

**Desarrollo de una herramienta
computacional para la
elaboración de diagramas
momento-curvatura de muros con
geometría compleja**

ESCUELA DE INGENIERÍA EN CONSTRUCCIÓN

CONSTANCIA DE PRESENTACIÓN PÚBLICA DEL TRABAJO FINAL DE GRADUACIÓN

Desarrollo de una herramienta computacional para la elaboración de diagramas momento-curvatura de muros con geometría compleja

Llevado a cabo por el estudiante:

Sánchez Soto Jose Pablo

Carné: 2019035465

Trabajo Final de Graduación presentado públicamente ante el Tribunal Evaluador el martes 30 de abril de 2024 como requisito parcial para optar por el grado de Licenciatura en Ingeniería en Construcción, del Instituto Tecnológico de Costa Rica.

En fe de lo anterior firman los siguientes integrantes del Tribunal evaluador:

MILTON ANTONIO SANDOVAL QUIROS (FIRMA)

Firmado digitalmente por MILTON ANTONIO SANDOVAL QUIROS (FIRMA)
Fecha: 2024.04.30 15:10:00 -06'00'

Ing. Milton Sandoval Quirós, MAE
Representante Director de la Escuela

ANGEL HUMBERTO NAVARRO MORA (FIRMA)

Firmado digitalmente por ANGEL HUMBERTO NAVARRO MORA (FIRMA)
Fecha: 2024.05.02 08:13:41 -06'00'

Ing. Ángel Navarro Mora, MSc.
Profesor Guía

GUSTAVO ADOLFO ROJAS MOYA (FIRMA)

Firmado digitalmente por GUSTAVO ADOLFO ROJAS MOYA (FIRMA)
Fecha: 2024.05.02 13:03:41 -06'00'

Ing. Gustavo Rojas Moya, MSc.
Profesor Lector

MAURICIO CARRANZA SOLANO (FIRMA)

Firmado digitalmente por MAURICIO CARRANZA SOLANO (FIRMA)
Fecha: 2024.05.02 14:12:15 -06'00'

Ing. Mauricio Carranza Solano
Profesor Observador

Resumen

El presente proyecto tiene como finalidad el desarrollo de una herramienta computacional para el cálculo automatizado del diagrama momento-curvatura de muros con geometría compleja, las cuales resultan de la conexión de muros rectos, esto con el fin de conocer el comportamiento de los elementos en el rango plástico y con ello, obtener información que pueda ser utilizada en el análisis no lineal de edificaciones. La herramienta fue construida a través del programa de análisis estructural OpenSeesPy en el lenguaje de programación Python y busca simplificar el proceso de cálculo a través de una interfaz intuitiva. La metodología aborda objetivos específicos, desde la evaluación de controles gráficos hasta la implementación de un modelo estructural.

Los resultados muestran la elección de la biblioteca PyQT y la estructura del programa. La validación incluye la comparación con datos obtenidos mediante el software XTRACT, destacando similitudes y diferencias en el comportamiento estructural, atribuidas en su gran mayoría a la discrepancia entre los modelos constitutivos que ofrecen ambos programas.

En general, el proyecto aspira a proporcionar a estudiantes e ingenieros una herramienta eficiente y amigable para el análisis no lineal de muros con variaciones geométricas.

Palabra clave: OpenSeesPy, muros, herramienta computacional, análisis no lineal.

Abstract

The present project aims to develop a computational tool for the automated calculation of the moment-curvature diagram for walls with complex geometry. These geometries are generated through the assembly of linear walls, with the purpose of understanding the behavior of the elements in the plastic range. This information is valuable for nonlinear analysis in building structures. The tool was constructed using the OpenSeesPy structural analysis program in the Python programming language and aims to simplify the calculation process through an intuitive interface. The methodology focuses on achieving specific objectives, from evaluating graphical controls to implementing a structural model.

Results showcase the use of the PyQT library and the program's structure. Validation involves comparison with data obtained from the XTRACT software, highlighting similarities and differences in structural behavior, mostly attributed to the discrepancy between the constitutive models provided by both programs.

In general, the project aspires to provide students and engineers with an efficient and user-friendly tool for the nonlinear analysis of walls with geometric variations.

Keyword: OpenSeesPy, walls, computational tool, nonlinear analysis.

Desarrollo de una herramienta computacional para la elaboración de diagramas momento-curvatura de muros con geometría compleja

JOSE PABLO SÁNCHEZ SOTO

Proyecto final de graduación para optar por el grado de
Licenciatura en Ingeniería en Construcción

DICIEMBRE 2023

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA EN CONSTRUCCIÓN

Contenido

<u>RESUMEN EJECUTIVO</u>	<u>2</u>
<u>INTRODUCCIÓN</u>	<u>4</u>
OBJETIVOS.....	5
ALCANCE Y LIMITACIONES.....	6
ANTECEDENTES	7
<u>CAPÍTULO 1: MARCO TEÓRICO.....</u>	<u>8</u>
1.1 PROGRAMA	8
1.2 LENGUAJE DE PROGRAMACIÓN.....	8
1.3 PROGRAMACIÓN ORIENTADA A OBJETOS.....	9
1.4 INTERFAZ GRAFICA	11
1.5 MODELACIÓN ESTRUCTURAL	12
1.6 DIAGRAMA MOMENTO CURVATURA	17
<u>CAPÍTULO 2: METODOLOGÍA</u>	<u>21</u>
2.1 ETAPAS DE LA INVESTIGACIÓN	21
2.2 ETAPA 1.....	21
2.3 ETAPA 2.....	26
2.4 ETAPA 3.....	34
<u>CAPÍTULO 3: RESULTADOS Y ANÁLISIS</u>	<u>40</u>
3.1 EVALUACIÓN DE CONTROLES GRÁFICOS	40
3.2 ESTRUCTURA DEL PROGRAMA	41
3.3 VALIDACIÓN.....	46
<u>CONCLUSIONES Y RECOMENDACIONES</u>	<u>58</u>
CONCLUSIONES.....	58
RECOMENDACIONES	60
<u>REFERENCIAS</u>	<u>61</u>
<u>APÉNDICES</u>	<u>64</u>

Resumen ejecutivo

El presente proyecto se enfoca en la creación de una herramienta computacional especializada en el análisis estructural de secciones de muros de concreto reforzado, con el principal propósito de generar de manera eficiente y precisa diagramas momento-curvatura para distintas configuraciones geométricas. La implementación de la herramienta se llevó a cabo de manera exitosa, destacando la atención puesta en el diseño de una interfaz intuitiva mediante PyQt que guía al usuario de manera fluida a lo largo del proceso analítico.

La herramienta, estructurada con principios de programación orientada a objetos y organizada en una jerarquía de scripts, se destaca por su eficiencia y la facilidad para futuras expansiones. El uso de OpenSeesPy para el análisis estructural, complementado con Python, NumPy y Matplotlib, permitió un análisis detallado y una visualización efectiva de los resultados.

La combinación de OpenSeesPy y Python no solo proporcionó un entorno de desarrollo poderoso, sino que también permitió aprovechar las capacidades de cálculo de OpenSeesPy para realizar análisis no lineales y generar diagramas momento-curvatura precisos. Este enfoque híbrido no solo facilita la implementación de métodos avanzados de análisis estructural, sino que también asegura la robustez y confiabilidad de la herramienta en la conceptualización de muros con geometría compleja.

El software desarrollado incorpora de manera efectiva el método de fibras, que se basa en la subdivisión de la sección especificada por el usuario. En este proceso, cada fibra resultante es asignada a un material específico, determinado por su ubicación en la sección. Para representar el comportamiento de los materiales, se emplearon herramientas poderosas como el comando Concrete01 de OpenSeesPy para el concreto y SteelMPF para el acero. Estas elecciones constitutivas permiten modelar de manera precisa y detallada las propiedades mecánicas de los materiales presentes en la sección, proporcionando así una representación fidedigna del comportamiento estructural durante el análisis.

El análisis estructural implementado en el software se ejecuta de manera integral, tomando en cuenta la carga axial constante definida por el usuario y un momento flector unitario. La carga axial proporciona un escenario realista, lo que permite al usuario introducir condiciones específicas para representar situaciones prácticas. Además, el momento flector unitario, que crece linealmente con el tiempo o avance del análisis, añade un componente dinámico al estudio. Esta combinación de cargas contribuye a la generación de diagramas momento-curvatura detallados, capturando la respuesta estructural bajo condiciones variadas

La validación de la herramienta se realizó mediante la comparación con datos generados por el software de análisis estructural de secciones transversales XTRACT, se evidenció una notoria similitud en los resultados de los diagramas momento-curvatura, con un porcentaje de error de 0,0165%. Además, se incorporó a la herramienta una función denominada "Calculadora Mander" para evaluar el aumento en resistencia y ductilidad del concreto confinado, cuyos resultados fueron validados mediante la comparación con un estudio previo realizado en la Escuela Politécnica Federal de Zúrich.

Introducción

Las normativas que regulan el abordaje de los métodos de análisis y el comportamiento sismo-resistente de las estructuras, tal como el Código Sísmico de Costa Rica 2010 Revisión 2014 (CSCR-10/14) y los Requisitos de Reglamento para Concreto Estructural (ACI 318-19), publicado por el Instituto Americano del Concreto (ACI, por sus siglas en inglés), adopta dentro de su filosofía de aplicación un nivel de daño a las edificaciones, en función de la importancia de estas, siempre priorizando proteger y salvaguardar la vida de los ocupantes. La definición del comportamiento no lineal que presentan los elementos utilizados para disipar la energía proveniente de las cargas sísmicas, a través de deformaciones inelásticas, permite contar con un mayor control del comportamiento esperado de la edificación ante solicitaciones sísmicas.

Ahora bien, previo al análisis de un sistema estructural global, es de suma importancia conocer el desempeño y la respuesta que va a experimentar cada uno de los elementos estructurales ante solicitaciones sísmicas. Esta práctica permite comprender mejor el funcionamiento individual de cada componente, lo que contribuye así a la optimización del diseño y la evaluación de la seguridad de una estructura. De forma análoga, determinar la capacidad en flexión y ductilidad de un elemento estructural induce a predecir el tipo de falla que se va a presentar cuando se alcance la capacidad de este.

Por lo anterior expuesto, a partir del presente proyecto, se desarrollará una herramienta computacional para la elaboración de diagramas momento-curvatura de muros con geometría compleja, o bien, con formas conocidas a nivel del mercado como tipo C, I, L, Z, T, etc. Este trabajo tiene como finalidad, proporcionar a los estudiantes y profesionales una herramienta que automatice el proceso de cálculo y análisis estructural necesario para conocer la capacidad máxima en flexión y la ductilidad de una sección, sirviendo, así como insumo para la formación y desarrollo profesional de las personas.

El desarrollo del proyecto aborda el objetivo 4 de la Agenda 2030 Desarrollo Sostenible (ODS): “Garantizar una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje”, meta 4.3: “De aquí a 2030, aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento”. A partir del desarrollo del presente proyecto, se generó una herramienta para el uso, tanto de la comunidad académica, como a profesionales dedicados al área de la ingeniería estructural, con el fin de contribuir a la comprensión del comportamiento plástico de muros con geometría compleja ante solicitaciones de carga. Finalmente, cumple con el objetivo 9 de la Agenda 2030 de Desarrollo Sostenible (ODS): “Construir

infraestructuras resilientes, promover la industrialización inclusiva y sostenible”, meta 9.b: “Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas”. Desde la creación de la herramienta computacional, se ha desarrollado un software de código abierto que sirve como insumo para el análisis estructural de edificaciones que presenten muros con geometría compleja.

Objetivos

Para el desarrollo del presente proyecto se plantearon los siguientes objetivos:

Objetivo general

- Desarrollar una herramienta computacional, para la elaboración de diagramas momento-curvatura de muros con geometría compleja, al adoptar los métodos de análisis establecidos en las normativas vigentes.

Objetivos específicos

- Diseñar una herramienta computacional en los lenguajes de programación Python y OpenSeesPy para la conceptualización de diagramas momento-curvatura de forma intuitiva, a partir de una interfaz amigable con el usuario.
- Desarrollar una programación para el procesamiento de datos, así como, el cálculo no lineal del muro estructural especificado por el usuario, a partir de las herramientas computacionales Python y OpenSeesPy, para el desarrollo del diagrama momento-curvatura.
- Validar los resultados proporcionados por la herramienta computacional desarrollada, a través de una comparación con datos extraídos del software de análisis estructural XTRACT, para una correcta verificación de los resultados calculados.

Alcance y limitaciones

El proyecto se centrará en el desarrollo de una herramienta computacional destinada a la automatización del cálculo no lineal para la confección de los diagramas momento-curvatura de muros con geometría compleja. Este documento detallará el proceso de desarrollo del programa, abordar desde su concepción hasta su implementación práctica.

Inicialmente, se pondrá énfasis en el análisis y diseño de una interfaz de usuario que garantice una experiencia óptima para el usuario. La interfaz será concebida para ser accesible y comprensible tanto para estudiantes como para ingenieros., incorporando elementos visuales y técnicos que facilitarán la interacción. El diseño y la elección de los paquetes para el desarrollo de controles gráficos se basará en una rigurosa evaluación por medio de una matriz de evaluación de desempeño como guía.

Posteriormente, se procede con la configuración del programa, que incluye la creación de una herramienta para el ingreso de las características geométricas de las diferentes secciones. Además, se lleva a cabo la programación del modelo estructural, que automatiza el cálculo no lineal mediante el uso del programa OpenSeesPy. Para efectos de la herramienta, las unidades se estandarizarán de forma tal que el esfuerzo se mida en kilogramos por centímetro cuadrado, la longitud en metros y la fuerza en tonelada fuerza.

Por último, se llevará a cabo la validación de la herramienta mediante la comparación de los resultados obtenidos con el software de análisis estructural XTRACT. Se evaluará el rendimiento de la herramienta utilizando diversos parámetros, entre los que se destacan el momento flector máximo y la ductilidad de la sección. Este proceso de validación garantizará la fiabilidad y eficacia de la herramienta en diferentes escenarios de aplicación.

En el uso de la herramienta, es importante tener en cuenta ciertas consideraciones inherentes al diseño. La herramienta está optimizada para zonas confinadas cuadradas dentro de una sección, pero permite la combinación de múltiples zonas para representar geometrías más complejas. Además, ofrece flexibilidad al usuario para incorporar diferentes geometrías mediante la unión de varias zonas confinadas. Sin embargo, es esencial tener en cuenta que, dentro de una sección, todas las zonas confinadas deben compartir propiedades mecánicas idénticas. Por otro lado, es importante señalar que la herramienta no contempla el efecto cortante en su análisis.

Finalmente, la herramienta proyecta el comportamiento mecánico de los materiales especificados por el usuario. Sin embargo, es importante considerar que la curva proyectada es una estimación de

comportamiento y no refleja en su totalidad dicho comportamiento. Esta representación gráfica se deriva de cálculos manuales orientados únicamente para este fin. Por otro lado, el programa hace referencia al modelo constitutivo de Kent-Scott-Park integrado en OpenSeesPy.

Antecedentes

Actualmente, existen diversos esfuerzos por generar programas de código abierto para facilitar la elaboración de diagramas momento-curvatura, en el ámbito internacional; Sánchez Becerra, K [1]. “Desarrollo de una herramienta computacional para conceptualización y elaboración de diagramas momento-curvatura en secciones de concreto reforzado” y Morales Ortiz, F [2]. “Desarrollo de una Herramienta Computacional para la Evaluación del Comportamiento No Lineal de Muros de Concreto Reforzado”; ambos desarrollados en la Escuela Colombiana de Ingeniería Julio Garavito. Estas publicaciones pretenden contribuir en el estudio del comportamiento inelástico de los muros estructurales, sin embargo, se limitan únicamente al análisis de elementos con secciones rectangulares.

A través de una revisión bibliográfica, se identificaron softwares de tipo comercial como SAP 2000, ETABS, Robot Structural, Xtract y Exacad, que, si bien es cierto, permiten realizar una evaluación del comportamiento inelástico de muros con geometría compleja, estos requieren de licencias de alto costo y puede presentar dificultades para la creación de secciones en cuanto a la determinación de la geometría y la disposición del refuerzo. Así mismo, la implementación de este tipo de elementos estructurales dentro de un modelo de análisis de una edificación puede llegar a representar un alto consumo de recursos, por lo que, el desarrollo de estas herramientas le propicia al usuario una alternativa para abordar los análisis estructurales de una forma práctica, simple y con menor demanda de insumos computacionales.

Capítulo 1: Marco teórico

Para el desarrollo de una herramienta computacional que favorezca la conceptualización y visualización de diagramas momento-curvatura de secciones complejas es necesario definir algunos fundamentos teóricos. Inicialmente, se abordan conceptos básicos en relación con el diseño y programación del software. Posteriormente, se presentan aspectos asociados al comportamiento no lineal y el modelado de las secciones, las cuales se abordaron por medio del método de fibras aplicado en el programa estructural Openseespy. Finalmente, se presentan los modelos constitutivos utilizados para la representación del comportamiento del concreto confinado e inconfinado, así como el acero de refuerzo.

1.1 Programa

“Un programa es una secuencia de instrucciones que especifica cómo alcanzar un resultado” [3]. En este contexto, una herramienta computacional viene a ser la consecuencia a un conjunto de indicaciones que un computador interpreta para realizar una determinada tarea. Generalmente, los softwares son diseñados con el fin de facilitar, automatizar y mejorar los procesos que de alguna forma serían difíciles de abordar manualmente por una persona, o bien, son propensos al error humano.

Para desarrollar un sistema computacional es necesario utilizar lenguajes de programación que le permitan a la computadora procesar o entender los diferentes comandos establecidos por el programador, para posteriormente ser ejecutados.

1.2 Lenguaje de Programación

“El lenguaje de programación es un conjunto que contiene elementos gramaticales y semánticos que sirven para definir estructuras válidas que le dan solución a un problema” [4]. Los lenguajes de programación se caracterizan por ser precisos y claros, es decir, cada comando tiene un único significado. Las sintaxis y semánticas propias de cada lenguaje definen la forma de ordenar y plasmar las instrucciones que definen un programa, con el fin de que el computador pueda interpretar el código de forma correcta.

1.2.1 Python

El lenguaje de código Python fue creado en 1990 por el informático Guido van Rossum. Python se considera un lenguaje interpretado porque sus programas los ejecuta un intérprete, en otras palabras, no hay que compilar ni enlazar códigos. Así mismo, se define como un lenguaje multiparadigma, dado que, permite la aplicación de diferentes estilos de programación; orientada a objetos, estructurada y funcional [3].

“Una de las ventajas significativas de Python sobre otros lenguajes de programación es la rica colección de tipos de datos integrados” [5]. Python como lenguaje de código proporciona a los programadores las bases y herramientas para manipular y representar información en una forma simplificada en los programas desarrollados.

1.2.2 Anaconda

Anaconda es una plataforma de distribución de datos, fundada en el año 2012 por Peter Wang y Travis Oliphant, y surge ante la necesidad de incorporar el lenguaje de código Python al análisis de datos empresariales, que se transformaba rápidamente resultados de las tendencias tecnológicas emergentes [6]. Esta herramienta permite gestionar entornos y paquetes, debido a la gran cantidad de bibliotecas y herramientas que incluye.

1.2.3 Spyder

Spyder es un entorno científico gratuito y de código abierto escrito en Python, para el lenguaje de programación Python, y diseñado para científicos, ingenieros y analistas de datos. Ofrece a su vez funciones avanzadas para la edición, análisis y depuración de código [7]. Esta herramienta fue creada y desarrollada inicialmente por Pierre Raybaut en 2009, sin embargo, desde el 2012 ha sido mantenido y mejorado por un equipo de la comunidad científica de Python [8]. Spyder como entorno de trabajo, ofrece un editor de código, consolas interactivas y visores de variables, lo cual favorece el desarrollo de programas.

1.3 Programación orientada a objetos

A diferencia de la programación estructurada, la programación orientada a objetos (OOP), por sus siglas en inglés, permite una mayor organización y manipulación de los datos, a partir de la segmentación de los programas a través de objetos. La OOP, se caracteriza por los siguientes principios clave que se definen a continuación [9]:

- **Abstracción:** permite introducir clases de cierto nivel que incluso no existen en la realidad.
- **Encapsulamiento:** mecanismo utilizado para proteger los datos de un objeto.
- **Herencia:** mecanismo mediante el cual se implementa la relación de generalización.
- **Polimorfismo:** propiedad de las clases de compartir un objeto en común y responder de manera diferente a la misma acción.

“Un programa orientado a objetos está formado por una colección de objetos que interactúan entre sí para representar o solucionar algún problema” [3]. La estructura general característica de la OOP consiste en un conjunto de objetos, compuestos por datos y métodos, donde cada método está formado por una secuencia de instrucciones [10]. Ahora bien, cada colección de objetos se encuentra interconectada, favoreciendo así el desarrollo de sistemas robustos, dado que, la lógica detrás de la programación se centra en establecer la relación entre los diferentes componentes, como se muestra en la figura 1.

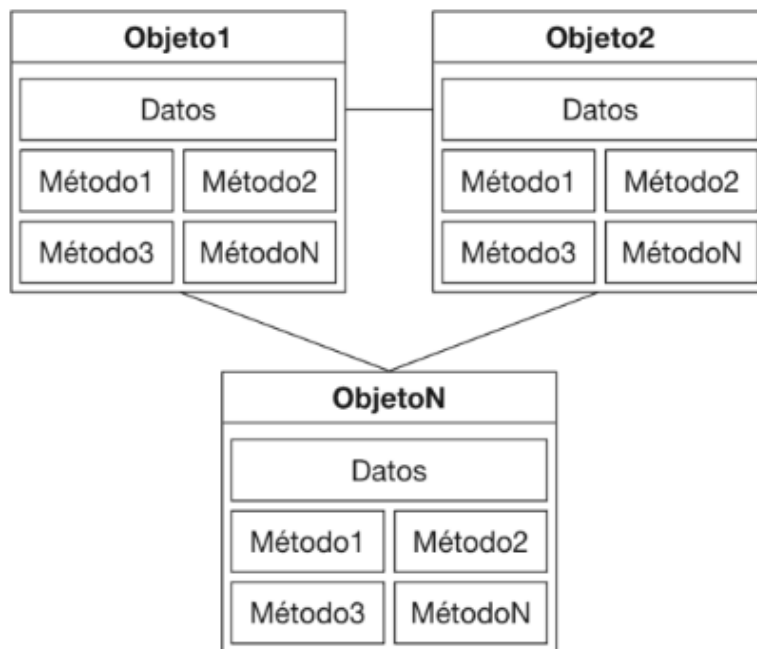


Fig 1. Estructura de la programación orientada a objetos. [10]

En otras palabras, la OOP consiste en un paradigma de la programación donde los “objetos” representan diferentes elementos. Por otro lado, cada objeto presenta características descritas como “datos” y pueden realizar acciones a través de “métodos”.

Parte fundamental de la OOP es la capacidad que ofrece la metodología para simplificar objetos a aspectos esenciales (abstracción), proteger datos y métodos dentro de los objetos (encapsulamiento), compartir características entre objetos (herencia) y propiciar una respuesta única de diferentes objetos ante la misma

acción (polimorfismo), estos principios clave benefician y facilitan la creación de programas organizados y mantenibles.

Ahora bien, una “clase” representa la unidad básica de la programación orientada a objetos en el lenguaje de código Python. Un objeto es una instancia de una clase. Una clase define el comportamiento de un objeto mediante las funciones del método [5]. Es decir, un objeto es la representación concreta de una clase y la relación entre ambos conceptos permite la creación y manipulación de instancias de una entidad con características y funcionalidades específicas, tal y como se ilustra en la figura 2.

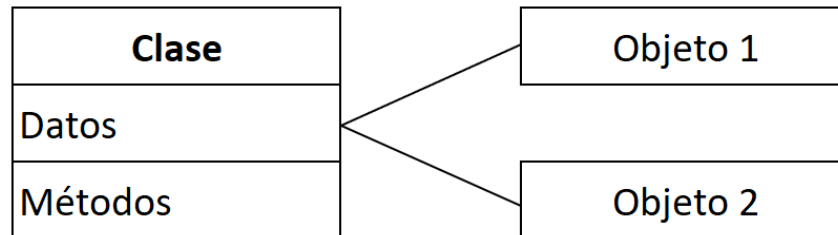


Fig. 2. Relación de objetos en una clase.

1.4 Interfaz Gráfica

Las interfaces de usuario son el medio de interacción entre las aplicaciones y las personas [11]. De manera análoga, las interfaces gráficas de usuario (GUI), por sus siglas en inglés, permiten operar las herramientas computacionales a través de elementos visuales. Estas a su vez, favorecen la usabilidad y funcionalidad de las aplicaciones, al impulsar al usuario hacia una interacción intuitiva con el software. En este contexto, la palabra intuitiva representa un concepto “familiar” [12]. Es decir, GUI por medio de sus componentes característicos, ha transformado el paradigma de interacción entre las personas y la tecnología, al hacer de esta un recurso accesible y amigable ante una gran variedad de población, independientemente de sus conocimientos técnicos.

Uno de los principales retos del diseño orientado a la interacción, consiste en desarrollar productos que cuenten con alta usabilidad, por esto se entiende que sean, fáciles de aprender, efectivos de usar y que le propicien una experiencia agradable al usuario [13]. Es decir, el desarrollo de GUI para herramientas computacionales no consiste únicamente en plasmar a través de elementos visuales las funciones que ofrece la aplicación, sino que, a su vez, es importante establecer un escenario “familiar” con el fin de proporcionar al usuario un panorama agradable para poder hacer uso del software.

1.5 Modelación Estructural

La modelación estructural se entiende como el proceso en el cual se realiza una idealización matemática que pretende representar la conducta real de una estructura [14]. Esta disciplina toma un papel relevante en el área de la ingeniería, dado que, permite predecir el comportamiento de estructuras, ante diferentes solicitaciones y condiciones de carga. Ahora bien, esta técnica utiliza modelos matemáticos y computacionales para representar y simular el comportamiento de los diferentes materiales que componen una determinada estructura.

