• Crujías</li> </ul>	<ul style="list-style-type: none"> <li>• Lista ordenada con casos de estudio</li> </ul>
Espacio de soluciones	Definir el espacio de soluciones	<ul style="list-style-type: none"> <li>• Claros</li> </ul>	<ul style="list-style-type: none"> <li>• Lista de soluciones con anchos, altos y Núm. de torones</li> </ul>
Modelo inicial	Realizar modelo inicial para ir cambiando las variables de decisión en los diferentes análisis, sin cerrar y abrir software	<ul style="list-style-type: none"> <li>• Uso</li> <li>• Niveles</li> <li>• Claros</li> <li>• Crujías</li> </ul>	<ul style="list-style-type: none"> <li>• Función SAP 2000</li> </ul>
Evaluar solución	Obtener resultados finales para cualquier solución	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Núm. Torones</li> <li>• Alpha</li> </ul>	<ul style="list-style-type: none"> <li>• Resultados Solución</li> </ul>
Cambio Viga	Cambiar las dimensiones de las vigas del modelo ya abierto a las de la solución en cuestión	<ul style="list-style-type: none"> <li>• B y h viga</li> </ul>	<ul style="list-style-type: none"> <li>• Función SAP 2000</li> </ul>
Cambio Torones	Cambiar la cantidad de torones del modelo ya abierto a la de la solución en cuestión	<ul style="list-style-type: none"> <li>• H viga</li> <li>• Núm. torones</li> <li>• Niveles</li> <li>• Claros</li> <li>• Crujías</li> </ul>	<ul style="list-style-type: none"> <li>• Excentricidad torones</li> <li>• Longitud torón</li> </ul>
Resultados Análisis	Obtención de resultados del análisis en curso	<ul style="list-style-type: none"> <li>• Excentricidad torones</li> <li>• Longitud torón</li> <li>• H viga</li> <li>• Ruta de Modelo</li> </ul>	<ul style="list-style-type: none"> <li>• Tabla de Resultados</li> <li>• <math>M_u</math></li> <li>• <math>V_u</math></li> <li>• Desp Max</li> <li>• Long Torón</li> <li>• Exc Torón</li> <li>• <math>V_d</math></li> <li>• <math>M_d</math></li> <li>• <math>M_u</math> Sos</li> <li>• <math>M_u</math> Tot</li> <li>• Theta Torón</li> </ul>

Continuación tabla 18

Proceso	Objetivo	Datos de entrada	Datos de salida
Diseño a flexión con presfuerzo	Obtener varillas de refuerzo por flexión	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• <math>M_u</math></li> <li>• Torones</li> <li>• Exc Torón</li> </ul>	<ul style="list-style-type: none"> <li>• <math>A_s</math></li> <li>• <math>\emptyset</math> y # de varillas</li> <li>• <math>M_r</math></li> <li>• FS</li> <li>• Cuantía</li> <li>• Penalización</li> <li>• Tipo de Varillas</li> </ul>
Diseño a cortante con presfuerzo	Obtener estribos de refuerzo por cortante	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• <math>V_u</math></li> <li>• <math>M_u</math></li> <li>• <math>V_d</math></li> <li>• <math>M_d</math></li> <li>• Exc Torón</li> <li>• Torones</li> <li>• Theta</li> <li>• <math>A_s</math></li> <li>• <math>\emptyset</math> de varillas</li> </ul>	<ul style="list-style-type: none"> <li>• <math>\emptyset</math> y separación de estribos</li> <li>• <math>V_r</math></li> <li>• FS</li> <li>• Penalización</li> </ul>
Diseño por esfuerzos permisibles	Verificar cumplimiento de tensiones y compresiones en el concreto	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Torones</li> <li>• Exc Torón</li> <li>• <math>M_u</math> Sos</li> <li>• <math>M_u</math> Tot</li> <li>• <math>M_d</math></li> <li>• <math>A_s</math></li> </ul>	<ul style="list-style-type: none"> <li>• Revisión</li> <li>• Penalización</li> <li>• Tabla Resultados</li> </ul>
Revisión de desplazamientos	Confirmar si el desplazamiento vertical es correcto	<ul style="list-style-type: none"> <li>• Desp máx.</li> <li>• Claros</li> </ul>	<ul style="list-style-type: none"> <li>• Revisión</li> <li>• Penalización</li> </ul>
Cuantificación de acero	Obtención del peso de acero de refuerzo	<ul style="list-style-type: none"> <li>• <math>\emptyset</math> y # de varillas</li> <li>• Claros</li> <li>• <math>\emptyset</math> y sep. de estribos</li> <li>• B y h viga</li> <li>• Torones</li> <li>• Long Torón</li> </ul>	<ul style="list-style-type: none"> <li>• Peso acero</li> </ul>

Continuación tabla 18

Proceso	Objetivo	Datos de entrada	Datos de salida
Cuantificación de concreto	Obtención del peso de concreto	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Crujías</li> <li>• Claros</li> </ul>	<ul style="list-style-type: none"> <li>• Peso concreto</li> </ul>
Fitness ponderado	Obtención del valor fitness de la solución evaluada	<ul style="list-style-type: none"> <li>• B y h viga</li> <li>• Torones</li> <li>• Alpha</li> </ul>	<ul style="list-style-type: none"> <li>• Fitness</li> <li>• Resultados Solución</li> </ul>
PSO	Obtención de mejor solución con algoritmo	<ul style="list-style-type: none"> <li>• <math>W_{max}</math></li> <li>• <math>W_{min}</math></li> <li>• <math>C_1</math></li> <li>• <math>C_2</math></li> <li>• Partículas</li> <li>• Iteraciones</li> </ul>	<ul style="list-style-type: none"> <li>• Mejor solución</li> </ul>

En donde:

- B: Ancho de la viga.
- h: Peralte de la viga.
- Desp máx: Desplazamiento vertical máximo (deflexión)
- $M_u$ : Momento último de diseño.
- $M_d$ : Momento debido a la carga muerta del elemento o peso propio.
- $M_u$  Sos: Momento actuante debido a la combinación de cargas sostenidas.
- $M_u$  Tot: Momento actuante total bajo la combinación de cargas totales.
- $M_r$ : Momento resistente nominal de la sección
- $V_u$ : Fuerza cortante última de diseño.
- $V_d$ : Fuerza cortante producida por el peso propio.
- $V_r$ : Fuerza cortante resistente nominal de la sección.
- FS: Factor de seguridad.

- $A_s$ : Área total de acero de refuerzo longitudinal pasivo.
- $\emptyset$ : Diámetro nominal de la varilla de acero de refuerzo.
- Exc Torón: Excentricidad de los torones medido desde el centroide de la sección.
- Long Torón: Longitud total de la trayectoria del torón.
- Theta Torón: Ángulo de inclinación del torón respecto al eje horizontal.
- PSO: Optimización por Enjambre de Partículas (Particle Swarm Optimization).
- Fitness: Valor de la función objetivo ponderada.
- Alpha: Factor de ponderación que define la importancia entre el consumo de concreto y el de acero en la función objetivo.
- $W_{max} / W_{min}$ : Límites máximo y mínimo para el factor de inercia del algoritmo.
- $C_1 / C_2$ : Coeficientes de aceleración cognitiva y social, respectivamente.

### **Etapas 3. Cuantificación de material y comparación de resultados**

Al término de las iteraciones del PSO se obtiene en la última iteración la solución que mejor se adaptó al valor buscado en la función objetivo. En este punto, se capturan las dimensiones de la sección transversal obtenida y la cantidad de torones propuestos para finalmente realizar la cuantificación del peso de concreto y aceros obtenidos para cada caso de estudio en sus dos análisis, el presforzado y el convencional. Los datos se colocan a manera de tabla comparativa y se grafican, para observar la variabilidad que existe para cada caso de estudio.

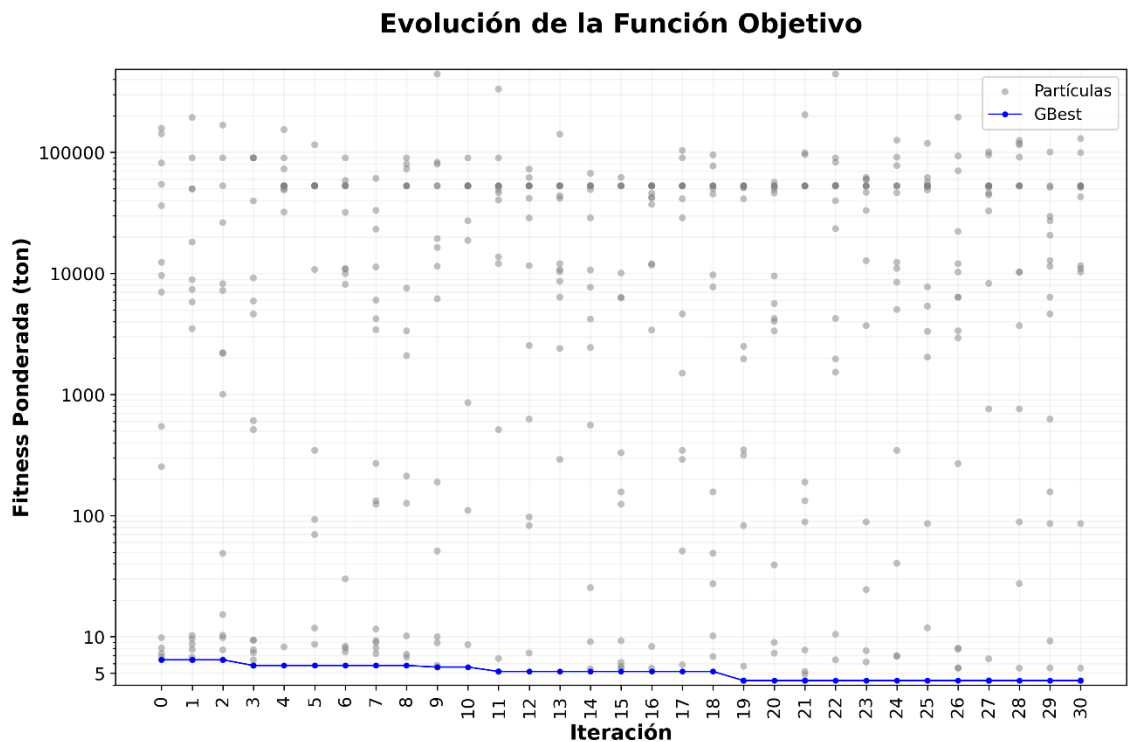
De esta manera, se cuantifica el porcentaje de ahorro de material asociado a la incorporación del postensado y al uso del algoritmo PSO, evaluando la efectividad de la metodología propuesta para cada caso de estudio.

## VI. Resultados y discusión.

Dentro de los resultados principales, se encuentran los que se refieren al comportamiento del algoritmo a lo largo del análisis para cada caso de estudio, para esto, se mostrará la evolución de la función objetivo con el paso de las iteraciones para cada caso de estudio de este trabajo.

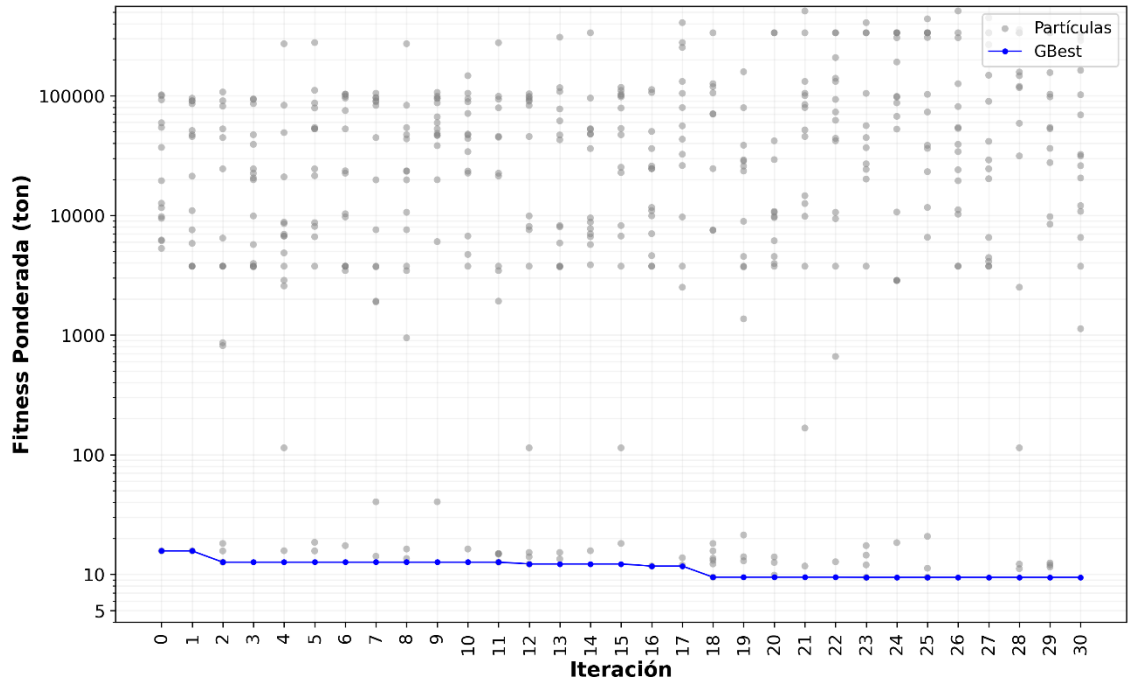
Una vez mostrado el comportamiento del algoritmo, se muestran los resultados principales que sirvieron para la comparativa base de este trabajo, como lo es la geometría o sección transversal de las vigas, y sus refuerzos de acero activo y pasivo para las distintas solicitaciones.

### Evolución de la función objetivo en casos de estudio



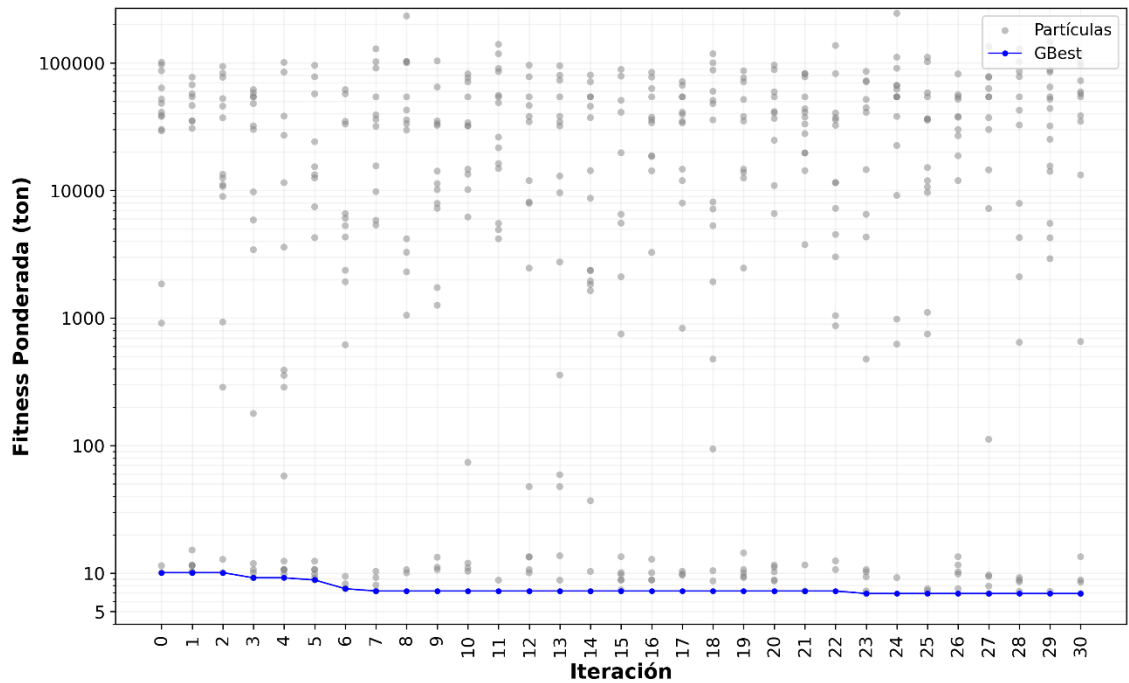
Gráfica 2. Caso 1; Hospital, 3 niveles, 9 m

### Evolución de la Función Objetivo



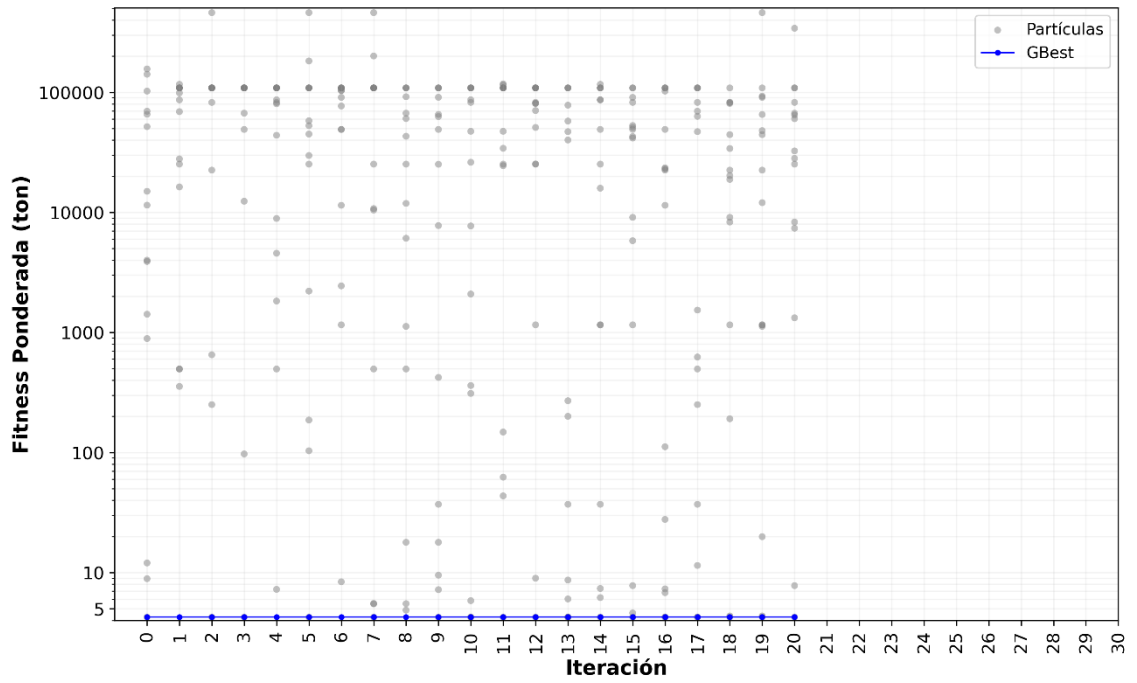
Gráfica 3. Caso 2; Hospital, 3 niveles, 11 m

### Evolución de la Función Objetivo



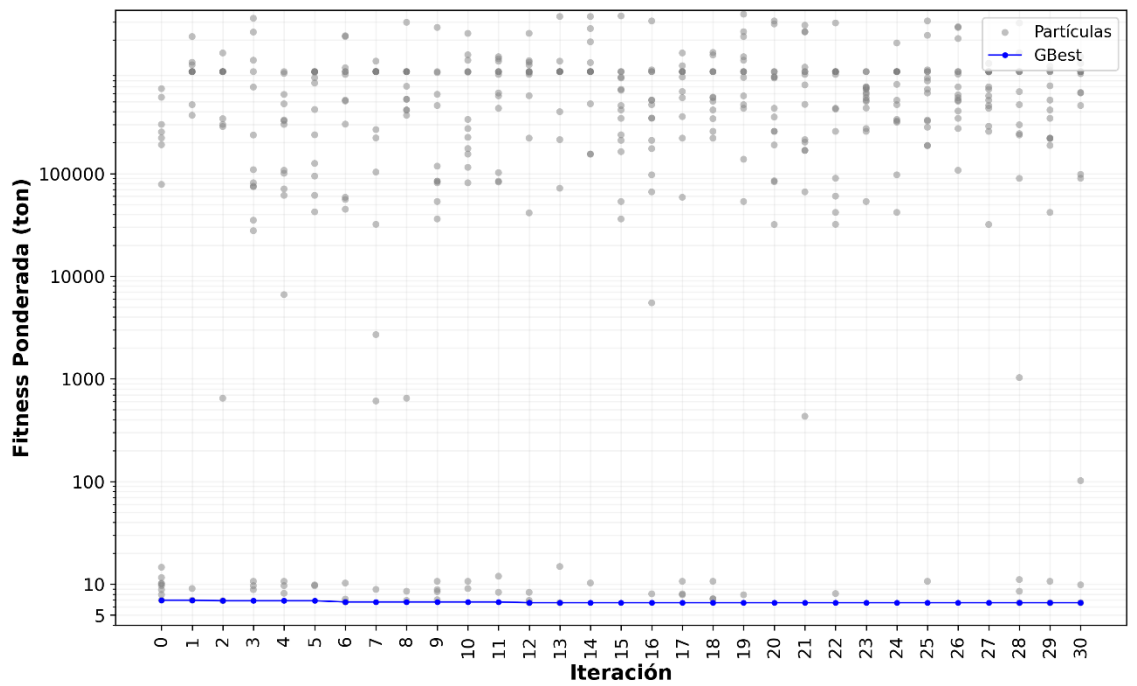
Gráfica 4. Caso 3; Hospital, 3 niveles, 13 m

**Evolución de la Función Objetivo**



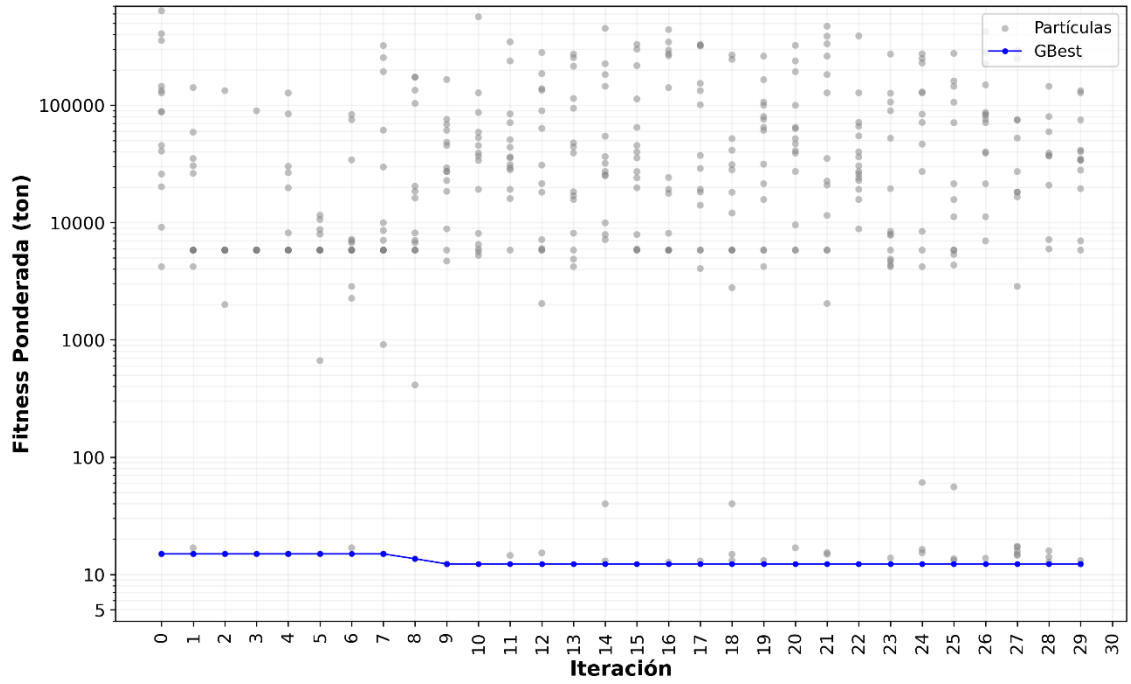
**Gráfica 5. Caso 4; Hospital, 5 niveles, 9 m**

**Evolución de la Función Objetivo**



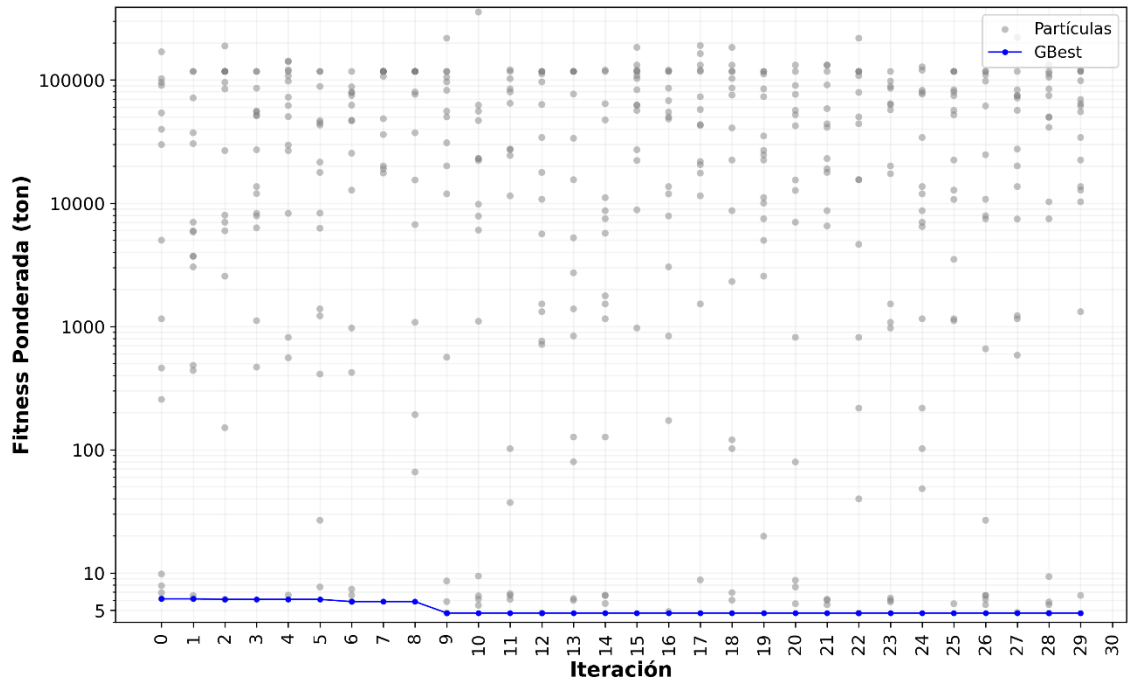
**Gráfica 6. Caso 5; Hospital, 5 niveles, 11 m**