1.5.1 Opensees

Opensees es un software de código abierto para desarrollar aplicaciones que simulan el desempeño de sistemas estructurales ante solicitaciones sísmicas [15]. Este programa cuyo nombre completo es “Open System for Earthquake Engineering Simulation”, y fue desarrollado a finales de la década de 1990, por investigadores en la Universidad de California Berkeley, y permite combinar modelos de elementos finitos y algoritmos de soluciones no lineales hacia sistemas de ecuaciones [16]. Dentro de los aspectos a resaltar que ofrece esta herramienta de análisis, es su capacidad para transformar modelos estructurales en sistemas globales de ecuaciones, que posteriormente resuelve con precisión numérica a través de diferentes algoritmos de solución, para obtener parámetros de importancia en la ingeniería estructural como deformaciones y esfuerzos internos en los elementos, es decir, transforma ecuaciones complejas en resultados visuales y cuantitativos comprensibles.

1.5.1.1 Openseespy

Openseespy es un intérprete Python de Opensees, desarrollado por el investigador Minjie Zhu en la Universidad Estatal de Obregón [17]. Esta interfaz de programación de aplicaciones destaca dentro de los programas de análisis estructural debido a la accesibilidad y capacidad para combinar la potencia de Opensees con la facilidad y los complementos que ofrece el Python como lenguaje de programación.

1.5.2 Método de Fibras

El método de fibras para el análisis no lineal de secciones transversales de elementos estructurales es un enfoque empleado en la ingeniería estructural en el cual la sección se divide en pequeños segmentos, a los cuales se les asigna características físicas particulares, y la interacción entre ellas permite conocer el comportamiento mecánico de la sección [18]. Cada una de las fibras, se caracteriza por su posición, área, material y estado de esfuerzo y deformación, dado que, estas se consideran independientes y se encuentran sometidas a diferentes niveles de carga.

Esta técnica, se basa en conceptos relacionados a la mecánica de materiales y análisis estructural, ya que es posible asignar diferentes niveles de rigidez a cada una de las fibras, lo cual permite comprender con detalle la respuesta de una sección de un elemento estructural ante solicitaciones de cargas no lineales como las que experimentan las edificaciones con la presencia de sismos o sobrecargas. Es importante destacar que, este método está sujeto a hipótesis donde únicamente interactúan fenómenos en una dirección, que coincide con la del momento.

Ahora bien, la no linealidad de las secciones se determina a partir de la caracterización mecánica de cada una de las fibras, donde a estas se les asigna un material cuyo comportamiento se conoce tanto en el rango elástico como inelástico, lo cual permite determinar el comportamiento no lineal de una determinada sección. En la figura 3, se ilustra la subdivisión en fibras de una sección, donde a cada una de estas se les asigna un material característico.

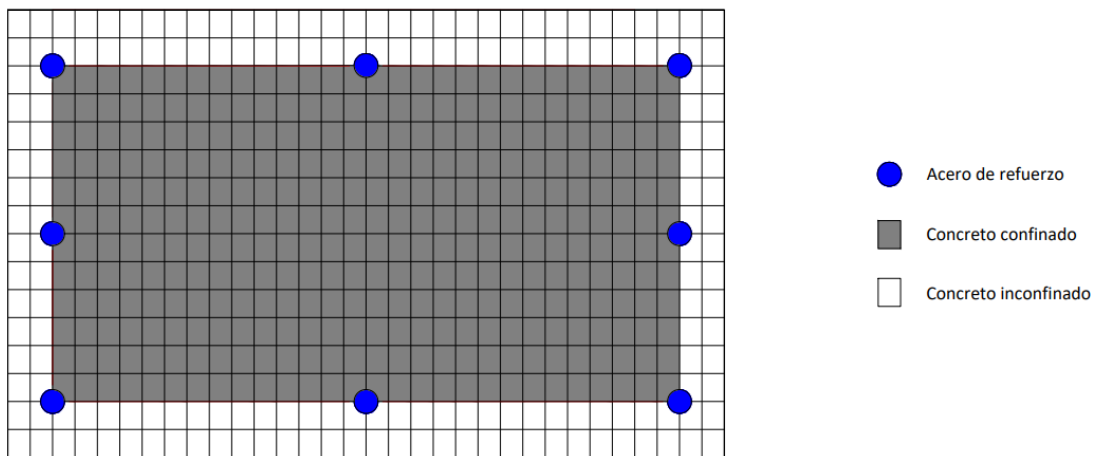


Fig. 3. Método de fibras.

1.5.3 Modelos Constitutivos

Los modelos constitutivos constan de expresiones matemáticas que expresan la relación esfuerzo-deformación de un determinado material [19]. Estos modelos se fundamentan en principios físicos y matemáticos que permiten representar el comportamiento de los materiales ante diferentes condiciones de carga. Así mismo, un modelo constitutivo debe propiciar la información necesaria para modelar en un escenario detallado y realista el comportamiento elástico e inelástico de un material.

1.5.3.1 Modelo Kent-Scott-Park

Este modelo constitutivo específico, conocido como Kent-Scott-Park, se caracteriza por su capacidad para modelar el comportamiento del concreto en condiciones uniaxiales. Introduce una rigidez de descarga/recarga degradada, lo cual significa que la rigidez disminuye después de la descarga y aumenta nuevamente durante la recarga. Además, no tiene en cuenta la resistencia a la tracción, lo que refleja su enfoque en comportamientos puramente compresivos del concreto [20]. La relación típica de tensión-deformación histerética para el material se muestra en la figura 4.

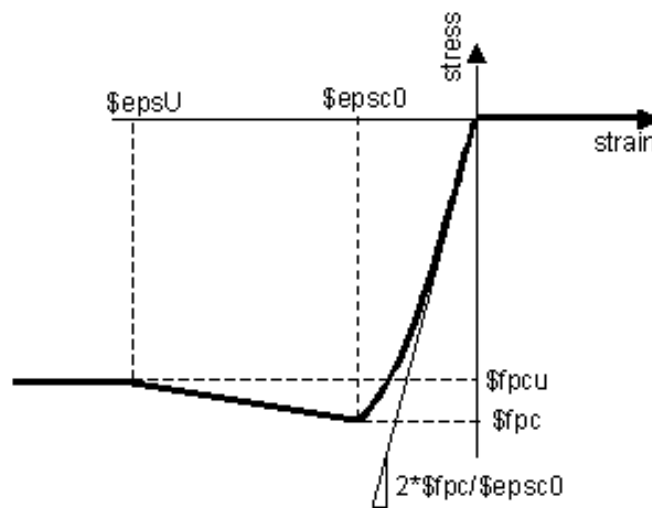


Fig. 4. Relación esfuerzo-deformación histerética para el material [15].

Donde:

- f_{pc} : Resistencia a la compresión del concreto a los 28 días.
- f_{pcu} : Resistencia al aplastamiento del concreto.
- ϵ_{psc0} : Deformación del concreto a la máxima resistencia.
- ϵ_{psU} : Deformación del concreto a la resistencia al aplastamiento del concreto.

1.5.3.2 Modelo de Mander, Priestley, Park (1988)

Esta representación matemática representa un modelo de tensión-deformación para el concreto sometido a carga compresiva uniaxial y confinado por refuerzo transversal [21]. Este modelo se ilustra en la figura 5.

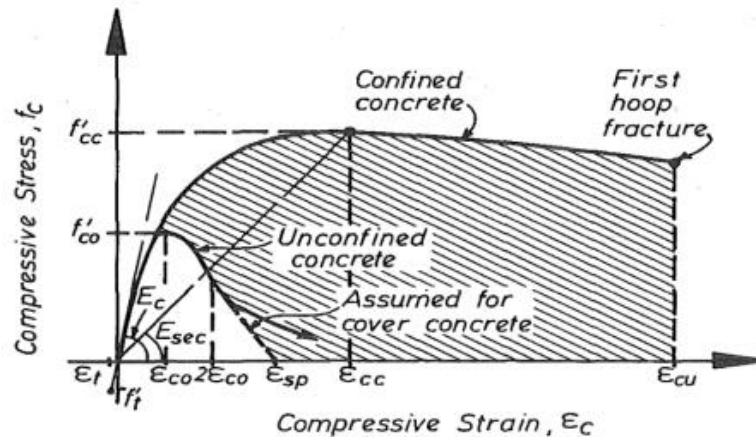


Fig. 5. Modelo de esfuerzo-deformación para una carga monótona de concreto confinado e inconfinado [21].

Donde:

- f'_{co} : Resistencia a la compresión del concreto inconfinado.
- f'_{cc} : Resistencia a la compresión del concreto confinado.
- ϵ_{co} : Deformación unitaria de la máxima resistencia a la compresión del concreto inconfinado.
- ϵ_{sp} : Deformación unitaria máxima del concreto inconfinado.
- ϵ_{cc} : Deformación unitaria de la máxima resistencia a la compresión del concreto confinado.
- ϵ_{cu} : Deformación unitaria máxima del concreto inconfinado.
- ϵ_t : Deformación unitaria de rotura por tensión del concreto.
- f'_t : Resistencia a la tensión del concreto confinado.
- E_c : Módulo de elasticidad del concreto.
- E_{sec} : Módulo de elasticidad tangente del concreto.

Según se observa en la figura 5, existe un incremento en la resistencia y la ductilidad del concreto debido al reforzamiento transversal de la sección. Esta mejora en la capacidad del material se propicia donde la tensión de confinamiento se ha desarrollado completamente debido a la acción del arqueado [21], según se muestra en la figura 6, para secciones con geometría rectangular.

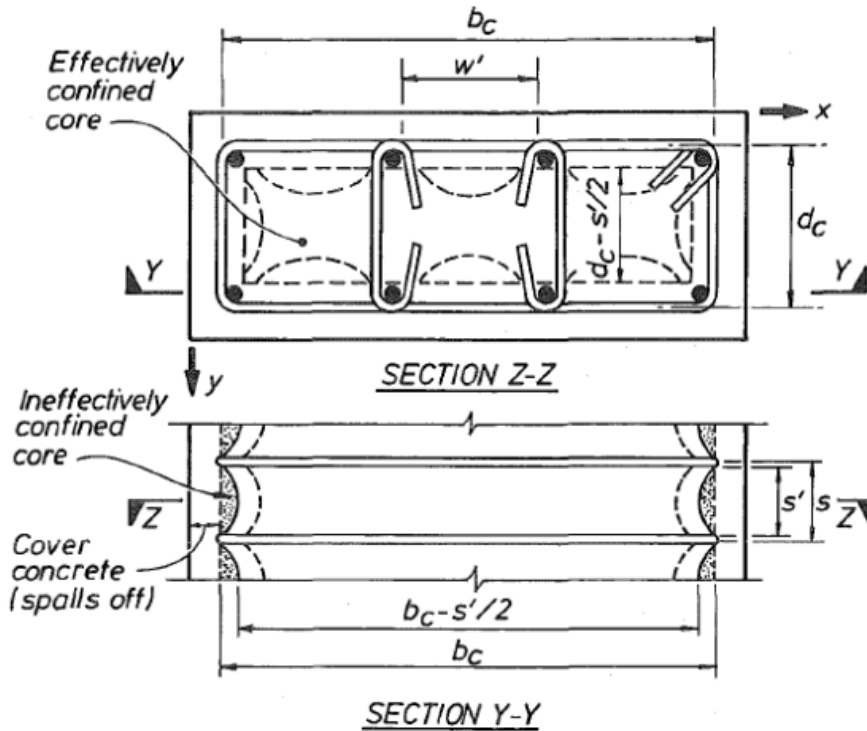


Fig. 6. Núcleo efectivamente confinado para refuerzo de acero de aro rectangular [21].

1.5.3.3 Modelo de Menegotto & Pinto (1973)

Este modelo constitutivo desarrollado en 1973, por Angelo M. Menegotto y Pedro E. Pinto, pretende describir el comportamiento del acero en el rango no lineal bajo cargas de tensión y compresión. Esta representación matemática es capaz de modelar tanto el endurecimiento cinemático e isotrópico, así como el efecto Bauschinger, lo que propicia una simulación precisa de los resultados experimentales [22]. La respuesta no lineal del acero se describe a partir de la siguiente ecuación [23].

$$\sigma = (1 - b) \cdot \frac{\varepsilon}{1 + \varepsilon^R} + (b \cdot \varepsilon)$$

Donde:

b : Define la pendiente de la línea trabajadora.

R : Define la variación después de cada inversión.

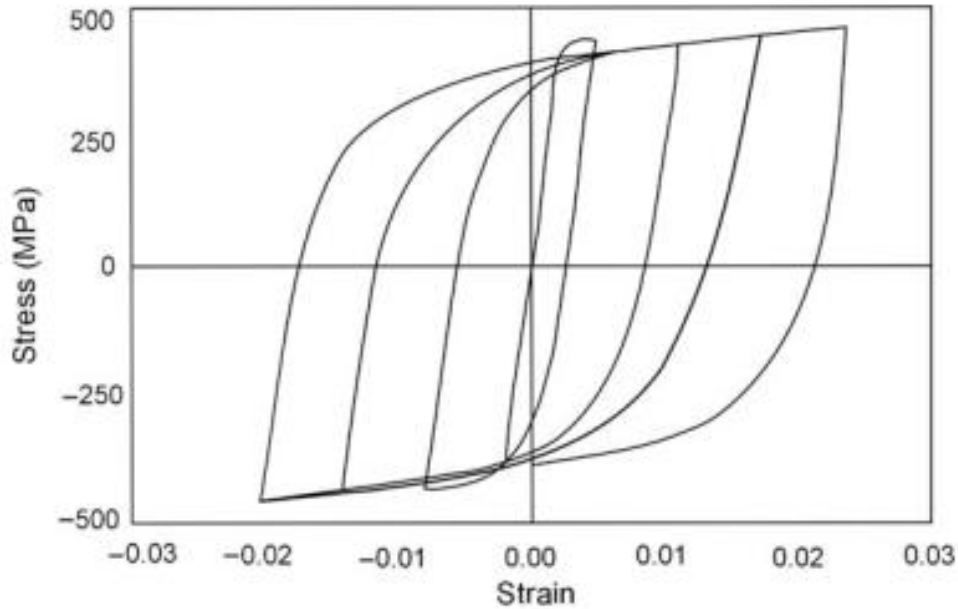


Fig. 7. Respuesta típica de esfuerzo-deformación cilíndrica [22].

En la figura 7, se ilustra una respuesta típica al comportamiento inelástico del acero, modelado a partir de lo dispuesto en el modelo de Menegotto & Pinto (1973), es importante destacar que el modelo asume una respuesta simétrica ante cargas de compresión y tensión [22].

1.6 Diagrama Momento Curvatura

El diagrama momento-curvatura es una representación gráfica que ilustra la relación entre el momento flector aplicado a una sección estructural y la curvatura resultante en esa sección. Este diagrama es fundamental para comprender el comportamiento estructural de un elemento bajo carga.

En el contexto del análisis estructural, el diagrama momento-curvatura proporciona información valiosa sobre la capacidad de deformación y la resistencia a la flexión de una sección. A medida que se aplica un momento a la sección, la curvatura resultante se representa en el eje de las ordenadas, mientras que el momento aplicado se representa en el eje de las abscisas.

Esta representación gráfica es de gran importancia en el diseño de estructuras ante cargas estáticas y dinámicas, ya que de forma rápida se visualiza que tan dúctil y resistente es un miembro [24]. La relación momento-curvatura, se encuentra en función de las características mecánicas de los materiales, la disposición de estos y la geometría de la sección.

El área bajo la curva momento-curvatura, representa la energía interna en la sección, es decir, la parte bajo la región elástica corresponde a la energía de deformación acumulada en el miembro, mientras que el área en la zona inelástica representa la energía disipada ante las deformaciones plástica [25].

Este tipo de representación gráfica es esencial para evaluar la capacidad resistente y la ductilidad de una sección frente a diferentes niveles de carga, lo que facilita la toma de decisiones en el diseño y la optimización de estructuras. En la figura 8, se presentan dos escenarios representados mediante gráficas. La gráfica A ilustra una sección con comportamiento dúctil. En contraste, la gráfica B muestra una sección frágil, evidenciada por una menor capacidad de disipación de energía.

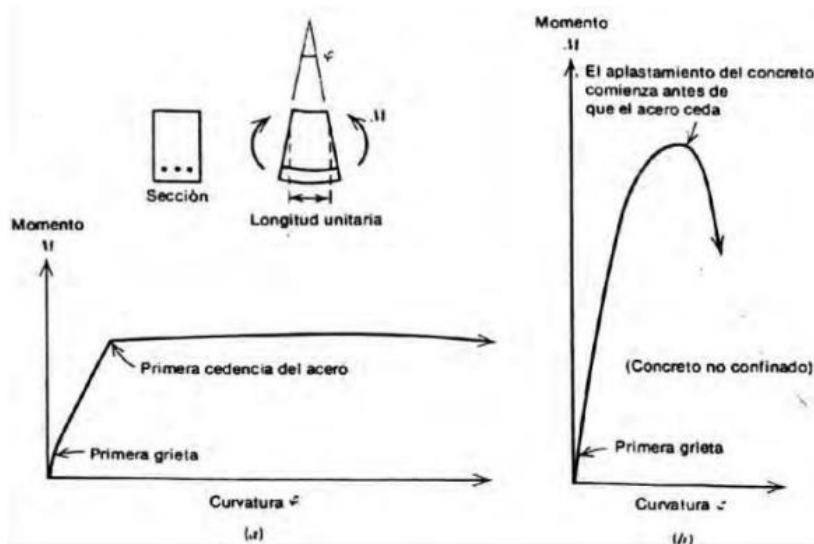


Fig. 8. Relación momento curvatura para secciones de viga simplemente reforzadas. A) sub reforzada. B) Sobre reforzada [26].

1.6.1 Estados límite del diagrama momento curvatura

1.6.1.1 Inicio del Agrietamiento

El inicio del agrietamiento se presenta cuando el concreto en la fibra extrema en tensión, alcanza su máxima resistencia a la tensión, por lo que, a raíz de la falla local se empiezan a experimentar fisuraciones en la sección.

1.6.1.2 Fluencia del acero

En este punto, la curva muestra una disminución en la pendiente, indicando que el acero ha alcanzado su límite elástico y ha entrado en la fase de deformación plástica. La fluencia del acero es un indicador crucial que muestra la transición final del comportamiento elástico al inelástico de la sección.

1.6.1.3 Agotamiento de la sección

El agotamiento de la sección representa el estado límite final en el diagrama momento-curvatura. En este punto, la curva alcanza su punto máximo y comienza a descender abruptamente, indicando que la sección ha alcanzado su capacidad máxima. En la figura 9, se muestran los estados límite del diagrama momento-curvatura, el último punto representa el estado en el que el concreto alcanza la máxima deformación a la compresión, o bien cuando el acero llega a la deformación de ruptura.

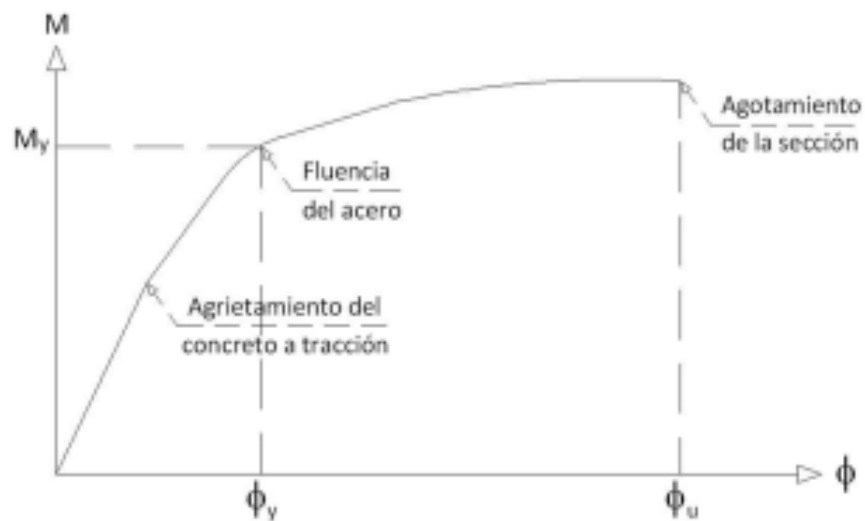


Fig. 9. Estados límite del diagrama momento-curvatura [27].

1.6.2 Ductilidad

La ductilidad en el contexto de un diagrama momento-curvatura se refiere a la capacidad de un elemento estructural para deformarse plásticamente antes de la falla. En términos más simples, es la medida de la

capacidad del miembro estructural para absorber energía incluso después de haber alcanzado la fluencia del acero.

En el diagrama momento-curvatura, la ductilidad se puede evaluar observando el área total bajo la curva hasta el punto de agotamiento de la sección. Cuanto mayor sea esta área, mayor será la ductilidad de la estructura. La ductilidad es un factor clave en el diseño sísmico y estructural, ya que estructuras más dúctiles tienden a absorber y disipar más energía durante eventos sísmicos o cargas inesperadas.

Una estructura que exhibe ductilidad proporciona ciertos beneficios, como la capacidad de soportar deformaciones considerables antes de la falla, facilitando la detección y reparación de daños antes de un colapso catastrófico. La ductilidad es, por lo tanto, un aspecto fundamental en la evaluación de la seguridad y el rendimiento estructural.

De acuerdo con las consideraciones establecidas en el capítulo 3 del Código Sísmico de Costa Rica 2010 revisión 2014 (CSCR 2010/14), en cualquier sistema sismorresistente todos sus elementos deben ser capaces de deformarse de manera compatible con los desplazamientos presentes durante la acción sísmica, sin que pierdan la capacidad de resistir las cargas gravitacionales presentes durante el sismo [28].

Capítulo 2: Metodología

2.1 Etapas de la investigación

El desarrollo del presente proyecto se realizó a partir de un proceso estructurado definido de acuerdo con los objetivos específicos planteados, este proceso consta de tres etapas las cuales se muestran en la figura 10.

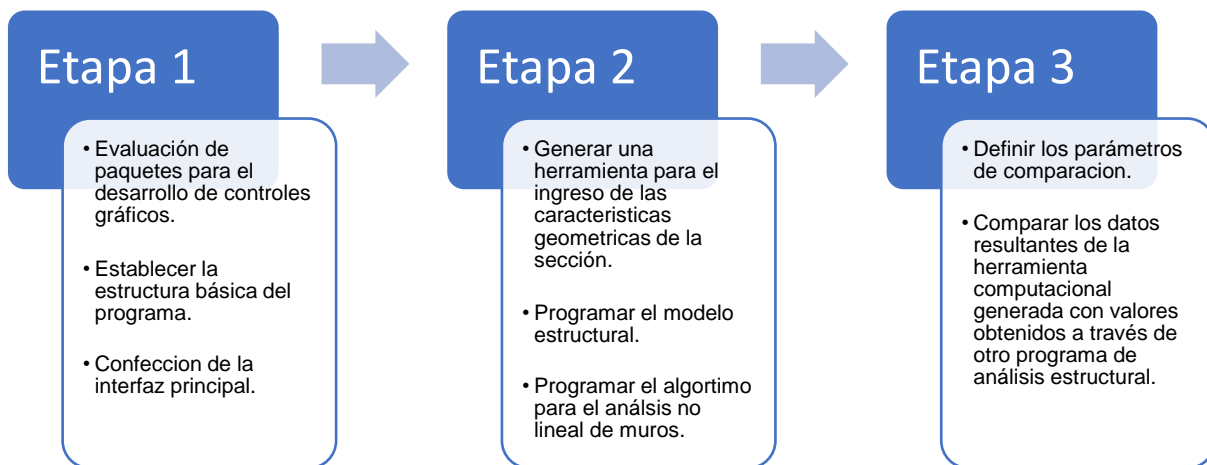


Fig. 10. Etapas de la metodología.

2.2 Etapa 1

2.2.1 Evaluación de controles gráficos

Se realizó un proceso en el cual se examinaron y evaluaron librerías o conjunto de componentes gráficos adaptados al lenguaje de programación Python. En este se revisaron factores como elementos gráficos disponibles, aspectos de funcionalidad, documentación, compatibilidad y licencias para operar. A continuación, se describen algunas de las características principales que detallan a las bibliotecas evaluadas.

2.2.1.1 PYQT

PYQT es un sistema desarrollado por la firma británica Riverbank Computing Limited, y representa un conjunto de enlaces de Python del marco de la aplicación QT, la cual contiene herramientas GUI. Esta aplicación cuenta con clases que crean un mecanismo signal/slot para comunicarse entre objetos, lo que facilita la creación de componentes de software reutilizables [29]. El desarrollo de una interfaz gráfica de usuario se puede realizar a través de un programa llamado “Qt designer” el cual permite componer una ventana de interfaz personalizada con widgets, sin el uso de líneas de programación, dado que, la herramienta es capaz de generar código Python, de acuerdo con el diseño plasmado por el usuario.

2.2.1.2 Tkinter

Tkinter es un recubrimiento del lenguaje de programación interpretado TCL, que permite utilizar la interfaz Tk desde Python. Es decir, se trata de una interfaz de programación de aplicaciones, escrito por Fredrik Lundh. Tkinter representa el paquete estándar de Python, por lo que, se encuentra disponible una vez instalado el propio entorno. [11] Esta herramienta no es una biblioteca única, se trata de unos pocos módulos distintivos entre sí, separados por funcionalidades y documentación propia [30].

2.2.1.3 Kivy

Kivy es una librería de código abierto de Python para el desarrollo rápido de aplicaciones equipadas con novedosas interfaces de usuario, como aplicaciones multitáctiles [31]. Esta herramienta cuenta con un lenguaje de descripción de interfaz, llamado Kivy Language (KV), el cual permite crear los árboles de widgets de forma declarativa y vincular las propiedades de estos elementos entre sí, agilizando los cambios y creación de prototipos [32]. En esta biblioteca los gráficos son procesados a través de OpenGL, lo que permite desarrollar interfaces gráficas de usuario funcionales en una gran variedad de plataformas, y estas van a presentar una apariencia similar independientemente del sistema operativo que se ejecute [11].