**Evolución de la Función Objetivo**



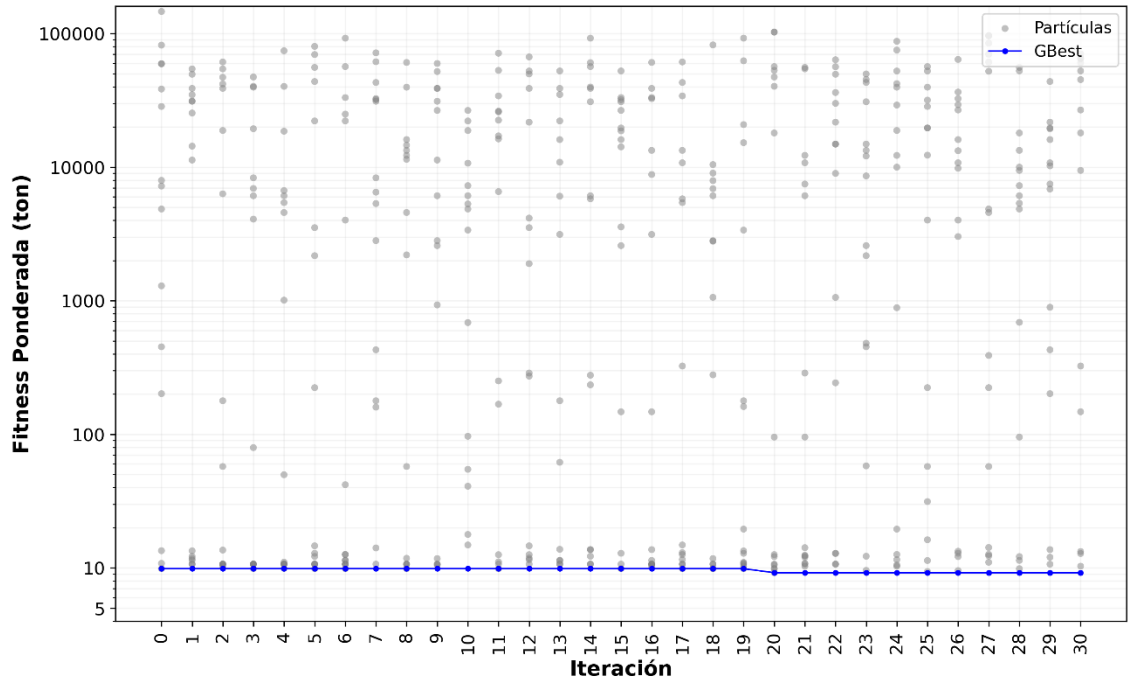
**Gráfica 7. Caso 6; Hospital, 5 niveles, 13 m**

**Evolución de la Función Objetivo**



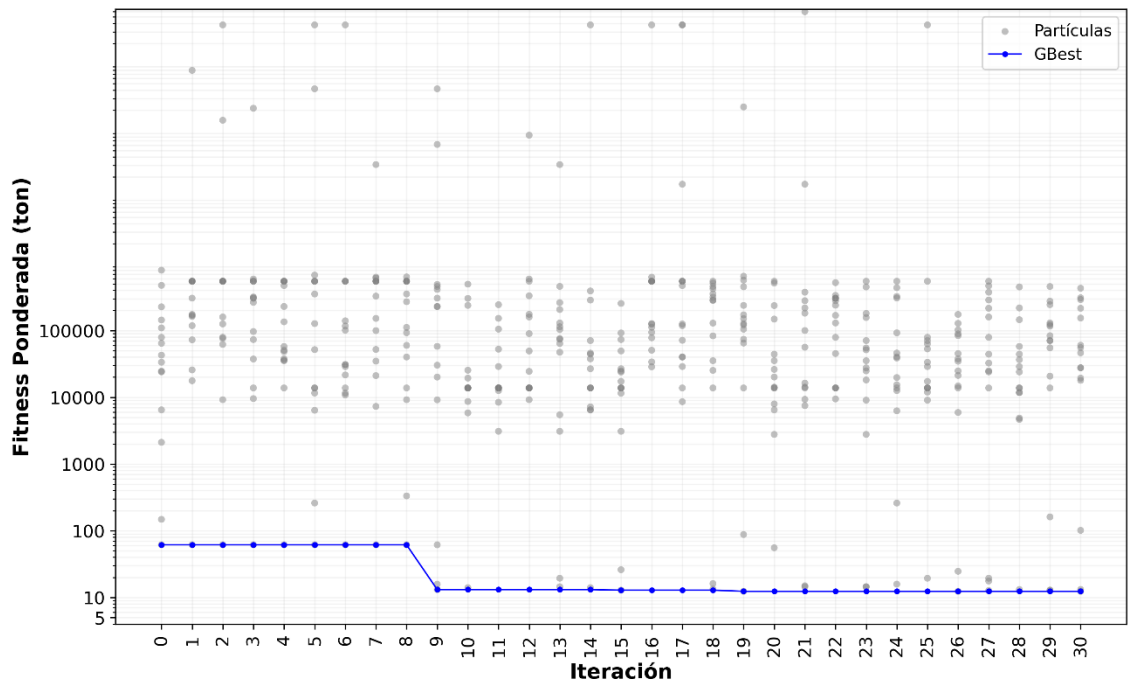
**Gráfica 8. Caso 7; Hospital, 7 niveles, 9 m**

### Evolución de la Función Objetivo



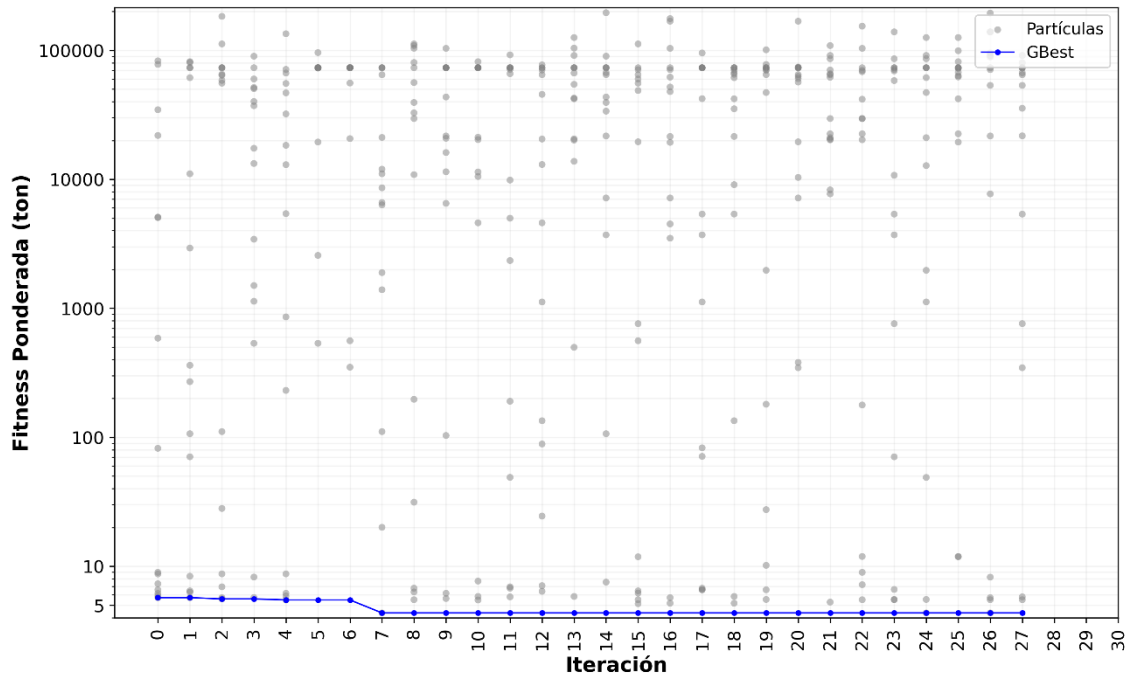
Gráfica 9. Caso 8; Hospital, 7 niveles, 11 m

### Evolución de la Función Objetivo



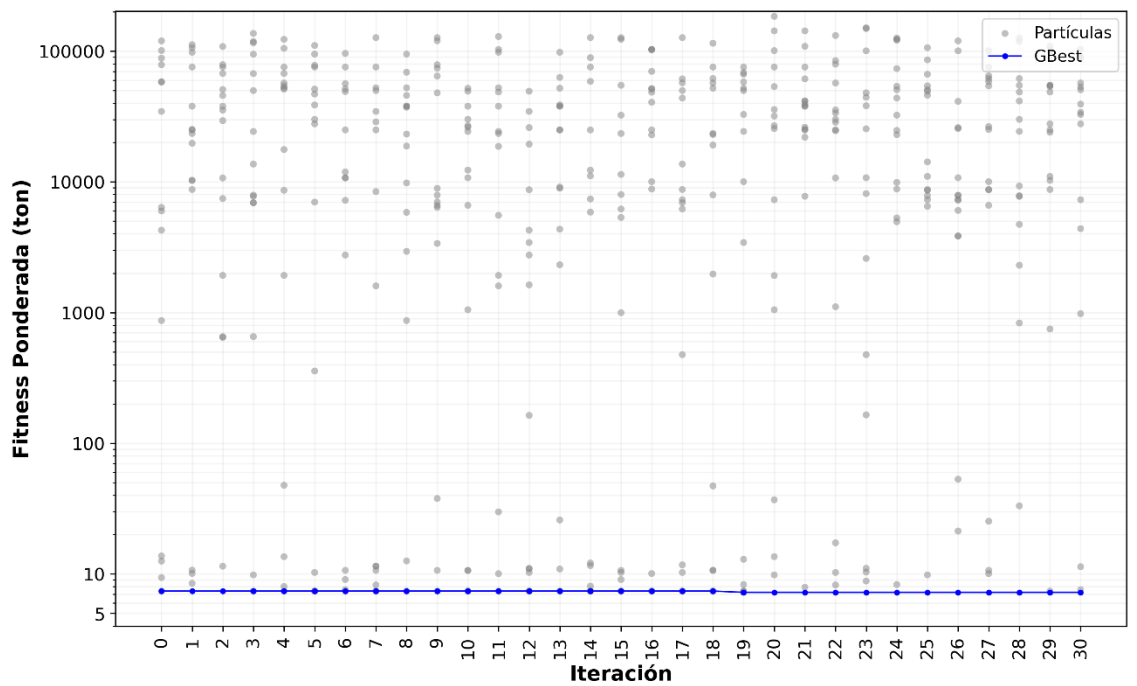
Gráfica 10. Caso 9; Hospital, 7 niveles, 13 m

**Evolución de la Función Objetivo**



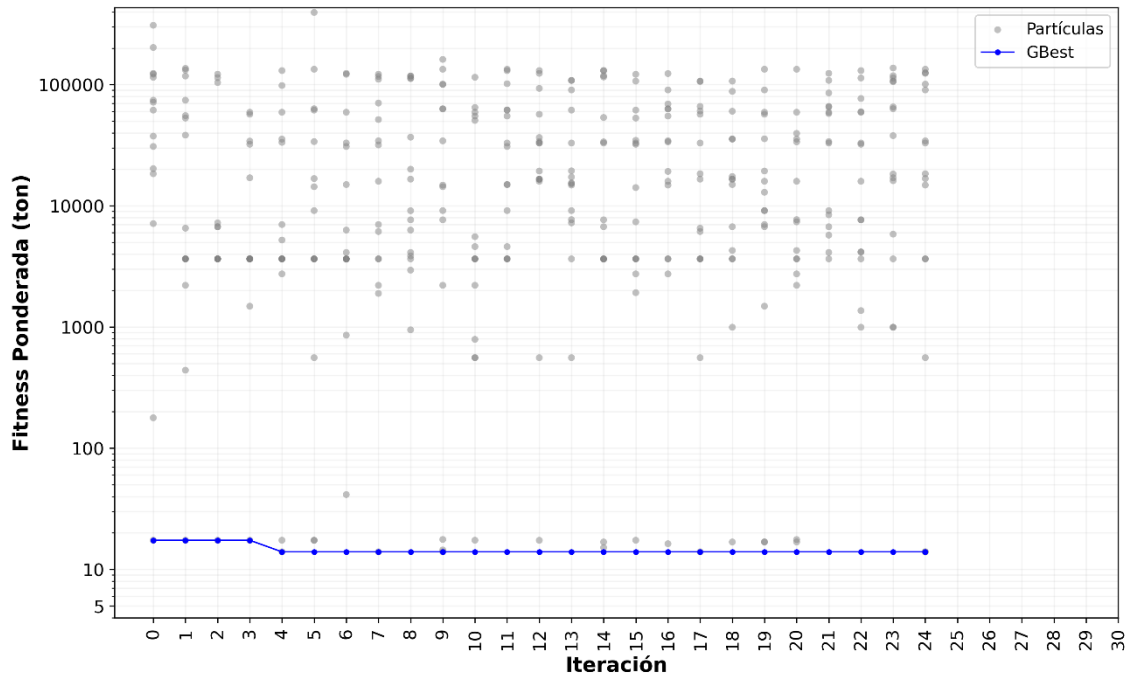
**Gráfica 11. Caso 10; Estacionamiento, 3 niveles, 9 m**

**Evolución de la Función Objetivo**



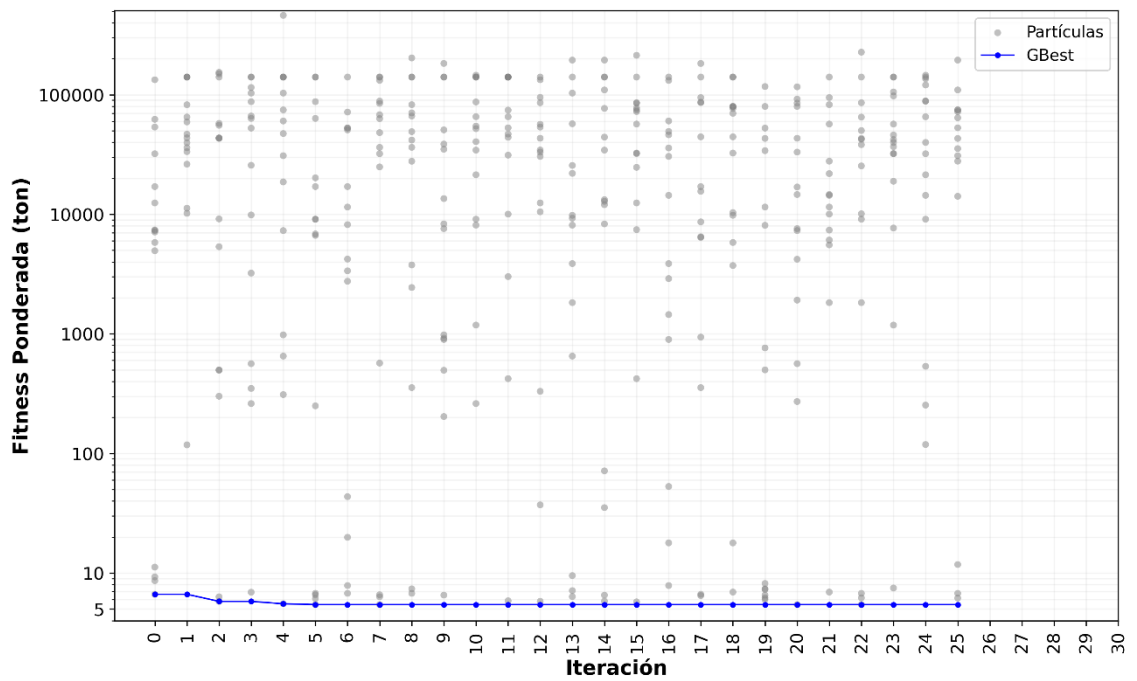
**Gráfica 12. Caso 11; Estacionamiento, 3 niveles, 11 m**

**Evolución de la Función Objetivo**



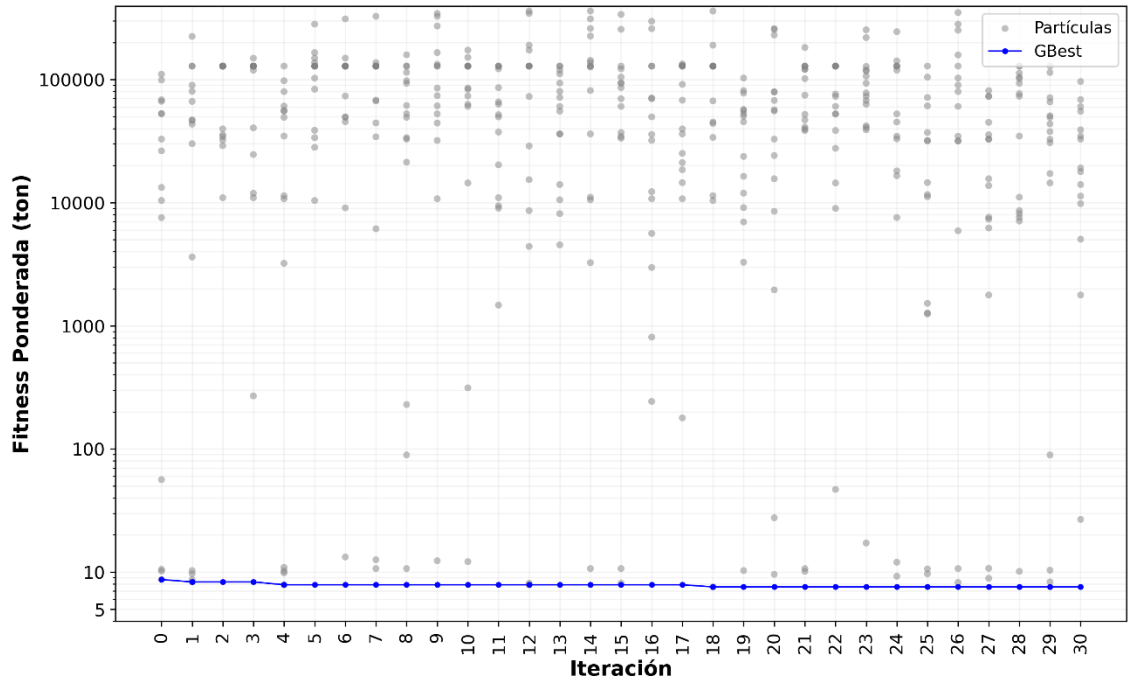
**Gráfica 13. Caso 12; Estacionamiento, 3 niveles, 13 m**

**Evolución de la Función Objetivo**



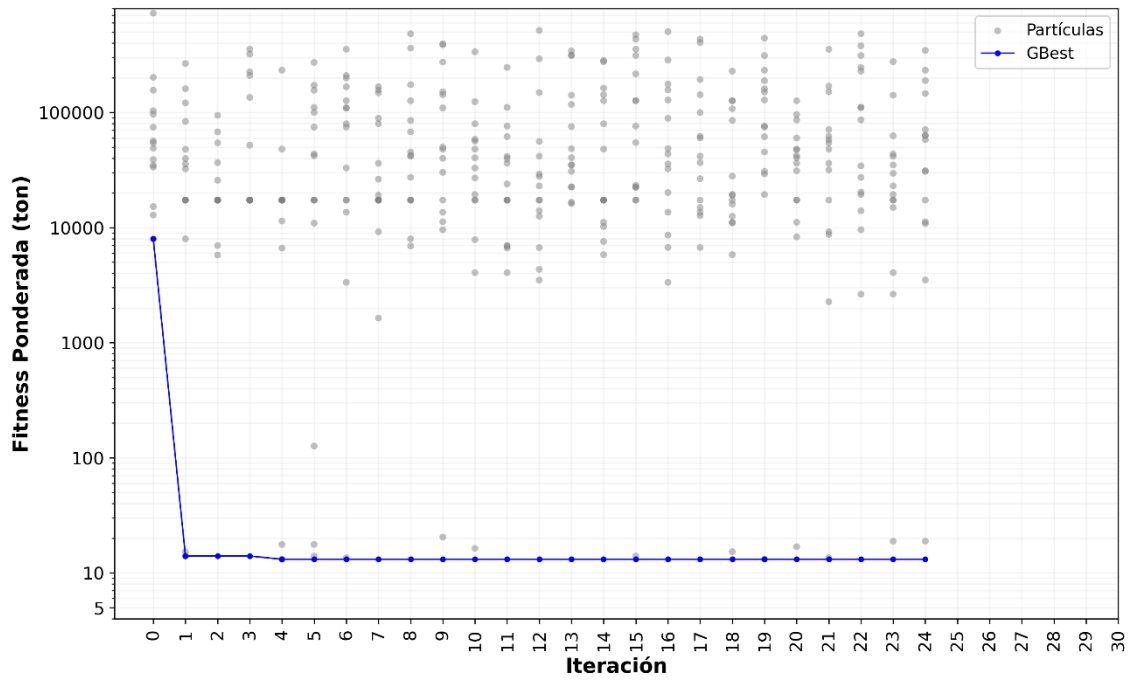
**Gráfica 14. Caso 13; Estacionamiento, 5 niveles, 9 m**

**Evolución de la Función Objetivo**



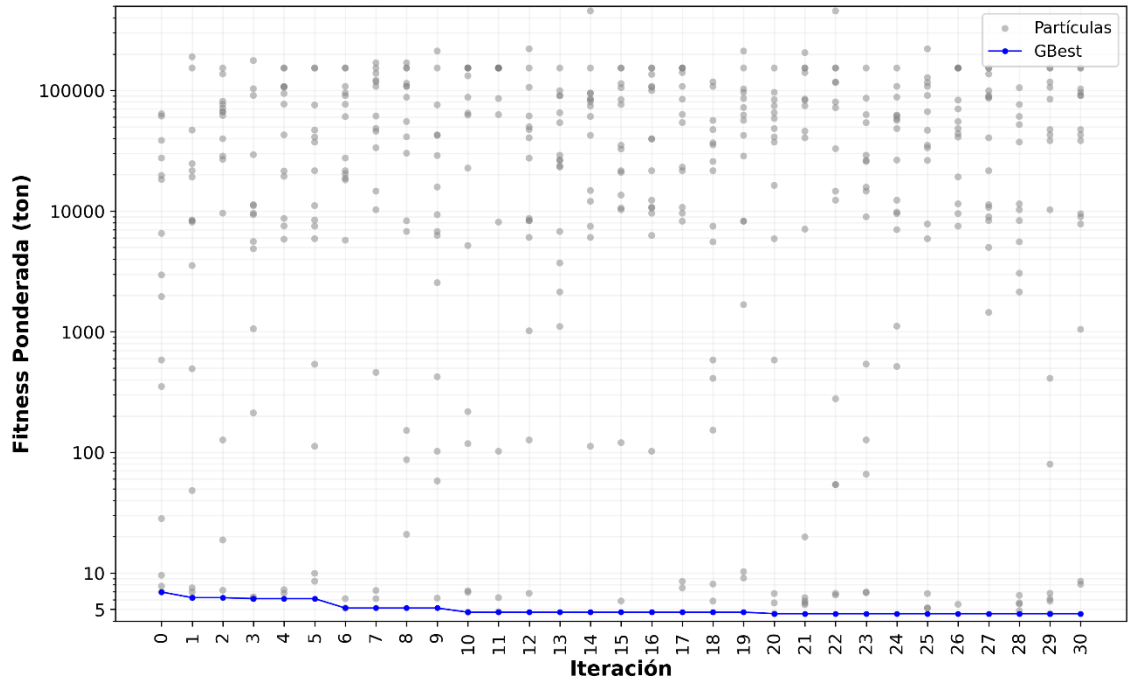
**Gráfica 15. Caso 14; Estacionamiento, 5 niveles, 11 m**

**Evolución de la Función Objetivo**



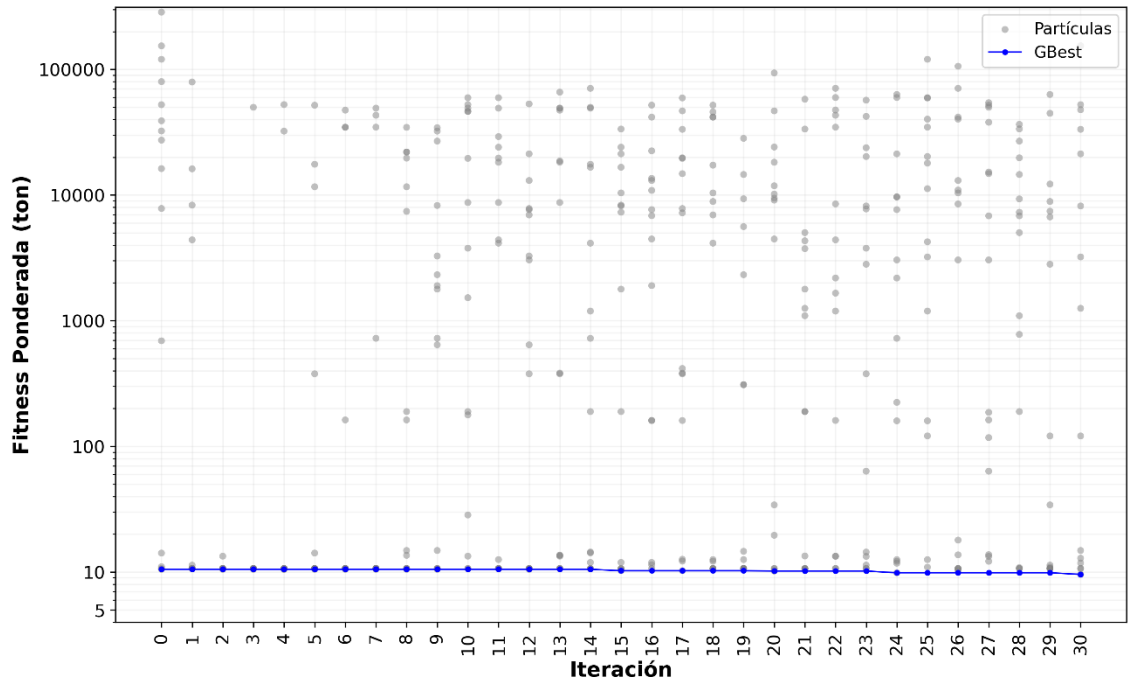
**Gráfica 16. Caso 15; Estacionamiento, 5 niveles, 13 m**

**Evolución de la Función Objetivo**



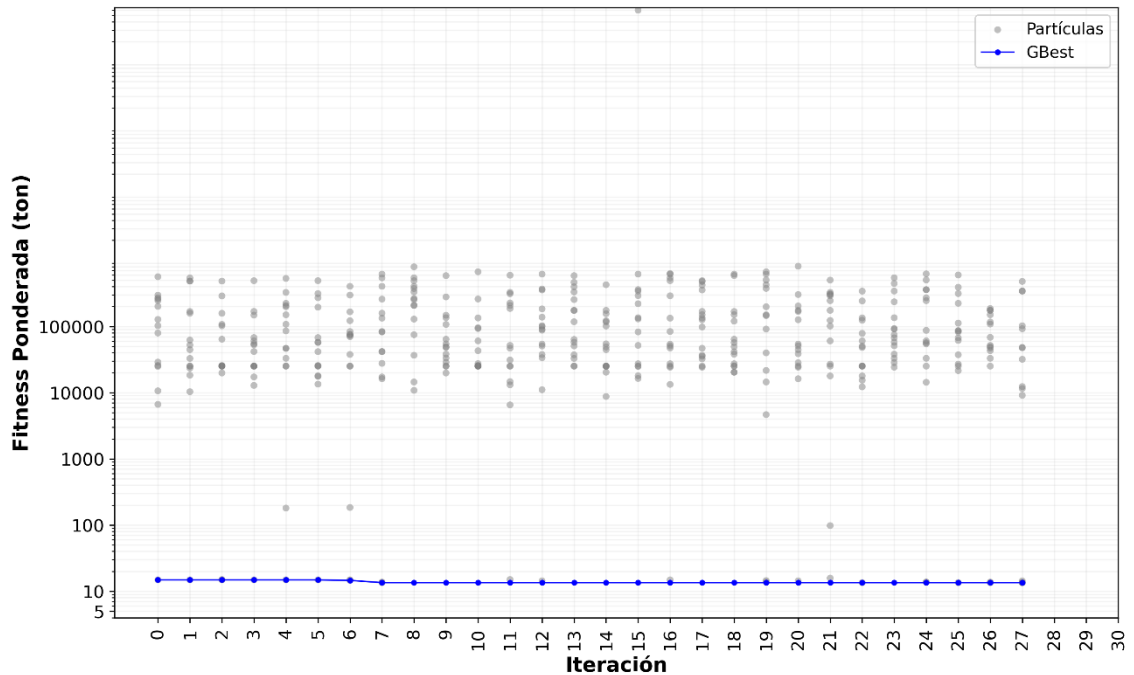
**Gráfica 17. Caso 16; Estacionamiento, 7 niveles, 9 m**

**Evolución de la Función Objetivo**



**Gráfica 18. Caso 17; Estacionamiento, 7 niveles, 11 m**

### Evolución de la Función Objetivo



Gráfica 19. Caso 18; Estacionamiento, 7 niveles, 13 m

De la Gráfica 2 a la Gráfica 19, se observa cómo se comporta la función objetivo con el paso de las iteraciones para distintos casos de estudio, de modo que en algunas se mantiene como mejor resultado global ( $g_{best}$ ) una solución, pero después de unas cuantas iteraciones, este va mejorando, lo cual se aprecia en los quiebres que lleva la gráfica.