2.2.1.4 Comparación

A través de la información recolectada, así como pequeños ejemplos ejecutados al momento del desarrollo del ejercicio, se establecieron diferentes niveles de desempeño descritos en la Tabla I.

TABLA I
NIVELES DE DESPEÑO

Criterio	Nivel de desempeño		
	2	1	0
Funcionalidad	Ofrece facilidad para integrar elementos de interfaz como botones, listas, cuadros de texto, formularios, elementos interactivos, entre otros.	Presenta dificultades para integrar elementos de interfaz como botones, listas, cuadros de texto, formularios, elementos interactivos, entre otros.	Ofrece poca variedad de elementos visuales
Documentación	Existe documentación actualizada en las páginas web de las organizaciones desarrolladoras, sobre ejemplos y problemas comunes, en la utilización de la herramienta.	Existe documentación desactualizada en las páginas web de las organizaciones desarrolladoras, sobre ejemplos y problemas comunes, en la utilización de la herramienta.	Únicamente se encuentra información referente a la utilización de la herramienta en blogs o video tutoriales.
Adaptabilidad a la herramienta	Presenta elementos visuales con apariencia profesional y ofrece facilidad para configurar ventanas emergentes a la aplicación principal.	Ofrece elementos visuales con apariencia profesional, sin embargo, presenta dificultades para adaptar ventanas emergentes a la aplicación principal.	Ofrece elementos visuales con apariencia poco atractiva.
Recursos computacionales	Bajo consumo de recursos computacionales, dado que ofrece soluciones simples a los diferentes requerimientos.	Moderado consumos de recursos computacionales.	Alto consumo de recursos computacionales debido a la complejidad de funcionalidades.
Licencias para operar	Opera bajo una licencia de código abierto.	Opera bajo una licencia por características, es decir, hay un costo asociado por el uso de algunas funcionalidades de la herramienta.	Opera bajo una licencia de uso comercial.

Nota: Se especifican los criterios asociados a los niveles a desempeño establecidos para la comparación de controles gráficos.

2.2.2 Estructura básica del programa

En el desarrollo de la herramienta computacional para la conceptualización de diagramas momentos curvatura de muros con geometría compleja, se utilizó el lenguaje de programación Python desde el entorno científico Spyder. Ahora bien, se empleó un estilo de programación orientado a objetos, con el fin de definir diferentes clases u objetos que responden a diversos módulos dentro de la aplicación. La estructura básica del programa que se siguió se ilustra en la figura 11.

```

# estatutos "import"

# comentarios característicos de la Clase 1
# definición de estatutos Clase 1
    # definición de estatutos función 1.1
    .
    .
    # definición de estatuto función 1.n
    .
# comentarios característicos de la Clase n
# definición de estatutos Clase n
    # definición de estatutos función n.1
    .
    .
    # definición de estatuto función n.n

# definición de estatutos programa principal (main)

```

Fig. 11. Estructura básica del programa.

Desde una perspectiva más general, el diseño y desarrollo del programa se realizó a partir de un enfoque organizativo, donde se generaron diferentes scripts los cuales presentan una organización estratégica que permite ordenar y asegurar que cada elemento cumpla con su función designada, y se integre armoniosamente con los demás códigos para lograr el buen funcionamiento la herramienta desarrollada.

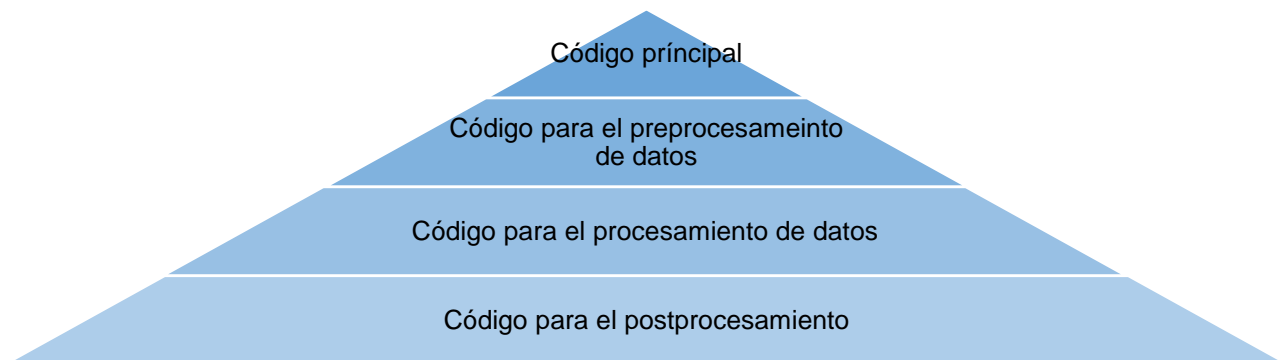


Fig. 12. Organización jerárquica de los scripts.

2.2.3 Confección de la interfaz principal

La interfaz gráfica de usuario fue desarrollada mediante la biblioteca de enlace PyQt5. En primera instancia, se generó una interfaz principal por medio de la aplicación Qt designer, esta ventana de contenido visual, pretende ser un bosquejo que contiene los elementos principales con los cuales va a interactuar el usuario, al hacer uso

de la herramienta computacional. El objetivo principal de la ejecución de este procedimiento consiste en plasmar el diseño principal de forma rápida y ágil, previo al inicio de la programación del algoritmo que orienta al desarrollo del software. Así mismo, a través de un proceso iterativo, se pretende alcanzar un diseño con alta usabilidad para establecer un software intuitivo ante los usuarios.

La principal consideración contemplada para la confección de la interfaz principal consiste en que el usuario deba seguir un flujo constante en el ingreso de datos y el uso de la herramienta, para minimizar el error y favorecer la usabilidad, según se muestra en la figura 13.

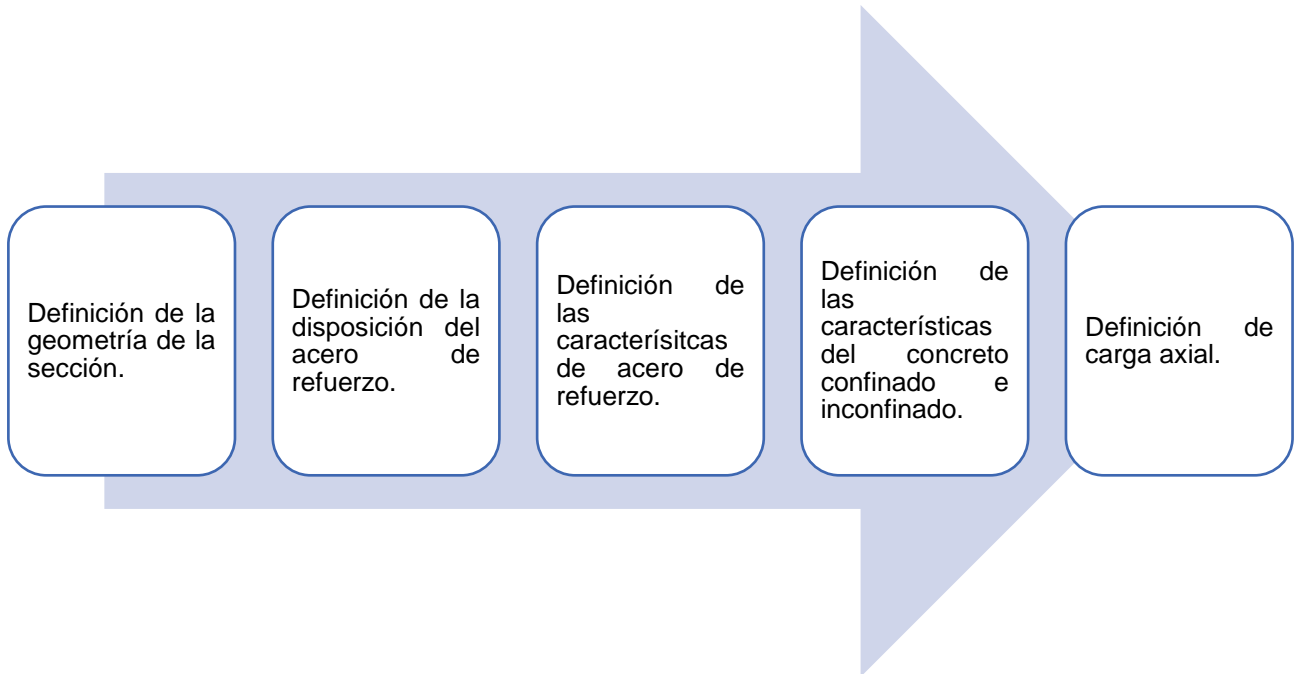


Fig. 13. Flujo continuo de uso de la herramienta.

2.3 Etapa 2

2.3.1 Conceptualización del modelo geométrico

Para efectos de la definición geométrica de cada sección, se generó un script el cual interpreta una lista de coordenadas en un espacio bidimensional, con el fin de establecer un polígono que conceptualiza la geometría de la sección. El ingreso de los datos se realiza a partir de una interfaz de usuario como la que se muestra en la figura 14.

	X (m)	Y (m)
Vértice 01		
Vértice 02		
Vértice 03		
Vértice 04		

Fig. 14. Ingreso de la geometría de la sección.

De manera análoga, se estableció un script, que ofrece una interfaz de usuario que permite ingresar datos relacionados con la disposición del refuerzo de acero, incluyendo la posición de cada varilla en un plano bidimensional y la respectiva área característica. Este enfoque modular, permite la inclusión de diferentes disposiciones y tamaños de barras, brindando flexibilidad en la conceptualización del modelo, la interfaz desarrollada para dicho fin se ilustra en la figura 15.

	X (m)	Y (m)	Area (cm2)
Varilla 01			
Varilla 02	0	0	0
Varilla 03	0	0	0
Varilla 04	0	0	0

Fig. 15. Ingreso de las características del refuerzo de acero.

2.3.2 Definición del modelo estructural

El modelo estructural se definió en programa OpenSeesPy, mediante el lenguaje de programación Python en el entorno científico SPYDER. El mismo fue modelado en un espacio bidimensional, con tres grados de libertad. Así mismo, se establecieron dos nodos ubicados en la misma coordenada (X, Y) y conectados por un elemento de longitud cero, con el fin de realizar un análisis por desplazamiento en dicho elemento, y con ello conocer la relación esfuerzo-deformación ante la aplicación de cargas.

Por otro lado, se definió una sección de tipo fibra, donde cada una de las fibras contine un material, un área y una posición específica en el un plano X, Y. Dentro del código de programación donde se establece el modelo estructural, se implementaron funciones específicas para interpretar las matrices de datos generadas por otro script. Este último script se encarga de descomponer la sección en fibras. El propósito de esta estructura es permitir que el programa asigne de manera adecuada las características individuales de cada fibra a una sección particular.

2.3.2.1 Método de fibras

Para la aplicación del método de fibras se creó un código de programación, que procesa los datos de la sección ingresados por el usuario para la generación del modelo estructural. En este contexto, su función principal consta en dividir la sección en una serie de fibras más pequeñas, lo que se logra a través de la subdivisión del polígono en un número definido de divisiones tanto en el eje X como en el eje Y.

La ejecución del script inicia con la recolección e interpretación de los datos característicos de la sección ingresados por el usuario. Posteriormente, se lleva a cabo la subdivisión, generando así una malla de fibras más pequeñas que abarcan la totalidad de la sección original, es importante resaltar que, el código separa las fibras generadas dentro de las zonas confinadas definidas por el usuario y las almacena en matrices separadas. Por último, el script genera diferentes matrices que contienen información específica de cada fibra. Estos datos incluyen las coordenadas del centro de cada fibra y su área característica.

Finalmente, las matrices generadas por este script se integran con el código que define el modelo estructural, para que este pueda utilizar la información de manera efectiva, asegurando un representación precisa y fiel del comportamiento estructural de la sección.

Nomenclatura de las matrices generadas

- **matrix_1**: Lista de coordenadas X, Y del vértice inferior izquierdo y superior derecho de cada fibra dentro de las áreas confinadas.

- **matrix_2:** Área de cada fibra dentro de las áreas confinadas.
- **matrix_3:** Lista de coordenadas X, Y del vértice inferior izquierdo y superior derecho de cada fibra dentro de las áreas confinadas.
- **matrix_4:** Área de cada fibra dentro de las áreas confinadas.
- **matrix_5:** Lista de coordenadas X, Y del centro de cada fibra dentro de las áreas confinadas.
- **matrix_5:** Lista de coordenadas X, Y del centro de cada fibra dentro de las áreas confinadas.

2.3.2.2 Definición de cargas

En primer lugar, se estableció una carga axial constante aplicada en el nodo 2 del modelo estructural. La magnitud de esta es establecida por el usuario, y se define mediante la creación de un patrón de carga y el uso de la serie de tiempo constante que se encuentra dentro de las funciones de Openseespy. La carga axial constante busca proporcionar las condiciones iniciales para el análisis estructural, generando una respuesta inicial del sistema bajo la acción de esta carga constante. El comportamiento del factor de carga a lo largo del tiempo se muestra en la figura 16.

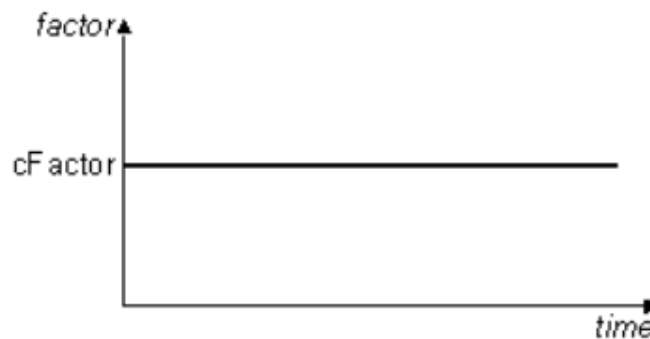


Fig. 16. Serie de tiempo constante.

Posteriormente, se introdujo un momento unitario en el nodo 2 que crece linealmente con el tiempo. Este momento unitario fue definido utilizando un patrón de carga y una serie de tiempo lineal. Este enfoque permite incrementar gradualmente el momento aplicado al nodo 2 a medida que avanza el análisis estructural, según se ilustra en la figura 17.

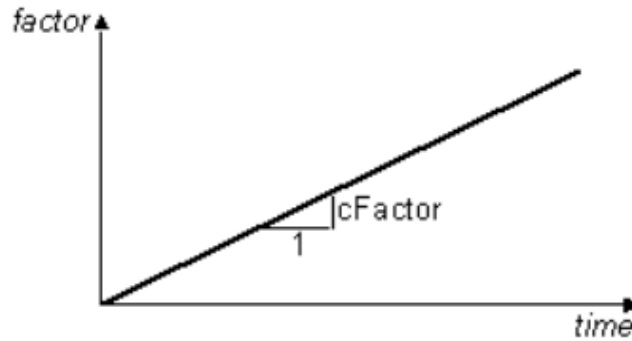


Fig. 17. Serie de tiempo lineal.

2.3.2.3 Propiedades mecánicas de los materiales

Dentro de la interfaz principal se añadieron elementos visuales con el fin de que el usuario inserte las propiedades mecánicas de los diferentes materiales que componen la sección en estudio. Estos valores son, almacenados de forma integral, para posteriormente ser utilizados en el análisis estructural. La herramienta contempla la utilización de los modelos constitutivos de Menegotto y Pinto para el acero de refuerzo, los cuales se implementan por medio del comando “SteelMPF” de Openseespy.

De manera análoga, se creó el espacio para que el usuario especifique las propiedades mecánicas del concreto confinado e inconfinado. Para el modelado del comportamiento no lineal del concreto, se emplea el modelo constitutivo de Kent-Scott-Park, implementado a través del comando “Concrete01” en Openseespy.

Ahora bien, la herramienta se diseñó con el fin de que el usuario ingrese las propiedades mecánicas del concreto confinado de forma manual. Sin embargo, dentro de la herramienta se implementó un botón nombrado “Calculadora Mander” que presenta como fin, calcular el aumento en resistencia y ductilidad que experimenta el concreto mediante la incorporación de acero de refuerzo, la interfaz que permite el uso de esta herramienta se muestra en la figura 18.

The image shows a software window titled "Calculadora Mander" with the following input fields and labels:

- f_c:** [input field] kgf/cm². Label: f_c: Resistencia a la compresión del concreto a los 28 días.
- As_x:** [input field] cm². Label: As_x: Área de acero de refuerzo transversal paralelo al eje X.
- As_y:** [input field] cm². Label: As_y: Área de acero de refuerzo transversal paralelo al eje Y.
- f_{yt}:** [input field] kgf/cm². Label: f_{yt}: Resistencia a la fluencia del acero de refuerzo transversal.
- s:** [input field] cm. Label: s: [input field] cm.
- b_{cx}:** [input field] cm. Label: b_{cx}: [input field] cm.
- b_{cy}:** [input field] cm. Label: b_{cy}: [input field] cm.

At the bottom, there are two buttons: "Calcular" and "Ayuda".

Fig. 18. Serie de tiempo lineal.

El confinamiento se calculó de acuerdo con lo propuesto en [33], que consiste en el modelo constitutivo de Mander, Priestley & Park, con una actualización en el método de cálculo de los coeficientes a partir de una gran cantidad de experimentos [34]. La ecuación del modelo que describe la resistencia de la compresión confinada de secciones rectangulares y en unidades de kg/cm², se muestra a continuación.

$$f'_{cc} = f'_c + k_1 \cdot f_{le}$$

Donde,

- f'_{cc} : Resistencia a la compresión confinada.
- f'_c : Resistencia a la compresión inconfiada.
- k_1 : Coeficiente igual a $9.94 \times f_{le}^{-0.17}$
- f_{le} : Presión uniforme equivalente de confinamiento.

Por otro lado, la ecuación del modelo que describe la deformación asociada a la resistencia a la compresión confinada del concreto se muestra a continuación.

$$\varepsilon_1 = \varepsilon'_c \cdot \left(1 + \frac{5 \cdot k_1 \cdot f_{le}}{f'_c} \right)$$

Donde,

- ε_1 : Deformación asociada a la resistencia a la compresión confinada.
- ε_c : Deformación asociada a la resistencia a la compresión inconfiada.

La presión efectiva se calcula a partir de la siguiente ecuación.

$$f_{le} = \frac{f_{lex} \cdot bcx + f_{ley} \cdot bcy}{bcx + bcy}$$

Donde,

- f_{le} : Presión efectiva.
- bcx : Dimensión del rectángulo de confinamiento, paralelo al eje X.
- bcy : Dimensión del rectángulo de confinamiento, paralelo al eje y.
- f_{lex} : Presión efectiva sobre bcx .
- f_{ley} : Presión efectiva sobre bcy .

Finalmente, las presiones efectivas sobre bcx y bcy , que se ilustran en la figura 19, se calculan a partir de la siguiente ecuación.

$$f_{lex}, f_{ley} = \frac{\Sigma(A_s \cdot f_y)}{s \cdot (bcx, bcy)}$$

Donde,

- A_s : Área del refuerzo transversal.
- f_y : Resistencia a la fluencia del acero transversal.
- s : Separación de los aros.

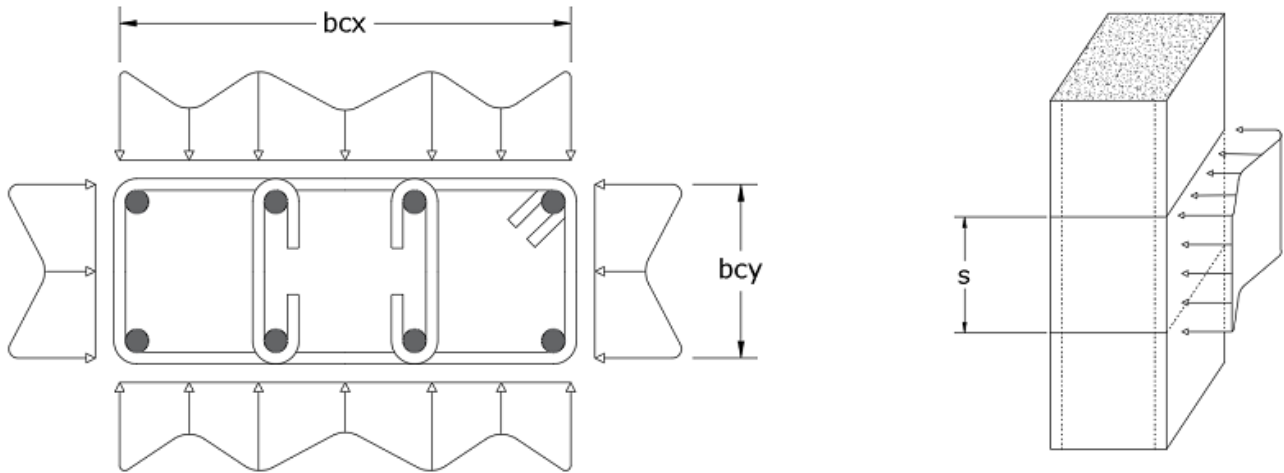


Fig. 19. Distribución de presiones lateral en elementos rectangulares.

2.3.3 Análisis por desplazamiento

Como parte del análisis estructural, se implementó una estrategia específica para evaluar el comportamiento del modelo en términos de desplazamiento. Este análisis se ejecuta a través de Openseespy mediante el uso de un integrador de control de desplazamiento en el nodo 2 del modelo estructural.

El análisis se llevó a cabo con un número de incrementos igual a 100. Lo que implica que el análisis estructural se divide en 100 pasos o intervalos, cada uno de los cuales se caracteriza por un pequeño incremento en el tiempo. Los incrementos se realizan gradualmente, permitiendo un monitoreo detallado del comportamiento de la estructura a medida que evoluciona en términos de desplazamiento a lo largo del análisis.

La elección de un número significativo de 100 incrementos busca favorecer la precisión del análisis y proporcionar una representación más detallada de la respuesta estructural. No obstante, a través de este número de incrementos también se optimiza la eficiencia computacional de la herramienta, equilibrando la precisión del análisis y la carga computacional asociada.

2.3.4 Diagrama momento-curvatura

El procedimiento utilizado para calcular la curva del diagrama momento-curvatura se resume a continuación.

1. Se recopilan y almacenan los datos ingresados por el usuario, que incluyen las dimensiones geométricas de la sección, y la posición y propiedades de los materiales que componen la estructura.

2. A través del script "Section_Designer.py" se implementa el método de fibras para dividir la sección en una red de pequeñas fibras. Lo que implica la subdivisión de la sección en un número definido de divisiones en los ejes X, Y.
3. Se generan matrices con información detallada de cada fibra, incluyendo las coordenadas de su centro y área específica. Estas matrices se utilizan para representar cada fibra en el modelo estructural.
4. Se desarrolla un modelo estructural utilizando las matrices generadas, donde cada fibra se asigna con sus propiedades específicas. Este modelo se construye contemplando la disposición y características de las fibras.
5. Se ejecuta un análisis por desplazamiento en el nodo al cual se le asignó la sección especificada por el usuario, con 100 incrementos.
6. Se almacenan los datos de esfuerzo-deformación.
7. Se grafica el diagrama momento-curvatura.

2.4 Etapa 3

2.4.1 Definición de modelos representativos

Se seleccionaron tres modelos representativos, cada uno caracterizado por diferencias en geometría, propiedades mecánicas de los materiales y disposición de los elementos estructurales. Estos modelos sirvieron como casos de estudio para evaluar la capacidad de la herramienta desarrollada ante diferentes escenarios.

2.4.1.1 Geometría

La geometría y disposición del acero de refuerzo de las secciones analizadas se muestra de manera visual en las figuras 20, 21, 22.

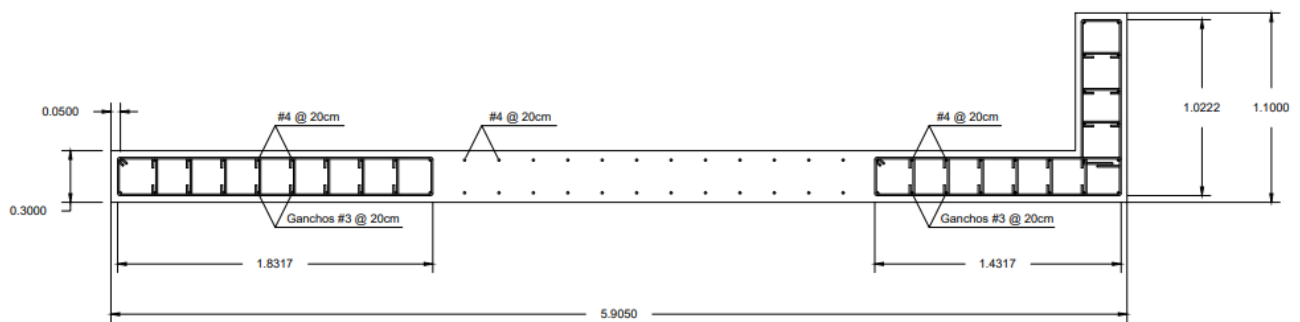


Fig. 20. Modelo 1.

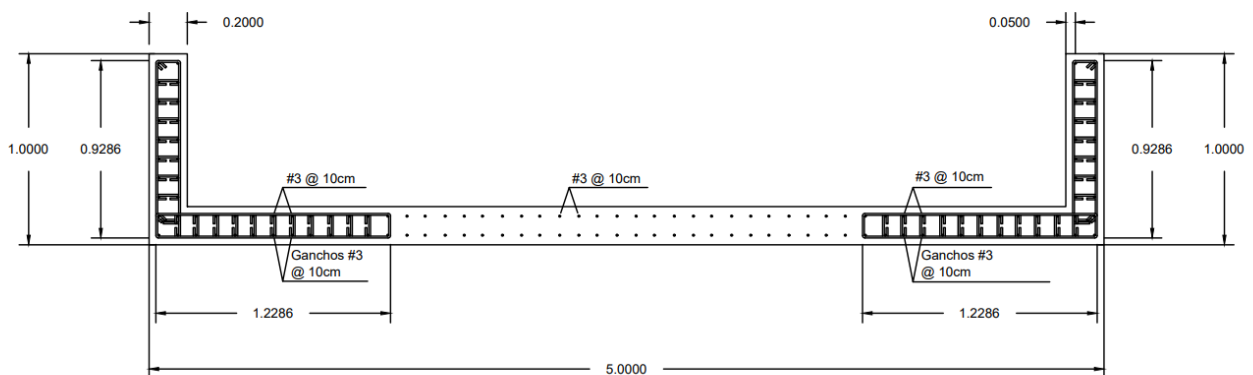


Fig. 21. Modelo 2.

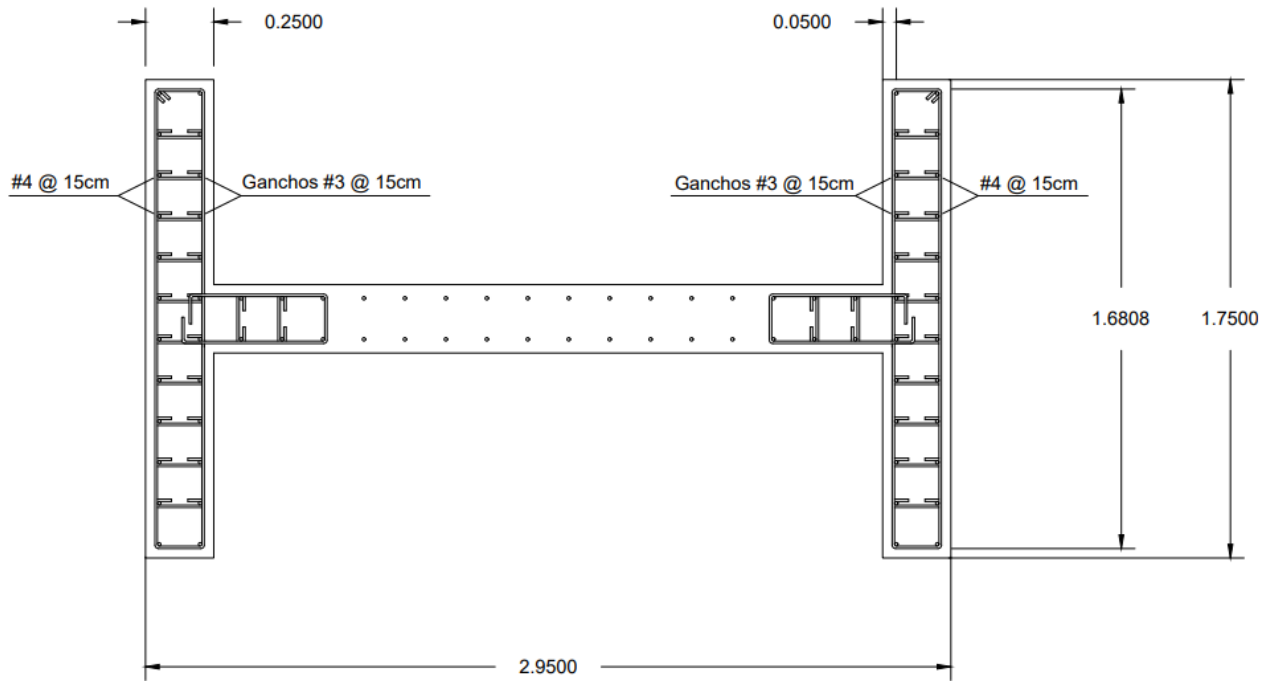


Fig. 22. Modelo 3.

2.4.1.2 Propiedades mecánicas de los materiales

En las tablas II, III, IV, V y VI, es posible observar el valor asignado para cada uno de los parámetros en el concreto confinado, inconfinado y acero de refuerzo.

TABLA II
CARACTERÍSTICAS DEL CONCRETO PARA MODELO 1

Parámetro	Descripción	Inconfinado	Confinado
f_{pc}	Resistencia a la compresión a los 28 días.	387 kgf/cm ²	280 kgf/cm ²
f_{pcu}	Resistencia al aplastamiento del concreto	387 kgf/cm ²	0 kgf/cm ²
ϵ_{psu}	Deformación en el punto de resistencia a la compresión a los 28 días.	0.002	0.002
ϵ_{psc}	Deformación en el punto de resistencia al aplastamiento.	0.006	0.004

Nota: Se muestran los datos característicos del concreto para el modelo 1.

TABLA III

CARACTERÍSTICAS DEL CONCRETO PARA MODELO 2

Parámetro	Descripción	Inconfinado	Confinado
fpc	Resistencia a la compresión a los 28 días.	450 kgf/cm ²	250 kgf/cm ²
fpcu	Resistencia al aplastamiento del concreto	450 kgf/cm ²	0 kgf/cm ²
epsu	Deformación en el punto de resistencia a la compresión a los 28 días.	0.002	0.002
epsc	Deformación en el punto de resistencia al aplastamiento.	0.009	0.004

Nota: Se muestran los datos característicos del concreto para el modelo 2.

TABLA IV

CARACTERÍSTICAS DEL CONCRETO PARA MODELO 3

Parámetro	Descripción	Inconfinado	Confinado
fpc	Resistencia a la compresión a los 28 días.	313 kgf/cm ²	210 kgf/cm ²
fpcu	Resistencia al aplastamiento del concreto	313 kgf/cm ²	0 kgf/cm ²
epsu	Deformación en el punto de resistencia a la compresión a los 28 días.	0.002	0.002
epsc	Deformación en el punto de resistencia al aplastamiento.	0.007	0.004

Nota: Se muestran los datos característicos del concreto para el modelo 3.

TABLA V

CARACTERÍSTICAS DEL ACERO PARA LOS MODELOS 1 y 2

Parámetro	Descripción	Valor
fyp	Resistencia a la tensión.	4200 kgf/cm ²
fyn	Resistencia a la compresión.	4200 kgf/cm ²
E0	Módulo de elasticidad.	2100000 kgf/cm ²
bp	Índice de endureciendo en tensión.	0.015
bn	Índice de endureciendo en compresión.	0.015

Nota: Se muestran los datos característicos del acero para los modelos 1 y 2.

TABLA VI

CARACTERÍSTICAS DEL ACERO PARA MODELOS 3

Parámetro	Descripción	Valor
fyp	Resistencia a la tensión.	2800 kgf/cm ²
fyn	Resistencia a la compresión.	2800 kgf/cm ²
E0	Módulo de elasticidad.	2100000 kgf/cm ²
bp	Índice de endureciendo en tensión.	0.007
bn	Índice de endureciendo en compresión.	0.007

Nota: Se muestran los datos característicos del acero para el modelo 3.

2.4.1.3 Carga axial asignada

Para efectos del modelado de las secciones definidas, se asignó una carga axial a cada uno de los modelos cuya magnitud se expone en la tabla VII.

TABLA VII
CARGAS AXIALES APLICADAS A MODELOS

Modelo	Carga axial
1	90 tonf
2	50 tonf
3	120 tonf

Nota: Se muestran las cargas axiales aplicadas a los diferentes modelos.

2.5.2 Generación de gráficos comparativos

Para efectos de la comparación de los resultados se modelaron las secciones estructurales definidas previamente en el software XTRACT, según se mencionó anteriormente. Los datos resultantes de los modelos se registraron, con el fin de captar las respuestas estructurales específicas generadas por el software XTRACT para cada uno de los casos en estudio, finalmente estos datos fueron comparados de forma directa con los resultados obtenidos a través de la herramienta computacional desarrollada. Se generaron gráficos en un

entorno común, utilizando Microsoft Excel, para visualizar de forma clara las diferencias y similitudes entre los resultados de cada herramienta.

La fase de graficación no solo constituye una herramienta visual esencial para evaluar de forma inmediata la confiabilidad de la herramienta desarrollada, sino que también se integra perfectamente en el proceso metodológico general. Este enfoque metodológico, funge como insumo clave para la validación de la herramienta desarrollada y la comprensión integral de su desempeño en relación con un estándar establecido.

2.5.3 Calculadora Mander

Para verificar la precisión de la "Calculadora Mander" incorporada en la herramienta computacional desarrollada, se realizará una comparación exhaustiva de sus resultados con los obtenidos en un estudio previo llevado a cabo en la Escuela Politécnica Federal de Zúrich. Este estudio, titulado "Ensayos cíclicos cuasiestáticos y análisis de rótulas plásticas de muros estructurales RC" [35], proporciona una base sólida para evaluar la capacidad de la herramienta en simular el comportamiento del concreto confinado en términos de aumento en la resistencia y ductilidad ante el refuerzo de acero.

Ahora bien, para efectos de la validación se tomó como referencia el muro ensayado y denominado WSH6, dentro del estudio. Las características geométricas y disposición del acero de refuerzo se ilustra en la siguiente figura.

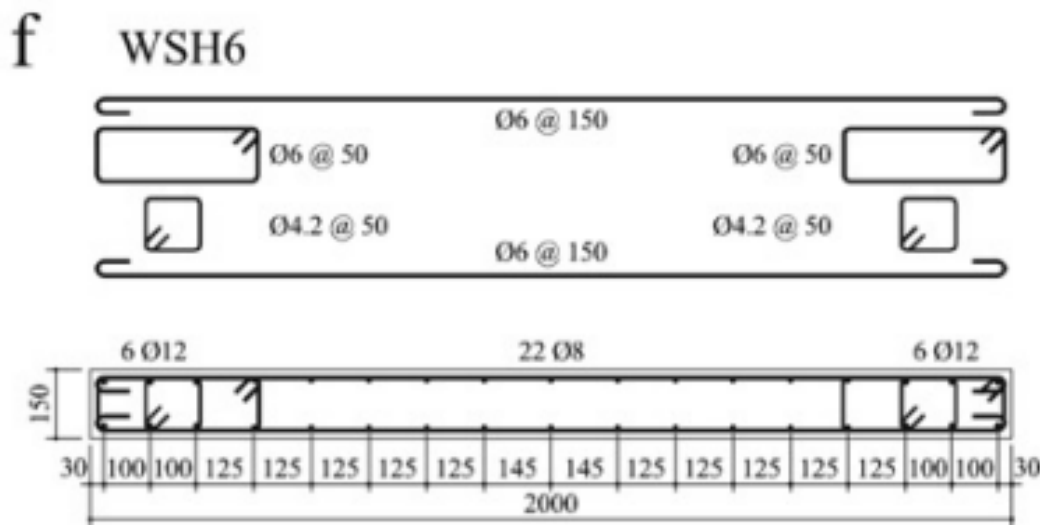


Fig. 23. Modelo WSH6 [35].

Por otro lado, las características mecánicas de los materiales se muestran en la tabla VIII.

TABLA VIII
CARACTERÍSTICAS MECANICAS DE LOS MATERIALES

Parámetro	Descripción	Valor
fc	Resistencia a la compresión a los 28 días.	45 N/mm ²
ec	Deformación máxima del concreto	0.002
fy	Resistencia a la fluencia del acero	583 N/mm ²

Nota: Se muestran las características mecánicas de los materiales reportados en el documento de apoyo.

Para efectos de la comparación entre resultados, se procedió a realizar una comparación entre los resultados obtenidos, se integraron los datos característicos del muro ensayado directamente en la herramienta computacional desarrollada. Luego, se llevó a cabo el cálculo del aumento en la resistencia y ductilidad del concreto confinado para finalmente, evaluar de manera cuantitativa las discrepancias entre los resultados de la herramienta y los datos experimentales, se aplicaron estrategias estadísticas en el proceso de comparación.

Capítulo 3: Resultados y análisis

En esta sección, se presentan los resultados con su respectivo análisis, los cuales se derivan de los objetivos específicos establecidos para el desarrollo del proyecto. En primer lugar, se muestran los resultados derivados de una exhaustiva evaluación de diversos paquetes de controles gráficos. Posteriormente, expone la estructura del programa y finalmente los resultados obtenidos a partir de la validación y comparación de la herramienta desarrollada con el estándar establecido.

3.1 Evaluación de controles gráficos

La metodología empleada en esta evaluación se basó en criterios diseñados para identificar la solución que mejor se adapte a las exigencias del proyecto. Cada uno de los controles gráficos se sometió a un análisis donde se tuvieron en cuenta parámetros como funcionalidad, documentación, adaptabilidad a la herramienta, recursos computacionales y licencias para operar. En la tabla IX, se presentan los resultados obtenidos a partir de la evaluación realizada, así se proporciona una visión objetiva de las fortalezas y limitaciones de cada opción evaluada.

TABLA IX
EVALUACIÓN DE LAS DIFERENTES LIBRERÍAS GRÁFICAS

Criterio	PyQt	Tkinter	Kivy
Funcionalidad	2	2	2
Documentación	2	2	1
Adaptabilidad a la herramienta	2	0	1
Recursos computacionales	1	2	0
Licencias para operar	2	2	2
Suma	9	8	6

Nota: Se muestra la evaluación asignada a cada una de las librerías gráficas.

Este proceso de evaluación y selección constituye la base sobre la cual se fundamenta la elección de la herramienta, asegurando que la solución adoptada responda de manera óptima a las necesidades específicas del proyecto.

La definición de los diferentes criterios de evaluación se estableció, contemplando la necesidad de proporcionar al usuario una herramienta no solo funcional, sino también amigable e intuitiva. La consideración principal fue garantizar que la interfaz diseñada facilitara el ingreso de datos, optimizando así el cálculo del diagrama momento-curvatura. En este contexto, se priorizaron criterios como la funcionalidad y adaptabilidad para asegurar que la herramienta desarrollada cuente con elementos visuales que beneficien la interacción eficaz del usuario con el programa, incluso para aquellos que cuenten con escasos niveles de experiencia.

Por otro lado, se evaluaron criterios técnicos, como el consumo de recursos computacionales, la documentación y las licencias para operar. Este enfoque estratégico en el establecimiento de criterios no solo garantiza la funcionalidad técnica de la herramienta, sino que también respalda su viabilidad práctica al considerar las necesidades y expectativas del usuario final. Este compromiso con criterios específicos refleja la dedicación al desarrollo de una herramienta que no solo cumple con los estándares técnicos, sino que también se adapta a la experiencia del usuario de manera integral.

Posterior a la evaluación realizada basada en los criterios establecidos en la metodología, PYQT emergió como la elección más destacada. Este sistema de enlaces de Python mostró un rendimiento superior en comparación con Tkinter y Kivy, según se muestra en la Tabla IX. Su selección se fundamenta en diversos factores clave, entre ellos la robustez de sus capacidades gráficas, la flexibilidad en la personalización de la interfaz y su integración con el lenguaje de programación Python y Openseespy. Además, PYQT se destacó por la capacidad de ofrecer elementos visuales que garantizan una experiencia del usuario intuitiva. Esta elección estratégica de PYQT como sistema para la elaboración de la interfaz gráfica de usuario de la herramienta computacional desarrollada respalda una visión que destaca en términos de eficiencia técnica y experiencia de usuario.

3.2 Estructura del programa

La implementación de la metodología propuesta, la cual se expone en la sección anterior, condujo al desarrollo de una herramienta computacional robusta y eficiente para la conceptualización de diagramas momento-curvatura en muros con geometría compleja. A continuación, se presentan los resultados y análisis detallado de la estructura del programa, partiendo de una visión general a la específica, resaltando los aspectos más relevantes y la consecución de los objetos establecidos.

3.2.1 Programación orientada a objetos

La elección de Python como lenguaje de programación y Spyder como entorno científico, proporcionó una plataforma sólida y flexible para la implementación del proyecto. La sintaxis clara, la compatibilidad con el software de análisis de estructural Openseespy y la amplia biblioteca de Python contribuyeron significativamente en la eficiencia y legibilidad del código. Por otro lado, Spyder ofreció un ambiente de desarrollo interactivo que facilitó la iteración y depuración de la herramienta.

Ahora bien, la adopción de un estilo de programación orientado a objetos demostró ser crucial para la sostenibilidad del código. La estructura propuesta, se compone de clases y objetos claramente definidos, y dedicados a cubrir y funciones específicas dentro del programa. Esta organización facilita la expansión futura del programa y contribuye a una mayor comprensión del código.

3.2.2 Organización jerárquica de scripts

La herramienta computacional desarrollada se compone de diferentes scripts, cada uno diseñado para funciones específicas. Esta organización jerárquica facilita la gestión y mantenimiento del código, lo que permite futuras inclusiones de nuevas funcionalidades y actualización a las existentes. En la figura 24, se ilustra la interconexión entre códigos en el programa, lo que representa así la cohesión estructural. Cada script contribuye de manera específica al flujo general del programa, al cumplir con funciones específicas.

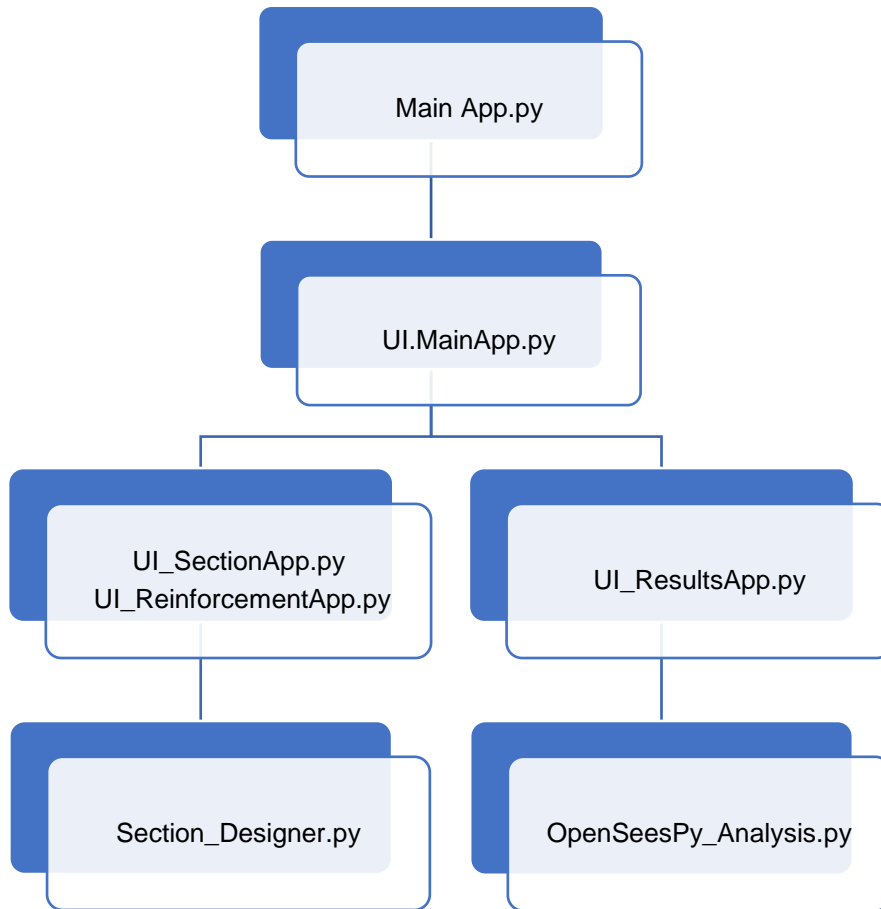


Fig. 24. Interconexión entre scripts.

Según se observa en la figura 24, la cual ilustra la interconexión entre los diversos scripts que componen la herramienta computacional, es posible resaltar la colaboración entre los diferentes elementos del código. Cada script ha sido diseñado con un propósito específico, contribuye así de manera singular al conjunto funcional del programa. Ahora bien, la conexión entre estos se realizó, de forma que, se asegure un flujo coherente de datos y funciones a lo largo de la ejecución del programa. El enfoque organizativo mostrado, se traduce en un desarrollo más ágil y eficiente, el cual promueve la expansión y el mejoramiento del proyecto.

3.2.3 Confección de la interfaz principal

La confección de la interfaz principal se realizó con un enfoque centrado en el usuario, de forma tal que este se sienta guiado a través de un flujo constante durante la entrada de datos, con el objetivo claro de minimizar errores y mejorar significativamente la usabilidad de la herramienta. El diseño de la interfaz principal se muestra en la figura 25.

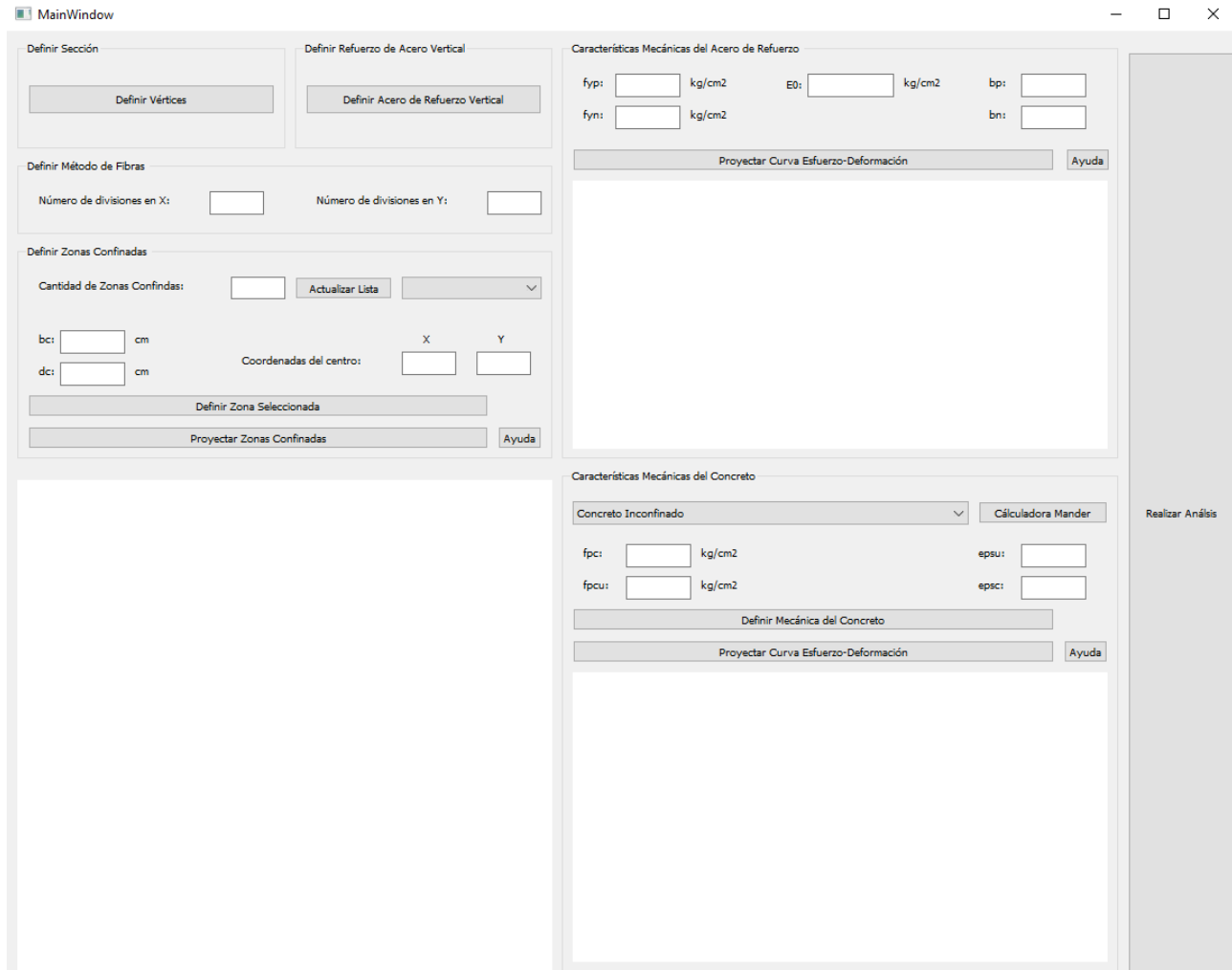


Fig. 25. Diseño de la interfaz principal.

El enfoque centrado en el usuario trajo a una interfaz que presenta instrucciones claras y un diseño lógico. Los elementos visuales, se distribuyeron de manera coherente, guiando al usuario de forma natural a medida que procede la entrada de datos requerida para el cálculo del diagrama momento-curvatura. Esta estructura secuencial no solo simplifica el proceso, sino que, además, reduce posibles errores al minimizar las oportunidades de desviación del flujo previsto.

Por otro lado, se le dio un énfasis particular en la retroalimentación visual, es decir, permitir que el usuario tenga una comprensión inmediata del progreso y la validez de los datos ingresados, a partir de los diferentes gráficos que se muestran durante la ejecución del programa, donde se representa la geometría de la sección, disposición del refuerzo de acero y características de mecánicas de los materiales. Botones de ayuda se implementaron estratégicamente para ofrecer orientación en tiempo real, lo que favorece la capacidad del usuario para interactuar de manera efectiva con la herramienta.

3.2.4 Componentes clave de la herramienta

Como se mencionó anteriormente, la herramienta computacional desarrollada se compone de diferentes scripts, que cuentan con funciones específicas, para abordar el proceso desde la validación de los datos de entrada hasta la generación del modelo estructural y la ejecución del análisis no lineal. A continuación, en la tabla X, se detalla de forma puntual, la visión detallada de cada script, destacando su contribución al funcionamiento global de la herramienta.

TABLA X
NIVELES DE DESPEÑO

Nombre del Script	Descripción del funcionamiento
Main App.py	Establece la base y ejecuta el programa.
UI_MainApp.py	Cuenta con la programación de todos los elementos visuales que componen la interfaz principal, almacena los datos que son ingresados dentro de la misma y ejecuta el análisis no lineal.
UI_SectionApp.py	Cuenta con la programación de un elemento de tabla donde el usuario ingresa los datos correspondientes a la geometría de la sección. Además, almacena dichos datos.
UI_Reinforcement.py	Cuenta con la programación de un elemento de tabla donde el usuario ingresa los datos correspondientes a la disposición del acero de refuerzo longitudinal. Además, almacena dichos datos.
UI_Results.py	Cuenta con la programación de la ventana emergente donde se muestra el diagrama momento-curvatura. Almacena la carga axial ingresada por el usuario y muestra el diagrama.
Section_Designer.py	Recolecta los datos ingresados por el usuario y aplica el método de fibras. Además, genera las matrices donde se almacenan los datos característicos de cada fibra.
OpenSeesPy_Analysis.py	Contiene la programación del análisis no lineal. Interpreta todos los datos ingresados por el usuario, para posteriormente ejecutar el cálculo a través de OpenSeesPy.