Es posible ver la velocidad de convergencia de las partículas, ya que en algunos casos como los de la Gráfica 5, la cantidad de iteraciones fue menor a las 30 que se pretendieron hacer, esto es debido a un criterio de parada que se implementó en el algoritmo, el cual, hace que cuando el  $g_{best}$  no cambie después de 20 iteraciones, el algoritmo se detenga y deje de iterar.

Lo anterior nos dice, que el algoritmo se comportó de una manera eficiente pues logró encontrar una buena solución en pocas iteraciones, ayudando a ahorrar en cuanto al esfuerzo computacional aplicado para dicho caso de estudio.

En zonas donde se tienen caídas bruscas en la gráfica del fitness, significa que es el punto en el que las partículas encontraron una zona factible, en la cual, ya no se están aplicando penalizaciones debido a la violación de algunas de las restricciones propuestas.

Por otro lado, en zonas donde la curva del fitness, se mantiene plana, o la caída es muy suave, significa que ya se entró en una zona de explotación, en la cual las partículas solo están buscando en una zona más reducida con el fin de pulir aún más la mejor solución encontrada.

Las gráficas llevan una escala logarítmica con el fin de observar de manera más cómoda todas las partículas debido a las grandes penalizaciones que algunas están teniendo, sin embargo, es posible observar cierta brecha vacía entre las soluciones que si están cumpliendo con las restricciones y las que están siendo penalizadas. Esto quiere decir que los rangos de penalización están actuando correctamente, dirigiendo a las partículas hacia regiones estructuralmente viables.

En algunos casos se observa cómo se sitúan diversas partículas dentro de esa brecha, esto es, debido a que son partículas cuya penalización no es tan fuerte como las que se encuentran con pesos en un rango de  $10^4$  ton, lo cual nos demuestra que el algoritmo no se estanca del todo en una sola región, sino que si está considerando un gran campo de soluciones a lo largo de sus iteraciones.

Se observa como a partir de la iteración 25, en términos generales, se alcanza una estabilidad del diseño óptimo, lo cual, demuestra la conveniencia de utilizar 30 iteraciones y evitar un mayor costo computacional innecesario.

Por otro lado, es entre las iteraciones 15 y 25 que las partículas pasan de una exploración global, a una explotación local, ya que la nube de partículas se empieza a concentrar en las franjas inferiores de la escala logarítmica.

Una vez obtenidos y analizados los resultados para cada caso de estudio fue posible comprender como se comporta el algoritmo.

### Sección transversal y cantidad de acero obtenidos.

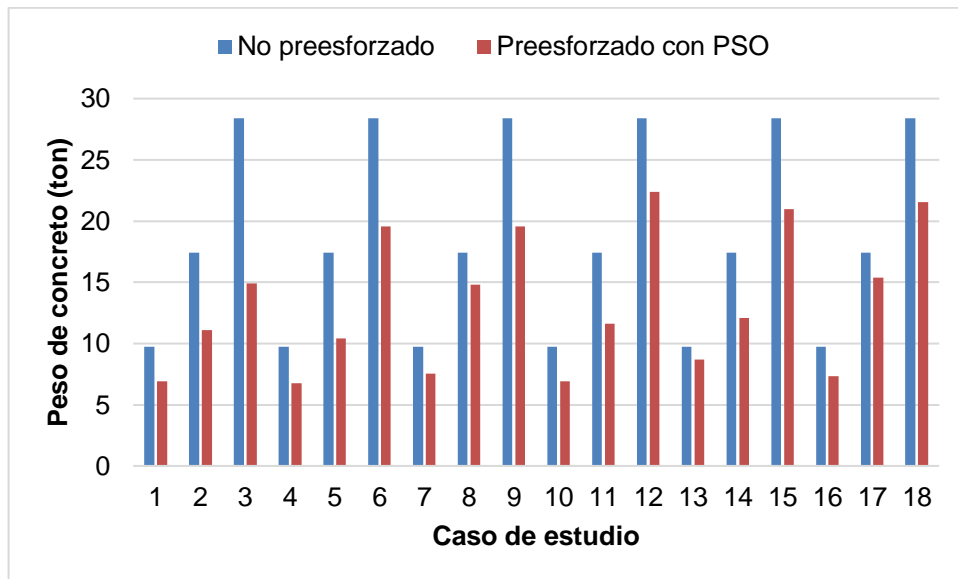
A continuación, se muestra en la Tabla 19 los resultados obtenidos para ambos diseños en cuanto a la geometría de la viga.

**Tabla 19. Sección transversal de viga para ambos diseños**

Caso	Sin presfuerzo		Con presfuerzo	
	Ancho (cm)	Alto (cm)	Ancho (cm)	Alto (cm)
1	25	45	25	32
2	30	55	25	42
3	35	65	26	46
4	25	45	26	30
5	30	55	28	35
6	35	65	28	56
7	25	45	25	35
8	30	55	25	56
9	35	65	28	56
10	25	45	25	32
11	30	55	25	44
12	35	65	28	64
13	25	45	28	36
14	30	55	26	44
15	35	65	28	60
16	25	45	25	34
17	30	55	26	56
18	35	65	32	54

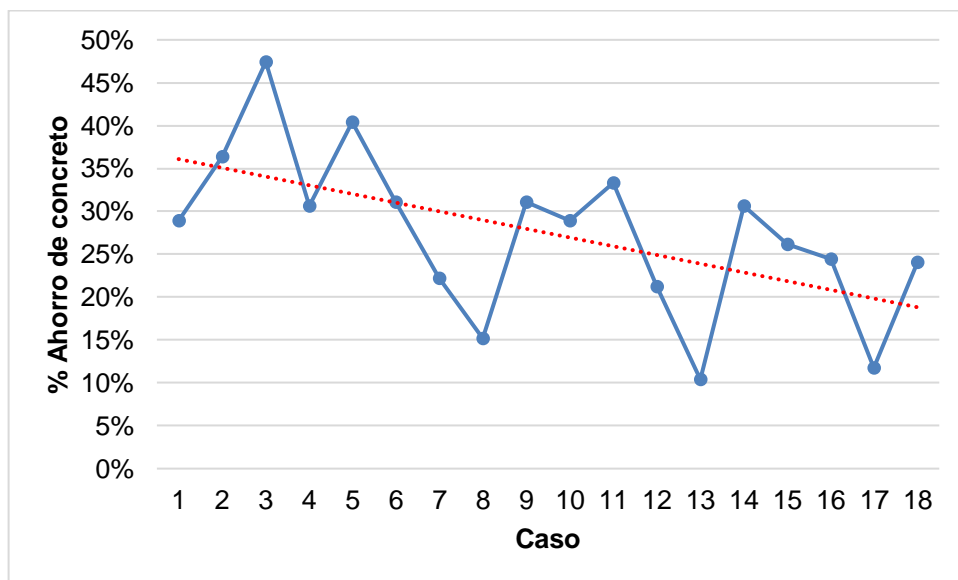
Se puede apreciar cómo es que las medidas de la viga de estudio si va disminuyendo en el diseño presforzado con PSO en relación al diseño convencional. Ahora se Muestra en la Gráfica 20, la comparativa del peso de concreto entre el diseño sin presfuerzo, y el presforzado optimizado con PSO.

Es posible observar los resultados detallados tanto para análisis sin presfuerzo como para el presforzado con PSO en el Anexo A, desde la geometría de las secciones, hasta los esfuerzos actuantes, resistentes, deformaciones y armados de acero para cada caso de estudio.



**Gráfica 20. Comparativa de volumen de concreto para diseño sin presfuerzo vs diseño con presfuerzo y PSO**

Es evidente que, si se está teniendo un ahorro en el diseño presfuerzo con PSO, la pregunta es, qué tanto se está ahorrando, esto se aprecia en la Gráfica 21.



**Gráfica 21. Porcentaje de ahorro de concreto**

De acuerdo a los datos obtenidos para todos los casos de estudio, en promedio se está ahorrando un 27% de concreto.

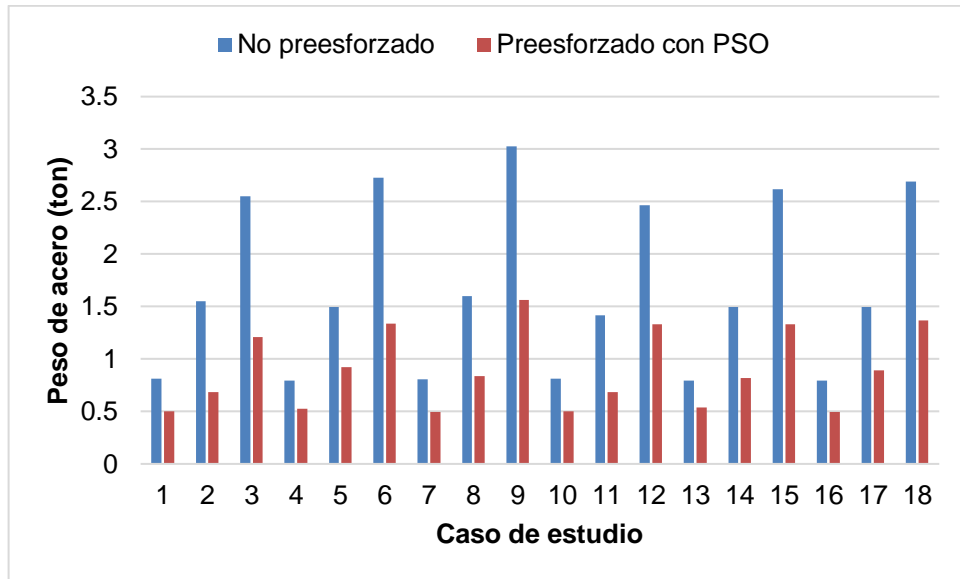
Ahora procederemos con la tabla comparativa de cantidades de acero longitudinal y transversal pasivo, así como los torones o acero activo para obtener las cantidades de acero que se tienen en cada caso para cada diseño en la Tabla 20, posteriormente, se muestran las cantidades de acero de ambos diseños en la Gráfica 22

**Tabla 20. Resultados de acero longitudinal, transversal y torones para ambos diseños.**

Caso	Diseño sin presfuerzo								Diseño presforzado optimizado con PSO								
	Lecho superior			Lecho inferior			Estribos		Lecho superior			Lecho inferior			Estribos		Tor
	Var	No	Tipo	Var	No	Tipo	No	Sep	Var	No	Tipo	Var	No	Tipo	No	Sep	
1	3	8	S	6	4	P	3	18	3	4	S	3	4	S	3	14	4
2	8	6	P	4	6	S	4	22	3	4	S	3	4	S	3	14	5
3	4	10	S	8	5	P	3	10	4	5	P	4	4	S	3	10	6
4	3	8	S	5	4	S	3	16	3	4	S	3	4	S	3	12	4
5	8	6	P	5	5	S	4	20	4	5	P	3	4	S	3	10	5
6	3	12	S	3	8	S	4	16	6	4	P	2	6	S	3	10	7
7	3	8	S	5	4	S	3	15	3	4	S	3	4	S	3	15	4
8	3	10	S	5	5	S	3	10	4	4	S	4	4	S	3	15	6
9	8	8	P	3	8	S	4	16	2	8	S	2	6	S	3	8	7
10	3	8	S	6	4	P	3	18	3	4	S	3	4	S	3	14	4
11	4	8	S	2	8	S	3	12	2	5	S	2	5	S	3	15	5
12	6	8	P	3	8	S	3	10	6	4	P	6	4	P	3	15	8
13	3	8	S	5	4	S	3	16	3	4	S	3	4	S	3	15	5
14	8	6	P	5	5	S	4	22	4	4	S	4	4	S	3	12	5
15	3	12	S	5	6	S	4	18	5	4	S	5	4	S	3	10	7
16	3	8	S	5	4	S	3	16	3	4	S	3	4	S	3	15	4
17	8	6	P	5	5	S	4	22	2	6	S	2	6	S	3	15	6
18	3	12	S	5	6	S	4	16	5	4	S	5	4	S	3	10	8

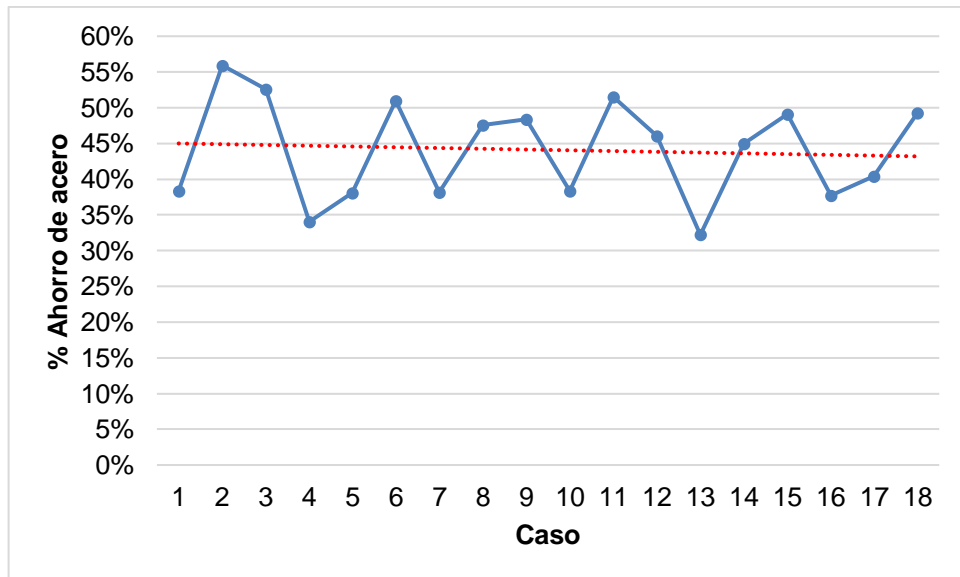
Donde:

- Var: Cantidad de varillas de acero pasivo
- No: Diámetro de varilla en octavos de pulgada
- Tipo: Acomodo de varillas (“S” varillas sencillas y “P” en paquetes de dos)
- Sep: Separación de estribos



**Gráfica 22. Comparativa de cantidad de acero para viga de estudio con diseño convencional vs diseño presforzado con PSO**

Al igual que el concreto, se aprecia una reducción, la cual se Muestra en porcentaje de ahorro en cada caso de estudio en la Gráfica 23.



**Gráfica 23. Porcentaje de ahorro de acero**

En este caso, se está logrando un ahorro promediado de 44% de acero.

## VII. Conclusiones y recomendaciones.

Se demostró que la aplicación del algoritmo PSO ha resultado ser una herramienta de gran utilidad para problemas de optimización estructural para trabes de concreto postensado, ya que, en la mayoría de los casos de estudio, se logró llegar a la convergencia en menos de 25 iteraciones. Esto se logró en gran parte gracias a la manera en que se manejaron las penalizaciones para las restricciones del problema de optimización, ya que al ser estas de carácter dinámico, es decir, variaban en la proporción en que tan buena o mala fuera una solución, se logró evitar que el algoritmo se estancara en óptimos locales y pudiera explorar en mayor forma el campo de soluciones regido por las revisiones tanto por diseño por esfuerzos permisibles (ASD) como por diseño por factores de carga y resistencia (LRFD).

Se lograron ahorros significativos en los materiales de las vigas, teniendo una reducción promedio del 27% en concreto y del 44% en acero en comparación del diseño convencional. Esto resalta que el ahorro no se debe simplemente a disminuir la sección transversal de la viga, sino también a la inclusión del presfuerzo, cuyo aporte en cuanto a flexión, cortante y desplazamientos, llevó a secciones más esbeltas y con menor cantidad de acero de refuerzo.

De esta manera básicamente en los diseños presforzados con PSO, se redujo el área de acero convencional al mínimo sin comprometer la seguridad de la edificación.

Gracias a ello, se puede decir que se cumplió con la hipótesis planteada, donde se buscaba llegar a una reducción de al menos un 20% en concreto y acero.

Se observó que, en claros mayores, como lo son los de 11 y de 13 metros, el algoritmo fue más eficiente, debido a que el peso propio de las vigas mismas, es un factor de gran dominancia, por otro lado, con respecto al uso de la edificación, al tratarse de hospitales el ahorro de material fue menos significativo que en

estacionamientos, lo cual nos dice que entre mayor sea la exigencia de cargas, el espacio de búsqueda óptimo se reduce.

Es necesario destacar que en el caso 5, como se puede observar en el apartado de Anexo A, la revisión de tensión en fibra inferior por esfuerzos permisibles no cumple por muy poco, esto se debe a la diferencia tan pequeña que hay entre lo admisible y lo actuante, de modo que la penalización cuadrática no impone un castigo tan severo en este caso, conservando esta solución como la óptima en relación a las demás analizadas en este caso.

Cabe mencionar que las soluciones obtenidas, más allá de tener un enfoque teórico para la culminación de este trabajo, son soluciones altamente calificadas para ser llevadas a la práctica profesional, pues se procuró el cumplimiento de todos los requisitos y estándares solicitados por la normativa aplicable para elementos de concreto reforzado con y sin presfuerzo. Desde la inclusión de varillas de acero corrugado de diámetros comerciales, hasta las revisiones de separaciones mínimas, máximas, recubrimientos, entre otros, donde se demuestra que el código realizado puede llevar a resultados aptos para ser aplicables en obra sin problemas de obstrucción del acero en el colado, y que cumplan con las resistencias solicitadas.

Este trabajo fue hecho basándose en las (NTC, 2023a), lo cual demuestra la correlación que puede haber entre las herramientas computacionales modernas y la normativa vigente aplicable a la construcción en nuestro país.

Es importante mencionar que este modelo se puede escalar a problemas de optimización de edificios completos, no solamente de un eje de crujeas para tener un proyecto integral de la estructura completa, además se puede hacer la verificación de la huella de carbono o un análisis de costos para complementar los resultados y tener una mejor evaluación de los beneficios de la metodología aplicada.

## VIII. Bibliografía.

- A. Kaveh. (2017). Applications of Metaheuristic Optimization Algorithms in Civil Engineering. En *Springer*. Springer. [https://doi.org/10.1007/978-3-319-48012-1\\_1](https://doi.org/10.1007/978-3-319-48012-1_1)
- Alcocer, S. M. (2012). *Comportamiento sísmico de elementos de concreto presforzado y postensado*.
- ANIPPAC. (2018). *Manual de diseño de estructuras prefabricadas y presforzadas*.
- Annicchiarico, W. D., & Cerrolaza, M. (2019). Improved Dynamical Particle Swarm Optimization Method for Structural Dynamics. *Mathematical Problems in Engineering*, 2019. <https://doi.org/10.1155/2019/8250185>
- Asgshahr, M. S. (2021). Reliability based design optimization of reinforced concrete frames using genetic algorithm. *Periodica Polytechnica Civil Engineering*, 65, 566–576. <https://doi.org/10.3311/PPci.17150>
- Ayyash, A. jbury N. A., & Hejazi, F. (2023). Development of hybrid performance-based optimization algorithm for structures equipped with vibration damper devices. *Archives of Civil and Mechanical Engineering*, 23(2). <https://doi.org/10.1007/s43452-023-00665-z>
- Deb, K. (2011). *Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction*. <http://www.iitk.ac.in/kangal/deb.htm>
- Dong, G., Garcia, R., Pilakoutas, K., & Hajirasouliha, I. (2024). A review of optimum seismic design of RC frames: State-of-the-art, challenges and future directions. En *Engineering Structures* (Vol. 315). Elsevier Ltd. <https://doi.org/10.1016/j.engstruct.2024.118455>
- Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle Swarm Optimisation: A historical review up to the current developments. En *Entropy* (Vol. 22, Número 3). MDPI AG. <https://doi.org/10.3390/E22030362>
- Guo, T., Chen, Z., Lu, S., & Yao, R. (2017). *Monitoring and analysis of long-term prestress losses in post-tensioned concrete beams*. <https://www.sciencedirect.com/science/article/pii/S0263224117304979>
- Hajirasouliha, I., Asadi, P., & Pilakoutas, K. (2012). An efficient performance-based seismic design method for reinforced concrete frames. *Earthquake Engineering and Structural Dynamics*, 41(4). <https://doi.org/10.1002/eqe.1150>
- IEA. (2022, septiembre 22). *Global cement demand for building construction in the Net Zero Scenario, 2000-2030*. Licence: CC BY 4.0.

- Jain, N. K., Nangia, U., & Jain, J. (2018). A Review of Particle Swarm Optimization. En *Journal of The Institution of Engineers (India): Series B* (Vol. 99, Número 4, pp. 407–411). Springer.  
<https://doi.org/10.1007/s40031-018-0323-y>
- Kennedy, J., & Eberhart, R. (2002). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1942–1948.  
<https://doi.org/10.1109/ICNN.1995.488968>
- Kral'ovanec, J., Moravčík, M., Bujňáková, P., & Jošt, J. (2021). Indirect determination of residual prestressing force in post-tensioned concrete beam. *Materials*, 14(6).  
<https://doi.org/10.3390/ma14061338>
- Mergos, P. E. (2018). Seismic design of reinforced concrete frames for minimum embodied CO2 emissions. *Energy and Buildings*, 162, 177–186.  
<https://doi.org/10.1016/j.enbuild.2017.12.039>
- Meziane, M. A., Mouloudi, Y., Bouchiba, B., & Laoufi, A. (2019). Impact of inertia weight strategies in particle swarm optimization for solving economic dispatch problem. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 377–383.  
<https://doi.org/10.11591/ijeecs.v13.i1.pp377-383>
- Nouri, A., Asadi, P., & Taheriyoun, M. (2020). Life-cycle sustainability design of RC frames under the seismic loads. *Asian Journal of Civil Engineering*, 21(2), 293–310.  
<https://doi.org/10.1007/s42107-019-00199-x>
- NTC. (2023a). *Normas Técnicas Complementarias para diseño y construcción de estructuras de concreto*. Secretaría de Obras y Servicios.
- NTC. (2023b). *Normas Técnicas Complementarias sobre criterios y acciones para el diseño estructural de edificaciones*.
- Park, H. W., Lee, D., Hwang, J. H., & Lee, S. C. (2026). Rational Estimation of Initial State of Post-tensioned Concrete Beams Using Field Measurement Data. *International Journal of Concrete Structures and Materials*, 20(1). <https://doi.org/10.1186/s40069-026-00910-8>
- Quintanilla Huayta, D. (2016). *Ensayos experimentales en vigas de concreto postensado con tendones adheridos y no adheridos*. Pontificia Universidad Católica de Perú.
- Reyes-Sierra, M., & Coello Coello, C. A. (2006). Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. En *International Journal of Computational Intelligence Research* (Vol. 2, Número 3). <http://delta.cs.cinvestav.mx/>
- Tuncer, E., Karageyik, C., Binici, B., Canbay, E., & Muka, N. (2025). BEHAVIOR OF POST-TENSIONED REINFORCED CONCRETE BEAMS UNDER CYCLIC LOADING. *Proceedings of*

- International Structural Engineering and Construction*, 12(1).  
[https://doi.org/10.14455/ISEC.2025.12\(1\).STR-66](https://doi.org/10.14455/ISEC.2025.12(1).STR-66)
- Velázquez, C. F., & Mejía, M. (2016). *Análisis del algoritmo de optimización por enjambre de partículas por medio de una aplicación gráfica 3D*.  
<https://www.researchgate.net/publication/303677894>
- Wang, T. H., Huang, R., & Wang, T. W. (2013). The variation of flexural rigidity for post-tensioned prestressed concrete beams. *Journal of Marine Science and Technology (Taiwan)*, 21(3), 300–308. <https://doi.org/10.6119/JMST-012-0508-2>
- Wang, X., & Qiu, X. (2013). Application of Particle Swarm Optimization for Enhanced Cyclic Steam Stimulation in a Offshore Heavy Oil Reservoir. *International Journal of Information Technology, Modeling and Computing*, 1(2), 37–47.  
<https://doi.org/10.5121/ijitmc.2013.1204>
- Wen, Y., Guo, R., Duan, Z., Tong, Y., Tang, X., Pan, T., & Fu, C. (2026). Machine learning model optimization with optuna for accurate prediction of strength and crack behavior in prestressed concrete beams. *Scientific Reports*. <https://doi.org/10.1038/s41598-026-36692-y>
- Yang, X. S., Bekdas, G., & Nigdeli, S. M. (2016). Metaheuristics and Optimization in Civil Engineering. En *Modeling and Optimization in Science and Technologies* (Vol. 7, pp. v–vi). Springer. <https://doi.org/10.1007/978-3-319-26245-1>
- Zhu, Q., Lin, Q., Chen, W., Wong, K. C., Coello Coello, C. A., Li, J., Chen, J., & Zhang, J. (2017). An External Archive-Guided Multiobjective Particle Swarm Optimization Algorithm. *IEEE Transactions on Cybernetics*, 47(9), 2794–2808.  
<https://doi.org/10.1109/TCYB.2017.2710133>