Nota: Se muestra la evaluación asignada a cada una de las librerías gráficas.

Como se detalla en la tabla X, cada script desempeña una función única y especializada dentro del proceso de análisis estructural. Estas funciones específicas se diseñaron con el fin de abordar los diferentes elementos necesarios para la automatización del cálculo del diagrama momento-curvatura, de manera separada, favoreciendo así el proceso de depuración del código y para facilitar las futuras actualizaciones a la herramienta computacional desarrollada.

La secuencia de scripts no solo garantiza un análisis estructural completo, sino que también una coordinación eficiente entre cada fase del proceso, demostrando el sistema modular y la funcionalidad distintiva de cada componente en la consecución de los objetivos de la herramienta.

3.2.5 Análisis estructural

El análisis no lineal, se realizó a través de los comandos proporcionados por OpenSeesPy. En este se configuraron dos tipos de carga, una carga axial constante y un momento unitario que crece linealmente con el tiempo. La definición de esta última carga proporciona una carga variable en la estructura, lo que resulta en un análisis que aborda el comportamiento no lineal de la misma. Este enfoque es fundamental en la evaluación de la sección bajo condiciones de carga variable o bien, la respuesta no lineal de la estructura.

Por otro lado, el integrador de control de desplazamiento se seleccionó con el fin de controlar el análisis con base en los desplazamientos del nodo al cual se le asignó la sección por calcular. Ahora bien, el análisis se llevó a cabo con 100 incrementos, lo cual garantiza un equilibrio entre la precisión del resultado y la eficiencia computacional al ejecutar el programa. Este enfoque es especialmente relevante en el desarrollo de herramientas computacionales para análisis estructural donde la eficiencia en términos de recursos es una consideración clave para la utilidad y aplicabilidad práctica de la herramienta.

3.3 Validación

3.3.1 Diagrama momento-curvatura

En esta fase de validación, se llevaron a cabo modelaciones específicas de secciones estructurales, se utiliza la herramienta computacional desarrollada, se abarcan los modelos detallados en la sección de la metodología. El objetivo principal fue comparar y contrastar los resultados obtenidos por la herramienta con aquellos generados por XTRACT, una herramienta reconocida en análisis estructural de secciones transversales. A continuación, en las figuras 26, 27, 28, 29, 30 y 31, se presenta el proceso para la obtención y los resultados

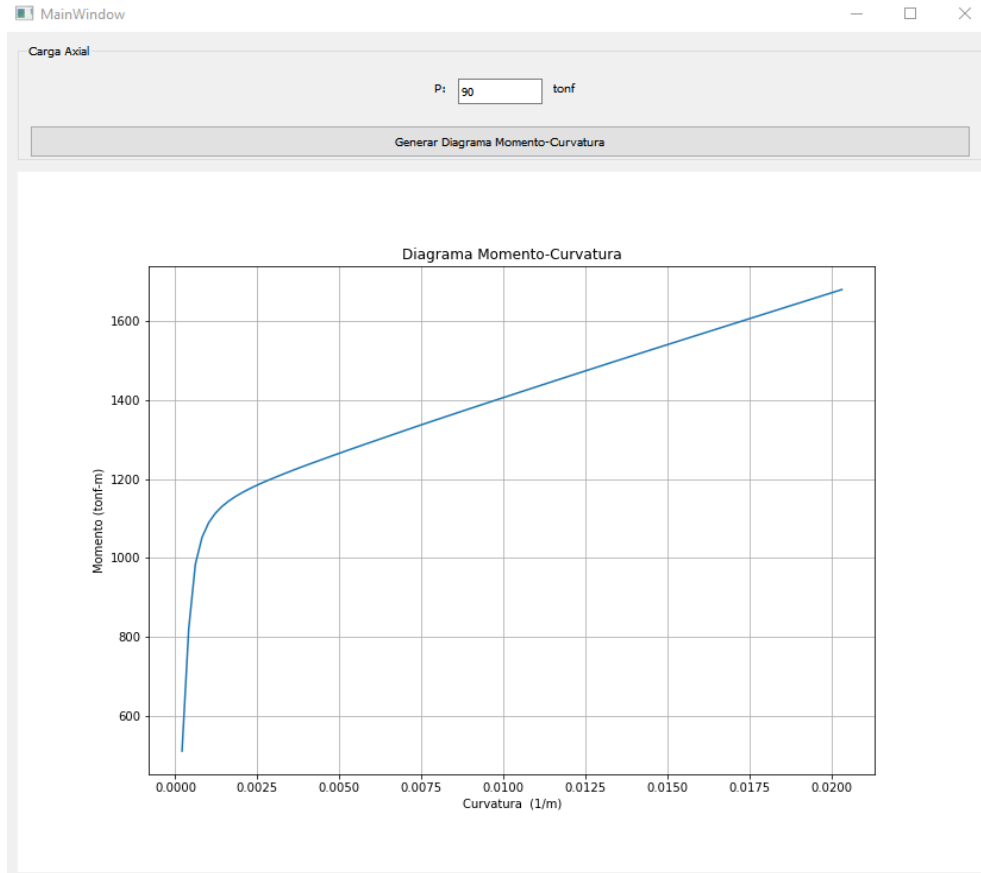


Fig. 27. Resultados del modelo 1.

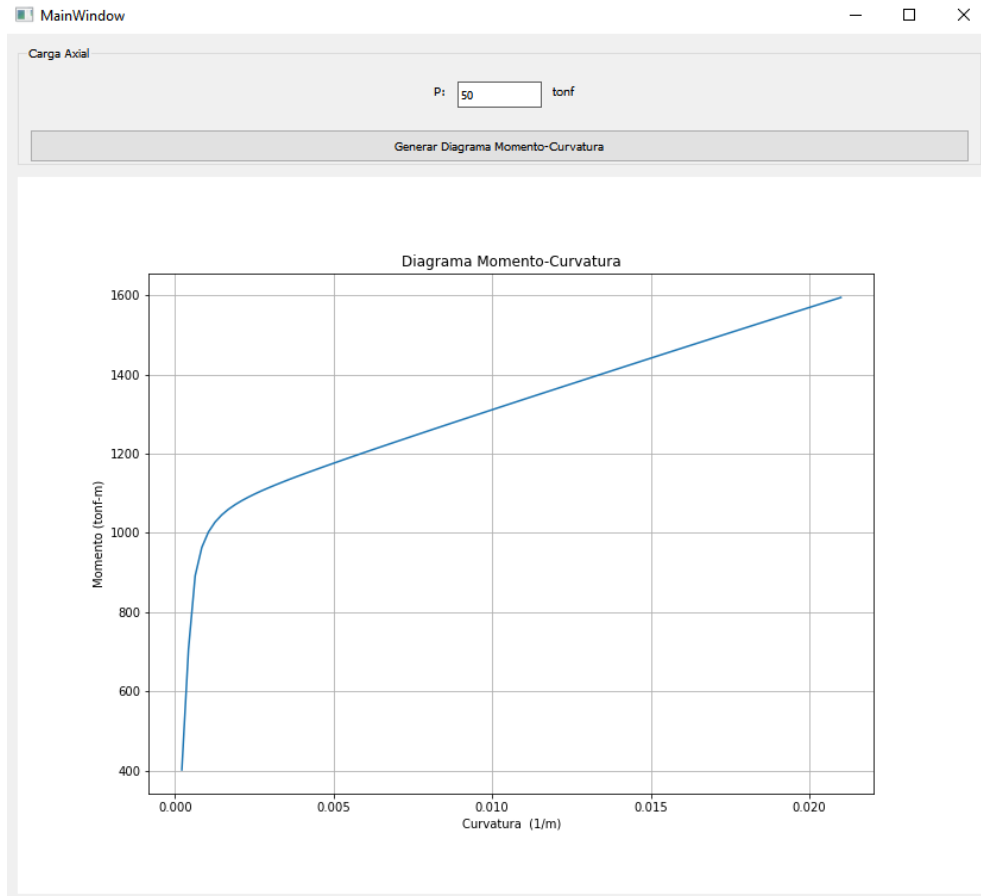


Fig. 29. Resultados del modelo 2.

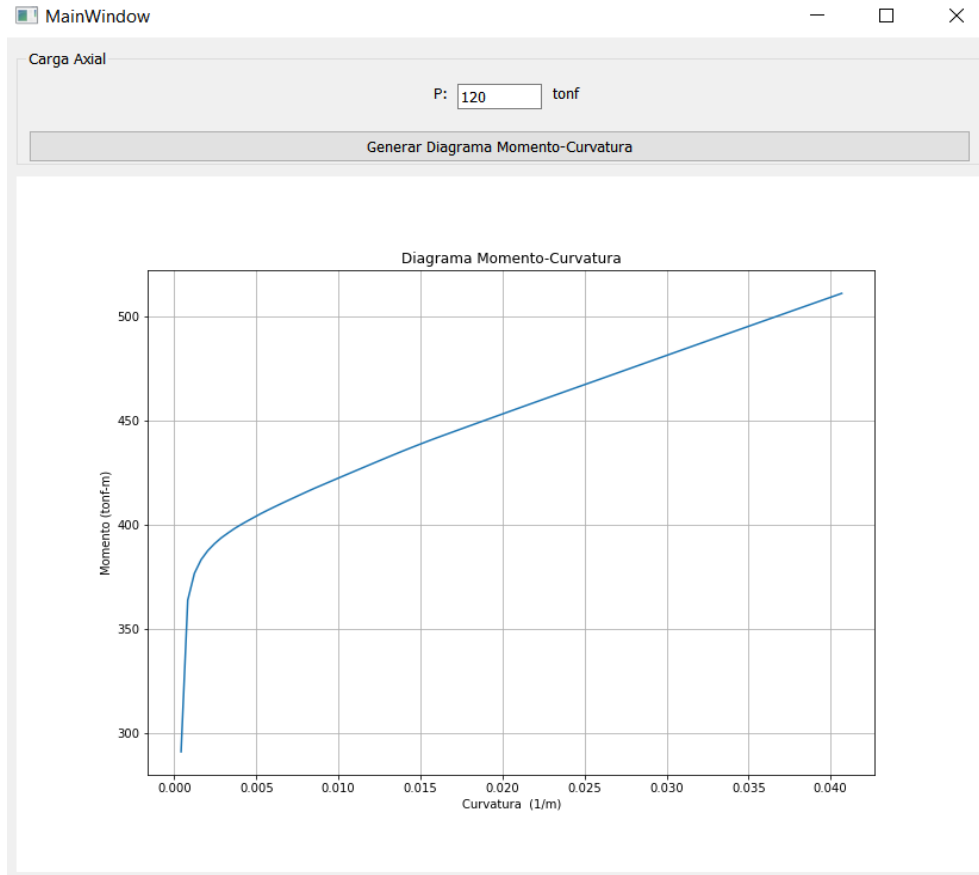


Fig. 31. Resultados del modelo 3.

La representación gráfica generada por el programa computacional desarrollado no solo ofrece una visualización detallada del comportamiento momento-curvatura de las secciones bajo análisis, sino que también proporciona datos cuantitativos cruciales. En este contexto, se extraen los momentos flectores máximos para cada uno de los modelos evaluados. En el caso del Modelo 1, se ha identificado un momento máximo de 1700 tonf-m, para el Modelo 2 se obtuvo un valor de 1600 tonf-m; el Modelo 3 exhibe un momento flector máximo de 510 tonf-m. Estos valores son esenciales para la evaluación precisa de la capacidad de carga de cada sección, y desempeñan un papel crítico en la selección de materiales y dimensiones adecuadas, garantizando así un rendimiento seguro y eficiente ante las condiciones de carga proyectadas. En particular, el análisis no lineal proporciona una ventaja significativa al permitir una evaluación más precisa del momento flector máximo, ya que considera la no linealidad del material y el comportamiento geométrico de la estructura. Esto resulta en una mayor precisión en la predicción del comportamiento estructural, lo que a su vez facilita la toma de decisiones informadas en el diseño y la selección de materiales, contribuyendo a la seguridad y eficiencia del proyecto.

Además, la información resultante de la herramienta computacional incluye las curvaturas asociadas a cada uno de los modelos evaluados. En el caso del Modelo 1, la curvatura máxima alcanzada es de 0.02,

mientras que para el Modelo 2 se registra una curvatura de 0.022. Por último, el Modelo 3 exhibe una curvatura de 0.04. Las curvaturas se expresan en unidades de 1/m, donde la medida representa la variación de la curvatura por unidad de longitud. Estos valores de curvatura son fundamentales para comprender la capacidad de deformación de cada sección estructural, proporcionando así información valiosa sobre la ductilidad y flexibilidad de los materiales utilizados.

La evaluación comparativa entre la herramienta computacional desarrollada y XTRACT se materializa a través de gráficos comparativos detallados que ilustran los momentos de flexión y las curvaturas generadas por ambas herramientas. Estos gráficos, elaborados con precisión en Microsoft Excel, ofrecen una visión completa y comparativa de las respuestas estructurales para cada modelo analizado. El análisis detallado de estos gráficos proporciona información clave sobre la coherencia y la precisión de la herramienta desarrollada en relación con XTRACT, y permite así una evaluación crítica y cuantitativa de su desempeño en diferentes contextos estructurales.

A continuación, en las figuras 32, 33 y 34, se presentan los gráficos momento-curvatura correspondientes a cada uno de los modelos analizados. Estos gráficos son fundamentales para discernir cualquier variación significativa entre los resultados de ambas herramientas, lo que contribuye así a la validación integral de la herramienta computacional desarrollada.

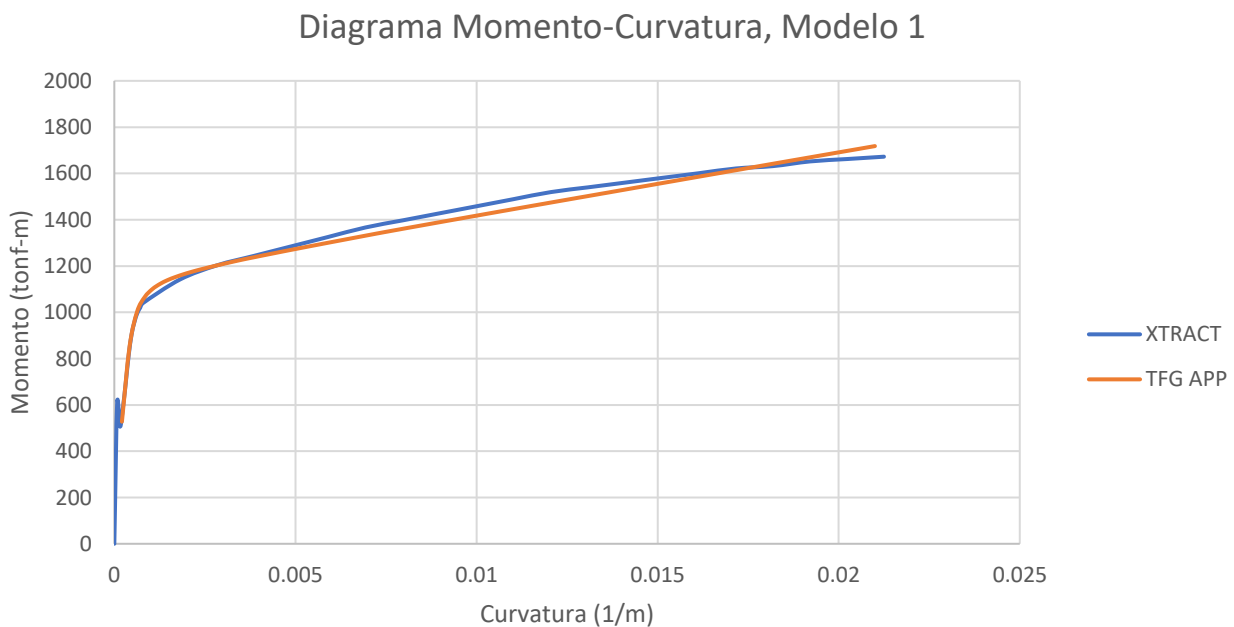


Fig. 32. Diagrama momento-curvatura, modelo 1.

Diagrama Momento-Curvatura, Modelo 2

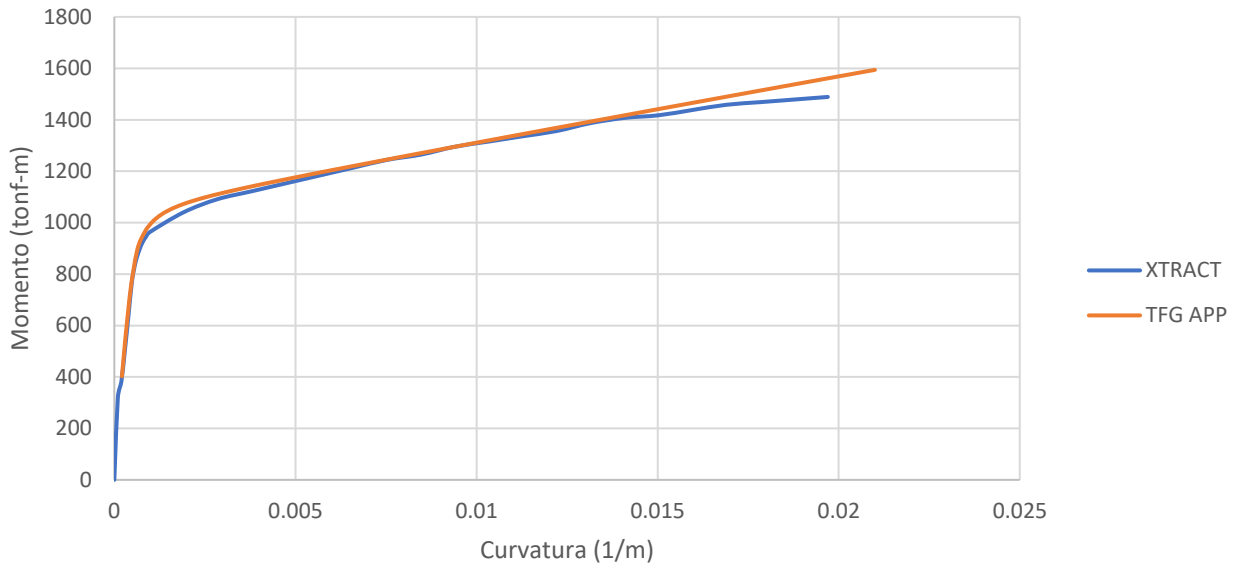


Fig. 33. Diagrama momento-curvatura, modelo 2.

Diagrama Momento-Curvatura, Modelo 3

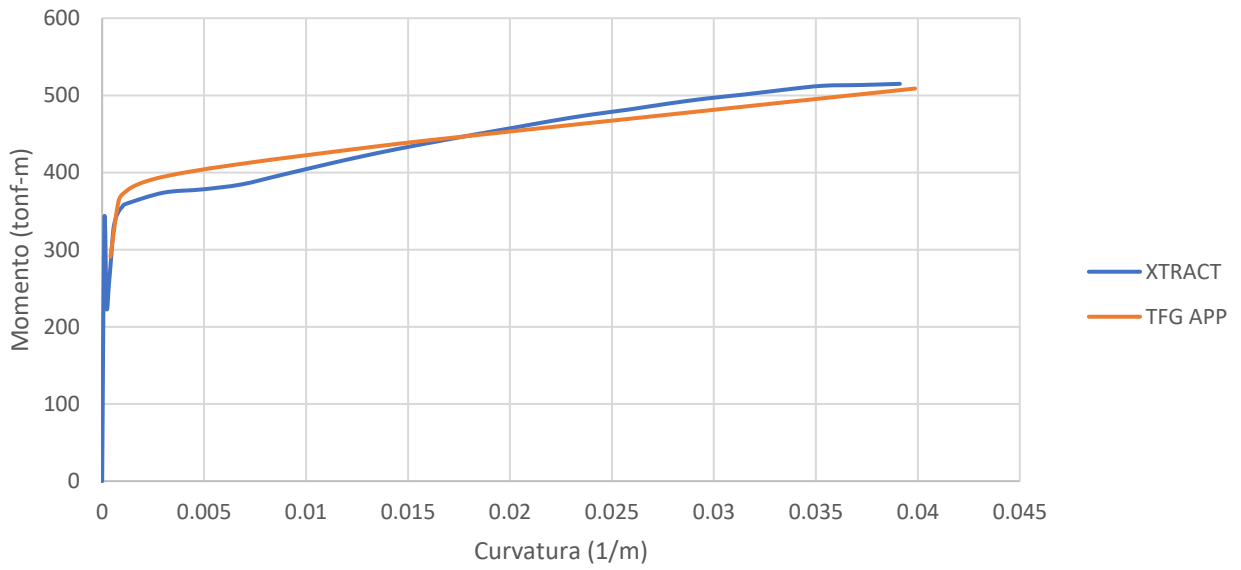


Fig. 34. Diagrama momento-curvatura, modelo 3.

Al examinar los resultados obtenidos de las gráficas momento-curvatura generadas por XTRACT y la herramienta computacional desarrollada para las tres secciones analizadas, se destaca una gran similitud en el comportamiento estructural. En muchos tramos de las curvas, las respuestas de ambas herramientas se superponen de manera significativa, lo que evidencia una concordancia notable. En particular, la zona anterior a la fluencia del acero muestra una gran concordancia, aunque al acercarnos al punto de fluencia se observa una pequeña variación, donde la herramienta computacional exhibe una capacidad de momento ligeramente superior.

Por otro lado, posterior al punto de fluencia, la curva generada por el software XTRACT presenta un comportamiento parabólico. En contraste, la herramienta desarrollada presenta un comportamiento más lineal en esta fase.

Las discrepancias señaladas anteriormente se atribuyen al modelo constitutivo utilizado para representar el comportamiento del acero en la herramienta computacional desarrollada. SteelMPF de OpenSeesPy incluye efectos de endurecimiento por deformación isotrópica, al presentar transiciones curvas con asíntotas de línea recta [36]. Las pendientes de estas rectas dependen del módulo de elasticidad y los índices de endurecimiento definidos por el usuario. Por otro lado, XTRACT utiliza un modelo de acero con endurecimiento por deformación parabólica [37], lo que muestra una transición lineal antes del esfuerzo de fluencia, seguida de una zona de plasticidad perfecta, posteriormente un endurecimiento por deformación y finalmente una etapa de estricción donde disminuye la capacidad del material hasta alcanzar la fractura.

En términos de ductilidad, ambos conjuntos de datos reflejan valores muy similares con pequeñas variaciones. Estas diferencias se atribuyen al enfoque único que emplea la herramienta computacional desarrollada para abordar las zonas confinadas y no confinadas de las secciones. Mientras ambos softwares aplican el método de fibras, la herramienta computacional considera el número de divisiones que el usuario estipula para dividir la sección, así mismo, únicamente permite incluir zonas confinadas rectangulares, lo que obliga al usuario a establecer múltiples zonas para representar ciertas geometrías. Esta particularidad puede inducir variaciones en la predicción del comportamiento del elemento estructural ante cargas monotónicas, lo que afecta la deformación última del concreto y, por ende, el valor de momento-curvatura de la sección. Estas observaciones resaltan la importancia de considerar el enfoque específico de cada herramienta al interpretar los resultados.

3.3.2 Calculadora Mander

Como resultado ante el cálculo realizado a través de la herramienta computacional desarrollada, que contempla los datos característicos del ensayo WSH6, publicado por Alessandro Dazio [35], se tiene figura 35, la cual muestra el aumento en resistencia y ductilidad del material especificado.

Fig. 35. Validación Calculadora Mander.

Ahora bien, los resultados ante la comparación de los datos extraídos del estudio titulado "Ensayos cíclicos cuasiestáticos y análisis de rótulas plásticas de muros estructurales RC", se resumen en la tabla XI.

TABLA XI
COMPARACIÓN DE RESULTADOS

Factor	Valor exacto (kgf/cm2)	Valor calculado (kgf/cm2)	% de error
Resistencia a la compresión	642.42	648.93	0.00010
Deformación máxima	0.00600	0.00614	0.00023

Nota: Se muestra la comparativa de datos entre los reportados en el ensayo y el calculado a través de la herramienta desarrollada.

Los resultados del análisis comparativo entre los datos obtenidos mediante la Calculadora Mander y los resultados experimentales del estudio de la Escuela Politécnica Federal de Zúrich revelan notables similitudes en el aumento de resistencia a la compresión y la deformación máxima del concreto confinado. La baja magnitud de los porcentajes de error calculados subraya la precisión de la herramienta desarrollada para estimar estos parámetros fundamentales. Sin embargo, es crucial considerar ciertos factores que podrían contribuir a las diferencias observadas. En primer lugar, se destaca el cambio en las unidades de medida, ya que los datos experimentales del estudio utilizan unidades distintas a las empleadas internamente por la Calculadora Mander. Este ajuste unitario se realizó meticulosamente para garantizar la coherencia en la comparación.

Además, es relevante señalar que la Calculadora Mander basa sus cálculos en un método que utiliza coeficientes derivados de una extensa cantidad de experimentos y datos recopilados [33]. En contraste, los detalles específicos sobre el modelo constitutivo empleado en el estudio experimental no fueron proporcionados en la publicación. Esta diferencia en los enfoques metodológicos puede influir en las discrepancias observadas y destaca la importancia de la transparencia en la documentación de los modelos constitutivos utilizados en investigaciones experimentales para lograr una comparación más precisa y completa.

Conclusiones y recomendaciones

Conclusiones

- El desarrollo de la herramienta computacional ha culminado de manera satisfactoria, al proporcionar a los usuarios la capacidad de generar diagramas momento-curvatura para diversas configuraciones geométricas de secciones de concreto reforzado. La flexibilidad de la herramienta, al permitir la definición de secciones con geometría compleja, la convierte en una herramienta valiosa para la conceptualización y evaluación de estructuras de concreto reforzado.
- Se implementó una interfaz gráfica desarrollada en el marco de trabajo PyQt. Esta se distingue por su intuitividad, que guía al usuario a través de un flujo continuo y lógico que simplifica el proceso de ingreso de datos y cálculo. La arquitectura amigable de la interfaz asegura que los usuarios se mantengan en el camino correcto sin desviarse del procedimiento, lo que facilita la generación de diagramas momento-curvatura de manera eficiente y precisa. La combinación de la potencia de cálculo de la herramienta con una interfaz intuitiva representa un avance significativo en la accesibilidad y usabilidad para ingenieros y estudiantes en el análisis de estructuras de concreto reforzado.
- Para el desarrollo de la herramienta, se adoptó una programación orientada a objetos (OOP). La implementación de la OOP permitió organizar de manera eficiente el código, al mejorar su comprensión y facilitar futuras expansiones y modificaciones. Se estableció una jerarquía de scripts, cada uno cumple así con un propósito específico dentro de la funcionalidad integral del programa. Esta estructura modular no solo contribuyó a una mayor claridad y mantenibilidad del código, sino que también proporciona una base sólida para incorporar nuevas funcionalidades o ajustar las existentes en el futuro.
- El desarrollo de la herramienta se llevó a cabo al aprovechar la potencia y flexibilidad del programa de análisis estructural OpenSeesPy. Integra las capacidades de OpenSeesPy con las funcionalidades de Python, se logra construir una herramienta robusta y versátil. La combinación de estas tecnologías permitió implementar el método de fibras para el análisis de secciones de concreto reforzado, así como la definición detallada de la geometría de estas secciones.
- El programa desarrollado ha demostrado ser una herramienta valiosa para la aproximación de los momentos flectores máximos junto con sus respectivas curvaturas en el análisis de secciones estructurales. La visualización detallada y la información cuantitativa proporcionada por la herramienta permiten una comprensión integral del comportamiento momento-curvatura, al destacar la

concordancia significativa con resultados obtenidos de manera independiente. La capacidad de extraer con precisión los momentos flectores máximos asociados con las respectivas curvaturas no solo fortalece la confiabilidad de la herramienta sino también su utilidad en el diseño estructural, donde la combinación de estos dos parámetros es esencial para evaluar la respuesta y la seguridad de las secciones bajo diversas condiciones de carga.

- La recopilación de los momentos flectores máximos y curvaturas en la evaluación estructural enriquece la comprensión del comportamiento de las secciones, al proporcionar una base sólida para la toma de decisiones informadas en el diseño y análisis de estructuras.
- La validación de la herramienta computacional frente al software XTRACT demostró coherencia en los resultados obtenidos para las secciones analizadas. La comparación detallada de los diagramas momento-curvatura reveló una gran similitud en el comportamiento estructural, con superposiciones significativas en muchos tramos de las curvas. Específicamente, la herramienta desarrollada presentó una capacidad de momento ligeramente superior en la zona posterior al punto de fluencia del acero.
- La “Calculadora Mander” integrada dentro de la herramienta computacional desarrollada, presentó en términos cuantitativos, porcentajes de error inferiores al 0.0003%, en la validación realizada, al destacar la coherencia y precisión del programa. A pesar de que las diferencias podrían atribuirse a factores como el cambio de unidades entre el estudio y la calculadora, y la utilización de modelos constitutivos diferentes, la tendencia general demostró una concordancia efectiva.

Recomendaciones

- Se recomienda a futuros desarrolladores buscar asesoramiento directo con el autor del proyecto al considerar una modificación o expansión del programa.
- Para mejorar la precisión del análisis, es crucial incorporar el efecto del cortante en futuras versiones de la herramienta.
- El estudio detallado del comportamiento de los modelos constitutivos integrados en el programa comprende un aspecto esencial en el uso de la herramienta para garantizar la correcta interpretación de los resultados y la entrada de datos al programa.
- Se considera importante que los usuarios presenten conocimientos previos en el análisis no lineal de elementos de concreto. Dado que la herramienta desarrollada se centra en proporcionar un enfoque automatizado para generar diagramas momento-curvatura, es esencial que los usuarios estén familiarizados con los principios fundamentales del análisis estructural y las características del comportamiento no lineal de los elementos de concreto reforzado. Este conocimiento previo permitirá a los usuarios interpretar de manera más efectiva los resultados generados por la herramienta y realizar ajustes pertinentes en función de su comprensión del comportamiento estructural.

Referencias

References

- [1] K. Sánchez Becerra, "Desarrollo De Una Herramienta Computacional Para Conceptualización Y Elaboración De Diagramas Momento Curvatura En Secciones De Concreto Reforzado." , Escuela Colombiana de Ingeniería Julio Garavito, 2018.
- [2] F. Morales Ortiz, "Desarrollo De Una Herramienta Computacional Para La Evaluación Del Comportamiento no Lineal De Muros De Concreto Reforzado." , Escuela Colombiana de Ingeniería Julio Garavito, 2021.
- [3] J. Solano, *Introducción a La Programación En Python . (2ª Ed ed.)* Costa Rica: Editorial Tecnológica de Costa Rica, 2017.
- [4] E. L. Pérez Montero and F. d. M. Hernández Pérez, "Object-oriented programming: easy to create," *I+t+c*, vol. 1, (13), 2019. Available: https://revistas.unicomfaucauca.edu.co/ojs/index.php/itc/article/view/itc2019_pag_96_100.
- [5] S. F. Lott, *Python Essentials*. 2015 Available: <https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1019626&lang=es&site=ehost-live>.
- [6] (). *Anaconda*. Available: <https://www.anaconda.com/>.
- [7] I. Spyder. (). *Spyder*. Available: <https://www.spyder-ide.org/>.
- [8] Sanchez Miquel, *Intelligent Decision Support Systems*. (1ª ed.) Suiza: Springer, 2022.
- [9] J. Cervantes *et al*, *Introducción a La Programación Orientada a Objetos*. (1ª ed.) México: Universidad Autónoma Metropolitana, 2016.
- [10] L. López, *Metodología De La Programación Orientada a Objetos*. (2ª ed.) México: Alfaomega, 2013.
- [11] T. Domínguez, *Desarrollo De Interfaces Gráficas En Python 3 Con Tkinter*. (1ª ed.) Marcombo, 2022.
- [12] J. Tidwell, *Designing Interfaces*. (2ª ed.) Canada: O' Reilly, 2010.
- [13] H. Sharp, Y. Rogers and J. Preece, *Interaction Design: Beyond-Computer Interaction*. (6ª ed.) Inglaterra: Wiley, 2017.
- [14] M. Á Gómez, "Diseño Estructural De Edificios Altos Tipo Torre Empleando Concreto De Alta Resistencia." , Universidad Nacional Autónoma de México, 2011.
- [15] (). *Welcome to the OpenSeesWiki*. Available: https://opensees.berkeley.edu/wiki/index.php/Main_Page.

- [16] M. Zhu, F. McKenna and M. H. Scott, "OpenSeesPy: Python library for the OpenSees finite element framework," *SoftwareX*, vol. 7, 2018. Available: <https://doi.org/10.1016/j.softx.2017.10.009>. DOI: 10.1016/j.softx.2017.10.009.
- [17] (). *OpenSeesPy*. Available: <https://openseespydoc.readthedocs.io/en/latest/#>.
- [18] J. A. Marín, "Análisis Del Método De Las Fibras Para La Revisión Y Diseño De Elementos De Hormigón Armado." , Escuela de Ingeniería de Antioquia, 2009.
- [19] O. González Cueto *et al*, "Análisis de los modelos constitutivos empleados para simular la compactación del suelo mediante el método de elementos finitos," *Revista Ciencias Técnicas Agropecuarias*, vol. 22, (3), 2013.
- [20] K. I. Demir and O. Jirsa James, "Behavior of Concrete Under Compressive Loadings," *Journal of the Structural Division*, vol. 95, (12), pp. 2543-2564, 1969. Available: <https://doi-org.ezproxy.itcr.ac.cr/10.1061/JSDEAG.0002424>. DOI: 10.1061/JSDEAG.0002424.
- [21] J. B. Mander, M. J. N. Priestley and R. Park, "Theoretical Stress-Strain Model for Confined Concrete," *J. Struct. Eng.*, vol. 114, (8), 1988. Available: [https://doi.org/10.1061/\(ASCE\)0733-9445\(1988\)114:8\(1804\)](https://doi.org/10.1061/(ASCE)0733-9445(1988)114:8(1804)). DOI: 10.1061/(ASCE)0733-9445(1988)114:8(1804).
- [22] K. Tsavdaridis, "Seismic analysis of Steel–Concrete composite buildings: Numerical modeling," in Anonymous 2014, . DOI: 10.1007/978-3-642-36197-5_125-1.
- [23] M. Menegotto and P. E. Pinto, "Method of analysis for cyclically loaded R.C. plane frames including changes in geometry and non-elastic behaviour of elements under combined normal force and bending," *Rapports Des Commissions De Travail AIPC = IVBH Berichte Der Arbeitskommissionen = IABSE Reports of the Working Commissions*, vol. 13, pp. 15, 1973. . DOI: 10.5169/seals-13741.
- [24] H. Romero and O. Quiroga, "Obtención Del Diagrama Momento - Curvatura En Muros De Concreto Reforzado." , Universidad Pontificia Boliviana, 2015.
- [25] P. Silva and S. Viñamagua, "Incidencia En La Ductilidad Y Capacidad Resistente De La Unión Viga-Columna De Hormigón Armado Con La Inclusión De Una Barra FRP Confinada Con Hormigón De Alta Resistencia." , Universidad Central del Ecuador, 2023.
- [26] R. Park and T. Paulay, *Reinforced Concrete Structures* . (4ª Ed ed.) México: Editorial Limusa, 1988.
- [27] A. Ospina, H. Urrego and J. C. Botero, "Application of a Method Based in Performance for the Seismic Analysis and Design of Reinforced Concrete Bridges," *Ingeniería Y Ciencia*, vol. 9, pp. 209-236, 2013.
- [28] Colegio Federado de Ingenieros y Arquitectos, Ed., *Código Sísmico De Costa Rica 2010 Revisión 2014*. (5ª Ed ed.) Cartago, Costa Rica: Editorial Tecnológica de Costa Rica, 2016.
- [29] (). *Riverbank Computing Limited*. Available: <https://www.riverbankcomputing.com>.
- [30] (). *Python* . Available: <https://www.python.org/>.
- [31] J. S. N. Valenzuela, *Python Aplicaciones Prácticas*. 2018.

[32] (). Kivy. Available: <https://kivy.org/>.

[33] S. Murat and R. Razvi Salim, "Strength and Ductility of Confined Concrete," *J. Struct. Eng.*, vol. 118, (6), pp. 1590-1607, 1992. Available: [https://doi-org.ezproxy.itcr.ac.cr/10.1061/\(ASCE\)0733-9445\(1992\)118:6\(1590\)](https://doi-org.ezproxy.itcr.ac.cr/10.1061/(ASCE)0733-9445(1992)118:6(1590)). DOI: 10.1061/(ASCE)0733-9445(1992)118:6(1590).

[34] Á Navarro, "Evaluación De La Fragilidad Ante Cargas Laterales De Tres Edificios Tipo Marco De Concreto Reforzado, Diseñados De Acuerdo Con El Código Sísmico De Costa Rica 2010. ." , Universidad de Costa Rica, 2019.

[35] A. Dazio, K. Beyer and H. Bachmann, "Quasi-static cyclic tests and plastic hinge analysis of RC structural walls," *Eng. Struct.*, vol. 31, 2009. . DOI: 10.1016/j.engstruct.2009.02.018.

[36] (). *SteelMPF - Menegotto and Pinto (1973) Model Extended by Filippou et al. (1983)*. Available: [https://opensees.berkeley.edu/wiki/index.php/SteelMPF_-_Menegotto_and_Pinto_\(1973\)_Model_Extended_by_Filippou_et_al._\(1983\)](https://opensees.berkeley.edu/wiki/index.php/SteelMPF_-_Menegotto_and_Pinto_(1973)_Model_Extended_by_Filippou_et_al._(1983)).

[37] C. B. Chadwell and R. A. Imbsen, "XTRACT: A Tool for Axial Force - Ultimate Curvature Interactions," *Structures 2004*, pp. 1-9, 2012. Available: [https://doi.org/10.1061/40700\(2004\)178](https://doi.org/10.1061/40700(2004)178). DOI: 10.1061/40700(2004)178.

Apéndices

Apéndice A. Manual de usuario.

Apéndice B. Resultados del análisis estructural del modelo 1 en XTRACT.

Apéndice C. Resultados del análisis estructural del modelo 2 en XTRACT.

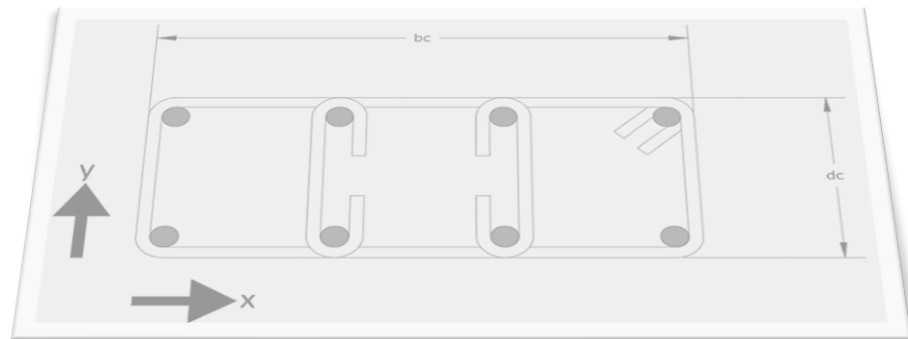
Apéndice D. Resultados del análisis estructural del modelo 3 en XTRACT.

Apéndice A

Manual de Usuario

Manual de Usuario

Herramienta computacional para la elaboración de diagramas momento-curvatura de muros con geometría compleja



Esta guía proporciona información detallada sobre el funcionamiento de la herramienta computacional, incluyendo un paso a paso para garantizar su correcto uso. Su objetivo principal es orientar al usuario en la manipulación adecuada de los componentes en la interfaz principal y las emergente.

Tabla de contenido

Instalación del programa.....	3
1. Descarga y extracción.....	3
2. Ejecución.....	3
Definición, Configuración y Análisis Estructural de Secciones.....	4
3. Definir Sección.....	5
4. Definir Refuerzo de Acero Vertical.....	6
5. Definir Método de Fibras.....	7
6. Definir Zonas Confinadas.....	8
7. Características Mecánicas del Acero de Refuerzo.....	9
8. Características Mecánicas del Concreto.....	10
8.1. Calculadora Mander.....	11
9. Realizar Análisis.....	12

Instalación del programa

1. Descarga y extracción

Para llevar a cabo la instalación de nuestro programa, es esencial contar con la carpeta que contiene el ejecutable. Asegúrate de haber descargado y extraído correctamente el archivo ZIP, y localiza la carpeta principal nombrada "Main App". Dentro de esta carpeta, verifiquemos la presencia de tres elementos clave: el archivo ejecutable de la aplicación, dos imágenes denominadas "bc_dc" y "bcx_dcy_s", y una subcarpeta guardada bajo el nombre "_internal". Esta estructura específica garantizará una instalación correcta y el pleno funcionamiento de nuestro programa.

2. Ejecución

- Sigue la ruta de carpetas como se indica a continuación:
 - [TFG J Sanchez eje] > [dist] > [Main App]
- Haz doble click en el ejecutable denominado "Main App" para iniciar la herramienta.

Definición, Configuración y Análisis Estructural de Secciones

A continuación, se muestran los pasos a seguir para ingresar y registrar los datos iniciales de manera efectiva.

The screenshot displays the 'MainWindow' interface with the following components and steps:

- Step 1:** 'Definir Sección' panel with a 'Definir Vértices' button.
- Step 2:** 'Definir Refuerzo de Acero Vertical' panel with a 'Definir Acero de Refuerzo Vertical' button.
- Step 3:** 'Definir Método de Fibras' panel with input fields for 'Número de divisiones en X' and 'Número de divisiones en Y'.
- Step 4:** 'Definir Zonas Confinadas' panel with a 'Cantidad de Zonas Confinadas' dropdown, 'Actualizar Lista' button, and input fields for 'bc', 'dc', and 'Coordenadas del centro' (X, Y).
- Step 5:** 'Características Mecánicas del Acero de Refuerzo' panel with input fields for f_{yp} , E_0 , f_{yn} , b_p , and b_n .
- Step 6:** 'Características Mecánicas del Concreto' panel with a 'Concreto Inconfinado' dropdown, 'Calculadora Mander' button, and input fields for f_{pc} , f_{pcu} , eps_u , and eps_c .
- Step 7:** A vertical sidebar on the right labeled 'Realizar Análisis'.

Buttons for 'Proyectar Curva Esfuerzo-Deformación' and 'Ayuda' are present in panels 5 and 6.

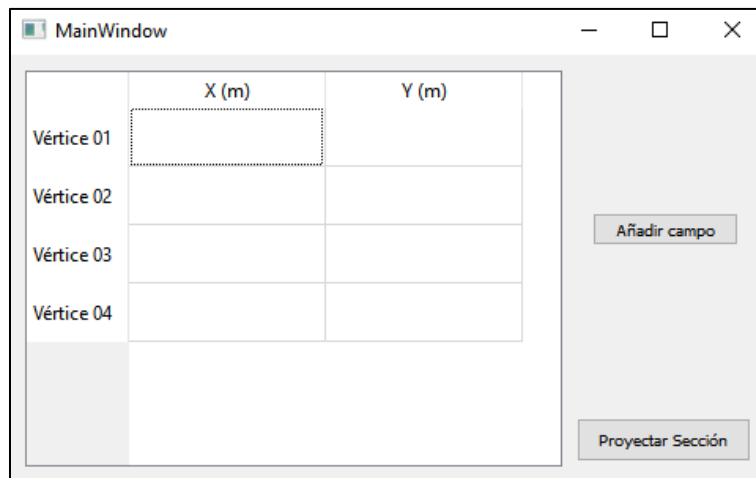
Fig. 1 Interfaz principal. Pasos para la ejecución.

3. Definir Sección

Al iniciar la herramienta, el usuario debe seleccionar el botón 'Definir Vértices' y completar las coordenadas X y Y para cada vértice en la ventana emergente. Es esencial seguir un orden que satisfaga la dirección en contra de las manecillas del reloj para garantizar la correcta disposición de los datos.

Se requiere que el usuario ingrese un mínimo de cuatro vértices para definir la sección. En caso de requerir agregar más vértices, debe seleccionar el botón 'Añadir Campo'. La cantidad de veces que deba presionar este botón corresponde a la cantidad de vértices adicionales que desea ingresar.

Una vez ingresadas las coordenadas en la tabla, el usuario debe seleccionar el botón 'Proyectar Sección' para graficar la geometría de la sección y, finalmente, cerrar la ventana emergente.



	X (m)	Y (m)
Vértice 01	<input type="text"/>	<input type="text"/>
Vértice 02	<input type="text"/>	<input type="text"/>
Vértice 03	<input type="text"/>	<input type="text"/>
Vértice 04	<input type="text"/>	<input type="text"/>

Fig. 2. Definir sección.

Nota: La sección debe ser definida de tal manera que el incremento de carga en el momento flector aplicado se realice en torno al eje Y.

4. Definir Refuerzo de Acero Vertical

Una vez que se ha definido la geometría de la sección, el usuario debe seleccionar el botón 'Definir Acero de Refuerzo Vertical'. En la ventana emergente correspondiente, se deben ingresar las coordenadas X y Y, junto con el área transversal de cada varilla de refuerzo. Similar al proceso anterior, se debe presionar el botón 'Añadir Campo' según la cantidad de varillas necesarias. Finalmente, para visualizar la posición de las varillas, se debe seleccionar 'Proyectar Sección' y cerrar la ventana emergente.



	X (m)	Y (m)	Area (cm2)
Varilla 01			

Añadir campo

Proyectar Refuerzo

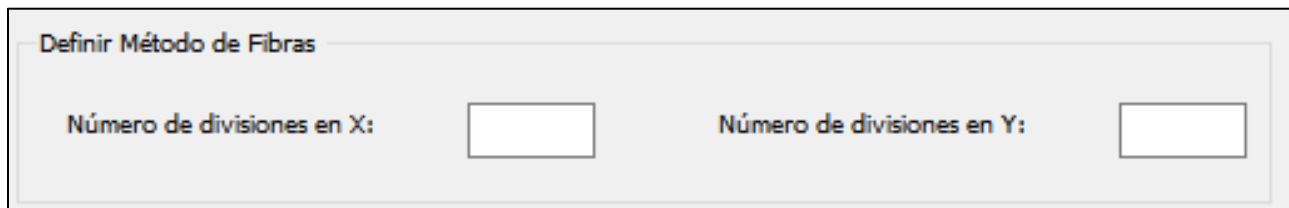
Fig. 3. Definir refuerzo de acero vertical.

5. Definir Método de Fibras

En esta fase del proyecto, el usuario debe ingresar el número de divisiones deseado tanto en el eje X como en el eje Y para definir el método de fibras.

Este número de divisiones es crucial, ya que determinará la discretización de la sección. Es importante destacar que el usuario debe asegurarse de que este número de divisiones sea suficiente para que el programa identifique de manera efectiva las zonas confinadas establecidas en el siguiente paso. Sin embargo, se enfatiza que no se debe seleccionar un número excesivo de divisiones, ya que esto podría afectar la agilidad computacional de la herramienta. Se recomienda al usuario iterar entre esta fase y la siguiente, donde se definen las zonas confinadas, para lograr un diagrama representativo de la sección.

Es responsabilidad del usuario ajustar cuidadosamente el número de divisiones, considerando el equilibrio entre la precisión del modelo y la eficiencia computacional, lo que facilitará un análisis exhaustivo y ágil del comportamiento estructural de la sección.



Definir Método de Fibras

Número de divisiones en X:

Número de divisiones en Y:

Fig. 4. Definir método de fibras.

6. Definir Zonas Confinadas

En esta fase, el usuario debe ingresar el número total de zonas confinadas presentes en la sección. Luego, se debe seleccionar el botón 'Actualizar Lista' para reflejar este número en la interfaz. En la lista generada, el usuario debe elegir la zona confinada a la cual se le van a ingresar los datos.

Es esencial destacar que la zona confinada debe estar seleccionada en la lista antes de ingresar los parámetros. El usuario debe proporcionar el ancho, el alto, y las coordenadas X y Y del centro de la zona. Una vez ingresados los datos, se debe seleccionar el botón 'Definir Zona Seleccionada'. Si el usuario omite esta acción, los datos no se guardarán.

Cabe mencionar que cambiar la zona en la lista es fundamental antes de realizar el proceso nuevamente, ya que, si se repite el procedimiento sin cambiar de zona, los nuevos datos sobrescribirán los previamente ingresados y almacenados. Como último paso, el usuario debe presionar el botón 'Proyectar Zonas Confinadas' para visualizar tanto las divisiones establecidas en el punto anterior como las zonas confinadas definidas. Para obtener orientación adicional, se ha agregado un botón de ayuda que muestra una figura explicativa sobre la disposición de los datos a ingresar.

The image shows a software dialog box titled "Definir Zonas Confinadas". It contains the following elements:

- A label "Cantidad de Zonas Confinadas:" followed by an empty text input field, a button labeled "Actualizar Lista", and a dropdown menu with a downward arrow.
- Two rows of input fields for dimensions: "bc:" followed by an empty text input field and "cm", and "dc:" followed by an empty text input field and "cm".
- A label "Coordenadas del centro:" followed by two empty text input fields labeled "X" and "Y".
- A wide button labeled "Definir Zona Seleccionada".
- A wide button labeled "Proyectar Zonas Confinadas" and a smaller button labeled "Ayuda" to its right.

Fig. 5. Definir zonas confinadas.

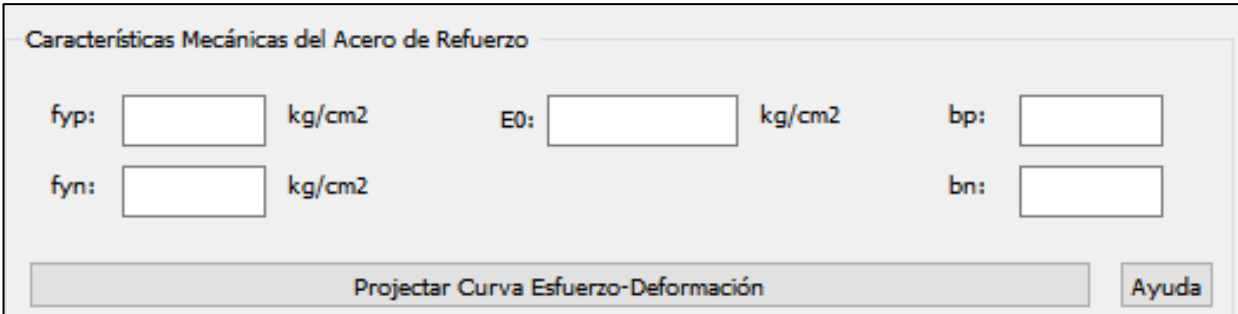
7. Características Mecánicas del Acero de Refuerzo

En esta etapa, el usuario debe ingresar las características mecánicas del acero de refuerzo. Después de proporcionar estos datos, se debe seleccionar el botón 'Proyectar Curva Esfuerzo-Deformación' para almacenar la información y visualizar la curva resultante.

Es esencial destacar que la curva proyectada es una aproximación del comportamiento del material y no representa el 100% del comportamiento real. Esto se debe a que el programa utiliza el modelo constitutivo de Menegotto-Pinto (1973) establecido para OpenSeesPy. Se recomienda al usuario tener en cuenta esta aproximación al interpretar los resultados obtenidos a través de la proyección de la curva esfuerzo-deformación.

La nomenclatura de las variables se describen a continuación:

- Fyp: resistencia a la tensión.
- Fyn: resistencia a la compresión.
- E0: módulo de elasticidad.
- bp: índice de endurecimiento a la tensión. (se recomienda bp = 0.015)
- bn: índice de endurecimiento a la compresión. (se recomienda bn = 0.015)



Características Mecánicas del Acero de Refuerzo

fyp: kg/cm² E0: kg/cm² bp:

fyn: kg/cm² bn:

Fig. 6. Características mecánicas del acero de refuerzo.