# **ANEXOS**

## ANEXO A.

### TABLAS DE RESULTADOS Y ESFUERZOS COMPLEMENTARIOS

Tabla A 1. Resultados principales para análisis sin presfuerzo

Caso	Uso	Niveles	Crujías	Claros (m)	Ancho Viga (cm)	Alto Viga (cm)	Concreto (ton)	Acero (ton)	¿Cumple Uz?
1	Hospital	3	4	9	25	45	9.72	0.811	Correcto
2	Hospital	3	4	11	30	55	17.42	1.548	Correcto
3	Hospital	3	4	13	35	65	28.39	2.553	Correcto
4	Hospital	5	4	9	25	45	9.72	0.793	Correcto
5	Hospital	5	4	11	30	55	17.42	1.493	Correcto
6	Hospital	5	4	13	35	65	28.39	2.729	Correcto
7	Hospital	7	4	9	25	45	9.72	0.804	Correcto
8	Hospital	7	4	11	30	55	17.42	1.601	Correcto
9	Hospital	7	4	13	35	65	28.39	3.027	Correcto
10	Estacionamiento	3	4	9	25	45	9.72	0.811	Correcto
11	Estacionamiento	3	4	11	30	55	17.42	1.413	Correcto
12	Estacionamiento	3	4	13	35	65	28.39	2.463	Correcto
13	Estacionamiento	5	4	9	25	45	9.72	0.793	Correcto
14	Estacionamiento	5	4	11	30	55	17.42	1.493	Correcto
15	Estacionamiento	5	4	13	35	65	28.39	2.617	Correcto
16	Estacionamiento	7	4	9	25	45	9.72	0.793	Correcto
17	Estacionamiento	7	4	11	30	55	17.42	1.493	Correcto
18	Estacionamiento	7	4	13	35	65	28.39	2.691	Correcto

Tabla A 2. Resultados técnicos para análisis sin presfuerzo

Caso	Lecho superior								Lecho inferior								Estribos					U <sub>z</sub>
	M <sub>u</sub>	A <sub>s</sub>	Var	Φ	M <sub>r</sub>	FS	ρ	Tipo	M <sub>u</sub>	A <sub>s</sub>	Var	Φ	M <sub>r</sub>	FS	ρ	Tipo	V <sub>u</sub>	Φ	Sep	V <sub>r</sub>	FS	
1	-19.0	15.2	3	8	21.4	1.1	0.015	S	9.8	7.6	6	4	11.2	1.1	0.007	P	14.0	3	18	14.1	1.0	0.62
2	-37.1	22.8	8	6	39.9	1.1	0.015	P	19.0	11.4	4	6	21.0	1.1	0.008	S	23.1	3	12	24.8	1.1	0.8
3	-66.0	31.7	4	10	66.3	1.0	0.015	S	34.0	15.8	8	5	34.8	1.0	0.007	P	33.9	3	10	35.4	1.0	1.05
4	-19.6	15.2	3	8	21.4	1.1	0.015	S	9.3	6.3	5	4	9.4	1.0	0.006	S	15.2	3	16	15.4	1.0	0.57
5	-38.9	22.8	8	6	39.9	1.0	0.015	P	18.2	9.9	5	5	18.3	1.0	0.007	S	24.4	3	12	24.8	1.0	0.76
6	-69.1	34.2	3	12	71.1	1.0	0.016	S	32.2	15.2	3	8	33.5	1.0	0.007	S	36.3	4	16	38.4	1.1	0.96
7	-20.4	15.2	3	8	21.4	1.1	0.015	S	9.1	6.3	5	4	9.4	1.0	0.006	S	15.5	3	15	16.1	1.0	0.57
8	-40.7	23.8	3	10	41.4	1.0	0.016	S	17.9	9.9	5	5	18.3	1.0	0.007	S	25.0	3	10	28.6	1.2	0.74
9	-71.8	40.5	8	8	82.5	1.2	0.019	P	31.7	15.2	3	8	33.5	1.1	0.007	S	37.1	3	8	42.2	1.1	0.95
10	-18.5	15.2	3	8	21.4	1.2	0.015	S	9.6	7.6	6	4	11.2	1.2	0.007	P	13.6	3	18	14.1	1.0	0.57
11	-35.8	20.3	4	8	35.9	1.0	0.013	S	18.4	10.1	2	8	18.7	1.0	0.007	S	22.2	3	12	24.8	1.1	0.74
12	-63.4	30.4	6	8	63.9	1.0	0.014	P	32.7	15.2	3	8	33.5	1.0	0.007	S	32.5	3	10	35.4	1.1	0.98
13	-19.0	15.2	3	8	21.4	1.1	0.015	S	9.0	6.3	5	4	9.4	1.1	0.006	S	14.7	3	16	15.4	1.0	0.53
14	-37.5	22.8	8	6	39.9	1.1	0.015	P	17.6	9.9	5	5	18.3	1.0	0.007	S	23.5	3	12	24.8	1.1	0.7
15	-66.4	34.2	3	12	71.1	1.1	0.016	S	31.0	14.3	5	6	31.5	1.0	0.007	S	34.8	4	18	35.1	1.0	0.89
16	-19.8	15.2	3	8	21.4	1.1	0.015	S	8.9	6.3	5	4	9.4	1.1	0.006	S	15.0	3	16	15.4	1.0	0.52
17	-38.7	22.8	8	6	39.9	1.0	0.015	P	17.4	9.9	5	5	18.3	1.1	0.007	S	23.9	3	12	24.8	1.0	0.7
18	-69.0	34.2	3	12	71.1	1.0	0.016	S	30.5	14.3	5	6	31.5	1.0	0.007	S	35.6	4	16	38.4	1.1	0.88

M<sub>u</sub>: Momento último (ton-m)      A<sub>s</sub>: Área de acero en cm<sup>2</sup>

Var: Número de varillas

Φ: Diámetro en octavos de pulg.

M<sub>r</sub>: Momento resistente (ton-m)      FS: Factor de seguridad

ρ: Cuantía

U<sub>z</sub>: Desplazamiento vertical

Tipo: Acomodo de varillas (S es sencillo, P en paquete)

V<sub>u</sub>: Cortante último (ton)

V<sub>r</sub>: Cortante resistente (ton)

**Tabla A 3. Resultados principales para análisis presforzado con PSO.**

<b>Caso</b>	<b>Uso</b>	<b>Niveles</b>	<b>Crujías</b>	<b>Claro (m)</b>	<b>Iteración</b>	<b>Ancho de viga (cm)</b>	<b>Alto de viga (cm)</b>	<b>Torones</b>	<b>Concreto (ton)</b>	<b>Acero (ton)</b>	<b>Fitness</b>
1	Hospital	3	4	9	30	25	32	4	6.91	0.50	4.35
2	Hospital	3	4	11	30	25	42	5	11.09	0.68	6.93
3	Hospital	3	4	13	30	26	46	6	14.93	1.21	9.44
4	Hospital	5	4	9	20	26	30	4	6.74	0.52	4.25
5	Hospital	5	4	11	30	28	35	5	10.39	0.92	6.60
6	Hospital	5	4	13	29	28	56	7	19.57	1.34	12.28
7	Hospital	7	4	9	29	25	35	4	7.56	0.50	4.73
8	Hospital	7	4	11	30	25	56	6	14.78	0.84	9.21
9	Hospital	7	4	13	30	28	56	7	19.57	1.56	12.37
10	Estacionamiento	3	4	9	27	25	32	4	6.91	0.50	4.35
11	Estacionamiento	3	4	11	30	25	44	5	11.62	0.69	7.24
12	Estacionamiento	3	4	13	24	28	64	8	22.36	1.33	13.95
13	Estacionamiento	5	4	9	25	28	36	5	8.71	0.54	5.44
14	Estacionamiento	5	4	11	30	26	44	5	12.08	0.82	7.58
15	Estacionamiento	5	4	13	24	28	60	7	20.97	1.33	13.11
16	Estacionamiento	7	4	9	30	25	34	4	7.34	0.49	4.60
17	Estacionamiento	7	4	11	30	26	56	6	15.38	0.89	9.58
18	Estacionamiento	7	4	13	27	32	54	8	21.57	1.36	13.48

Tabla A 4. Resultados técnicos para análisis presforzado con PSO

Caso	Tor	Lecho Superior								Lecho Inferior								Estribos					Desp	Revisión ASD		
		M <sub>u</sub>	A <sub>s</sub>	Var	Φ	M <sub>r</sub>	FS	ρ	Tipo	M <sub>u</sub>	A <sub>s</sub>	Var	Φ	M <sub>r</sub>	FS	ρ	Tipo	V <sub>u</sub>	Φ	Sep	V <sub>r</sub>	FS	U <sub>z</sub>	ASD Inf	ASD Sup	
1	4	-11.6	3.8	3	4	16.0	1.4	0.005	S	5.2	3.8	3	4	16.0	3.1	0.005	S	10.6	3	14	14.2	1.3	0.68	V	V	V
2	5	-24.3	3.8	3	4	27.2	1.1	0.004	S	10.2	3.8	3	4	27.2	2.7	0.004	S	19.1	3	14	19.3	1.0	0.83	V	V	V
3	6	-38.6	7.9	4	5	39.3	1.0	0.007	P	14.0	5.1	4	4	36.2	2.6	0.005	S	26.8	3	10	27.0	1.0	1.29	V	V	V
4	4	-12.4	3.8	3	4	14.6	1.2	0.006	S	4.3	3.8	3	4	14.6	3.4	0.006	S	11.8	3	12	14.8	1.3	0.63	V	V	V
5	5	-23.0	7.9	4	5	24.6	1.1	0.009	P	7.0	3.8	3	4	21.4	3.1	0.004	S	18.5	3	10	20.4	1.1	0.93	V	F	V
6	7	-55.8	7.6	6	4	57.1	1.0	0.005	P	21.5	5.7	2	6	54.4	2.5	0.004	S	33.4	3	10	34.3	1.0	0.78	V	V	V
7	4	-15.3	3.8	3	4	18.3	1.2	0.005	S	5.8	3.8	3	4	18.3	3.2	0.005	S	13.4	3	15	15.1	1.1	0.49	V	V	V
8	6	-38.0	5.1	4	4	47.4	1.3	0.004	S	16.5	5.1	4	4	47.4	2.9	0.004	S	24.5	3	15	25.3	1.0	0.33	V	V	V
9	7	-58.4	10.1	2	8	60.5	1.0	0.007	S	21.2	5.7	2	6	54.4	2.6	0.004	S	34.3	3	8	40.1	1.2	0.75	V	V	V
10	4	-11.3	3.8	3	4	16.0	1.4	0.005	S	5.1	3.8	3	4	16.0	3.1	0.005	S	10.3	3	14	14.2	1.4	0.60	V	V	V
11	5	-24.6	4.0	2	5	29.3	1.2	0.004	S	11.0	4.0	2	5	29.3	2.7	0.004	S	18.9	3	15	19.5	1.0	0.68	V	V	V
12	8	-55.4	7.6	6	4	73.9	1.3	0.005	P	28.5	7.6	6	4	73.9	2.6	0.005	P	30.2	3	15	30.6	1.0	0.57	V	V	V
13	5	-14.7	3.8	3	4	22.4	1.5	0.004	S	6.6	3.8	3	4	22.4	3.4	0.004	S	12.7	3	15	16.3	1.3	0.33	V	V	V
14	5	-28.5	5.1	4	4	30.8	1.1	0.005	S	10.8	5.1	4	4	30.8	2.9	0.005	S	20.7	3	12	22.8	1.1	0.61	V	V	V
15	7	-56.7	6.3	5	4	60.8	1.1	0.004	S	23.7	6.3	5	4	60.8	2.6	0.004	S	33.1	3	10	36.9	1.1	0.59	V	V	V
16	4	-14.5	3.8	3	4	17.6	1.2	0.005	S	5.3	3.8	3	4	17.6	3.3	0.005	S	12.9	3	15	14.6	1.1	0.44	V	V	V
17	6	-35.9	5.7	2	6	48.8	1.4	0.004	S	16.6	5.7	2	6	48.8	2.9	0.004	S	23.3	3	15	25.7	1.1	0.30	V	V	V
18	8	-55.7	6.3	5	4	58.8	1.1	0.004	S	20.8	6.3	5	4	58.8	2.8	0.004	S	32.4	3	10	34.4	1.1	0.61	V	V	V

M<sub>u</sub>: Momento último (ton-m)      A<sub>s</sub>: Área de acero en cm<sup>2</sup>      Var: Número de varillas      Φ: Diámetro en octavos de pulg.

M<sub>r</sub>: Momento resistente (ton-m)      FS: Factor de seguridad      ρ: Cuantía      U<sub>z</sub>: Desplazamiento vertical

Tipo: Acomodo de varillas (S es sencillo, P en paquete)      V<sub>u</sub>: Cortante último (ton)      V<sub>r</sub>: Cortante resistente (ton)

Tor: Torones      ASD Inf/Sup: Revisión por ASD en fibra inferior y superior respectivamente (V cumple, F no cumple)

Tabla A 5. Análisis de esfuerzos permisibles para caso 5

Etapa	Fibra	Esfuerzo kg/cm <sup>2</sup>	EP compresión	EP tensión	Dif tension	Cumple	Penal	Signo	Ancho	Alto	Tor
Transferencia	Inf	-192.15	-192	16	0	FALSO	6.10E-05	Pos	28	35	5
Transferencia	Sup	46.72	-192	16	7526.4	VERDADERO	0	Pos	28	35	5
Servicio cargas sostenidas	Inf	-151.83	-180	32	0	VERDADERO	0	Pos	28	35	5
Servicio cargas sostenidas	Sup	21.7	-180	32	0	VERDADERO	0	Pos	28	35	5
Servicio cargas totales	Inf	-126.11	-240	32	0	VERDADERO	0	Pos	28	35	5
Servicio cargas totales	Sup	-4.01	-240	32	0	VERDADERO	0	Pos	28	35	5
Transferencia	Inf	-38.65	-192	16	0	VERDADERO	0	Neg	28	35	5
Transferencia	Sup	-106.79	-192	16	0	VERDADERO	0	Neg	28	35	5
Servicio cargas sostenidas	Inf	-100.05	-180	32	0	VERDADERO	0	Neg	28	35	5
Servicio cargas sostenidas	Sup	-30.08	-180	32	0	VERDADERO	0	Neg	28	35	5
Servicio cargas totales	Inf	-183.14	-240	32	0	VERDADERO	0	Neg	28	35	5
Servicio cargas totales	Sup	53.01	-240	32	5147.45	VERDADERO	0	Neg	28	35	5

**EP:** Esfuerzo permisible en kg/cm<sup>2</sup>

**Dif tensión:** Tensión absorbida por acero pasivo para sección en transición

**Penal:** Penalización

**Signo:** Pos para análisis en centro de claro, Neg para análisis en apoyos

**Inf/Sup:** Fibras inferior y superior respectivamente

**Tor:** Cantidad de torones

## ANEXO B.

### CODIGO FUENTE EN PYTHON

#### Código para generación de casos de estudio

```
def CasosDeEstudio():  
  
    Uso=['Hospital','Estacionamiento']  
    Niveles=[3,5,7]  
    Crujias=[4]  
    Claros=[900,1100,1300]  
  
    Casos_De_Estudio=[]  
  
    for i in range(len(Uso)):  
        for j in range(len(Niveles)):  
            for k in range(len(Crujias)):  
                for l in range(len(Claros)):  
                    Caso=[Uso[i],Niveles[j],Crujias[k],Claros[l]]  
                    Casos_De_Estudio.append(Caso)  
  
    return(Casos_De_Estudio)
```

## Código para análisis estructural sin presfuerzo

```
def AnalisisSinPresfuerzo(Usos,Niveles,Crujias,Claros):

    # CONEXION A SAP 2000

    import os, sys, math, comtypes.client
    import pandas as pd
    import numpy as np
    from Analisis_Estructural.ResultadosAnalisis import AnalisisRapido as AnalisisRapido

    AttachToInstance = False
    APIPath=r'C:\Users\jmhdz\Documents\Maestría\Tesis\Experimentacion\Analisis Sin
Presfuerzo'
    os.makedirs(APIPath+os.sep+f'Analisis {Usos}_{Niveles}N_{Crujias}CR_{Claros/100} M')
    ModelPath = APIPath+os.sep+f'Analisis {Usos}_{Niveles}N_{Crujias}CR_{Claros/100}
M'+os.sep+f'{Usos}_{Niveles}N_{Crujias}CR_{Claros} M.sdb'

    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)

    if AttachToInstance:
        mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")
    else:
        mySapObject = helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")
        mySapObject.ApplicationStart()

    SapModel = mySapObject.SapModel
    SapModel.InitializeNewModel()
    mySapObject.Hide()
    SapModel.File.NewBlank()
    SapModel.SetPresentUnits(14)
    AnalisisRapido()
    '''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
    RutaEspectro=r'C:\Users\jmhdz\Documents\Maestría\Tesis\Experimentacion\Archivos para
apoyo de codigo\Espectro Modificado Qro.txt'

    '''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/'''
    # DEFINICION DE CARGAS Y FACTORES DE CARGA

    Carga={ 'Viva':{      'Oficinas':      {'W':100E-4, 'Wa':180E-4, 'Wm':250E-4},
                        'Hospital':      {'W':80E-4, 'Wa':100E-4, 'Wm':190E-4},
                        'Estacionamiento': {'W':40E-4, 'Wa':100E-4, 'Wm':250E-4},
                        'Azotea':        {'W':15E-4, 'Wa':70E-4, 'Wm':100E-4}},

            'Muerta':{    'Entrepiso':120E-4,
                        'Azotea': 220E-4}}

    CV_Media_Ent= Carga['Viva'][Usos]['W']
    CV_Inst_Ent= Carga['Viva'][Usos]['Wa']
    CV_Max_Ent= Carga['Viva'][Usos]['Wm']
    CV_Media_Az= Carga['Viva']['Azotea']['W']
    CV_Inst_Az= Carga['Viva']['Azotea']['Wa']
    CV_Max_Az= Carga['Viva']['Azotea']['Wm']
    Cm_Entrepiso= Carga['Muerta']['Entrepiso']
    Cm_Azotea= Carga['Muerta']['Azotea']
```

```

PatronesCargaViva=[CV_Media_Ent, CV_Inst_Ent, CV_Max_Ent, CV_Media_Az, CV_Inst_Az, CV_Max_Az]
NomPatronesCargaViva=['W', 'Wa', 'Wm', 'W', 'Wa', 'Wm']
PatronesCargaMuerta=[Cm_Entrepiso, Cm_Azotea]

if Uso=='Hospital':
    FC_Muerta=1.5
    FC_Viva=1.7
else:
    FC_Muerta=1.3
    FC_Viva=1.5

'''
# DEFINICION DE SECCIONES

# SECCION VIGAS / PREDIMENSIONAMIENTO ACI 318/25 - NTC 2023

AltoViga=math.ceil((Claros/21)/5)*5
AnchoViga=math.ceil((AltoViga/2)/5)*5

Viga='V-' +str(AnchoViga)+'x'+str(AltoViga)
SapModel.PropFrame.SetRectangle('Viga', '4000psi', AltoViga, AnchoViga)

# SECCION LOSA NERVADA / PREDIMENSIONAMIENTO ACI 318/25 - NTC 2023

# PeralteNerv=math.ceil((Claros*(0.8+(4200/14000))/(36+9*.6))/5)*5
PeralteNerv=math.floor((Claros/33)/5)*5
AnchoNerv=15
SepNerv=75
def esp_equiv_losa_nerv():

    b=AnchoNerv/2
    s=SepNerv-AnchoNerv
    h=PeralteNerv
    c=5
    A1,A2,A3=b*h, s*c, b*h
    Y1,Y2,Y3=h/2, h-c/2, h/2
    Yc=np.dot([A1, A2, A3], [Y1, Y2, Y3])/np.sum([A1, A2, A3])
    Ix1, Ix2, Ix3=(b*h**3)/12+A1*(Yc-Y1)**2, (s*c**3)/12+A2*(Y2-
Yc)**2, (b*h**3)/12+A3*(Yc-Y3)**2
    Ix=np.sum([Ix1, Ix2, Ix3])
    HEq=(12*Ix/SepNerv)**(1/3)
    AreaNerv=np.sum([A1, A2, A3])
    AreaEspEq=HEq*SepNerv
    FactorPesoVol=AreaNerv/AreaEspEq
    PesoVolEq=FactorPesoVol*2.4E-3
    PesoLosaEq=(AreaEspEq*PesoVolEq/SepNerv)

    return(HEq, PesoVolEq, PesoLosaEq)

EspEqReticular=esp_equiv_losa_nerv()[0]
PesoLosa=esp_equiv_losa_nerv()[2]
PesoVolReticular=esp_equiv_losa_nerv()[1]

Losa='Losa reticular h= '+str(PeralteNerv)+' cm'
SapModel.PropArea.SetShell_1(Losa, 1, False, '4000psi', 0, EspEqReticular, EspEqReticular, -
1, "Reticular", 'Default')

```



```

SapModel.SelectObj.ClearSelection()

# ASIGNACION MATERIAL A LOSAS
SapModel.SelectObj.PropertyArea(Losa,False)
SapModel.AreaObj.SetMaterialOverwrite('','CONC 400 LOSA RETICULAR',2)
SapModel.SelectObj.ClearSelection()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE GRUPO PARA VIGAS Y NODOS DE ESTUDIO

SapModel.GroupDef.SetGroup('GrupoVigas',-1,True)
SapModel.SelectObj.CoordinateRange(-LongTotX/2,LongTotX/2,-
5,5,5,AlturaEnt,False,'GLOBAL',False,False,True,False,False,False)
SapModel.FrameObj.SetGroupAssign('','GrupoVigas',False,2)
SapModel.SelectObj.ClearSelection()

# DISCRETIZACION DE VIGAS DE ESTUDIO
Columnas=Niveles*(Crujias+1)**2
VigasAnt=Niveles*Crujias**2/2
GrupoVigas=[int(Columnas+VigasAnt+1)]

for VigaGrupo in range(Crujias-1):
    VigaNueva=int(GrupoVigas[VigaGrupo]+Niveles)
    GrupoVigas.append(VigaNueva)
GrupoVigasString=[str(elemento) for elemento in GrupoVigas]

for viga in GrupoVigasString:
    SapModel.EditFrame.DivideByRatio(viga,10,1,[])

SapModel.FrameObj.SetOutputStations('GrupoVigas',2,0,1,True,True,1)

# DEFINICION DE NODOS DE ESTUDIO
SapModel.GroupDef.SetGroup('GrupoNodos',-1,True)
SapModel.SelectObj.CoordinateRange(-LongTotX/2,LongTotX/2,-
5,5,5,AlturaEnt,False,'GLOBAL',False,True,False,False,False,False)
SapModel.PointObj.SetGroupAssign('','GrupoNodos',False,2)
SapModel.SelectObj.ClearSelection()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE PATRONES DE CARGA

SapModel.LoadPatterns.ChangeName('DEAD','Pp')
SapModel.LoadPatterns.Add('Cm',2)
SapModel.LoadPatterns.Add('W',3)
SapModel.LoadPatterns.Add('Wa',3)
SapModel.LoadPatterns.Add('Wm',3)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE FUENTES DE MASA

FuenteDeMasa=['Pp','Cm','Wa']
SapModel.SourceMass.ChangeName('MSSSRC1','Fuente De Masa')
SapModel.SourceMass.SetMassSource('Fuente De
Masa',False,False,True,True,3,FuenteDeMasa,[1,1,1])

```



```

Diafragmas=[]
for nivel in range(Niveles):
    SapModel.ConstraintDef.SetDiaphragm(f'Diafragma N{nivel+1}',3,'GLOBAL')
    Diafragmas.append(f'Diafragma N{nivel+1}')
    nodo=nivel+2
    SapModel.SelectObj.PlaneXY(str(nodo))
    SapModel.PointObj.SetConstraint('',Diafragmas[nivel],2,True)
    SapModel.SelectObj.ClearSelection()

SapModel.GroupDef.SetGroup('Losas Entrepiso')
SapModel.GroupDef.SetGroup('Losas Azotea')

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# APLICACION DE CARGAS

SapModel.SelectObj.CoordinateRange(-LongTotX,LongTotX,-LongTotY,LongTotY,0,(Niveles-
1)*AlturaEnt,False,'GLOBAL',True,False,False,True,False,False)
SapModel.AreaObj.SetGroupAssign('','Losas Entrepiso',False,2)
SapModel.SelectObj.CoordinateRange(-LongTotX,LongTotX,-
LongTotY,LongTotY,0,Niveles*AlturaEnt,False,'GLOBAL',True,False,False,True,False,False)
SapModel.SelectObj.Group('Losas Entrepiso',True)
SapModel.AreaObj.SetGroupAssign('','Losas Azotea',False,2)
SapModel.SelectObj.ClearSelection()

NomPatronesCargaViva=['W','Wa','Wm','W','Wa','Wm']

for carga_entrepiso in range(3):
    SapModel.AreaObj.SetLoadUniform('Losas
Entrepiso',NomPatronesCargaViva[carga_entrepiso],-
1*PatronesCargaViva[carga_entrepiso],6,False,'GLOBAL',1)

    SapModel.AreaObj.SetLoadUniform('Losas Entrepiso','Cm',-
1*PatronesCargaMuerta[0],6,False,'GLOBAL',1)

    for carga_azotea in range(3,6):
        SapModel.AreaObj.SetLoadUniform('Losas
Azotea',NomPatronesCargaViva[carga_azotea],-
1*PatronesCargaViva[carga_azotea],6,False,'GLOBAL',1)

        SapModel.AreaObj.SetLoadUniform('Losas Azotea','Cm',-
1*PatronesCargaMuerta[1],6,False,'GLOBAL',1)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# MESH PARA LOSAS

SapModel.AreaObj.SetAutoMesh('ALL', 1, 10, 10, False, False, False, False, 0,
0, False, False, False, "ALL", False, 0, 1)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# CORRER ANALISIS

SapModel.File.Save(ModelPath)
SapModel.Analyze.RunAnalysis()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''

```

```

# EXTRACCION DE RESULTADOS DE ESFUERZOS CON ENVOLVENTE
def Resultados(Combo):
    SapModel.SetPresentUnits(12)
    NumberResults = 0
    Obj, Elm, LoadCase, StepType, StepNum, ObjSta, ElmSta, P, V2, V3, T, M2, M3 = [],
[], [], [], [], [], [], [], [], [], [], []

    SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
    SapModel.Results.Setup.SetComboSelectedForOutput(Combo)

    [NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType, StepNum, P, V2, V3,
T, M2, M3, ret] = SapModel.Results.FrameForce(
    'GrupoVigas', 2, NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase,
StepType, StepNum, P, V2, V3, T, M2, M3)

    TablaEsfuerzos =
pd.DataFrame({'CasoCarga':LoadCase, 'Viga':Obj, 'TipoValor':StepType, 'Estacion':ObjSta, 'M3':
:M3, 'V2':V2})
    return TablaEsfuerzos

def _add_dist_sin_warning(df):
    partes = []
    for _, g in df.groupby('CasoCarga', sort=False):
        orden = sorted(g['Viga'].unique())
        offset = 0.0
        offsets = {}
        for v in orden:
            offsets[v] = offset
            L = g.loc[g['Viga'] == v, 'Estacion'].max()
            offset += float(L)
        gg = g.copy()
        gg['Distancia'] = gg['Estacion'] + gg['Viga'].map(offsets)
        partes.append(gg)
    return pd.concat(partes, ignore_index=True)

TablaEsfuerzos = Resultados('ENVOLVENTE')

# ORDENAMIENTO DE TABLA DE ESFUERZOS
TablaEsfuerzos['Viga'] = pd.to_numeric(TablaEsfuerzos['Viga'], errors='coerce')
TablaEsfuerzos['Estacion'] = pd.to_numeric(TablaEsfuerzos['Estacion'],
errors='coerce').round(6)
TablaEsfuerzos['M3'] = pd.to_numeric(TablaEsfuerzos['M3'], errors='coerce')
TablaEsfuerzos['V2'] = pd.to_numeric(TablaEsfuerzos['V2'], errors='coerce')

gcols = ['CasoCarga', 'Viga', 'Estacion']
ext = (TablaEsfuerzos.groupby(gcols)
    .agg(Mu_maximo=('M3', 'max'), Mu_minimo=('M3', 'min'),
        Vu_maximo=('V2', 'max'), Vu_minimo=('V2', 'min'))
    .reset_index())

ext['Vu'] = np.maximum(ext['Vu_maximo'].abs(), ext['Vu_minimo'].abs())
ext = _add_dist_sin_warning(ext)

TablaEsf =
(ext[['CasoCarga', 'Viga', 'Distancia', 'Mu_maximo', 'Mu_minimo', 'Vu_maximo', 'Vu_minimo', 'Vu'
]])
    .sort_values(['CasoCarga', 'Distancia', 'Viga'])
    .reset_index(drop=True))

```

```

t = TablaEsf.copy()
t['Distancia'] = t['Distancia'].round(6)
agg_cols = {'Mu_maximo':'max', 'Mu_minimo':'min', 'Vu_maximo':'max', 'Vu_minimo':'min'}

TablaResultados = (t.groupby(['CasoCarga', 'Distancia'],
as_index=False).agg(agg_cols))
TablaResultados['Vu'] = TablaResultados[['Vu_maximo', 'Vu_minimo']].abs().max(axis=1)
TablaResultados =
TablaResultados.sort_values(['CasoCarga', 'Distancia']).reset_index(drop=True)

# OBTENCION DE ESFUERZOS DE DISEÑO
MuPos = round(float(TablaResultados['Mu_maximo'].max()),2)
MuNeg = round(float(TablaResultados['Mu_minimo'].min()),2)
Vu = round(float(TablaResultados['Vu'].max()).round(2)),2)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# EXTRACCION DE RESULTADOS DE DESPLAZAMIENTOS

SapModel.SetPresentUnits(14)
SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
SapModel.Results.Setup.SetComboSelectedForOutput('ELS')
NumberResults = 0
Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3 = [], [], [], [], [],
[], [], [], [], [], []
(NumberResults, Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3, ret) =
SapModel.Results.JointDispl('GrupoNodos', 2, NumberResults, Obj, Elm, LoadCase, StepType,
StepNum, U1, U2, U3, R1, R2, R3)

# ORDENAMIENTO DE TABLA DE DESPLAZAMIENTOS
df=pd.DataFrame({'Nodo': Obj, 'U3': U3})
df['Nodo'] = df['Nodo'].astype(str)
df = df.drop_duplicates(subset=['Nodo'], keep='first').reset_index(drop=True)

xs = []
for n in df['Nodo']:
    x, y, z, _ = SapModel.PointObj.GetCoordCartesian(n)
    xs.append(float(x))

df['X'] = xs
df = df.sort_values('X', kind='mergesort').reset_index(drop=True)
x0 = df['X'].min()
df['Distancia'] = (df['X'] - x0).round(6)
TablaDesplazamientos = df[['Nodo', 'Distancia', 'U3']]

# ADICION A TABLA PRINCIPAL
TablaResultados['Uz']=TablaDesplazamientos['U3'].values

# OBTENCION DE DESPLAZAMIENTO MAXIMO
DespMax=round(float((TablaDesplazamientos['U3'].min()*-1)),2)

mySapObject.ApplicationExit(False)
SapModel = None
mySapObject = None

# BORRADO DE ARCHIVOS PARA EVITAR LLENAR MEMORIA
import shutil, time
carpeta = os.path.dirname(ModelPath)

```

```
for _ in range(5):
    try:
        if os.path.isdir(carpeta):
            shutil.rmtree(carpeta, ignore_errors=False)
            break
    except PermissionError:
        time.sleep(0.3)
    except FileNotFoundError:
        break

return(AnchoViga, AltoViga, TablaResultados, MuPos, MuNeg, Vu, DespMax)

# Devuelve Ancho de viga (cm), Alto de viga (cm), Tabla de esfuerzos y
desplazamientos(ton y m), Momento ultimo Positivo (t-m), Momento ultimo Negativo (t-m)
# Cortante ultimo (t), Desplazamiento maximo (cm)
```



```

PatronesCargaViva=[CV_Media_Ent,CV_Inst_Ent,CV_Max_Ent,CV_Media_Az,CV_Inst_Az,CV_Max_
Az]
NomPatronesCargaViva=['W', 'Wa', 'Wm', 'W', 'Wa', 'Wm']
PatronesCargaMuerta=[Cm_Entrepiso,Cm_Azotea]

if Uso=='Hospital':
    FC_Muerta=1.5
    FC_Viva=1.7
else:
    FC_Muerta=1.3
    FC_Viva=1.5

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE SECCIONES

# SECCION VIGAS

Viga='V- '+str(AnchoViga)+'x'+str(AltoViga)
SapModel.PropFrame.SetRectangle('Viga','4000psi',AltoViga,AnchoViga)

# SECCION LOSA NERVADA / PREDIMENSIONAMIENTO - NTC 2023

PeralteNerv=math.floor((Claros/33)/5)*5
AnchoNerv=15
SepNerv=75
def esp_equiv_losa_nerv():

    b=AnchoNerv/2
    s=SepNerv-AnchoNerv
    h=PeralteNerv
    c=5
    A1,A2,A3=b*h,s*c,b*h
    Y1,Y2,Y3=h/2,h-c/2,h/2
    Yc=np.dot([A1,A2,A3],[Y1,Y2,Y3])/np.sum([A1,A2,A3])
    Ix1,Ix2,Ix3=(b*h**3)/12+A1*(Yc-Y1)**2,(s*c**3)/12+A2*(Y2-
Yc)**2,(b*h**3)/12+A3*(Yc-Y3)**2
    Ix=np.sum([Ix1,Ix2,Ix3])
    HEq=(12*Ix/SepNerv)**(1/3)
    AreaNerv=np.sum([A1,A2,A3])
    AreaEspEq=HEq*SepNerv
    FactorPesoVol=AreaNerv/AreaEspEq
    PesoVolEq=FactorPesoVol*2.4E-3
    PesoLosaEq=(AreaEspEq*PesoVolEq/SepNerv)

    return(HEq,PesoVolEq,PesoLosaEq)

EspEqReticular=esp_equiv_losa_nerv()[0]
PesoLosa=esp_equiv_losa_nerv()[2]
PesoVolReticular=esp_equiv_losa_nerv()[1]

Losa='Losa reticular h= '+str(PeralteNerv)+' cm'
SapModel.PropArea.SetShell_1(Losa,1,False,'4000psi',0,EspEqReticular,EspEqReticular,-
1,"Reticular",'Default')
RigFl=0.8
mods=[1,1,1,RigFl,RigFl,RigFl,1,1,1,1]
SapModel.PropArea.SetModifiers(Losa,mods)

```

```

# SECCION COLUMNAS

AreaTribCol=Niveles*Claros*Claros/2
PuCol=AreaTribCol*(FC_Muerta*(PesoLosa+0.012)+FC_Viva*(CV_Max_Ent))
AnchoColumna=math.ceil(np.sqrt(PuCol/(0.3*400)))/5)*5+2
AltoColumna=AnchoColumna
Columna='C-'+str(AnchoColumna)+'x'+str(AltoColumna)
SapModel.PropFrame.SetRectangle(Columna,'400psi',AnchoColumna,AltoColumna)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# GENERACIÓN DE MODELO DE ANALISIS

AlturaEnt=400
DimX=Claros
DimY=Claros*2/3
LongTotX=DimX*Crujias
LongTotY=DimY*Crujias

SapModel.File.New3DFrame(2,Niveles,AlturaEnt,Crujias,DimX,Crujias,DimY,True,'Viga',Co
lumna,Losa,1,1)
SapModel.SelectObj.SupportedPoints([True,True,True,False,False,False])
SapModel.PointObj.SetRestraint('',[True,True,True,True,True,True],2)
SapModel.SelectObj.ClearSelection()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION MATERIALES Y ASIGNACION A ELEMENTOS

# MATERIAL VIGAS Y COLUMNAS
SapModel.PropMaterial.AddMaterial('',2,'United States','Customary','f'c 5000
psi','CONC 400 VIGAS Y COLUMNAS')
SapModel.PropMaterial.SetMPIsotropic('CONC 400 VIGAS Y COLUMNAS',280000,0.2,9.900E-
06)
SapModel.PropMaterial.SetWeightAndMass('CONC 400 VIGAS Y COLUMNAS',1,2.4E-3)

# MATERIAL LOSAS
SapModel.PropMaterial.AddMaterial('',2,'United States','Customary','f'c 5000
psi','CONC 400 LOSA RETICULAR')
SapModel.PropMaterial.SetMPIsotropic('CONC 400 LOSA RETICULAR',280000,0.2,9.900E-06)
SapModel.PropMaterial.SetWeightAndMass('CONC 400 LOSA RETICULAR',1,PesoVolReticular)

# ASIGNACION MATERIAL A VIGAS
SapModel.SelectObj.PropertyFrame('Viga',False)
SapModel.FrameObj.SetMaterialOverwrite('','CONC 400 VIGAS Y COLUMNAS',2)
SapModel.SelectObj.ClearSelection()
SapModel.PropMaterial.AddMaterial('Acero G60',6,'United States','ASTM A615','Grade
60','Acero G60')
SapModel.PropMaterial.SetORebar_1('Acero G60',4200,6300,4200,6300,1,1,0.01,0.09,-
0.1,False)
SapModel.PropFrame.SetRebarBeam('Viga','Acero G60','Acero G60',5,5,0,0,0)

# ASIGNACION MATERIAL A COLUMNAS
SapModel.SelectObj.PropertyFrame(Columna,False)
SapModel.FrameObj.SetMaterialOverwrite('','CONC 400 VIGAS Y COLUMNAS',2)
SapModel.SelectObj.ClearSelection()

# ASIGNACION MATERIAL A LOSAS
SapModel.SelectObj.PropertyArea(Losa,False)

```

```

SapModel.AreaObj.SetMaterialOverwrite('', 'CONC 400 LOSA RETICULAR', 2)
SapModel.SelectObj.ClearSelection()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE GRUPOS PARA VIGAS DE ESTUDIO

SapModel.GroupDef.SetGroup('GrupoVigas', -1, True)
SapModel.SelectObj.CoordinateRange(-LongTotX/2, LongTotX/2, -
5, 5, 5, AlturaEnt, False, 'GLOBAL', False, False, True, False, False, False)
SapModel.FrameObj.SetGroupAssign('', 'GrupoVigas', False, 2)
SapModel.SelectObj.ClearSelection()

# DISCRETIZACION DE VIGAS DE ESTUDIO
Columnas=Niveles*(Crujias+1)**2
VigasAnt=Niveles*Crujias**2/2
GrupoVigas=[int(Columnas+VigasAnt+1)]

for VigaGrupo in range(Crujias-1):
    VigaNueva=int(GrupoVigas[VigaGrupo]+Niveles)
    GrupoVigas.append(VigaNueva)
GrupoVigasString=[str(elemento) for elemento in GrupoVigas]

for viga in GrupoVigasString:
    SapModel.EditFrame.DivideByRatio(viga, 10, 1, [])

SapModel.FrameObj.SetOutputStations('GrupoVigas', 2, 0, 1, True, True, 1)

# DEFINICION DE NODOS DE ESTUDIO
SapModel.GroupDef.SetGroup('GrupoNodos', -1, True)
SapModel.SelectObj.CoordinateRange(-LongTotX/2, LongTotX/2, -
5, 5, 5, AlturaEnt, False, 'GLOBAL', False, True, False, False, False)
SapModel.PointObj.SetGroupAssign('', 'GrupoNodos', False, 2)
SapModel.SelectObj.ClearSelection()

# DEFINICION DE GRUPO PARA TODAS LAS VIGAS
SapModel.GroupDef.SetGroup('GrupoVigasGlobal', -1, True)
SapModel.SelectObj.PropertyFrame('Viga')
SapModel.FrameObj.SetGroupAssign('', 'GrupoVigasGlobal', False, 2)
SapModel.SelectObj.ClearSelection()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE PATRONES DE CARGA

SapModel.LoadPatterns.ChangeName('DEAD', 'Pp')
SapModel.LoadPatterns.Add('Cm', 2)
SapModel.LoadPatterns.Add('W', 3)
SapModel.LoadPatterns.Add('Wa', 3)
SapModel.LoadPatterns.Add('Wm', 3)
SapModel.LoadPatterns.Add('Presfuerzo', 34)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE FUENTES DE MASA

FuenteDeMasa=['Pp', 'Cm', 'Wa']
SapModel.SourceMass.ChangeName('MSSSRC1', 'Fuente De Masa')

```

```

SapModel.SourceMass.SetMassSource('Fuente De
Masa', False, False, True, True, 3, FuenteDeMasa, [1, 1, 1])

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE CASOS DE CARGA

SapModel.LoadCases.ChangeName('DEAD', 'Pp')
SapModel.LoadCases.ChangeName('MODAL', 'Modal')
SapModel.LoadCases.ModalEigen.SetNumberModes('Modal', Niveles*3, Niveles)

SapModel.Func.Delete('UNIFRS')
SapModel.Func.FuncRS.SetFromFile('EspectroElastico', RutaEspectro, 0, 0.05, 2)

def EspectroElastico(Caso, Eje, Espectro, Angulo):
    SapModel.LoadCases.ResponseSpectrum.SetCase(Caso)
    SapModel.LoadCases.ResponseSpectrum.SetLoads(Caso, 1, [Eje], [Espectro], [981], ['Global'], [Angulo])

EspectroElastico('Sismo +X', 'U1', 'EspectroElastico', 0)
EspectroElastico('Sismo -X', 'U1', 'EspectroElastico', 180)
EspectroElastico('Sismo +Y', 'U2', 'EspectroElastico', 0)
EspectroElastico('Sismo -Y', 'U2', 'EspectroElastico', 180)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE COMBINACIONES DE CARGA

FactoresDeCarga={
    'ELS':{'Pp':1, 'Cm':1, 'W':1, 'Presfuerzo':1},
    'ELU':{'Pp':FC_Muerta, 'Cm':FC_Muerta, 'Wm':FC_Viva},
    'ACC +SX':{'Pp':1.1, 'Cm':1.1, 'Wa':1.1, 'Sismo +X':1.1},
    'ACC -SX':{'Pp':1.1, 'Cm':1.1, 'Wa':1.1, 'Sismo -X':1.1},
    'ACC +SY':{'Pp':1.1, 'Cm':1.1, 'Wa':1.1, 'Sismo +Y':1.1},
    'ACC -SY':{'Pp':1.1, 'Cm':1.1, 'Wa':1.1, 'Sismo -Y':1.1},
    'ASD Cargas sostenidas':{'Pp':1, 'Cm':1},
    'ASD Cargas totales':{'Pp':1, 'Cm':1, 'Wm':1}}

def CombinacionCargaLineal(Combo):
    SapModel.RespCombo.Add(Combo, 0)
    for factor in FactoresDeCarga[Combo]:
        SapModel.RespCombo.SetCaseList(Combo, 0, factor, FactoresDeCarga[Combo][
factor])

CombinacionCargaLineal('ELS')
CombinacionCargaLineal('ELU')
CombinacionCargaLineal('ACC +SX')
CombinacionCargaLineal('ACC -SX')
CombinacionCargaLineal('ACC +SY')
CombinacionCargaLineal('ACC -SY')
CombinacionCargaLineal('ASD Cargas sostenidas')
CombinacionCargaLineal('ASD Cargas totales')

Combinaciones=list(SapModel.RespCombo.GetNameList()[1])
SapModel.RespCombo.Add('ENVOLVENTE', 1)
for combo in Combinaciones[1:6]:
    SapModel.RespCombo.SetCaseList('ENVOLVENTE', 1, combo, 1)

```

```

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DEFINICION DE DIAFRAGMAS

Diafragmas=[]
for nivel in range(Niveles):
    SapModel.ConstraintDef.SetDiaphragm(f'Diafragma N{nivel+1}',3,'GLOBAL')
    Diafragmas.append(f'Diafragma N{nivel+1}')
    nodo=nivel+2
    SapModel.SelectObj.PlaneXY(str(nodo))
    SapModel.PointObj.SetConstraint('',Diafragmas[nivel],2,True)
    SapModel.SelectObj.ClearSelection()

SapModel.GroupDef.SetGroup('Losas Entrepiso')
SapModel.GroupDef.SetGroup('Losas Azotea')

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# APLICACION DE CARGAS

SapModel.SelectObj.CoordinateRange(-LongTotX,LongTotX,-LongTotY,LongTotY,0,(Niveles-
1)*AlturaEnt,False,'GLOBAL',True,False,False,True,False,False)
SapModel.AreaObj.SetGroupAssign('','Losas Entrepiso',False,2)
SapModel.SelectObj.CoordinateRange(-LongTotX,LongTotX,-
LongTotY,LongTotY,0,Niveles*AlturaEnt,False,'GLOBAL',True,False,False,True,False,False)
SapModel.SelectObj.Group('Losas Entrepiso',True)
SapModel.AreaObj.SetGroupAssign('','Losas Azotea',False,2)
SapModel.SelectObj.ClearSelection()

for carga_entrepiso in range(3):
    SapModel.AreaObj.SetLoadUniform('Losas
Entrepiso',NomPatronesCargaViva[carga_entrepiso],-
1*PatronesCargaViva[carga_entrepiso],6,False,'GLOBAL',1)

SapModel.AreaObj.SetLoadUniform('Losas Entrepiso','Cm',-
1*PatronesCargaMuerta[0],6,False,'GLOBAL',1)

NomPatronesCargaViva=['W','Wa','Wm','W','Wa','Wm']
for carga_azotea in range(3,6):
    SapModel.AreaObj.SetLoadUniform('Losas
Azotea',NomPatronesCargaViva[carga_azotea],-
1*PatronesCargaViva[carga_azotea],6,False,'GLOBAL',1)

SapModel.AreaObj.SetLoadUniform('Losas Azotea','Cm',-
1*PatronesCargaMuerta[1],6,False,'GLOBAL',1)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/'''
# APLICACION PRESFUERZO

H_Toron=((AltoViga/2)-5)/2.54

if Crujias==4 and Claros==900:    hoja=0
if Crujias==4 and Claros==1100:   hoja=1
if Crujias==4 and Claros==1300:   hoja=2
if Crujias==6 and Claros==900:    hoja=3
if Crujias==6 and Claros==1100:   hoja=4
if Crujias==6 and Claros==1300:   hoja=5
if Crujias==8 and Claros==900:    hoja=6

```

```

if Crujias==8 and Claros==1100:    hoja=7
if Crujias==8 and Claros==1300:    hoja=8

df=pd.read_excel(RutaCoordenadasTorones, sheet_name=hoja, header=None)

Coord_X=df[0].tolist()
Coord_Y=df[1].tolist()
Coord_X=[cx/2.54 for cx in Coord_X]
Coord_Y=[cy*H_Toron for cy in Coord_Y]
Coord_Z=[0 for cz in range(len(Coord_X))]

MyType=[0 for mt in range(len(Coord_X))]
MyType[0]=1
i=1
while i<len(Coord_X):
    if i%2==0:
        MyType[i]=6
    else:
        MyType[i]=7
    i+=1
NumPuntos=len(Coord_X)

# COORDENADAS DE TORON
Coord_X_Exc=[ccx*2.54 for ccx in Coord_X]
Coord_Y_Exc=[ccy*2.54 for ccy in Coord_Y]

Coord_X_Estaciones=[i*(Claros/10) for i in range(int(Crujias*10)+1)]

x = np.asarray(Coord_X_Exc, dtype=float)
y = np.asarray(Coord_Y_Exc, dtype=float)
xq = np.asarray(Coord_X_Estaciones, dtype=float)
yq = np.interp(xq, x, y)
Coord_Y_Estaciones = yq.tolist()

Exc_Torones=pd.DataFrame({'X':Coord_X_Estaciones, 'Y':Coord_Y_Estaciones})

x = Exc_Torones['X'].to_numpy(dtype=float)
y = Exc_Torones['Y'].to_numpy(dtype=float)
n = len(Exc_Torones)

SumaLong=[]
for indice in range(1,n):
    SumaLong.append(np.sqrt((y[indice]-y[indice-1])**2+(x[indice]-x[indice-1])**2))

LongToron=round(float(sum(SumaLong)),2)

dx = np.empty(n, dtype=float)
dy = np.empty(n, dtype=float)

dx[1:-1] = x[2:] - x[:-2]
dy[1:-1] = y[2:] - y[:-2]
dx[0] = x[1] - x[0]
dy[0] = y[1] - y[0]
dx[-1] = x[-1] - x[-2]
dy[-1] = y[-1] - y[-2]

theta_rad = np.arctan2(dy, dx)
theta_deg = np.degrees(theta_rad)
Exc_Torones['Theta_Toron'] = theta_deg

```



## Código para extracción de resultados de cada análisis en PSO

```
def ResultadosAnalisis(Exc_Torones, LongToron, AltoViga, RutaModelo):

    import pandas as pd; import numpy as np
    import comtypes.client
    # CONEXION A SAP 2000
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
    mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")
    SapModel = mySapObject.SapModel
    # CORRER ANALISIS
    SapModel.File.Save(RutaModelo)
    SapModel.Analyze.RunAnalysis()

    # EXTRACCION DE RESULTADOS DE ESFUERZOS CON ENVOLVENTE
    def Resultados(Combo):
        SapModel.SetPresentUnits(12)
        NumberResults = 0
        Obj, Elm, LoadCase, StepType, StepNum, ObjSta, ElmSta, P, V2, V3, T, M2, M3 = [],
        [], [], [], [], [], [], [], [], [], [], []

        SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
        SapModel.Results.Setup.SetComboSelectedForOutput(Combo)

        [NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType, StepNum, P, V2, V3,
        T, M2, M3, ret] = SapModel.Results.FrameForce(
            'GrupoVigas', 2, NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase,
            StepType, StepNum, P, V2, V3, T, M2, M3)

        TablaEsfuerzos =
        pd.DataFrame({'CasoCarga':LoadCase, 'Viga':Obj, 'TipoValor':StepType, 'Estacion':ObjSta, 'M3':
        :M3, 'V2':V2})
        return TablaEsfuerzos

    def _add_dist_sin_warning(df):
        partes = []
        for _, g in df.groupby('CasoCarga', sort=False):
            orden = sorted(g['Viga'].unique())
            offset = 0.0
            offsets = {}
            for v in orden:
                offsets[v] = offset
                L = g.loc[g['Viga'] == v, 'Estacion'].max()
                offset += float(L)
            gg = g.copy()
            gg['Distancia'] = gg['Estacion'] + gg['Viga'].map(offsets)
            partes.append(gg)
        return pd.concat(partes, ignore_index=True)

    TablaEsfuerzos = Resultados('ENVOLVENTE')

    # ORDENAMIENTO DE TABLA DE ESFUERZOS
    TablaEsfuerzos['Viga'] = pd.to_numeric(TablaEsfuerzos['Viga'], errors='coerce')
    TablaEsfuerzos['Estacion'] = pd.to_numeric(TablaEsfuerzos['Estacion'],
    errors='coerce').round(6)
    TablaEsfuerzos['M3'] = pd.to_numeric(TablaEsfuerzos['M3'], errors='coerce')
    TablaEsfuerzos['V2'] = pd.to_numeric(TablaEsfuerzos['V2'], errors='coerce')
```

```

gcols = ['CasoCarga', 'Viga', 'Estacion']
ext = (TablaEsfuerzos.groupby(gcols)
      .agg(Mu_maximo=('M3', 'max'), Mu_minimo=('M3', 'min'),
          Vu_maximo=('V2', 'max'), Vu_minimo=('V2', 'min'))
      .reset_index())

ext['Vu'] = np.maximum(ext['Vu_maximo'].abs(), ext['Vu_minimo'].abs())
ext = _add_dist_sin_warning(ext)

TablaEsf =
(ext[['CasoCarga', 'Viga', 'Distancia', 'Mu_maximo', 'Mu_minimo', 'Vu_maximo', 'Vu_minimo', 'Vu'
]])
      .sort_values(['CasoCarga', 'Distancia', 'Viga'])
      .reset_index(drop=True)

t = TablaEsf.copy()
t['Distancia'] = t['Distancia'].round(6)
agg_cols = {'Mu_maximo': 'max', 'Mu_minimo': 'min', 'Vu_maximo': 'max', 'Vu_minimo': 'min'}

TablaResultados = (t.groupby(['CasoCarga', 'Distancia'],
as_index=False).agg(agg_cols))
TablaResultados['Vu'] = TablaResultados[['Vu_maximo', 'Vu_minimo']].abs().max(axis=1)
TablaResultados =
TablaResultados.sort_values(['CasoCarga', 'Distancia']).reset_index(drop=True)

# OBTENCION DE ESFUERZOS DE DISEÑO
MuPos = round(float(TablaResultados['Mu_maximo'].max()),2)
MuNeg = round(float(TablaResultados['Mu_minimo'].min()),2)
Vu = round(float(TablaResultados['Vu'].max().round(2)),2)

Indice_MuPos = TablaResultados['Mu_maximo'].idxmax()
Indice_MuNeg = TablaResultados['Mu_minimo'].idxmin()
Indice_Vu = TablaResultados['Vu'].idxmax()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# EXTRACCION RESULTADOS DE ESFUERZOS DEBIDO A PESO PROPIO
SapModel.SetPresentUnits(12)
SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
SapModel.Results.Setup.SetCaseSelectedForOutput('Pp')

NumberResults = 0
Obj, Elm, LoadCase, StepType, StepNum, ObjSta, ElmSta, P, V2, V3, T, M2, M3 = [], [],
[], [], [], [], [], [], [], [], []
[NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType, StepNum, P, V2, V3, T,
M2, M3, ret] = SapModel.Results.FrameForce(
'GrupoVigas', 2, NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType,
StepNum, P, V2, V3, T, M2, M3)

TablaPP =
pd.DataFrame({'CasoCarga': LoadCase, 'Viga': Obj, 'Estacion': ObjSta, 'Md': M3, 'Vd': V2})
TablaPP['Viga'] = pd.to_numeric(TablaPP['Viga'], errors='coerce')
TablaPP['Estacion'] = pd.to_numeric(TablaPP['Estacion'], errors='coerce').round(6)
TablaPP['Md'] = pd.to_numeric(TablaPP['Md'], errors='coerce')
TablaPP['Vd'] = pd.to_numeric(TablaPP['Vd'], errors='coerce')
TablaPP = (TablaPP.sort_values(['Viga', 'Estacion'])
      .drop_duplicates(subset=['Viga', 'Estacion'], keep='first')
      .reset_index(drop=True))

```

```

TablaPP = _add_dist_sin_warning(TablaPP)
TablaPP_collapsed = (TablaPP.groupby('Distancia',
as_index=False).agg(Md=('Md','first'), Vd=('Vd','first')))

TablaResultados['Distancia'] = TablaResultados['Distancia'].round(6)
TablaPP_collapsed['Distancia'] = TablaPP_collapsed['Distancia'].round(6)
TablaResultados = TablaResultados.merge(TablaPP_collapsed, on='Distancia',
how='left')

MdPos = round(float(TablaResultados.loc[Indice_MuPos, 'Md']),2)
MdNeg = round(float(TablaResultados.loc[Indice_MuNeg, 'Md']),2)

Vd = round(abs(float(TablaResultados.loc[Indice_MuNeg, 'Vd'])),2)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# EXTRACCIÓN RESULTADOS DE ESFUERZOS: ASD Cargas sostenidas y totales
def _momentos_combo_asd(nombre_combo: str, nombre_col_m: str):
    SapModel.SetPresentUnits(12)
    SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
    SapModel.Results.Setup.SetComboSelectedForOutput(nombre_combo)

    NumberResults = 0
    Obj, Elm, LoadCase, StepType, StepNum, ObjSta, ElmSta, P, V2, V3, T, M2, M3 = [],
[], [], [], [], [], [], [], [], [], [], []
    [NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType, StepNum, P, V2, V3,
T, M2, M3, ret] = SapModel.Results.FrameForce(
        'GrupoVigas', 2, NumberResults, Obj, ObjSta, Elm, ElmSta, LoadCase, StepType,
StepNum, P, V2, V3, T, M2, M3)

    dfm = pd.DataFrame({'CasoCarga':LoadCase, 'Viga':Obj, 'Estacion':ObjSta,
nombre_col_m:M3})
    dfm['Viga'] = pd.to_numeric(dfm['Viga'], errors='coerce')
    dfm['Estacion'] = pd.to_numeric(dfm['Estacion'], errors='coerce').round(6)
    dfm[nombre_col_m] = pd.to_numeric(dfm[nombre_col_m], errors='coerce')
    dfm = (dfm.sort_values(['Viga', 'Estacion'])
        .drop_duplicates(subset=['Viga', 'Estacion'], keep='first')
        .reset_index(drop=True))
    dfm = _add_dist_sin_warning(dfm)
    dfm = (dfm.groupby('Distancia', as_index=False)
        .agg(**{nombre_col_m: (nombre_col_m, 'first')}))
    dfm['Distancia'] = dfm['Distancia'].round(6)
    return dfm[['Distancia', nombre_col_m]]

Tabla_ASD_SOS = _momentos_combo_asd('ASD Cargas sostenidas', 'M_ASD_Sos')
Tabla_ASD_TOT = _momentos_combo_asd('ASD Cargas totales', 'M_ASD_Tot')

TablaResultados = (TablaResultados.merge(Tabla_ASD_SOS, on='Distancia',
how='left').merge(Tabla_ASD_TOT, on='Distancia', how='left'))

Mu_Sos_Pos = round(float(TablaResultados.loc[Indice_MuPos, 'M_ASD_Sos']),2)
Mu_Sos_Neg = round(float(TablaResultados.loc[Indice_MuNeg, 'M_ASD_Sos']),2)

Mu_Tot_Pos = round(float(TablaResultados.loc[Indice_MuPos, 'M_ASD_Tot']),2)
Mu_Tot_Neg = round(float(TablaResultados.loc[Indice_MuNeg, 'M_ASD_Tot']),2)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''

```

```

# EXTRACCION DE RESULTADOS DE DESPLAZAMIENTOS

SapModel.SetPresentUnits(14)
SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
SapModel.Results.Setup.SetComboSelectedForOutput('ELS')

NumberResults = 0
Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3 = [], [], [], [], [],
[], [], [], [], [], []
(NumberResults, Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3, ret) =
SapModel.Results.JointDispl('GrupoNodos', 2, NumberResults, Obj, Elm, LoadCase, StepType,
StepNum, U1, U2, U3, R1, R2, R3)

# ORDENAMIENTO DE TABLA DE DESPLAZAMIENTOS
df=pd.DataFrame({'Nodo': Obj, 'U3': U3})
df['Nodo'] = df['Nodo'].astype(str)
df = df.drop_duplicates(subset=['Nodo'], keep='first').reset_index(drop=True)

xs = []
for n in df['Nodo']:
    x, y, z, _ = SapModel.PointObj.GetCoordCartesian(n)
    xs.append(float(x))

df['X'] = xs
df = df.sort_values('X', kind='mergesort').reset_index(drop=True)
x0 = df['X'].min()
df['Distancia'] = (df['X'] - x0).round(6)
TablaDesplazamientos = df[['Nodo', 'Distancia', 'U3']]

# ADICION A TABLA PRINCIPAL
TablaResultados['Uz'] = TablaDesplazamientos['U3'].values
TablaResultados['Exc_Torones'] = Exc_Torones['Y'].values
TablaResultados['Theta_Toron'] = Exc_Torones['Theta_Toron'].values

Exc_Toron_Pos = -1 * ((AltoViga / 2) - 5)
Exc_Toron_Neg = ((AltoViga / 2) - 5)
Theta_Toron_Neg = TablaResultados.loc[Indice_Vu, 'Exc_Torones']

# OBTENCION DE DESPLAZAMIENTO MAXIMO
DespMax = round(float((TablaDesplazamientos['U3'].min() * -1)), 2)

return(TablaResultados, MuPos, MuNeg, Vu, DespMax, LongToron, Exc_Toron_Pos,
Exc_Toron_Neg, Vd, MdNeg, MdPos, Mu_Sos_Pos, Mu_Sos_Neg, Mu_Tot_Pos, Mu_Tot_Neg,
Theta_Toron_Neg)

```

## Funciones adicionales utilizadas en algoritmo de PSO

```
def CambioViga(AnchoViga,AltoViga):
    import comtypes.client
    # CONEXION A SAP 2000
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
    mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")
    SapModel = mySapObject.SapModel

    SapModel.SetModelIsLocked(False)
    SapModel.SetPresentUnits(14)

    # Cambio de dimensiones de viga
    SapModel.PropFrame.SetRectangle('Viga','4000psi',AltoViga,AnchoViga)

def CambioTorones(AltoViga,Torones,Niveles,Claros,Crujias):

    import numpy as np; import pandas as pd
    import comtypes.client
    # CONEXION A SAP 2000
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
    mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")
    SapModel = mySapObject.SapModel
    SapModel.SetModelIsLocked(False)
    SapModel.SetPresentUnits(14)
    RutaCoordenadasTorones=r'C:\Users\jmhdz\Documents\Maestría\Tesis\Experimentacion\Archivos para apoyo de codigo\Coordenadas tendones.xlsx'

    # APLICACION PRESFUERZO

    H_Toron=((AltoViga/2)-5)/2.54

    if Crujias==4 and Claros==900:        hoja=0
    if Crujias==4 and Claros==1100:       hoja=1
    if Crujias==4 and Claros==1300:       hoja=2
    if Crujias==6 and Claros==900:        hoja=3
    if Crujias==6 and Claros==1100:       hoja=4
    if Crujias==6 and Claros==1300:       hoja=5
    if Crujias==8 and Claros==900:        hoja=6
    if Crujias==8 and Claros==1100:       hoja=7
    if Crujias==8 and Claros==1300:       hoja=8

    df=pd.read_excel(RutaCoordenadasTorones,sheet_name=hoja,header=None)

    Coord_X=df[0].tolist()
    Coord_Y=df[1].tolist()
    Coord_X=[cx/2.54 for cx in Coord_X]
    Coord_Y=[cy*H_Toron for cy in Coord_Y]
    Coord_Z=[0 for cz in range(len(Coord_X))]

    MyType=[0 for mt in range(len(Coord_X))]
    MyType[0]=1
    i=1
    while i<len(Coord_X):
        if i%2==0:
            MyType[i]=6
```



## Código para cuantificación de acero considerando presfuerzo

```
def CuantificacionAceroConPresf(Var_Pos, Var_Neg, Diam_Pos, Diam_Neg, Crujias, Claros,
Est_Diam, Est_Sep, AnchoViga, AltoViga, Torones, LongToron):

    import numpy as np
    import math

    def CuantificacionAceroLongPos(Var_Pos, Diam_Pos, Crujias, Claros):
        if Diam_Pos == '0' or Var_Pos == 0:
            return 0.0

        PesoVar = {'4':0.996, '5':1.56, '6':2.25, '8':3.975, '10':6.225, '12':8.938}
        DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175,
'12':3.810}

        if int(Diam_Pos)<=8:
            DiamInterno=6*DiametroVar[Diam_Pos]
        else:
            DiamInterno=8*DiametroVar[Diam_Pos]
        PerimInterno=(np.pi*DiamInterno)

        LongEjes=Claros*Crujias
        LongCurva=PerimInterno/2
        LongDoblez=12*DiametroVar[Diam_Pos]*2

        LongTotal=LongEjes+LongCurva+LongDoblez
        PesoVarilla=PesoVar[Diam_Pos]*LongTotal*0.01

        return(round(Var_Pos*PesoVarilla*0.001,3))

    def CuantificacionAceroLongNeg(Var_Neg, Diam_Neg, Crujias, Claros):
        if Diam_Neg == '0' or Var_Neg == 0:
            return 0.0

        PesoVar = {'4':0.996, '5':1.56, '6':2.25, '8':3.975, '10':6.225, '12':8.938}
        DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175,
'12':3.810}

        if int(Diam_Neg)<=8:
            DiamInterno=6*DiametroVar[Diam_Neg]
        else:
            DiamInterno=8*DiametroVar[Diam_Neg]
        PerimInterno=(np.pi*DiamInterno)

        LongEjes=Claros*Crujias
        LongCurva=PerimInterno/2
        LongDoblez=12*DiametroVar[Diam_Neg]*2

        LongTotal=LongEjes+LongCurva+LongDoblez
        PesoVarilla=PesoVar[Diam_Neg]*LongTotal*0.01

        return(round(Var_Neg*PesoVarilla*0.001,3))

    def CuantificacionEstribos(Est_Diam, Est_Sep, AnchoViga, AltoViga, Crujias, Claros):

        PesoEst = {'3':0.557, '4':0.996}
        Diametro = {'3':0.952, '4':1.270}
```

```

DiamInt=4*Diametro[Est_Diam]
PerimInt=DiamInt**2*np.pi/4

Rectos=(AltoViga-5-DiamInt)*2+(AnchoViga-5-DiamInt)*2
Curvas90=PerimInt*3/4
Curvas135=PerimInt*3*2/8
Curvas=Curvas90+Curvas135
Dobleses=max(6*Diametro[Est_Diam], 8)*2
LongEst=Rectos+Curvas+Dobleses
PesoEstribo=PesoEst[Est_Diam]*LongEst*0.01

Estribos=math.ceil(Crujias*Claros/Est_Sep)

return(round(Estribos*PesoEstribo*0.001,3))

def CuantificacionPresfuerzo(Torones, LongToron):
    PesoToron=0.84 #kg/m
    return(round(PesoToron*0.001*Torones*LongToron*0.01,4))

AsLongPos=CuantificacionAceroLongPos(Var_Pos, Diam_Pos, Crujias, Claros)
AsLongNeg=CuantificacionAceroLongNeg(Var_Neg, Diam_Neg, Crujias, Claros)
AsEst=CuantificacionEstribos(Est_Diam, Est_Sep, AnchoViga, AltoViga, Crujias, Claros)
AsPresf=CuantificacionPresfuerzo(Torones, LongToron)

return AsLongPos+AsLongNeg+AsEst+AsPresf

# Diam_Var es STRING
# Claros en CM

# Devuelve Toneladas de acero

```

## Código para cuantificación de acero sin presfuerzo

```
def CuantificacionAceroSinPresf(Var_Pos, Var_Neg, Diam_Pos, Diam_Neg, Crujias, Claros,
Est_Diam, Est_Sep, AnchoViga, AltoViga):

    import numpy as np
    import math

    def CuantificacionAceroLongPos(Var_Pos, Diam_Pos, Crujias, Claros):

        PesoVar = {'4':0.996, '5':1.56, '6':2.25, '8':3.975, '10':6.225, '12':8.938}
        DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175,
'12':3.810}

        if int(Diam_Pos)<=8:
            DiamInterno=6*DiametroVar[Diam_Pos]
        else:
            DiamInterno=8*DiametroVar[Diam_Pos]
        PerimInterno=(np.pi*DiamInterno)

        LongEjes=Claros*Crujias
        LongCurva=PerimInterno/2
        LongDoblez=12*DiametroVar[Diam_Pos]*2

        LongTotal=LongEjes+LongCurva+LongDoblez
        PesoVarilla=PesoVar[Diam_Pos]*LongTotal*0.01

        return(round(Var_Pos*PesoVarilla*0.001,3))

    def CuantificacionAceroLongNeg(Var_Neg, Diam_Neg, Crujias, Claros):

        PesoVar = {'4':0.996, '5':1.56, '6':2.25, '8':3.975, '10':6.225, '12':8.938}
        DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175,
'12':3.810}

        if int(Diam_Neg)<=8:
            DiamInterno=6*DiametroVar[Diam_Neg]
        else:
            DiamInterno=8*DiametroVar[Diam_Neg]
        PerimInterno=(np.pi*DiamInterno)

        LongEjes=Claros*Crujias
        LongCurva=PerimInterno/2
        LongDoblez=12*DiametroVar[Diam_Neg]*2

        LongTotal=LongEjes+LongCurva+LongDoblez
        PesoVarilla=PesoVar[Diam_Neg]*LongTotal*0.01

        return(round(Var_Neg*PesoVarilla*0.001,3))

    def CuantificacionEstribos(Est_Diam, Est_Sep, AnchoViga, AltoViga, Crujias, Claros):

        PesoEst = {'3':0.557, '4':0.996}
        Diametro = {'3':0.952, '4':1.270}

        DiamInt=4*Diametro[Est_Diam]
        PerimInt=DiamInt**2*np.pi/4
```

```

Rectos=(AltoViga-5-DiamInt)*2+(AnchoViga-5-DiamInt)*2
Curvas90=PerimInt*3/4
Curvas135=PerimInt*3*2/8
Curvas=Curvas90+Curvas135
Dobleces=max(6*Diametro[Est_Diam], 8)*2
LongEst=Rectos+Curvas+Dobleces
PesoEstribo=PesoEst[Est_Diam]*LongEst*0.01

Estribos=math.ceil(Crujias*Claros/Est_Sep)

return(round(Estribos*PesoEstribo*0.001,3))

AsLongPos=CuantificacionAceroLongPos(Var_Pos, Diam_Pos, Crujias, Claros)
AsLongNeg=CuantificacionAceroLongNeg(Var_Neg, Diam_Neg, Crujias, Claros)
AsEst=CuantificacionEstribos(Est_Diam, Est_Sep, AnchoViga, AltoViga, Crujias, Claros)

return AsLongPos+AsLongNeg+AsEst

# Diam_Var es STRING
# Claros en CM

# Devuelve Toneladas de acero

```

## Código para cuantificación de concreto

```

def CuantificacionConcreto(AnchoViga, AltoViga, Crujias, Claros):

    PesoVolConc=2.4

    return(round(AnchoViga*AltoViga*Claros*Crujias*10**(-6)*PesoVolConc,3))

# AnchoViga, AltoViga y Claros en CM

# Devuelve Concreto en TON
print(CuantificacionConcreto(25,45,4,900))

```

## Código para diseño de cortante sin presfuerzo

```
def DiseñoCortanteSinPresfuerzo(AnchoViga,AltoViga,Vu,cuantia,Diam_Var):
    import numpy as np
    import math

    # GEOMETRIA Y MATERIALES
    fc = 400; fy = 4200.0
    b = AnchoViga; h = AltoViga; r = 4.0
    d = h - r
    FR = 0.75
    TON_to_KG = 1e3
    TMA = 0.75*2.54

    # AREAS DE VARILLAS
    AreaVar = {'3':0.7125, '4':1.267}

    # CALCULO DE VCR
    Vcr=min(FR*0.5*np.sqrt(fc)*b*d, FR*2*cuantia**(1/3)*np.sqrt(fc))
    Vcr_max=FR*1.25*np.sqrt(fc)*b*d
    Vcr_min=FR*0.25*np.sqrt(fc)*b*d

    if Vcr_max>=Vcr>=Vcr_min:
        Vcr=Vcr
    elif Vcr<Vcr_min:
        Vcr=Vcr_min
    else:
        Vcr=Vcr_max

    Vc=Vcr/FR
    Vcr=float(round(Vcr,2))

    # REVISION VU
    if Vu*TON_to_KG>FR*(Vc+2.2*np.sqrt(fc)*b*d):
        raise RuntimeError("Modificar dimensiones de viga")

    # CALCULO DE VSR
    Vsr=Vu*TON_to_KG-Vcr

    # SEPARACION REQUERIDA CON EST NO.3 Y NO.4
    Sep_3 = 999.0
    Sep_4 = 999.0

    if Vsr > 0:
        Sep_3 = max(math.floor((FR*2*AreaVar['3']*fy*d/Vsr)/2)*2,
math.floor((FR*2*AreaVar['3']*fy*d/Vsr)/5)*5)
        Sep_4 = max(math.floor((FR*2*AreaVar['4']*fy*d/Vsr)/2)*2,
math.floor((FR*2*AreaVar['4']*fy*d/Vsr)/5)*5)
    else:
        Sep_3 = 999.0
        Sep_4 = 999.0

    # SEPARACION MAXIMA
    if Vsr<=1.1*np.sqrt(fc)*b*d:
        Sep_max=d/2
    else:
        Sep_max=d/4
```

```

Sep_max_3=min(Sep_max,16*int(Diam_Var)*2.54/8, 48*3*2.54/8)
Sep_max_4=min(Sep_max,16*int(Diam_Var)*2.54/8, 48*4*2.54/8)

# SEPARACION MINIMA
Sep_min=1.5*TMA

# SELECCION DE DIAMETRO Y SEPARACION DE ESTRIBO
if Vsr<0:
    if int(Diam_Var)>10:
        Diam_est='4'
        Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
    else:
        Diam_est='3'
        Sep_est=max(math.floor(Sep_max_3/2)*2, math.floor(Sep_max_3/5)*5)

else:
    if int(Diam_Var)>10:
        Diam_est='4'
        if Sep_4>=Sep_max_4:
            Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)

        elif Sep_min<=Sep_4<Sep_max_4:
            Sep_est=Sep_4

        else:
            raise RuntimeError('Separacion no cumple con diametro del No.4')

    else:
        if Sep_3>=Sep_max_3:
            Diam_est='3'
            Sep_est=max(math.floor(Sep_max_3/2)*2, math.floor(Sep_max_3/5)*5)

        elif Sep_min<=Sep_3<Sep_max_3:
            Diam_est='3'
            Sep_est=Sep_3

        else:
            if Sep_4>=Sep_max_4:
                Diam_est='4'
                Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)

            elif Sep_min<=Sep_4<Sep_max_4:
                Diam_est='4'
                Sep_est=Sep_4

            else:
                raise RuntimeError('Separacion no cumple con ningun diametro
disponible de estribo')

# REVISION DE AV MIN
Av_min=max(0.2*np.sqrt(fc)*b*Sep_est/fy, 3.5*b*Sep_est/fy)
Av=2*AreaVar[Diam_est]
As_est=Av/2

if Av<Av_min:
    raise RuntimeError('Aumentar diametro de estribo o reducir separacion')

# CALCULO DE VSR
Vsr=FR*Av*fy*d/Sep_est

```

```
# CALCULO DE VR
Vr=float(round((Vcr+Vsr)/TON_to_KG,2))

# CALCULO DE FS
FS_Cort=float(round(Vr/Vu,2))

return(Diam_est, Sep_est, As_est, Vr, FS_Cort)

# AnchoViga y AltoViga en cm
# Vu en ton
# Diam Var es str

# Devuelve Diam de estribo, Separacion de estribos (cm), As de estribo (cm), Cortante
resistente (ton-m), Factor de seguridad
```

## Código para diseño a flexión sin presfuerzo

```
def DiseñoFlexionSinPresfuerzo(AnchoViga, AltoViga, Mu):
    import numpy as np, math

    # GEOMETRÍA Y MATERIALES
    fc = 400; fy = 4200.0
    b = float(AnchoViga); h = float(AltoViga); r = 4.0
    d = h - r
    Beta1 = 0.85 if fc >= 300 else 1.05 - fc / 1400.0
    fc_2 = 0.85 * fc
    FR = 0.90
    TONM_to_KGCM = 1e5
    TMA = 0.75 * 2.54
    cc_libre = 2.5
    d_est = 0.9
    ancho_util = b - 2 * (cc_libre + d_est)

    AreaVar = {'4':1.267, '5':1.979, '6':2.85, '8':5.067, '10':7.917, '12':11.401}
    DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175, '12':3.810}

    # 1. LÍMITES DE ACERO (NTC-2023)
    As_bal = (fc_2/fy) * ((6000.0*Beta1)/(fy + 6000.0)) * b * d
    As_max = 0.90 * As_bal
    As_min = max(0.80 * np.sqrt(fc) * b * d / fy, 14.0 * b * d / fy)

    # Momento Resistente Máximo (Ductilidad)
    a_max = (As_max * fy) / (fc_2 * b)
    Mr_max = FR * (As_max * fy * (d - a_max / 2.0)) / TONM_to_KGCM

    # 2. VALIDACIÓN INICIAL DE FACTIBILIDAD
    Mu_abs = abs(Mu)
    if Mu_abs > Mr_max:
        # En lugar de penalizar, devolvemos valores nulos
        return 0.0, 0, '0', round(Mr_max, 2), 0.0, 0.0, "Inviabile"

    # 3. CÁLCULO DE ACERO REQUERIDO
    A_q = FR * fy**2 / (2 * fc_2 * b)
    B_q = -FR * fy * d
    C_q = Mu_abs * TONM_to_KGCM
    roots = np.roots([A_q, B_q, C_q])

    try:
        As_req_teorico = min(r.real for r in roots if np.isreal(r) and r > 0)
    except ValueError:
        return 0.0, 0, '0', 0.0, 0.0, 0.0, "Error Raíces"

    As_diseño = max(As_req_teorico, As_min)

    # 4. BÚSQUEDA DE LA MEJOR COMBINACIÓN (Sencillas vs Paquetes)
    candidatos = []
    for key, area_bar in AreaVar.items():
        db = DiametroVar[key]
        for tipo in ['S', 'P']:
            if tipo == 'S':
                n_total = max(2, math.ceil(As_diseño / area_bar))
                n_apoyos = n_total
                db_eq = db
```

```

else:
    n_paquetes = max(2, math.ceil(As_diseño / (2 * area_bar)))
    n_total = n_paquetes * 2
    n_apoyos = n_paquetes
    db_eq = db * math.sqrt(2)

s_min = max(db_eq, 1.5 * TMA, 2.5)
s_clear = (ancho_util - n_apoyos * db_eq) / (n_apoyos - 1) if n_apoyos > 1
else 0

if s_clear >= s_min:
    As_tot = n_total * area_bar
    if As_tot <= As_max:
        a = (As_tot * fy) / (fc_2 * b)
        Mr_calc = FR * (As_tot * fy * (d - a / 2.0)) / TONM_to_KGCM

        if Mr_calc >= Mu_abs:
            candidatos.append({
                'As': As_tot, 'n': n_total, 'diam': key,
                'Mr': Mr_calc, 'tipo': tipo
            })
            break

if not candidatos:
    return 0.0, 0, '0', 0.0, 0.0, 0.0, "No cabe"

# 5. RETORNO DEL ÓPTIMO
candidatos.sort(key=lambda x: (round(x['As'], 3), 0 if x['tipo'] == 'S' else 1,
int(x['diam'])))
mejor = candidatos[0]

As_prov = round(mejor['As'], 2)
FS_Flex = round(mejor['Mr'] / Mu_abs, 2) if Mu_abs > 0 else 0
cuantia = round(As_prov / (b * d), 4)
tipo_armado = "Sencillas" if mejor['tipo'] == 'S' else "Paquetes de 2"

return (As_prov, mejor['n'], mejor['diam'], round(mejor['Mr'], 2), FS_Flex, cuantia,
tipo_armado)

```

## Código para revisión de desplazamientos

```

def Desplazamientos(DespMax, Claros):

    Desp_Perm=Claros/480+0.3

    ratio = abs(DespMax) / Desp_Perm
    P_Desp = 1e2

    if abs(DespMax) > Desp_Perm:
        Penalizacion = P_Desp * (ratio - 1)**2
    else:
        Penalizacion = 0

    return(abs(DespMax)<=Desp_Perm, Penalizacion)

''' Devuelve: Booleano de revision de desplazamiento'''

```

## Código para diseño por esfuerzos permisibles

```

def EsfuerzosPermisibles(AnchoViga, AltoViga, Torones, Exc_Torones, Mu_Sos, Mu_Tot, Md,
As_Pos, As_Neg):

    import numpy as np; import pandas as pd

    TONM_a_KGCM=1e5

    # GEOMETRIA
    h=AltoViga; b=AnchoViga
    A=b*h; I=b*h**3/12; y=h/2; e=Exc_Torones

    # MATERIALES
    fc=400; fci=0.8*fc; fpj=15200
    A_Toron=0.987; Aps=Torones*A_Toron

    # PERDIDAS, PRESFUERZO Y FUERZAS EFECTIVAS
    Perd_Inm=0.05; Perd_Dif=0.10; Perd_Tot=Perd_Inm+Perd_Dif
    fpe_transf=fpj*(1-Perd_Inm); fpe_serv=fpj*(1-Perd_Tot)
    Pe_transf=fpe_transf*Aps; Pe_serv=fpe_serv*Aps

    # CALCULO DE ESFUERZOS
    def Esfuerzo(Fibra, P, M):
        Com_P=-P/A
        if Fibra=='Inf':
            Mom_P=P*e*y/I
            Mom_M=M*y/I
        else:
            Mom_P=-P*e*y/I
            Mom_M=-M*y/I
        return(Com_P+Mom_P+Mom_M)

    Esf_Inf_Tran = Esfuerzo('Inf', Pe_transf, Md*TONM_a_KGCM)
    Esf_Sup_Tran = Esfuerzo('Sup', Pe_transf, Md*TONM_a_KGCM)
    Esf_Inf_Serv_Sos= Esfuerzo('Inf', Pe_serv, Mu_Sos*TONM_a_KGCM)
    Esf_Sup_Serv_Sos= Esfuerzo('Sup', Pe_serv, Mu_Sos*TONM_a_KGCM)
    Esf_Inf_Serv_Tot= Esfuerzo('Inf', Pe_serv, Mu_Tot*TONM_a_KGCM)
    Esf_Sup_Serv_Tot= Esfuerzo('Sup', Pe_serv, Mu_Tot*TONM_a_KGCM)

    # ESFUERZOS PERMISIBLES
    EP_Com_Tran=0.6*fci*-1; EP_Com_Ser_Tot=0.60*fc*-
1; EP_Com_Ser_Sos=0.45*fc*-1
    EP_Ten_Tran=float(0.8*np.sqrt(fc)); EP_Ten_Ser=float(1.6*np.sqrt(fc))

    # TENSION SOPORTADA POR ACERO DULCE
    fy=4200; fs=0.6*fy
    T_As_Neg=fs*As_Neg; T_As_Pos=fs*As_Pos

    # TENSIONES LIMITE PARA DEFINIR CLASE DE ELEMENTO PRESFORZADO
    LI_Clase=float(2*np.sqrt(fc)); LS_Clase=float(3.2*np.sqrt(fc))

    # TABLA DE ESFUERZOS
    Esfuerzos = [
        round(Esf_Inf_Tran,2),
        round(Esf_Sup_Tran,2),
        round(Esf_Inf_Serv_Sos,2),
        round(Esf_Sup_Serv_Sos,2),

```

```

        round(Esf_Inf_Serv_Tot,2),
        round(Esf_Sup_Serv_Tot,2)
    ]

    Etapa = pd.Series(data=[
        'Transferencia',
        'Transferencia',
        'Servicio cargas sostenidas',
        'Servicio cargas sostenidas',
        'Servicio cargas totales',
        'Servicio cargas totales'
    ])

    Fibra = pd.Series(data=[
        'Inferior',
        'Superior',
        'Inferior',
        'Superior',
        'Inferior',
        'Superior'
    ])

    EP_Compresion = pd.Series(data=[
        round(EP_Com_Tran,2),
        round(EP_Com_Tran,2),
        round(EP_Com_Ser_Sos,2),
        round(EP_Com_Ser_Sos,2),
        round(EP_Com_Ser_Tot,2),
        round(EP_Com_Ser_Tot,2)
    ])

    EP_Tension = pd.Series(data=[
        round(EP_Ten_Tran,2),
        round(EP_Ten_Tran,2),
        round(EP_Ten_Ser,2),
        round(EP_Ten_Ser,2),
        round(EP_Ten_Ser,2),
        round(EP_Ten_Ser,2)
    ])

    TablaEsfuerzos = pd.DataFrame({
        'ETAPA': Etapa,
        'FIBRA': Fibra,
        'ESFUERZO': Esfuerzos,
        'EP COMPRESION': EP_Compresion,
        'EP TENSION': EP_Tension
    })

    Cumple      = []
    Penal       = []
    T_Diferencia = []

    # DEFINICION DE PENALIZACIONES
    P_COMP = 1e2
    P_SA   = 1e2
    P_T    = 1e2
    P_A    = 1e2

```

```

for i in range(6):
    esf      = TablaEsfuerzos.iloc[i, 2] # ESFUERZO
    ep_comp  = TablaEsfuerzos.iloc[i, 3] # EP COMPRESION (negativo)
    ep_ten   = TablaEsfuerzos.iloc[i, 4] # EP TENSION (positivo)
    fibra    = TablaEsfuerzos.iloc[i, 1] # FIBRA

    if esf < 0: # compresión
        limit  = abs(ep_comp)
        esf_abs = abs(esf)
        t_diferencia = 0

        if esf_abs <= limit:
            c, pen = True, 0.0
        else:
            c = False
            ratio = esf_abs / limit
            pen = P_COMP * (ratio - 1.0)**2

    elif esf > 0: # tensión
        limit = ep_ten

        if esf <= limit:
            t_diferencia = 0
            c, pen = True, 0.0
        else:

            if esf < LI_Clase:
                t_diferencia = (esf - limit)*A/4
                c = False
                ratio = esf / limit
                pen = P_SA * (ratio - 1.0)**2

            elif LI_Clase <= esf <= LS_Clase:
                t_diferencia = (esf - limit)*A/4

                if fibra == 'Inferior':
                    if t_diferencia <= T_As_Pos:
                        c, pen = True, 0.0
                    else:
                        c = False
                        ratio = esf / limit
                        pen = P_T * (ratio - 1.0)**2
                else:
                    if t_diferencia <= T_As_Neg:
                        c, pen = True, 0.0
                    else:
                        c = False
                        ratio = esf / limit
                        pen = P_A * (ratio - 1.0)**2

            elif esf > LS_Clase:
                t_diferencia = (esf - limit)*A/4
                c = False
                ratio = esf / limit
                pen = P_COMP * (ratio - 1.0)**2

    else: # esf == 0
        c, pen = True, 0.0
        t_diferencia = 0

    Cumple.append(bool(c))

```

```

    Penal.append(pen)
    T_Diferencia.append(t_diferencia)

    TablaEsfuerzos['Tension diferencia'] = T_Diferencia
    TablaEsfuerzos['Cumple']           = Cumple
    TablaEsfuerzos['Penalizacion']     = Penal

    Revision          = bool(TablaEsfuerzos['Cumple'].all())
    PenalizacionTotal = float(sum(Penal))

    return (Revision, PenalizacionTotal, TablaEsfuerzos)

# Devuelve Revision / Tabla de Esfuerzos / Esfuerzos (Inf Tran, Sup Tran, Inf Serv Sos,
Sup Serv Sos, Inf Serv Tot, Sup Serv Tot) /
# Esfuerzos permisibles (Comp en Tran, Ten en Tran, Com en Serv Tot, Com en Serv Sos, Ten
en Serv)

```

## Código para diseño de cortante con presfuerzo

```
def DiseñoCortanteConPresfuerzo(AnchoViga, AltoViga, Vu, Mu, Vd, Md, Exc_Toron, Torones,
Theta_Toron, As_conv, Diam_Var):
    import numpy as np
    import math

    # GEOMETRIA Y MATERIALES
    fc = 400; fy = 4200.0
    b = AnchoViga; h = AltoViga; r = 4.0
    d = h - r; yt=h/2
    rp = yt-abs(Exc_Toron); dp = max(h-rp, 0.8*h)
    I=b*h**3/12
    FR = 0.75
    TON_M_to_KG_CM = 1e5
    TON_to_KG = 1e3
    TMA = 0.75*2.54

    # DATOS PRESFUERZO
    Ac=b*h
    Aps=Torones*0.987
    fsr=19000
    Perdidas=0.20
    fsj=0.8*fsr
    fe=fsj*(1-Perdidas)
    Pe=fe*Aps

    # PENALIZACIONES
    P_Vu = 1e2
    P_Est= 1e2
    P_Av = 1e2

    # AREAS DE VARILLAS
    AreaVar = {'3':0.7125, '4':1.267}

    # COMPONENTE DE PRESFUERZO
    theta = math.radians(Theta_Toron)
    Vp = Pe*math.sin(theta)
    fpc = (Pe)/Ac

    # CALCULO DE VCR
    fd=abs(Md*TON_M_to_KG_CM)*yt/I
    Mcre=(I/yt)*(1.6*np.sqrt(fc)+fe-fd)

    if Aps*fe<0.4*(Aps*fsr+As_conv*fy):
        Vci=FR*max(0.16*np.sqrt(fc)*b*dp+Vd*TON_to_KG+(Vu*TON_to_KG*Mcre/(Mu*TON_M_to_KG_
CM)), 0.45*np.sqrt(fc)*b*d)
    else:
        Vci=FR*max(0.16*np.sqrt(fc)*b*dp+Vd*TON_to_KG+(Vu*TON_to_KG*Mcre/(Mu*TON_M_to_KG_
CM)), 0.50*np.sqrt(fc)*b*d)

    Vcw=FR*(0.93*np.sqrt(fc)+0.3*fpc)*b*dp+Vp

    Vcr=min(Vci, Vcw)
    Vc=Vcr
    Vcr=float(round(Vcr,2))

    # REVISION VU
```

```

if Vu*TON_to_KG>(Vc+2.2*np.sqrt(fc)*b*d):
    ratio_Vu = Vu*TON_to_KG / ((Vc+2.2*np.sqrt(fc)*b*d))
    pen_Vu = P_Vu * (ratio_Vu - 1) **2
else:
    pen_Vu = 0

# CALCULO DE VSR
Vsr=Vu*TON_to_KG-Vcr

# SEPARACION REQUERIDA CON EST NO.3 Y NO.4
Sep_3 = 999.0
Sep_4 = 999.0

if Vsr > 0:
    Sep_3 = max(math.floor((FR*2*AreaVar['3']*fy*d/Vsr)/2)*2,
math.floor((FR*2*AreaVar['3']*fy*d/Vsr)/5)*5)
    Sep_4 = max(math.floor((FR*2*AreaVar['4']*fy*d/Vsr)/2)*2,
math.floor((FR*2*AreaVar['4']*fy*d/Vsr)/5)*5)
    Sep_2_3 = max(math.floor((FR*4*AreaVar['3']*fy*d/Vsr)/2)*2,
math.floor((FR*4*AreaVar['3']*fy*d/Vsr)/5)*5)
    Sep_2_4 = max(math.floor((FR*4*AreaVar['4']*fy*d/Vsr)/2)*2,
math.floor((FR*4*AreaVar['4']*fy*d/Vsr)/5)*5)

else:
    Sep_3 = 999.0
    Sep_4 = 999.0

# SEPARACION MAXIMA
if Vsr<=1.1*np.sqrt(fc)*b*d:
    Sep_max=d/2
else:
    Sep_max=d/4

Sep_max_3=min(Sep_max,16*3*2.54/8, 48*3*2.54/8)
Sep_max_4=min(Sep_max,16*4*2.54/8, 48*4*2.54/8)

# SEPARACION MINIMA
Sep_min=1.5*TMA

# SELECCION DE DIAMETRO Y SEPARACION DE ESTRIBO
if Vsr<0:
    pen_Est = 0
    if int(Diam_Var)>10:
        Diam_est='4'
        Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
    else:
        Diam_est='3'
        Sep_est=max(math.floor(Sep_max_3/2)*2, math.floor(Sep_max_3/5)*5)

else:
    if int(Diam_Var)>10:
        Diam_est='4'
        if Sep_4>=Sep_max_4:
            Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
            pen_Est = 0

        elif Sep_min<=Sep_4<Sep_max_4:
            Sep_est=Sep_4
            pen_Est = 0

```

```

else:
    Diam_est='4 doble'
    if Sep_2_4>Sep_max_4:
        Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
        pen_Est = 0

    elif Sep_min<=Sep_2_4<Sep_max_4:
        Sep_est=Sep_2_4
        pen_Est = 0

    else:
        Sep_est=Sep_2_4
        ratio_Est = Sep_min/Sep_2_4
        pen_Est = P_Est * (ratio_Est-1)**2

else:
    if Sep_3>=Sep_max_3:
        Diam_est='3'
        Sep_est=max(math.floor(Sep_max_3/2)*2, math.floor(Sep_max_3/5)*5)
        pen_Est = 0

    elif Sep_min<=Sep_3<Sep_max_3:
        Diam_est='3'
        Sep_est=Sep_3
        pen_Est = 0

    else:
        if Sep_4>=Sep_max_4:
            Diam_est='4'
            Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
            pen_Est = 0

        elif Sep_min<=Sep_4<Sep_max_4:
            Diam_est='4'
            Sep_est=Sep_4
            pen_Est = 0

        else:
            Diam_est='4 doble'

            if Sep_2_4>Sep_max_4:
                Sep_est=max(math.floor(Sep_max_4/2)*2, math.floor(Sep_max_4/5)*5)
                pen_Est = 0

            elif Sep_min<=Sep_2_4<Sep_max_4:
                Sep_est=Sep_2_4
                pen_Est = 0

            else:
                Sep_est=Sep_2_4
                ratio_Est = Sep_min/Sep_2_4
                pen_Est = P_Est * (ratio_Est-1)**2

# REVISION DE AV MIN
Av_min=max(0.2*np.sqrt(fc)*b*Sep_est/fy, 3.5*b*Sep_est/fy)
if Diam_est!='4 doble':
    Av=2*AreaVar[Diam_est]
    As_est=Av/2

```

```

else:
    Av=4*AreaVar['4']
    As_est=Av/2

if Av<Av_min:
    ratio_Av = Av_min/Av
    pen_Av = P_Av * (ratio_Av-1) **2
else:
    pen_Av = 0

# CALCULO DE VSR
Vsr=FR*Av*fy*d/Sep_est

# CALCULO DE VR
Vr=float(round((Vcr+Vsr)/TON_to_KG,2))

# CALCULO DE FS
FS_Cort=float(round(Vr/Vu,2))

# CALCULO DE PENALIZACIONES
PenalizacionTotal=pen_Av+pen_Est+pen_Vu

return(Diam_est, Sep_est, Vr, FS_Cort, PenalizacionTotal)

# AnchoViga y AltoViga en cm
# Vu, Vd en ton
# Mu, Md en ton-m
# Exc_Toron en cm
# As_conv en cm2
# Diam Var es str

# Devuelve Diam de estribo, Separacion de estribos (cm), Cortante resistente (ton-m),
Factor de seguridad

```

## Código para diseño a flexión con presfuerzo

```
def DiseñoFlexionConPresfuerzo(AnchoViga, AltoViga, Mu, Torones, Exc_Toron):
    import numpy as np
    import math

    # CONSTANTES Y MATERIALES
    TONM_a_KGCM = 1e5
    FR = 0.90
    fc = 400; fy = 4200.0; fpu = 19000.0
    b = float(AnchoViga); h = float(AltoViga); r = 4.0
    d = h - r; yt = h/2.0
    Beta1 = 0.85 if fc >= 300 else (1.05 - fc/1400.0)
    fc_2 = 0.85 * fc
    TMA = 0.75 * 2.54

    # 1. PRESFUERZO
    # dp es la distancia desde la fibra extrema en compresión al centroide del acero de
    # presfuerzo
    dp = h - (yt + Exc_Toron) if Mu > 0 else h - (yt - Exc_Toron)
    Aps = Torones * 0.987
    # Esfuerzo en el acero de presfuerzo en el estado de falla (fps) aproximado por NTC
    fps = fpu * (1 - 0.5 * (Aps * fpu) / (b * dp * fc_2)) # Simplificación NTC
    fps = max(0.0, min(fps, 0.9 * fpu))
    T_ps = Aps * fps

    # 2. LÍMITE DE DUCTILIDAD (ACERO MÁXIMO)
    c_b = (6000.0 / (fy + 6000.0)) * d
    a_b = Beta1 * c_b
    As_max = (0.90 * fc_2 * b * a_b - T_ps) / fy

    a_max = (T_ps + max(0, As_max) * fy) / (fc_2 * b)
    Mr_max_posible = FR * (T_ps * (dp - a_max/2.0) + max(0, As_max) * fy * (d -
a_max/2.0)) / TONM_a_KGCM

    # 3. REVISIÓN CRÍTICA Y PENALIZACIÓN
    Mu_abs = abs(Mu)
    Mn_req = Mu_abs * TONM_a_KGCM / FR
    a_aprox = 0.2 * d
    As_req = (Mn_req - T_ps*(dp - a_aprox/2.0)) / (fy * (d - a_aprox/2.0))
    As_min = max(0.80*np.sqrt(fc)*b*d/fy, 14.0*b*d/fy)
    As_diseno = max(As_req, As_min)

    if Mu_abs > Mr_max_posible:
        ratio_falla = Mu_abs / Mr_max_posible
        penal_M = (ratio_falla ** 2) * 1e2
        return (As_req, 0, '0', round(Mr_max_posible, 2), 0.0, 0.0, penal_M,
"Ductilidad")

    # 4. DISEÑO DE ACERO CONVENCIONAL
    AreaVar = {'4':1.267, '5':1.979, '6':2.85, '8':5.067, '10':7.917, '12':11.401}
    DiametroVar = {'4':1.270, '5':1.588, '6':1.905, '8':2.540, '10':3.175, '12':3.810}
    ancho_util = b - 2*(2.5 + 0.9)

    candidatos = []
    peor_separacion_ratio = 0.0 # Para penalizar si ninguna cabe

    for clave, av in AreaVar.items():
```

```

db = DiametroVar[clave]
for tipo in ['S', 'P']:
    n_total = max(2, math.ceil(As_diseno / (av if tipo=='S' else 2*av)))
    n_total = n_total if tipo=='S' else n_total * 2
    n_apoyos = n_total if tipo=='S' else n_total / 2
    db_eq = db if tipo=='S' else db * math.sqrt(2)

    s_min = max(db_eq, 1.5 * TMA, 2.5)
    s_clear = (ancho_util - n_apoyos * db_eq) / (n_apoyos - 1) if n_apoyos > 1
else 0

# Guardamos el ratio de qué tan lejos estamos de que quepa
ratio_s = s_min / s_clear if s_clear > 0 else 5.0
if ratio_s > peor_separacion_ratio: peor_separacion_ratio = ratio_s

if s_clear >= s_min:
    As_tot = n_total * av
    if As_tot <= As_max:
        a_real = (T_ps + As_tot * fy) / (fc_2 * b)
        Mr_final = FR * (T_ps * (dp - a_real/2.0) + As_tot * fy * (d -
a_real/2.0)) / TONM_a_KGCM

        if Mr_final >= Mu_abs:
            candidatos.append({'As': As_tot, 'n': n_total, 'diam': clave,
'Mr': Mr_final, 'tipo': tipo})
            break

if not candidatos:
    # Penalización dinámica por espacio: si s_clear fue muy pequeño, la penalización
es muy alta
    penal_espacio = (peor_separacion_ratio ** 2) * 1e4
    return (As_diseno, 0, '8', 0.1, 0.0, 0.0, penal_espacio, "No cabe")

# Selección del mejor candidato (Menor Área)
candidatos.sort(key=lambda x: (round(x['As'], 3), 0 if x['tipo']=='S' else 1,
int(x['diam'])))
mejor = candidatos[0]

return (round(mejor['As'], 2), mejor['n'], mejor['diam'], round(mejor['Mr'], 2),
round(mejor['Mr']/Mu_abs, 2), round(mejor['As']/(b*d), 4), 0.0,
"Sencillas" if mejor['tipo'] == 'S' else "Paquetes de 2")

```

## Código para algoritmo general de análisis sin presfuerzo

```
import pandas as pd
from Analisis_Estructural.Casos import CasosDeEstudio as Casos
from Analisis_Estructural.AnalisisSinPresfuerzo import AnalisisSinPresfuerzo as
AnaSinPres
from Diseño_Estructural.DiseñoFlexionSinPresfuerzo import DiseñoFlexionSinPresfuerzo as
DisFlexSinPres
from Diseño_Estructural.DiseñoCortanteSinPresfuerzo import DiseñoCortanteSinPresfuerzo as
DisCortSinPres
from Diseño_Estructural.DiseñoDesplazamientos import Desplazamientos as DisDesp
from Cuantificacion.CuantificacionConcreto import CuantificacionConcreto as CuanCon
from Cuantificacion.CuantificacionAceroSinPresf import CuantificacionAceroSinPresf as
CuanAs
from Analisis_Estructural.ResultadosAnalisis import AnalisisRapido as AnalisisRapido

# DEFINICION DE CASO DE ESTUDIO
CasosDeEstudio=Casos()

# LISTAS DE RESULTADOS
L_Usos, L_Niveles, L_Crujias, L_Claros, L_AnchoViga, L_AltoViga = [], [], [], [], [], []
L_MuNeg, L_As_Neg, L_Var_Neg, L_Diam_Neg, L_M_Res_Neg, L_FS_Flex_Neg, L_Cuant_Neg,
L_Tipo_Var_Neg= [], [], [], [], [], [], [], []
L_MuPos, L_As_Pos, L_Var_Pos, L_Diam_Pos, L_M_Res_Pos, L_FS_Flex_Pos, L_Cuant_Pos,
L_Tipo_Var_Pos= [], [], [], [], [], [], [], []
L_Vu, L_Est_Diam, L_Vr, L_FS_Cort, L_Est_Sep, L_DespMax, L_Rev_Despl = [], [], [], [], [],
[], []
L_Concreto, L_Acero = [], []
L_TablaRes=[]

for i in range(len(CasosDeEstudio)):

    Usos, Niveles, Crujias, Claros = CasosDeEstudio[i][0], CasosDeEstudio[i][1],
CasosDeEstudio[i][2], CasosDeEstudio[i][3]

    # RESULTADOS DE ANALISIS ESTRUCTURAL
    AnchoViga, AltoViga, TablaResultados, MuPos, MuNeg, Vu, DespMax = AnaSinPres(Usos,
Niveles, Crujias, Claros)

    # RESULTADOS DE DISEÑO A FLEXION
    As_Pos, Var_Pos, Diam_Pos, Mr_Pos, FS_Flex_Pos, Cuant_Pos, Tipo_Var_Pos =
DisFlexSinPres(AnchoViga, AltoViga, MuPos)
    As_Neg, Var_Neg, Diam_Neg, Mr_Neg, FS_Flex_Neg, Cuant_Neg, Tipo_Var_Neg =
DisFlexSinPres(AnchoViga, AltoViga, MuNeg)

    # RESULTADOS DE DISEÑO POR CORTANTE
    Est_Diam, Est_Sep, Est_As, Vr, FS_Cort = DisCortSinPres(AnchoViga, AltoViga, Vu,
Cuant_Neg, Diam_Neg)

    # RESULTADOS DESPLAZAMIENTOS
    Rev_Despl = DisDesp(DespMax, Claros)

    # CUANTIFICACIONES
    PesoConcreto_Bruto = CuanCon(AnchoViga, AltoViga, Crujias, Claros)
    PesoAcero = CuanAs(Var_Pos, Var_Neg, Diam_Pos, Diam_Neg, Crujias, Claros, Est_Diam,
Est_Sep, AnchoViga, AltoViga)
```

```

PesoConcreto = PesoConcreto_Bruto - PesoAcero

# LISTAS DE RESULTADOS
L_Uso.append(Uso), L_Niveles.append(Niveles), L_Crujias.append(Crujias),
L_Claros.append(round(Claros*0.01,0))
L_AnchoViga.append(AnchoViga), L_AltoViga.append(AltoViga)

L_MuNeg.append(MuNeg), L_As_Neg.append(As_Neg), L_Var_Neg.append(Var_Neg),
L_Diam_Neg.append(Diam_Neg), L_M_Res_Pos.append(Mr_Pos),
L_FS_Flex_Pos.append(FS_Flex_Pos), L_Cuant_Pos.append(Cuant_Pos),
L_Tipo_Var_Pos.append(Tipo_Var_Pos)
L_MuPos.append(MuPos), L_As_Pos.append(As_Pos), L_Var_Pos.append(Var_Pos),
L_Diam_Pos.append(Diam_Pos), L_M_Res_Neg.append(Mr_Neg),
L_FS_Flex_Neg.append(FS_Flex_Neg), L_Cuant_Neg.append(Cuant_Neg),
L_Tipo_Var_Neg.append(Tipo_Var_Neg)

L_Vu.append(Vu), L_Est_Diam.append(Est_Diam), L_Est_Sep.append(Est_Sep),
L_Vr.append(Vr), L_FS_Cort.append(FS_Cort),
L_DespMax.append(DespMax), L_Rev_Desp.append(Rev_Desp)

L_Concreto.append(PesoConcreto), L_Acero.append(PesoAcero)
L_TablaRes.append(TablaResultados)

ResultPrinc=pd.DataFrame(list(zip(L_Uso, L_Niveles, L_Crujias, L_Claros, L_AnchoViga,
L_AltoViga, L_Concreto, L_Acero)),
                           columns=['Uso', 'Niveles', 'Crujias', 'Claros (m)', 'Ancho Viga
(cm)', 'Alto Viga (cm)', 'Concreto (ton)', 'Acero (ton)'])

ResultSecun=pd.DataFrame(list(zip(L_Uso, L_Niveles, L_Crujias, L_Claros, L_AnchoViga,
L_AltoViga,
                                L_MuNeg, L_As_Neg, L_Var_Neg, L_Diam_Neg, L_M_Res_Neg,
L_FS_Flex_Neg, L_Cuant_Neg, L_Tipo_Var_Neg,
                                L_MuPos, L_As_Pos, L_Var_Pos, L_Diam_Pos, L_M_Res_Pos,
L_FS_Flex_Pos, L_Cuant_Pos, L_Tipo_Var_Pos,
                                L_Vu, L_Est_Diam, L_Est_Sep, L_Vr, L_FS_Cort, L_DespMax,
L_Rev_Desp)),
                           columns=['Uso', 'Niveles', 'Crujias', 'Claros (m)', 'Ancho Viga
(cm)', 'Alto Viga (cm)',
                                'Mu (-) tm', 'As (-) cm2', 'Varillas (-)', 'Ø (-)', 'Mr
(-) tm', 'FS (-)', 'Cuantia (-)', 'Tipo var (-)',
                                'Mu (+) tm', 'As (+) cm2', 'Varillas (+)', 'Ø (+)', 'Mr
(+) tm', 'FS (+)', 'Cuantia (+)', 'Tipo var (+)',
                                'Vu (t)', 'Ø Estribo', 'Separacion (cm)', 'Vr (t)', 'FS
Vr', 'Uz max (cm)', '¿Cumple Uz?'])

with pd.ExcelWriter(r'C:\Users\jmhhdz\Documents\Maestría\Tesis\Experimentacion\Análisis
Sin Presfuerzo\Análisis.xlsx') as writer:
    ResultPrinc.to_excel(writer, sheet_name='Resultados principales', index=False)
    ResultSecun.to_excel(writer, sheet_name='Resultados secundarios', index=False)

```



```

print("")
print(f'Inicio de Analisis de {Uso}_{Niveles}N_{Crujias}CR_{int(Claros/100)}M')
print("")

# ESPACIO DE SOLUCIONES: ANCHO, ALTO, TORONES

if Claros==900:
    rangos_altos = {
        25: (30, 65),
        26: (30, 65),
        28: (30, 65),
        30: (30, 65),
        32: (32, 65),
        34: (34, 65),
        35: (35, 65),
    }
elif Claros==1100:
    rangos_altos = {
        25: (35, 65),
        26: (35, 65),
        28: (35, 65),
        30: (35, 65),
        32: (35, 65),
        34: (35, 65),
        35: (35, 65),
    }
else:
    rangos_altos = {
        25: (40, 65),
        26: (40, 65),
        28: (40, 65),
        30: (40, 65),
        32: (40, 65),
        34: (40, 65),
        35: (40, 65),
    }

MaxTorones = {
    25: 6,
    26: 6,
    28: 8,
    30: 8,
    32: 8,
    34: 8,
    35: 10,
}

def es_multiplo_2_o_5(h):
    return (h % 2 == 0) or (h % 5 == 0)

geometrias = []
for ancho, (h_min, h_max) in rangos_altos.items():
    for h in range(h_min, h_max + 1):
        if es_multiplo_2_o_5(h):
            geometrias.append({'Ancho': ancho, 'Alto': h})

CombinacionesGeom = pd.DataFrame(geometrias)

nuevas_filas = []

```

```

for _, fila in CombinacionesGeom.iterrows():
    ancho = int(fila['Ancho'])
    alto = int(fila['Alto'])
    max_t = MaxTorones.get(ancho, 1)
    for n_toron in range(1, max_t + 1):
        nuevas_filas.append({'Ancho': ancho, 'Alto': alto, 'Torones': n_toron})

Posiciones = pd.DataFrame(nuevas_filas)
Posiciones = Posiciones.sort_values(by='Alto', ascending=True)
n_pos = len(Posiciones)
print("Total de soluciones (ancho, alto, torones):", n_pos)
print("")

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# LISTAS DE RESULTADOS (UNA FILA POR ITERACIÓN - GBest)

L_Uso, L_Niveles, L_Crujias, L_Claros = [], [], [], []
L_AnchoViga, L_AltoViga, L_Torones = [], [], []

L_MuNeg, L_As_Neg, L_Var_Neg, L_Diam_Neg, L_Mr_Neg, L_FS_Flex_Neg, L_Cuant_Neg,
L_Tipo_Var_Neg = [], [], [], [], [], [], [], []
L_MuPos, L_As_Pos, L_Var_Pos, L_Diam_Pos, L_Mr_Pos, L_FS_Flex_Pos, L_Cuant_Pos,
L_Tipo_Var_Pos = [], [], [], [], [], [], [], []
L_Vu, L_Est_Diam, L_Est_Sep, L_Vr, L_FS_Cort = [], [], [], [], []

L_DespMax, L_Rev_Desp = [], []
L_ASD_Pos, L_ASD_Neg = [], []
L_TablaASD = []

L_Concreto, L_Acero = [], []

L_TablaRes = []

# Para ploteo
pos_hist = []
gbest_hist = []
F_hist = []
Pc_hist = []
Pa_hist = []
F_part_hist= []

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# MODELACION INICIAL
ModIni(Uso,Niveles,Crujias,Claros)
AnalisisRapido()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# EVALUACION DE SOLUCION

def evaluar_solucion(AnchoViga, AltoViga, Torones, alpha):

    # ACTUALIZACION DE SECCION TRANSVERSAL DE VIGA
    CambioViga(AnchoViga,AltoViga)

    # ACTUALIZACION DE CANTIDAD Y TRAYECTORIA DE TORONES
    Exc_Torones,LongToron=CambioTorones(AltoViga,Torones,Niveles,Claros,Crujias)

```

```

# OBTENCION DE RESULTADOS DE ANALISIS
(TablaResultados, MuPos, MuNeg, Vu, DespMax, LongToron,Exc_Toron_Pos,
Exc_Toron_Neg, VdNeg, MdNeg, MdPos,
Mu_Sos_Pos, Mu_Sos_Neg, Mu_Tot_Pos,
Mu_Tot_Neg,Theta_Toron_Neg)=ResultadosAnalisis(Exc_Torones,LongToron,AltoViga,RutaModelo)

# LIMPIEZA DE RESULTADOS PARA SIGUIENTE ITERACION
LimpiezaResultados()

# DISEÑO A FLEXIÓN
As_Pos, Var_Pos, Diam_Pos, Mr_Pos, FS_Flex_Pos, Cuant_Pos, Penal_Mu_Pos,
Tipo_Var_Pos = DisFlexConPres(AnchoViga, AltoViga, MuPos, Torones, Exc_Toron_Pos)
As_Neg, Var_Neg, Diam_Neg, Mr_Neg, FS_Flex_Neg, Cuant_Neg, Penal_Mu_Neg,
Tipo_Var_Neg = DisFlexConPres(AnchoViga, AltoViga, MuNeg, Torones, Exc_Toron_Neg)

if Penal_Mu_Pos > 0:
    Var_Pos=math.ceil((As_Pos/5.067)/1);    As_Pos=Var_Pos*5.067;    Diam_Pos =
'g'
if Penal_Mu_Neg > 0:
    Var_Neg=math.ceil((As_Neg/5.067)/1);    As_Neg=Var_Neg*5.067;    Diam_Neg =
'g'

# DISEÑO A CORTANTE
Est_Diam, Est_Sep, Vr, FS_Cort, Penal_Vu = DisCortConPres(AnchoViga, AltoViga,
Vu, MuNeg, VdNeg, MdNeg,Exc_Toron_Neg, Torones, Theta_Toron_Neg, As_Neg, Diam_Neg)

# REVISION ASD
Revision_ASDFPos, Pen_ASDFPos, Tabla_ASDFPos = EP(AnchoViga, AltoViga,
Torones,Exc_Toron_Pos, Mu_Sos_Pos, Mu_Tot_Pos, MdPos, As_Pos, As_Neg)
Revision_ASDFNeg, Pen_ASDFNeg, Tabla_ASDFNeg = EP(AnchoViga, AltoViga,
Torones,Exc_Toron_Neg, Mu_Sos_Neg, Mu_Tot_Neg, MdNeg, As_Pos, As_Neg)

# REVISION DESPLAZAMIENTOS
Rev_Desp, Penal_Desp = DisDesp(DespMax, Claros)

# FACTORES PARA PENALIZACIONES
FACTOR_MU = 1e2;    FACTOR_VU = 1e2;    FACTOR_ASD = 1e3;    FACTOR_DESP =
1e2

# PENALIZACION POR MU
Penal_Mu = Penal_Mu_Pos + Penal_Mu_Neg
Peso_Mu = FACTOR_MU * Penal_Mu

# PENALIZACION POR VU
Peso_Vu = FACTOR_VU * Penal_Vu

# PENALIZACION POR ASD
Penal_ASD = Pen_ASDFPos + Pen_ASDFNeg
Peso_ASD = FACTOR_ASD * Penal_ASD

# PENALIZACION POR DESPLAZAMIENTOS
PesoDesp = FACTOR_DESP * Penal_Desp

# PENALIZACION TOTAL
PenalizacionTotal = Peso_Mu + Peso_Vu + Peso_ASD + PesoDesp

# CUANTIFICACION DE CONCRETO Y ACERO CON PENALIZACIONES

```

```

        PesoConcreto = CuanCon(AnchoViga, AltoViga, Crujias, Claros) + PenalizacionTotal
* alpha
        PesoAcero = CuanAs(Var_Pos, Var_Neg, Diam_Pos, Diam_Neg, Crujias, Claros,
Est_Diam, Est_Sep, AnchoViga, AltoViga, Torones, LongToron) + PenalizacionTotal * (1-
alpha)

        # DICCIONARIO CON RESULTADOS FINALES
res = {
        'Uso': Uso,
        'Niveles': Niveles,
        'Crujias': Crujias,
        'Claros_m': round(Claros * 0.01, 0),
        'AnchoViga': AnchoViga,
        'AltoViga': AltoViga,
        'Torones': Torones,

        'MuNeg': MuNeg, 'AsNeg': As_Neg, 'VarNeg': Var_Neg, 'DiamNeg': Diam_Neg,
'Mr_Neg': Mr_Neg, 'FS_Flex_Neg': FS_Flex_Neg, 'Cuant_Neg': Cuant_Neg, 'Tipo var (-
)':Tipo_Var_Neg,
        'MuPos': MuPos, 'AsPos': As_Pos, 'VarPos': Var_Pos, 'DiamPos': Diam_Pos,
'Mr_Pos': Mr_Pos, 'FS_Flex_Pos': FS_Flex_Pos, 'Cuant_Pos': Cuant_Pos, 'Tipo var
(+)':Tipo_Var_Pos,

        'Vu': Vu, 'EstDiam': Est_Diam, 'EstSep': Est_Sep, 'Vr': Vr, 'FS_Cort':
FS_Cort,
        'DespMax': DespMax, 'RevDesp': Rev_Desp,

        'PesoConcreto': PesoConcreto,
        'PesoAcero': PesoAcero,

        'ASDPos': Revision_ASDFlexPos,
        'ASDNeg': Revision_ASDFlexNeg,

        'TablaResultados': TablaResultados,
        'TablaASDPos': Tabla_ASDFlexPos.copy(),
        'TablaASDNeg': Tabla_ASDFlexNeg.copy()
}

    return res

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# FUNCION FITNESS PONDERADA

def fitness_ponderado(AnchoViga, AltoViga, Torones, alpha):

    res = evaluar_solucion(AnchoViga, AltoViga, Torones, alpha)
    Pc = res['PesoConcreto']
    Pa = res['PesoAcero']
    F = alpha * Pc + (1.0 - alpha) * Pa

    return F, res

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# ADICION DE RESULTADOS A LISTAS

def append_res_to_lists(res):
    L_Uso.append(res['Uso'])

```

```

L_Niveles.append(res['Niveles'])
L_Crujias.append(res['Crujias'])
L_Claros.append(res['Claros_m'])
L_AnchoViga.append(res['AnchoViga'])
L_AltoViga.append(res['AltoViga'])
L_Torones.append(res['Torones'])

L_MuNeg.append(res['MuNeg'])
L_As_Neg.append(res['AsNeg'])
L_Var_Neg.append(res['VarNeg'])
L_Diam_Neg.append(res['DiamNeg'])
L_Mr_Neg.append(res['Mr_Neg'])
L_FS_Flex_Neg.append(res['FS_Flex_Neg'])
L_Cuant_Neg.append(res['Cuant_Neg'])
L_Tipo_Var_Neg.append(res['Tipo var (-)'])

L_MuPos.append(res['MuPos'])
L_As_Pos.append(res['AsPos'])
L_Var_Pos.append(res['VarPos'])
L_Diam_Pos.append(res['DiamPos'])
L_Mr_Pos.append(res['Mr_Pos'])
L_FS_Flex_Pos.append(res['FS_Flex_Pos'])
L_Cuant_Pos.append(res['Cuant_Pos'])
L_Tipo_Var_Pos.append(res['Tipo var (+)'])

L_Vu.append(res['Vu'])
L_Est_Diam.append(res['EstDiam'])
L_Est_Sep.append(res['EstSep'])
L_Vr.append(res['Vr'])
L_FS_Cort.append(res['FS_Cort'])

L_DespMax.append(res['DespMax'])
L_Rev_Desp.append(res['RevDesp'])

L_Concreto.append(res['PesoConcreto'])
L_Acero.append(res['PesoAcero'])

L_ASD_Pos.append(res['ASDPos'])
L_ASD_Neg.append(res['ASDNeg'])
L_TablaRes.append(res['TablaResultados'])

# CACHÉ PARA EVITAR REANÁLISIS
cache_soluciones = {} # Diccionario para evitar repetir SAP2000
conteo_sap_real = 0 # Contador de análisis nuevos
soluciones_validas_vistas = set()

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# PSO
# PARÁMETROS
w_max = 0.9
w_min = 0.4
c1 = 1.5
c2 = 2.5

num_particles = 15
num_iterations = 30
dim = 1

```

```

xmin = 0
xmax = n_pos - 1

alpha = 0.6

pos = np.random.uniform(xmin, xmax, (num_particles, dim))
vel = np.zeros((num_particles, dim))

pbest_pos = pos.copy()
pbest_val = np.zeros(num_particles)
pbest_res = [None] * num_particles

idx_particulas = np.zeros(num_particles, dtype=int)

# pbest inicial
for i in range(num_particles):
    idx = int(round(pos[i, 0]))
    idx = max(xmin, min(xmax, idx))
    idx_particulas[i] = idx

    fila = Posiciones.iloc[idx]
    a_i = fila['Ancho']
    h_i = fila['Alto']
    t_i = fila['Torones']

    F_i, res_i = fitness_ponderado(a_i, h_i, t_i, alpha)
    pbest_val[i] = F_i
    pbest_res[i] = res_i
    pbest_pos[i, 0] = idx

gbest_index = int(np.argmin(pbest_val))
gbest_val = pbest_val[gbest_index]
gbest_pos = pbest_pos[gbest_index].copy()
gbest_res = pbest_res[gbest_index]

append_res_to_lists(gbest_res)

fila_g_init = Posiciones.iloc[int(gbest_pos[0])]
for signo, tabla in [('Pos', 'TablaASDPos'), ('Neg', 'TablaASDNeg')]:
    asd_init = gbest_res[tabla].copy()
    asd_init['Iter'] = 0
    asd_init['Signo'] = signo
    asd_init['Ancho'] = fila_g_init['Ancho']
    asd_init['Alto'] = fila_g_init['Alto']
    asd_init['Torones'] = fila_g_init['Torones']
    L_TablaASD.append(asd_init)

fila_g = Posiciones.iloc[int(gbest_pos[0])]

print(f"PSO inicio: mejor indice = {int(gbest_pos[0])}, "
      f"GBest ≈ (b={fila_g['Ancho']}, h={fila_g['Alto']}, torones={fila_g['Torones']}), "
      f"F = {gbest_val:.4f}")
print("")

# GUARDADO DE ENJAMBRE INICIAL (ITERACION 0) ANTES DE ACTUALIZAR PARTICULAS

anchos_init = Posiciones.iloc[idx_particulas]['Ancho'].to_numpy()
altos_init = Posiciones.iloc[idx_particulas]['Alto'].to_numpy()

```

```

torones_init= Posiciones.iloc[idx_particulas]['Torones'].to_numpy()
F_part_hist.append(pbest_val.copy())
pos_hist.append(np.column_stack([anchos_init, altos_init, torones_init]))

gbest_hist.append([fila_g['Ancho'], fila_g['Alto'], fila_g['Torones']])
F_hist.append(gbest_val)
Pc_hist.append(gbest_res['PesoConcreto'])
Pa_hist.append(gbest_res['PesoAcero'])

# DEFINICION DE CRITERIO DE PARADA SI SE REPITE 10 VECES SEGUIDAS GBEST

iteraciones_sin_mejora = 0
paciencia = 20
gbest_val_anterior = gbest_val

# BUCLE PRINCIPAL PARA EVALUACION DE PARTICULAS

for t in range(num_iteraciones):

    if num_iteraciones > 1:
        w = w_max - (w_max - w_min) * (t / (num_iteraciones - 1))
    else:
        w = w_min

    fitness_de_esta_pasada = []

    for i in range(num_particles):
        r1 = np.random.rand(dim)
        r2 = np.random.rand(dim)

        cognitivo = c1 * r1 * (pbest_pos[i] - pos[i])
        social     = c2 * r2 * (gbest_pos     - pos[i])
        inercia    = w * vel[i]

        vel[i] = inercia + cognitivo + social

        pos[i] = pos[i] + vel[i]
        pos[i, 0] = max(xmin, min(xmax, pos[i, 0]))

        idx = int(round(pos[i, 0]))
        idx = max(xmin, min(xmax, idx))
        idx_particulas[i] = idx

        fila = Posiciones.iloc[idx]
        a_i = fila['Ancho']
        h_i = fila['Alto']
        t_i = fila['Torones']

    # REVISION DE CACHE EN BUCLE DE PARTICULAS ANTES DE HACER ANALISIS EN SAP
2000
    clave = (a_i, h_i, t_i)
    if clave in cache_soluciones:
        F_i, res_i = cache_soluciones[clave]
    else:
        F_i, res_i = fitness_ponderado(a_i, h_i, t_i, alpha)
        cache_soluciones[clave] = (F_i, res_i)
        conteo_sap_real += 1

    if F_i < 500:

```

```

        soluciones_validas_vistas.add(clave)

# ADICION DE FITNESS DE PARTICULA A LISTA
fitness_de_esta_pasada.append(F_i)

# REVISION DE PBEST PARA PARTICULA EN CUESTION
if F_i < pbest_val[i]:
    pbest_val[i] = F_i
    pbest_pos[i, 0] = idx
    pbest_res[i] = res_i

# REVISION DE GBEST
best_idx_iter = int(np.argmin(pbest_val))
if pbest_val[best_idx_iter] < gbest_val:
    gbest_val = pbest_val[best_idx_iter]
    gbest_pos = pbest_pos[best_idx_iter].copy()
    gbest_res = pbest_res[best_idx_iter]

# OBTENCION DE RESULTADOS DE PARTICULA GBEST DE ESTA CORRIDA
fila_g = Posiciones.iloc[int(gbest_pos[0])]
print(f"Iteración {t+1}/{num_itearations} - gbest indice = {int(gbest_pos[0])}, "
      f"(b={fila_g['Ancho']}, h={fila_g['Alto']}, torones={fila_g['Torones']}), "
      f"F = {gbest_val:.4f}")

# Guardado GBest de esta iteración
append_res_to_lists(gbest_res)

# ASD TABLA (una fila por iteracion)
asd_pos = gbest_res['TablaASDPos'].copy()
asd_pos['Iter'] = t + 1
asd_pos['Signo'] = 'Pos'
asd_pos['Ancho'] = fila_g['Ancho']
asd_pos['Alto'] = fila_g['Alto']
asd_pos['Torones'] = fila_g['Torones']

asd_neg = gbest_res['TablaASDNeg'].copy()
asd_neg['Iter'] = t + 1
asd_neg['Signo'] = 'Neg'
asd_neg['Ancho'] = fila_g['Ancho']
asd_neg['Alto'] = fila_g['Alto']
asd_neg['Torones'] = fila_g['Torones']

L_TablaASD.append(asd_pos)
L_TablaASD.append(asd_neg)

# GUARDADO DE POSICIONES DE ENJAMBRE DE ESTA ITERACION
anchos_part = Posiciones.iloc[idx_particulas]['Ancho'].to_numpy()
altos_part = Posiciones.iloc[idx_particulas]['Alto'].to_numpy()
torones_part = Posiciones.iloc[idx_particulas]['Torones'].to_numpy() #
AÑADI LISTA DE TORONES PARA PARTICULAS
pos_hist.append(np.column_stack([anchos_part, altos_part, torones_part])) #
AÑADI TORONES A LAS COLUMNAS DEL HISTORIAL DE PARTICULAS

gbest_hist.append([fila_g['Ancho'], fila_g['Alto'], fila_g['Torones']]) #
AÑADI TORONES A LAS COLUMNAS DEL HISTORIAL DE gbest
F_hist.append(gbest_val)
Pc_hist.append(gbest_res['PesoConcreto'])
Pa_hist.append(gbest_res['PesoAcero'])
F_part_hist.append(fitness_de_esta_pasada)

```

```

# VERIFICACIÓN DE CRITERIO DE PARADA

if abs(gbest_val - gbest_val_anterior) < 1e-6:
    iteraciones_sin_mejora += 1
else:
    iteraciones_sin_mejora = 0 # Si hubo mejora, reiniciamos el contador
    gbest_val_anterior = gbest_val # Actualizamos el valor de referencia

if iteraciones_sin_mejora >= paciencia:
    print(f"Convergencia alcanzada en la iteración {t+1}")
    break # Sale del bucle for principal

# CONVERSION DE GBEST A ARRAY DE NUMPY
gbest_hist_np = np.array(gbest_hist)

# CIERRE DE SAP
CierreSAP()

# RESULTADOS FINALES DE ITERACION
fila_g = Posiciones.iloc[int(gbest_pos[0])]
print("\nPSO terminado.")
print(f"Mejor indice = {int(gbest_pos[0])} en Posiciones")
print("Mejor combinación (ancho, alto, torones) =",
      fila_g.to_dict())
print(f"Función agregada F = {gbest_val:.4f}")
print(f"Peso de concreto = {gbest_res['PesoConcreto']:.4f}, "
      f"Peso de acero = {gbest_res['PesoAcero']:.4f}")

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# DATAFRAMES & EXCEL

ResultPrinc = pd.DataFrame({
    'Uso': [L_Uso[0]] * len(F_hist),
    'Niveles': [L_Niveles[0]] * len(F_hist),
    'Crujías': [L_Crujias[0]] * len(F_hist),
    'Claro (m)': [L_Claros[0]] * len(F_hist),
    'Iteración': range(len(F_hist)),
    'Ancho de viga (cm)': gbest_hist_np[:, 0],
    'Alto de viga (cm)': gbest_hist_np[:, 1],
    'Torones': gbest_hist_np[:, 2].astype(int),
    'Concreto (ton)': Pc_hist,
    'Acero (ton)': Pa_hist,
    'Fitness': F_hist})

ResultSecun = pd.DataFrame(list(zip(
    L_Uso, L_Niveles, L_Crujias, L_Claros,
    L_AnchoViga, L_AltoViga, L_Torones,
    L_MuNeg, L_As_Neg, L_Var_Neg, L_Diam_Neg, L_Mr_Neg, L_FS_Flex_Neg, L_Cuant_Neg,
L_Tipo_Var_Neg,
    L_MuPos, L_As_Pos, L_Var_Pos, L_Diam_Pos, L_Mr_Pos, L_FS_Flex_Pos, L_Cuant_Pos,
L_Tipo_Var_Pos,
    L_Vu, L_Est_Diam, L_Est_Sep, L_Vr, L_FS_Cort, L_DespMax, L_Rev_Desp,
    L_ASD_Pos, L_ASD_Neg)),
    columns=['Uso', 'Niveles', 'Crujías', 'Claro (m)',
            'Ancho de viga (cm)', 'Alto de viga (cm)', 'Torones',

```

```

        'Mu (-) tm', 'As (-) cm2', 'Varillas (-)', 'φ (-)', 'Mr (-) tm', 'FS (-)',
'Cuantia (-)', 'Tipo var (-)',
        'Mu (+) tm', 'As (+) cm2', 'Varillas (+)', 'φ (+)', 'Mr (+) tm', 'FS (+)',
'Cuantia (+)', 'Tipo var (+)',
        'Vu (t)', 'Estribos φ', 'Separación (cm)', 'Vr (t)', 'FS Vr', 'Uz max
(cm)', 'Uz OK?',
        'ASD inferior', 'ASD superior'])

TablaASD = pd.concat(L_TablaASD, ignore_index=True) if L_TablaASD else pd.DataFrame()

# OBTENCION DE ESTADISTICAS DE EFICIENCIA
iteracion_final = t + 1 # t es el índice del bucle donde ocurrió el break
comb_intentadas = num_particulas * iteracion_final
comb_max_teoricas = num_particulas * num_iterations

# 1. Eficiencia de Memoria (Caché)
ahorro_cache = (1 - (conteo_sap_real / comb_intentadas)) * 100

# 2. Eficiencia de Tiempo (Parada Temprana)
ahorro_early_stop = (1 - (iteracion_final / num_iterations)) * 100

# 3. Ahorro Global (Combinado)
ahorro_total = (1 - (conteo_sap_real / comb_max_teoricas)) * 100

# CREACIÓN DE LA TABLA PARA EXCEL
datos_eficiencia = {
    "Concepto": [
        "Iteraciones máximas teóricas: ",
        "Combinaciones máximas teóricas: ",
        "Iteración de Parada (Convergencia): ",
        "Combinaciones evaluadas: ",
        "Análisis reales en SAP2000: ",
        "Ahorro por Memoria: ",
        "Ahorro por parada temprana: ",
        "Ahorro global de tiempo: ",
        "Soluciones que cumplen: "
    ],
    "Valor": [
        num_iterations,
        comb_max_teoricas,
        iteracion_final,
        comb_intentadas,
        conteo_sap_real,
        f"{ahorro_cache:.2f}%",
        f"{ahorro_early_stop:.2f}%",
        f"{ahorro_total:.2f}%",
        len(soluciones_validas_vistas)
    ]
}

df_eficiencia=pd.DataFrame(datos_eficiencia)

print("")
print("ESTADÍSTICAS DE EFICIENCIA")
print("")
print(df_eficiencia)

# GUARDADO DE ARCHIVO DE EXCEL

```

```

excel_path = os.path.join(RutaCarpeta, 'Análisis.xlsx')

with pd.ExcelWriter(excel_path) as writer:
    ResultPrinc.to_excel(writer, sheet_name='Resultados principales', index=False)
    ResultSecun.to_excel(writer, sheet_name='Resultados secundarios', index=False)
    TablaASD.to_excel(writer, sheet_name='ASD Esfuerzos', index=False)
    df_eficiencia.to_excel(writer, sheet_name='Eficiencia', index=False)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# PLOTEO DE DISTRIBUCION DE PARTICULAS EN ESPACIO DE SOLUCIONES

matplotlib.use('Agg')

# Preparación de datos
pos_hist = np.array(pos_hist)
gbest_hist = np.array(gbest_hist)
F_hist = np.array(F_hist)
Pc_hist = np.array(Pc_hist)
Pa_hist = np.array(Pa_hist)

'''/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/'''
# GRÁFICA: HISTORIA DE CONVERGENCIA

matplotlib.use('Agg')
plt.figure(figsize=(12, 8))
# Ajuste de márgenes para que no se corte el texto
plt.subplots_adjust(top=0.88, bottom=0.15, left=0.15, right=0.95)

# CONFIGURACIÓN DE TAMAÑOS

TAM_TITULO = 18
TAM_EJES = 14
TAM_TICKS = 12
TAM_TEXTO_INFO = 11

# 1. Graficar la nube de puntos de TODAS las partículas (Gris)
for t in range(len(F_part_hist)):
    # Creamos el eje X (iteración actual 't') repetido para cada partícula
    x_iter = [t] * len(F_part_hist[t])
    # Obtenemos la lista de fitness de todas las partículas en la iteración 't'
    y_fit = F_part_hist[t]

    # Graficamos los puntos:
    plt.scatter(x_iter, y_fit, color='gray', s=22, alpha=0.5, edgecolors='none',
label='Partículas' if t == 0 else "")

# 2. Graficar la línea del MEJOR GLOBAL (GBest) resaltada (Rojo)
# Recorremos el historial del GBest
plt.plot(range(len(F_hist)), F_hist, color='blue', linewidth=1, marker='o',
markersize=3, label='GBest')

y_max_real = max([max(iteracion) for iteracion in F_part_hist])
plt.yscale('log')
plt.ylim(bottom=4, top=y_max_real * 1.1)
major_ticks = [5, 10, 100, 1000, 10000, 100000]
plt.yticks(major_ticks, [str(t) for t in major_ticks])

```

```

plt.grid(True, which="both", ls="-", alpha=0.15)
plt.gca().xaxis.set_major_locator(plt.MaxNLocator(integer=True))

# Títulos y etiquetas con tamaños grandes
plt.title('Evolución de la Función Objetivo', fontsize=TAM_TITULO, pad=25,
weight='bold')
plt.xlabel('Iteración', fontsize=TAM_EJES, weight='bold')
plt.ylabel('Fitness Ponderada (ton)', fontsize=TAM_EJES, weight='bold')

# Forzamos que los ticks del eje X sean números enteros (0, 1, 2, 3...)
plt.xticks(range(num_iterations + 1), fontsize=12, rotation=90)
plt.yticks(fontsize=TAM_TICKS)

# Leyenda profesional
plt.legend(fontsize=TAM_TEXTO_INFO, loc='upper right', frameon=True)

# 4. Guardar resultado en alta resolución para la tesis
ruta_conv_final = os.path.join(RutaCarpeta, 'Convergencia_Rojo_Gris.png')
plt.savefig(ruta_conv_final, dpi=300, bbox_inches='tight')
plt.close()
print("")
print('-----')
print("")

```