8. Características Mecánicas del Concreto

En esta sección, el usuario debe seleccionar primero el material de concreto, ya sea confinado o inconfinado, de la lista y ajustar los parámetros según sea necesario. Después de ingresar los datos, se debe seleccionar el botón 'Definir Mecánica Concreto'. Luego, se procede a seleccionar el otro material en la lista y repetir el proceso de ajuste de parámetros, seguido por la selección del mismo botón.

Una vez que ambos materiales han sido incorporados con sus respectivos datos, se debe seleccionar el botón 'Proyectar Curva Esfuerzo-Deformación'. Es importante destacar que la curva proyectada es una aproximación del comportamiento real y no refleja al 100% el comportamiento del material utilizado en la herramienta. El programa hace referencia al modelo constitutivo de Kent-Scott-Park programado en OpenSeesPy. Se recomienda al usuario tener en cuenta esta aproximación al interpretar los resultados derivados de la proyección de la curva esfuerzo-deformación.

La nomenclatura de las variables se describen a continuación:

- f_{pc} : resistencia a la compresión a los 28 días.
- f_{pcu} : resistencia al aplastamiento del concreto.
- ϵ_{psu} : deformación en el punto de resistencia a la compresión a los 28 días.
- ϵ_{psc} : deformación en el punto de resistencia al aplastamiento del concreto.

The screenshot displays a software window titled "Características Mecánicas del Concreto". At the top left, there is a dropdown menu currently showing "Concreto Inconfinado". To its right is a button labeled "Cálculadora Mander". Below these are four input fields arranged in two columns. The left column has "fpc:" and "fpcu:" labels, each followed by a text box and the unit "kg/cm2". The right column has "epsu:" and "epsc:" labels, each followed by a text box. At the bottom of the window, there are three buttons: "Definir Mecanica Concreto" (centered), "Proyectar Curva Esfuerzo-Deformación" (left-aligned), and "Ayuda" (right-aligned).

Fig. 6. Características mecánicas del concreto.

8.1. Calculadora Mander

La aplicación presenta un recurso adicional llamado 'Calculadora Mander'. Al seleccionar este botón, se abrirá una ventana emergente que brinda al usuario la capacidad de calcular el aumento en la resistencia y la ductilidad del concreto debido al confinamiento.

Dentro de la ventana de la calculadora, el usuario simplemente debe completar las casillas con los parámetros especificados y seleccionar el botón 'Calcular'. Se destaca la importancia de utilizar el botón de ayuda proporcionado para obtener una comprensión detallada de la nomenclatura y las variables utilizadas.

Esta funcionalidad adicional proporciona al usuario una herramienta práctica para realizar cálculos específicos relacionados con el confinamiento del concreto, mejorando así la comprensión de los efectos en la sección analizada.

MainWindow

Calculadora Mander

f'c: kgf/cm2 f'c: Resistencia a la compresión del concreto a los 28 días

Asx: cm2 Asx: Área de acero de refuerzo transversal paralelo al eje X.

Asy: cm2 Asy: Área de acero de refuerzo transversal paralelo al eje Y.

fyt: kgf/cm2 fyt: Resistencia a la fluencia del acero de refuerzo transversal.

s: cm bcx: cm bcy: cm

Calcular Ayuda

Resultados

f'cc:

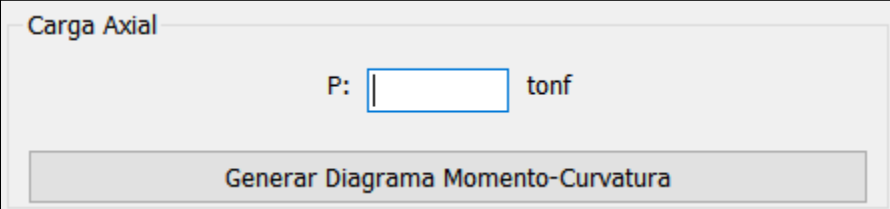
e'cc:

Fig. 7. Calculadora Mander

9. Realizar Análisis

Como paso final, el usuario deberá presionar el botón 'Realizar Análisis'. Esto abrirá una ventana emergente donde se deberá especificar la carga axial deseada. Una vez definida la carga axial, se selecciona el botón 'Generar Diagrama Momento-Curvatura' para graficar el diagrama momento-curvatura de la sección especificada por el usuario.

Este proceso permite al usuario llevar a cabo un análisis estructural completo y obtener un diagrama momento-curvatura que refleje el comportamiento de la sección bajo la carga axial definida.



Carga Axial

P: tonf

Generar Diagrama Momento-Curvatura

Fig. 8. Carga axial.

Apéndice B

Resultados del análisis estructural del modelo 1 en XTRACT.

XTRACT Analysis Report -

For use only in an academic or research setting.

Jose Pablo S.

TEC

4/12/2023

Section Name: Section1

Loading Name: 90 ton

Analysis Type: Moment Curvature

Modelo 1

Page __ of __

Section Details:

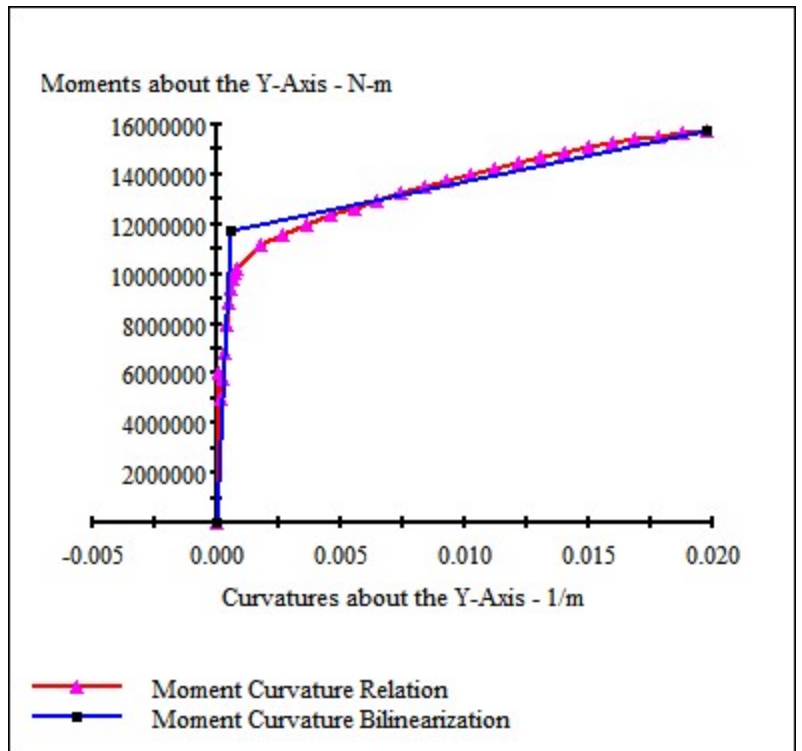
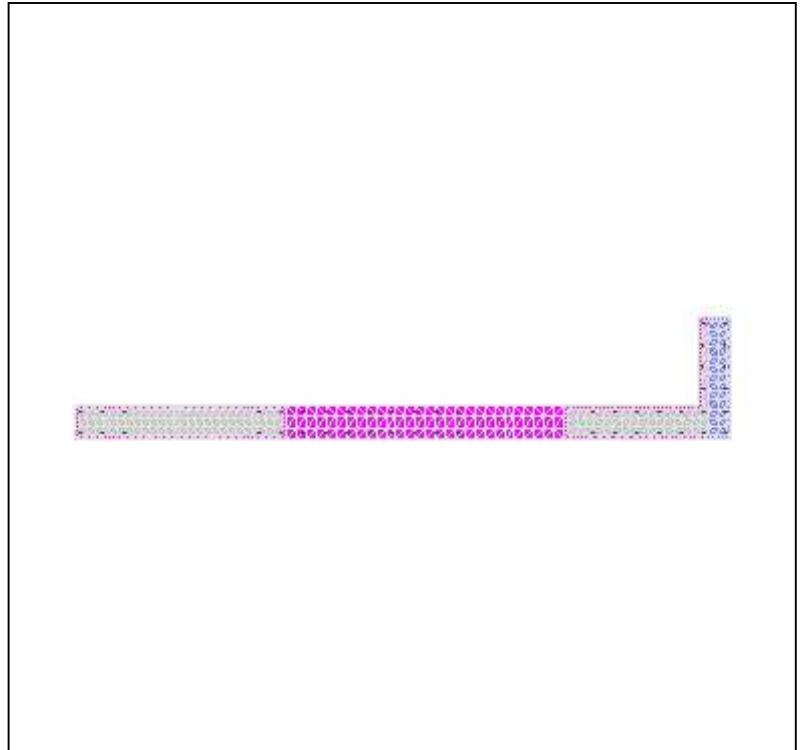
X Centroid: 3.287 m
Y Centroid: .2158 m
Section Area: 2.011 m²

Loading Details:

Constant Load - P: 882.6E+3 N
Incrementing Loads: Myy Only
Number of Points: 31
Analysis Strategy: Displacement Control

Analysis Results:

Failing Material: C. Inconfinado 280
Failure Strain: 4.000E-3 Compression
Curvature at Initial Load: -.9923E-10 1/m
Curvature at First Yield: .3963E-3 1/m
Ultimate Curvature: 19.73E-3 1/m
Moment at First Yield: 7.987E+6 N-m
Ultimate Moment: 1.57E+7 N-m
Centroid Strain at Yield: .7190E-3 Ten
Centroid Strain at Ultimate: 47.40E-3 Ten
N.A. at First Yield: 1.814 m
N.A. at Ultimate: 2.402 m
Energy per Length: 266.8E+3 N
Effective Yield Curvature: .5844E-3 1/m
Effective Yield Moment: 1.18E+7 N-m
Over Strength Factor: 1.336
EI Effective: 2.02E+10 N-m²
Yield EI Effective: 2.07E+8 N-m²
Bilinear Harding Slope: 1.027 %
Curvature Ductility: 33.76



Comments:

User Comments

XTRACT Analysis Report -

Jose Pablo S.

For use only in an academic or research setting.

TEC

Section Name: Section1

4/12/2023

Loading Name: 90 ton

Modelo 1

Analysis Type: Moment Curvature

Page __ of __

Max. C. Inconfinad 280 Strain (strain)	Min. C. Inconfinad 280 Strain (strain)	Max. C.Confinad 280 Strain (strain)	Min. C.Confinad 280 Strain (strain)	Max. Steel 4200 Strain (strain)	Min. Steel 4200 Strain (strain)	Max. C. Inconfinad 280 Stress (Pa)	Min. C. Inconfinad 280 Stress (Pa)	Max. C.Confinad 280 Stress (Pa)	Min. C.Confinad 280 Stress (Pa)
0	-17.17E-6	0	-17.17E-6	0	-17.17E-6	0	-425.5E+3	0	-425.1E+3
.3041E-3	-.1619E-3	.3002E-3	-.1579E-3	.3008E-3	-.1589E-3	0	-3.999E+6	0	-3.831E+6
.7399E-3	-.1920E-3	.7320E-3	-.1840E-3	.7332E-3	-.1862E-3	0	-4.738E+6	0	-4.445E+6
1.168E-3	-.2295E-3	1.157E-3	-.2175E-3	1.158E-3	-.2207E-3	0	-5.652E+6	0	-5.220E+6
1.593E-3	-.2706E-3	1.578E-3	-.2545E-3	1.580E-3	-.2589E-3	0	-6.645E+6	0	-6.064E+6
2.016E-3	-.3135E-3	1.997E-3	-.2934E-3	2.000E-3	-.2988E-3	0	-7.671E+6	0	-6.933E+6
2.443E-3	-.3524E-3	2.420E-3	-.3283E-3	2.423E-3	-.3348E-3	0	-8.590E+6	0	-7.699E+6
2.875E-3	-.3864E-3	2.848E-3	-.3583E-3	2.852E-3	-.3658E-3	0	-9.384E+6	0	-8.346E+6
3.311E-3	-.4162E-3	3.280E-3	-.3841E-3	3.285E-3	-.3928E-3	0	-10.07E+6	0	-8.896E+6
3.750E-3	-.4438E-3	3.714E-3	-.4077E-3	3.720E-3	-.4174E-3	0	-10.70E+6	0	-9.390E+6
4.191E-3	-.4682E-3	4.152E-3	-.4280E-3	4.158E-3	-.4388E-3	0	-11.25E+6	0	-9.810E+6
9.538E-3	-.6882E-3	9.452E-3	-.6002E-3	9.465E-3	-.6238E-3	0	-15.88E+6	0	-13.18E+6
14.93E-3	-.8623E-3	14.80E-3	-.7264E-3	14.82E-3	-.7629E-3	0	-19.04E+6	0	-15.43E+6
20.33E-3	-1.026E-3	20.15E-3	-.8424E-3	20.18E-3	-.8918E-3	0	-21.55E+6	0	-17.35E+6
25.74E-3	-1.185E-3	25.51E-3	-.9535E-3	25.55E-3	-1.016E-3	0	-23.52E+6	0	-19.05E+6
31.16E-3	-1.334E-3	30.88E-3	-1.055E-3	30.93E-3	-1.130E-3	0	-24.96E+6	0	-20.48E+6
36.58E-3	-1.474E-3	36.26E-3	-1.146E-3	36.31E-3	-1.234E-3	0	-25.98E+6	0	-21.70E+6
42.00E-3	-1.626E-3	41.63E-3	-1.250E-3	41.69E-3	-1.351E-3	0	-26.75E+6	0	-22.99E+6
47.42E-3	-1.772E-3	47.01E-3	-1.348E-3	47.07E-3	-1.462E-3	0	-27.21E+6	0	-24.12E+6
52.85E-3	-1.903E-3	52.39E-3	-1.431E-3	52.46E-3	-1.558E-3	0	-27.42E+6	0	-25.02E+6
58.28E-3	-2.048E-3	57.77E-3	-1.528E-3	57.85E-3	-1.668E-3	0	-27.45E+6	0	-26.01E+6
63.70E-3	-2.194E-3	63.14E-3	-1.626E-3	63.23E-3	-1.779E-3	0	-27.31E+6	0	-26.93E+6
69.15E-3	-2.311E-3	68.54E-3	-1.696E-3	68.64E-3	-1.861E-3	0	-27.10E+6	0	-27.55E+6
74.58E-3	-2.445E-3	73.93E-3	-1.782E-3	74.03E-3	-1.960E-3	0	-26.77E+6	0	-28.27E+6
80.00E-3	-2.590E-3	79.30E-3	-1.879E-3	79.41E-3	-2.070E-3	0	-26.33E+6	0	-29.03E+6
85.41E-3	-2.743E-3	84.67E-3	-1.984E-3	84.79E-3	-2.188E-3	0	-25.80E+6	0	-29.79E+6
90.83E-3	-2.892E-3	90.04E-3	-2.085E-3	90.16E-3	-2.302E-3	0	-25.23E+6	0	-30.48E+6
96.23E-3	-3.060E-3	95.39E-3	-2.205E-3	95.52E-3	-2.434E-3	0	-23.32E+6	0	-31.22E+6
.1016	-3.271E-3	.1007	-2.368E-3	.1008	-2.611E-3	0	-18.08E+6	0	-32.13E+6
.1069	-3.563E-3	.1059	-2.613E-3	.1061	-2.868E-3	0	-10.83E+6	0	-33.30E+6
.1120	-4.000E-3	.1110	-3.001E-3	.1112	-3.269E-3	0	-3945	0	-34.75E+6

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section1

Loading Name: 90 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 1

Page __ of __

Max. C. Inconfinad 280 Strain (strain)	Min. C. Inconfinad 280 Strain (strain)	Max. C.Confinad 280 Strain (strain)	Min. C.Confinad 280 Strain (strain)	Max. Steel 4200 Strain (strain)	Min. Steel 4200 Strain (strain)	Max. C. Inconfinad 280 Stress (Pa)	Min. C. Inconfinad 280 Stress (Pa)	Max. C.Confinad 280 Stress (Pa)	Min. C.Confinad 280 Stress (Pa)
.1120	-4.000E-3	.1110	-3.001E-3	.1112	-3.269E-3	0	0	0	-34.75E+6

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section1

Loading Name: 90 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 1

Page __ of __

Max. Steel 4200 Stress (Pa)	Min. Steel 4200 Stress (Pa)	Myy (N-m)	Kyy (1/m)
0	-3.537E+6	0	-.9923E-10
6.19E+7	-32.73E+6	6.000E+6	79.25E-6
1.51E+8	-38.34E+6	4.963E+6	.1585E-3
2.39E+8	-45.46E+6	5.764E+6	.2378E-3
3.25E+8	-53.31E+6	6.839E+6	.3170E-3
4.12E+8	-61.53E+6	7.986E+6	.3963E-3
4.12E+8	-68.94E+6	8.843E+6	.4755E-3
4.12E+8	-75.33E+6	9.400E+6	.5548E-3
4.12E+8	-80.89E+6	9.772E+6	.6340E-3
4.12E+8	-85.96E+6	1.00E+7	.7133E-3
4.12E+8	-90.37E+6	1.02E+7	.7926E-3
4.30E+8	-1.28E+8	1.12E+7	1.739E-3
4.51E+8	-1.57E+8	1.16E+7	2.686E-3
4.71E+8	-1.84E+8	1.20E+7	3.633E-3
4.90E+8	-2.09E+8	1.23E+7	4.580E-3
5.08E+8	-2.33E+8	1.27E+7	5.527E-3
5.25E+8	-2.54E+8	1.30E+7	6.473E-3
5.42E+8	-2.78E+8	1.33E+7	7.420E-3
5.58E+8	-3.01E+8	1.35E+7	8.367E-3
5.72E+8	-3.21E+8	1.38E+7	9.314E-3
5.86E+8	-3.43E+8	1.40E+7	10.26E-3
5.99E+8	-3.66E+8	1.43E+7	11.21E-3
6.12E+8	-3.83E+8	1.45E+7	12.15E-3
6.23E+8	-4.04E+8	1.47E+7	13.10E-3
6.33E+8	-4.12E+8	1.49E+7	14.05E-3
6.43E+8	-4.12E+8	1.51E+7	14.99E-3
6.51E+8	-4.12E+8	1.52E+7	15.94E-3
6.59E+8	-4.12E+8	1.54E+7	16.89E-3
6.66E+8	-4.12E+8	1.56E+7	17.84E-3
6.71E+8	-4.12E+8	1.57E+7	18.78E-3
6.76E+8	-4.12E+8	1.57E+7	19.73E-3

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section1

Loading Name: 90 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 1

Page __ of __

Max. Steel 4200 Stress (Pa)	Min. Steel 4200 Stress (Pa)	Myy (N-m)	Kyy (1/m)
6.76E+8	-4.12E+8	1.57E+7	19.73E-3

Apéndice C

Resultados del análisis estructural del modelo 2 en XTRACT.

XTRACT Analysis Report -

For use only in an academic or research setting.

Jose Pablo S.

TEC

4/12/2023

Section Name: Section2

Loading Name: 50 ton

Analysis Type: Moment Curvature

Modelo 2

Page __ of __

Section Details:

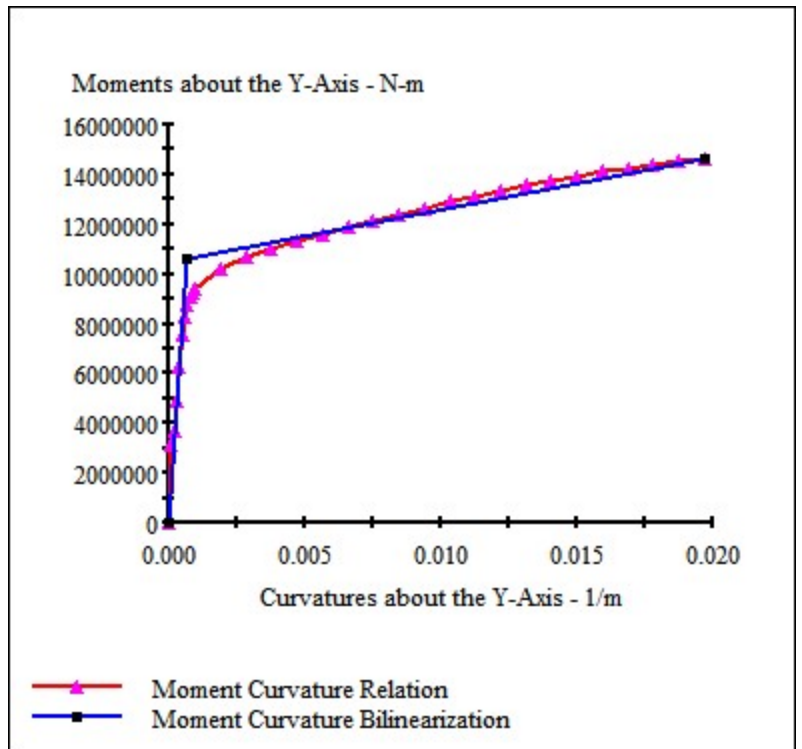
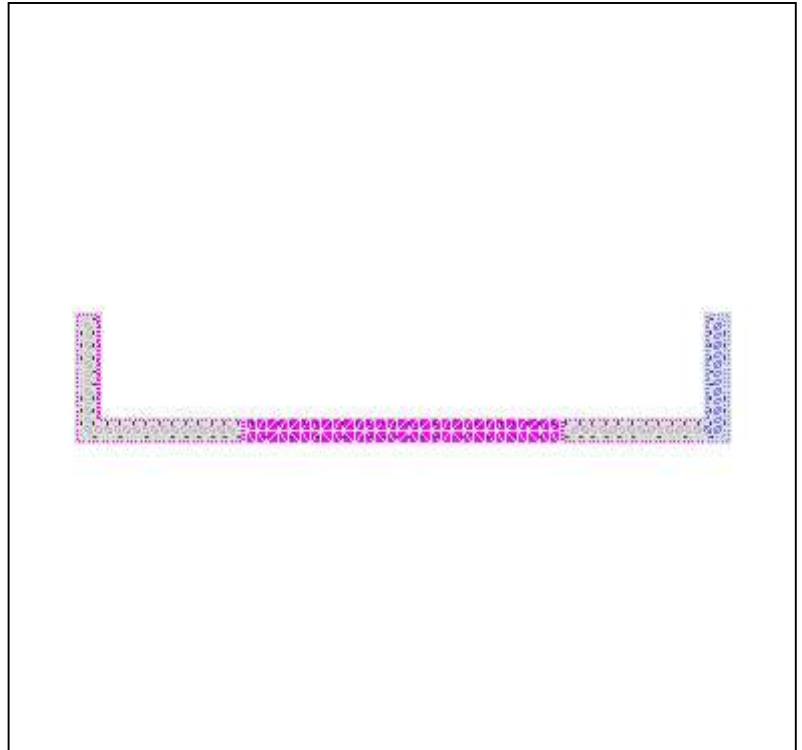
X Centroid: 2.500 m
Y Centroid: .2225 m
Section Area: 1.320 m²

Loading Details:

Constant Load - P: 444.8E+3 N
Incrementing Loads: Myy Only
Number of Points: 31
Analysis Strategy: Displacement Control

Analysis Results:

Failing Material: C.Inconfinado 250
Failure Strain: 4.000E-3 Compression
Curvature at Initial Load: .6137E-13 1/m
Curvature at First Yield: .4868E-3 1/m
Ultimate Curvature: 19.70E-3 1/m
Moment at First Yield: 7.562E+6 N-m
Ultimate Moment: 1.46E+7 N-m
Centroid Strain at Yield: .8074E-3 Ten
Centroid Strain at Ultimate: 45.02E-3 Ten
N.A. at First Yield: 1.659 m
N.A. at Ultimate: 2.285 m
Energy per Length: 244.2E+3 N
Effective Yield Curvature: .6856E-3 1/m
Effective Yield Moment: 1.07E+7 N-m
Over Strength Factor: 1.375
EI Effective: 1.55E+10 N-m²
Yield EI Effective: 2.10E+8 N-m²
Bilinear Harding Slope: 1.353 %
Curvature Ductility: 28.74



XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section2

Loading Name: 50 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 2

Page __ of __

Max. C.Confinad 250 Strain (strain)	Min. C.Confinad 250 Strain (strain)	Max. C.Inconfin 250 Strain (strain)	Min. C.Inconfin 250 Strain (strain)	Max. Steel 4200 Strain (strain)	Min. Steel 4200 Strain (strain)	Max. C.Confinad 250 Stress (Pa)	Min. C.Confinad 250 Stress (Pa)	Max. C.Inconfin 250 Stress (Pa)	Min. C.Inconfin 250 Stress (Pa)
0	-13.36E-6	0	-13.36E-6	0	-13.36E-6	0	-330.8E+3	0	-312.8E+3
.3267E-3	-1.507E-3	.3302E-3	-1.542E-3	.3265E-3	-1.505E-3	0	-3.633E+6	0	-3.595E+6
.7548E-3	-1.999E-3	.7618E-3	-2.069E-3	.7544E-3	-1.995E-3	0	-4.766E+6	0	-4.808E+6
1.174E-3	-2.585E-3	1.184E-3	-2.690E-3	1.173E-3	-2.578E-3	0	-6.075E+6	0	-6.213E+6
1.589E-3	-3.208E-3	1.603E-3	-3.348E-3	1.588E-3	-3.199E-3	0	-7.421E+6	0	-7.674E+6
2.001E-3	-3.861E-3	2.018E-3	-4.037E-3	2.000E-3	-3.851E-3	0	-8.783E+6	0	-9.161E+6
2.428E-3	-4.365E-3	2.449E-3	-4.575E-3	2.426E-3	-4.353E-3	0	-9.797E+6	0	-10.29E+6
2.863E-3	-4.781E-3	2.888E-3	-5.026E-3	2.862E-3	-4.767E-3	0	-10.61E+6	0	-11.21E+6
3.303E-3	-5.156E-3	3.331E-3	-5.436E-3	3.302E-3	-5.139E-3	0	-11.33E+6	0	-12.02E+6
3.749E-3	-5.473E-3	3.780E-3	-5.789E-3	3.747E-3	-5.455E-3	0	-11.92E+6	0	-12.70E+6
4.197E-3	-5.765E-3	4.232E-3	-6.115E-3	4.195E-3	-5.744E-3	0	-12.46E+6	0	-13.31E+6
8.589E-3	-7.760E-3	8.658E-3	-8.447E-3	8.585E-3	-7.719E-3	0	-15.87E+6	0	-17.21E+6
13.04E-3	-9.180E-3	13.14E-3	-1.020E-3	13.03E-3	-9.120E-3	0	-18.05E+6	0	-19.56E+6
17.51E-3	-1.036E-3	17.65E-3	-1.172E-3	17.50E-3	-1.028E-3	0	-19.72E+6	0	-21.16E+6
21.99E-3	-1.148E-3	22.16E-3	-1.317E-3	21.98E-3	-1.137E-3	0	-21.19E+6	0	-22.37E+6
26.48E-3	-1.252E-3	26.68E-3	-1.455E-3	26.47E-3	-1.240E-3	0	-22.47E+6	0	-23.23E+6
30.96E-3	-1.358E-3	31.20E-3	-1.595E-3	30.95E-3	-1.344E-3	0	-23.68E+6	0	-23.85E+6
35.45E-3	-1.461E-3	35.72E-3	-1.731E-3	35.44E-3	-1.445E-3	0	-24.79E+6	0	-24.24E+6
39.94E-3	-1.567E-3	40.24E-3	-1.871E-3	39.92E-3	-1.549E-3	0	-25.85E+6	0	-24.46E+6
44.43E-3	-1.669E-3	44.76E-3	-2.007E-3	44.41E-3	-1.649E-3	0	-26.82E+6	0	-24.52E+6
48.91E-3	-1.775E-3	49.28E-3	-2.147E-3	48.89E-3	-1.753E-3	0	-27.76E+6	0	-24.45E+6
53.39E-3	-1.883E-3	53.80E-3	-2.288E-3	53.37E-3	-1.859E-3	0	-28.65E+6	0	-24.27E+6
57.88E-3	-1.991E-3	58.32E-3	-2.430E-3	57.85E-3	-1.965E-3	0	-29.48E+6	0	-24.01E+6
62.35E-3	-2.106E-3	62.83E-3	-2.579E-3	62.33E-3	-2.078E-3	0	-30.32E+6	0	-23.66E+6
66.83E-3	-2.224E-3	67.33E-3	-2.730E-3	66.80E-3	-2.194E-3	0	-31.11E+6	0	-23.26E+6
71.30E-3	-2.341E-3	71.84E-3	-2.881E-3	71.27E-3	-2.309E-3	0	-31.85E+6	0	-22.81E+6
75.76E-3	-2.469E-3	76.34E-3	-3.043E-3	75.73E-3	-2.435E-3	0	-32.60E+6	0	-21.47E+6
80.20E-3	-2.622E-3	80.81E-3	-3.229E-3	80.17E-3	-2.586E-3	0	-33.42E+6	0	-17.29E+6
84.59E-3	-2.826E-3	85.23E-3	-3.467E-3	84.55E-3	-2.788E-3	0	-34.42E+6	0	-11.95E+6
88.95E-3	-3.056E-3	89.63E-3	-3.731E-3	88.91E-3	-3.016E-3	0	-35.40E+6	0	-6.039E+6
93.31E-3	-3.290E-3	94.02E-3	-3.999E-3	93.27E-3	-3.248E-3	0	-36.28E+6	0	-29.46E+3

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section2

Loading Name: 50 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 2

Page __ of __

Max. C.Confinad 250 Strain (strain)	Min. C.Confinad 250 Strain (strain)	Max. C.Inconfin 250 Strain (strain)	Min. C.Inconfin 250 Strain (strain)	Max. Steel 4200 Strain (strain)	Min. Steel 4200 Strain (strain)	Max. C.Confinad 250 Stress (Pa)	Min. C.Confinad 250 Stress (Pa)	Max. C.Inconfin 250 Stress (Pa)	Min. C.Inconfin 250 Stress (Pa)
93.34E-3	-3.291E-3	94.05E-3	-4.000E-3	93.29E-3	-3.249E-3	0	-36.29E+6	0	0

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section2

Loading Name: 50 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 2

Page __ of __

Max. Steel 4200 Stress (Pa)	Min. Steel 4200 Stress (Pa)	Myy (N-m)	Kyy (1/m)
0	-2.751E+6	0	.6137E-13
6.72E+7	-30.99E+6	3.132E+6	97.34E-6
1.55E+8	-41.09E+6	3.738E+6	.1947E-3
2.42E+8	-53.10E+6	4.944E+6	.2920E-3
3.27E+8	-65.89E+6	6.238E+6	.3893E-3
4.12E+8	-79.31E+6	7.562E+6	.4867E-3
4.12E+8	-89.64E+6	8.284E+6	.5840E-3
4.12E+8	-98.17E+6	8.729E+6	.6814E-3
4.12E+8	-1.06E+8	9.055E+6	.7787E-3
4.12E+8	-1.12E+8	9.271E+6	.8760E-3
4.12E+8	-1.18E+8	9.440E+6	.9734E-3
4.26E+8	-1.59E+8	1.02E+7	1.910E-3
4.44E+8	-1.88E+8	1.07E+7	2.846E-3
4.61E+8	-2.12E+8	1.10E+7	3.782E-3
4.77E+8	-2.34E+8	1.13E+7	4.718E-3
4.93E+8	-2.55E+8	1.16E+7	5.654E-3
5.08E+8	-2.77E+8	1.19E+7	6.590E-3
5.23E+8	-2.98E+8	1.22E+7	7.527E-3
5.37E+8	-3.19E+8	1.24E+7	8.463E-3
5.50E+8	-3.40E+8	1.27E+7	9.399E-3
5.63E+8	-3.61E+8	1.29E+7	10.34E-3
5.75E+8	-3.83E+8	1.31E+7	11.27E-3
5.86E+8	-4.05E+8	1.33E+7	12.21E-3
5.97E+8	-4.12E+8	1.36E+7	13.14E-3
6.07E+8	-4.12E+8	1.38E+7	14.08E-3
6.17E+8	-4.12E+8	1.39E+7	15.02E-3
6.26E+8	-4.12E+8	1.41E+7	15.95E-3
6.34E+8	-4.12E+8	1.43E+7	16.89E-3
6.42E+8	-4.12E+8	1.44E+7	17.82E-3
6.49E+8	-4.12E+8	1.45E+7	18.76E-3
6.56E+8	-4.12E+8	1.46E+7	19.70E-3

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section2

Loading Name: 50 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 2

Page __ of __

Max. Steel 4200 Stress (Pa)	Min. Steel 4200 Stress (Pa)	Myy (N-m)	Kyy (1/m)
6.56E+8	-4.12E+8	1.46E+7	19.70E-3

Apéndice D

Resultados del análisis estructural del modelo 3 en XTRACT.

XTRACT Analysis Report -

For use only in an academic or research setting.

Jose Pablo S.

TEC

4/12/2023

Section Name: Section3

Loading Name: 120 ton

Analysis Type: Moment Curvature

Modelo 3

Page __ of __

Section Details:

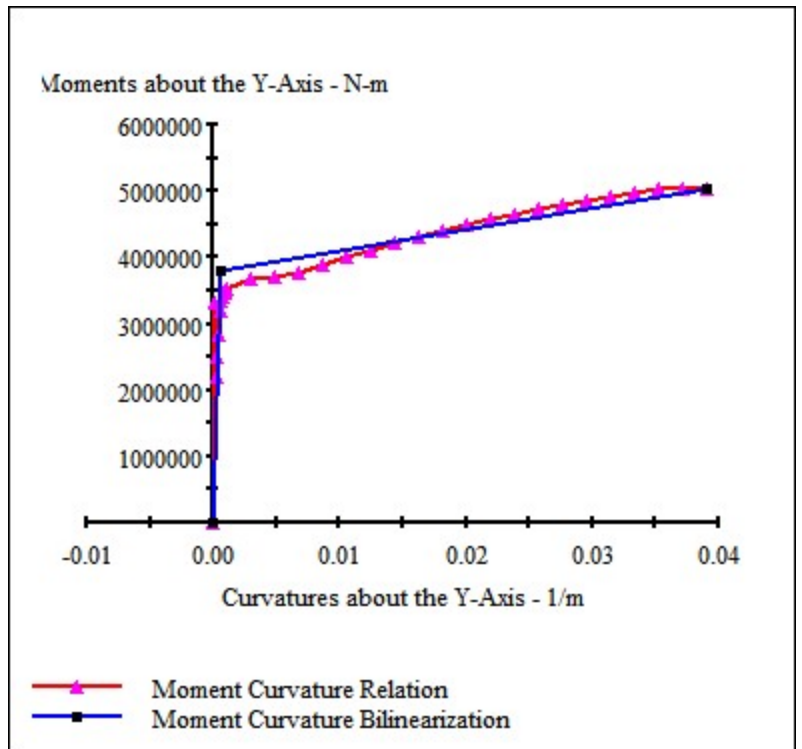
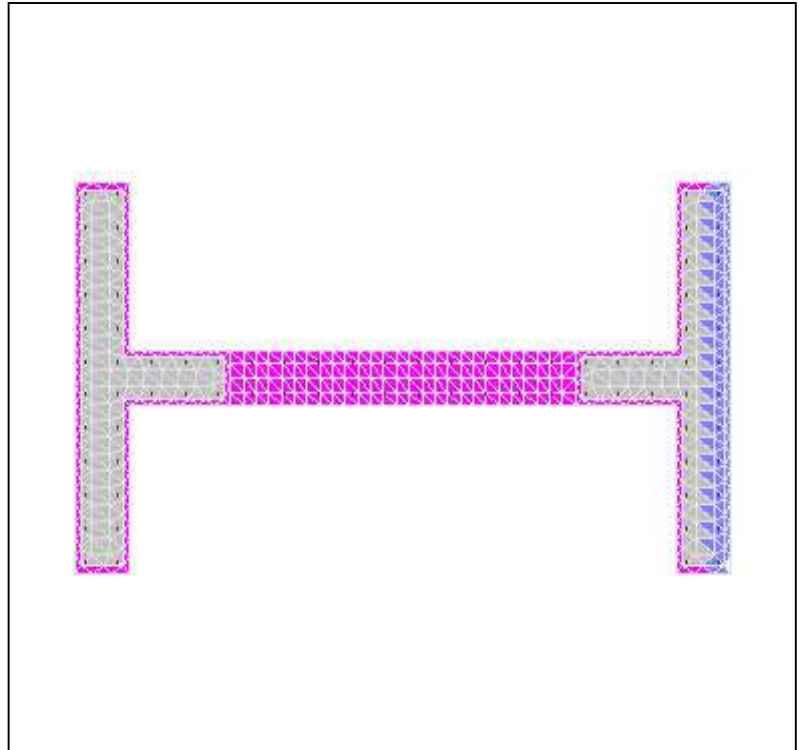
X Centroid: 1.475 m
Y Centroid: .8750 m
Section Area: 1.488 m²

Loading Details:

Constant Load - P: 1.068E+6 N
Incrementing Loads: Myy Only
Number of Points: 30
Analysis Strategy: Displacement Control

Analysis Results:

Failing Material: C. Inconfinado 210
Failure Strain: 4.000E-3 Compression
Curvature at Initial Load: -.3483E-16 1/m
Curvature at First Yield: .5502E-3 1/m
Ultimate Curvature: 39.12E-3 1/m
Moment at First Yield: 3.202E+6 N-m
Ultimate Moment: 5.050E+6 N-m
Centroid Strain at Yield: .5509E-3 Ten
Centroid Strain at Ultimate: 53.30E-3 Ten
N.A. at First Yield: 1.001 m
N.A. at Ultimate: 1.362 m
Energy per Length: 171.3E+3 N
Effective Yield Curvature: .6518E-3 1/m
Effective Yield Moment: 3.793E+6 N-m
Over Strength Factor: 1.331
EI Effective: 5.82E+9 N-m²
Yield EI Effective: 3.27E+7 N-m²
Bilinear Harding Slope: .5615 %
Curvature Ductility: 60.01



XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section3

Loading Name: 120 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 3

Page __ of __

Max. C. Confinado 210 Strain (strain)	Min. C. Confinado 210 Strain (strain)	Max. C. Inconfinado 210 Strain (strain)	Min. C. Inconfinado 210 Strain (strain)	Max. Steel 2800 Strain (strain)	Min. Steel 2800 Strain (strain)	Max. C. Confinado 210 Stress (Pa)	Min. C. Confinado 210 Stress (Pa)	Max. C. Inconfinado 210 Stress (Pa)	Min. C. Inconfinado 210 Stress (Pa)
0	-32.46E-6	0	-32.46E-6	0	-32.46E-6	0	-693.5E+3	0	-696.3E+3
.1523E-3	-.1620E-3	.1563E-3	-.1660E-3	.1519E-3	-.1616E-3	850.2E+3	-3.362E+6	764.6E+3	-3.529E+6
.4683E-3	-.1602E-3	.4763E-3	-.1682E-3	.4676E-3	-.1595E-3	0	-3.328E+6	0	-3.576E+6
.7584E-3	-.1845E-3	.7703E-3	-.1964E-3	.7572E-3	-.1833E-3	0	-3.808E+6	0	-4.163E+6
1.048E-3	-.2092E-3	1.064E-3	-.2251E-3	1.046E-3	-.2077E-3	0	-4.289E+6	0	-4.754E+6
1.336E-3	-.2349E-3	1.356E-3	-.2549E-3	1.335E-3	-.2331E-3	0	-4.784E+6	0	-5.359E+6
1.634E-3	-.2521E-3	1.658E-3	-.2760E-3	1.631E-3	-.2498E-3	0	-5.109E+6	0	-5.784E+6
1.933E-3	-.2669E-3	1.961E-3	-.2948E-3	1.930E-3	-.2642E-3	0	-5.386E+6	0	-6.158E+6
2.234E-3	-.2801E-3	2.266E-3	-.3120E-3	2.231E-3	-.2771E-3	0	-5.632E+6	0	-6.497E+6
2.536E-3	-.2922E-3	2.572E-3	-.3281E-3	2.533E-3	-.2889E-3	0	-5.856E+6	0	-6.813E+6
2.839E-3	-.3042E-3	2.879E-3	-.3441E-3	2.835E-3	-.3004E-3	0	-6.074E+6	0	-7.121E+6
8.103E-3	-.4713E-3	8.212E-3	-.5801E-3	8.093E-3	-.4610E-3	0	-8.953E+6	0	-11.31E+6
13.43E-3	-.5755E-3	13.61E-3	-.7532E-3	13.41E-3	-.5588E-3	0	-10.59E+6	0	-13.86E+6
18.80E-3	-.6387E-3	19.05E-3	-.8854E-3	18.78E-3	-.6155E-3	0	-11.52E+6	0	-15.49E+6
24.14E-3	-.7289E-3	24.46E-3	-1.044E-3	24.11E-3	-.6992E-3	0	-12.77E+6	0	-17.10E+6
29.48E-3	-.8250E-3	29.86E-3	-1.210E-3	29.44E-3	-.7888E-3	0	-14.02E+6	0	-18.38E+6
34.83E-3	-.9024E-3	35.28E-3	-1.356E-3	34.79E-3	-.8597E-3	0	-14.97E+6	0	-19.22E+6
40.21E-3	-.9571E-3	40.73E-3	-1.480E-3	40.16E-3	-.9080E-3	0	-15.60E+6	0	-19.75E+6
45.57E-3	-1.028E-3	46.16E-3	-1.620E-3	45.51E-3	-.9729E-3	0	-16.39E+6	0	-20.18E+6
50.93E-3	-1.100E-3	51.59E-3	-1.761E-3	50.86E-3	-1.038E-3	0	-17.14E+6	0	-20.44E+6
56.28E-3	-1.176E-3	57.01E-3	-1.906E-3	56.21E-3	-1.108E-3	0	-17.89E+6	0	-20.57E+6
61.63E-3	-1.258E-3	62.43E-3	-2.056E-3	61.56E-3	-1.183E-3	0	-18.66E+6	0	-20.59E+6
66.98E-3	-1.341E-3	67.85E-3	-2.208E-3	66.90E-3	-1.259E-3	0	-19.38E+6	0	-20.50E+6
72.33E-3	-1.424E-3	73.26E-3	-2.360E-3	72.24E-3	-1.336E-3	0	-20.07E+6	0	-20.34E+6
77.66E-3	-1.520E-3	78.67E-3	-2.525E-3	77.57E-3	-1.426E-3	0	-20.81E+6	0	-20.09E+6
83.00E-3	-1.618E-3	84.07E-3	-2.692E-3	82.90E-3	-1.517E-3	0	-21.51E+6	0	-19.79E+6
88.50E-3	-1.542E-3	89.65E-3	-2.685E-3	88.40E-3	-1.435E-3	0	-20.98E+6	0	-19.80E+6
93.86E-3	-1.619E-3	95.07E-3	-2.831E-3	93.74E-3	-1.505E-3	0	-21.52E+6	0	-19.51E+6
99.22E-3	-1.693E-3	.1005	-2.974E-3	99.10E-3	-1.573E-3	0	-22.02E+6	0	-19.20E+6
.1041	-2.215E-3	.1055	-3.564E-3	.1040	-2.088E-3	0	-24.86E+6	0	-8.336E+6
.1092	-2.582E-3	.1106	-4.000E-3	.1090	-2.448E-3	0	-26.30E+6	0	0

XTRACT Analysis Report -

For use only in an academic or research setting.

Section Name: Section3

Loading Name: 120 ton

Analysis Type: Moment Curvature

Jose Pablo S.

TEC

4/12/2023

Modelo 3

Page __ of __

Max. Steel 2800 Stress (Pa)	Min. Steel 2800 Stress (Pa)	Myy (N-m)	Kyy (1/m)
0	-6.687E+6	0	-.3483E-16
3.13E+7	-33.29E+6	3.316E+6	.1100E-3
9.63E+7	-32.85E+6	2.215E+6	.2200E-3
1.56E+8	-37.77E+6	2.498E+6	.3300E-3
2.16E+8	-42.78E+6	2.837E+6	.4400E-3
2.75E+8	-48.01E+6	3.202E+6	.5501E-3
2.75E+8	-51.47E+6	3.336E+6	.6601E-3
2.75E+8	-54.43E+6	3.408E+6	.7701E-3
2.75E+8	-57.08E+6	3.456E+6	.8801E-3
2.75E+8	-59.50E+6	3.487E+6	.9901E-3
2.75E+8	-61.89E+6	3.518E+6	1.100E-3
2.75E+8	-94.98E+6	3.666E+6	3.001E-3
2.75E+8	-1.15E+8	3.708E+6	4.903E-3
2.87E+8	-1.27E+8	3.770E+6	6.804E-3
3.04E+8	-1.44E+8	3.886E+6	8.705E-3
3.20E+8	-1.62E+8	4.003E+6	10.61E-3
3.36E+8	-1.77E+8	4.115E+6	12.51E-3
3.50E+8	-1.87E+8	4.218E+6	14.41E-3
3.64E+8	-2.00E+8	4.312E+6	16.31E-3
3.77E+8	-2.14E+8	4.403E+6	18.21E-3
3.90E+8	-2.28E+8	4.489E+6	20.11E-3
4.01E+8	-2.44E+8	4.577E+6	22.01E-3
4.12E+8	-2.59E+8	4.656E+6	23.91E-3
4.22E+8	-2.75E+8	4.723E+6	25.82E-3
4.31E+8	-2.75E+8	4.797E+6	27.72E-3
4.40E+8	-2.75E+8	4.863E+6	29.62E-3
4.48E+8	-2.75E+8	4.915E+6	31.52E-3
4.55E+8	-2.75E+8	4.974E+6	33.42E-3
4.61E+8	-2.75E+8	5.025E+6	35.32E-3
4.66E+8	-2.75E+8	5.035E+6	37.22E-3
4.70E+8	-2.75E+8	5.050E+6	39.12E-3

Anexo 1

Main App.py

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow
from PyQt5.QtCore import
QFile
from UI_MainApp import Ui_MainWindow

class MainWindow(QMainWindow):
    def
__init__(self):
    super(MainWindow, self).__init__()
    self.ui =
Ui_MainWindow()
    self.ui.setupUi(self)

    def closeEvent(self, event):
        #
Este método se llama cuando se intenta cerrar la ventana
        # Aquí puedes cerrar la
aplicación correctamente
        self.destroy()

if __name__ == "__main__":

app = QApplication(sys.argv)

    window = MainWindow()
    window.show()

sys.exit(app.exec())
```

Anexo 2

UI_MainApp.py

```
# -*- coding: utf-8
 -*-
```

```
#####
##
Form generated from reading UI file 'Main App.ui'
##
## Created by: Qt User Interface
Compiler version 6.6.0
##
## WARNING! All changes made in this file will be lost when
recompiling UI
file!
#####
```

```
from
PyQt5.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,
    QMetaObject, QObject,
QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PyQt5.QtGui import (QBrush, QColor,
    QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage,
QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient,
QTransform)
from PyQt5.QtWidgets import (QApplication, QComboBox, QFrame, QGroupBox,
    QLabel, QLineEdit, QMainWindow, QPushButton,
    QSizePolicy, QWidget, QVBoxLayout)
```

```
import
matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAff as
FigureCanvas
```

```
from Units import *
from Section_Designer import *
from UI_SectionApp import
Ui_SectionWindow
from UI_ReinforcementApp import Ui_ReinforcementWindow
from
UI_ConfineZoneHelp import Ui_ConfZoneHelpWindow
from UI_ConcreteHelp import
Ui_ConcreteHelpWindow
from UI_ReinforcementHelp import Ui_ReinforcementHelpWindow
from
UI_ManderCalculator import Ui_CalculadoraWindow
from UI_ResultsApp import
Ui_ResultsWindow
from OpenSeesPy_Analysis import *
```

```
class Ui_MainWindow(object):
```

```
def setupUi(self, MainWindow):
    if not MainWindow.setObjectName():
```

```
MainWindow.setObjectName(u"MainWindow")
```

```
    #Variables
```

```
self.dicZoneConf = {}
self.dicMecanicCon = {}
```

```
self.dicMecanicCon["Concreto Confinado"] = [0,0,0,0]
```

```
self.dicMecanicCon["Concreto Inconfinado"] = [0,0,0,0]
self.coord_x = []
```

```
self.coord_y = []
self.num_x_divisions = 0
self.num_y_divisions = 0
```

```

self.conf_zone_info = []
    self.fyp = 0
    self.fyn = 0
    self.E0 = 0

self.bp = 0
self.bn = 0
self.P = 0

self.reinforcing_steel = []
    self.matrix_1 = []
    self.matrix_2 = []

self.matrix_3 = []
    self.matrix_4 = []
    self.matrix_5 = []

self.matrix_6 = []

    self.canvasGeom = None
    self.canvasReinf = None

self.canvasConc = None

        MainWindow.resize(1155, 890)

self.centralwidget = QWidget(MainWindow)

self.centralwidget.setObjectName(u"centralwidget")
    self.groupBox_DefSect =
QGroupBox(self.centralwidget)

self.groupBox_DefSect.setObjectName(u"groupBox_DefSect")

self.groupBox_DefSect.setGeometry(QRect(10, 10, 251, 101))
    self.btn_DefVert =
QPushButton(self.groupBox_DefSect)

self.btn_DefVert.setObjectName(u"btn_DefVert")

self.btn_DefVert.setGeometry(QRect(10, 40, 231, 28))
    self.groupBox_DefLonStReinf =
QGroupBox(self.centralwidget)

self.groupBox_DefLonStReinf.setObjectName(u"groupBox_DefLonStReinf")

self.groupBox_DefLonStReinf.setGeometry(QRect(270, 10, 241, 101))

self.btn_DefLonStReinf = QPushButton(self.groupBox_DefLonStReinf)

self.btn_DefLonStReinf.setObjectName(u"btn_DefLonStReinf")

self.btn_DefLonStReinf.setGeometry(QRect(10, 40, 221, 28))
    self.groupBox_DefConfZon =
QGroupBox(self.centralwidget)

self.groupBox_DefConfZon.setObjectName(u"groupBox_DefConfZon")

self.groupBox_DefConfZon.setGeometry(QRect(10, 200, 501, 201))
    self.label_NumConfZon =
QLabel(self.groupBox_DefConfZon)

self.label_NumConfZon.setObjectName(u"label_NumConfZon")

self.label_NumConfZon.setGeometry(QRect(20, 30, 171, 16))
    self.lineEdit_NumConfZon =
QLineEdit(self.groupBox_DefConfZon)

self.lineEdit_NumConfZon.setObjectName(u"lineEdit_NumConfZon")

self.lineEdit_NumConfZon.setGeometry(QRect(200, 30, 51, 21))
    self.btn_ActConfZonList =
QPushButton(self.groupBox_DefConfZon)

```

```
self.btn_ActConfZonList.setObjectName(u"btn_ActConfZonList")

self.btn_ActConfZonList.setGeometry(QRect(260, 30, 91, 21))
    self.comboBox_NumConfZon =
QComboBox(self.groupBox_DefConfZon)

self.comboBox_NumConfZon.setObjectName(u"comboBox_NumConfZon")

self.comboBox_NumConfZon.setGeometry(QRect(360, 30, 131, 21))
    self.label_ConfZonWidth
= QLabel(self.groupBox_DefConfZon)

self.label_ConfZonWidth.setObjectName(u"label_ConfZonWidth")

self.label_ConfZonWidth.setGeometry(QRect(20, 80, 21, 16))
    self.lineEdit_ConfZonWith =
QLineEdit(self.groupBox_DefConfZon)

self.lineEdit_ConfZonWith.setObjectName(u"lineEdit_ConfZonWith")

self.lineEdit_ConfZonWith.setGeometry(QRect(40, 80, 61, 22))

self.label_ConfZonWidth_unit = QLabel(self.groupBox_DefConfZon)

self.label_ConfZonWidth_unit.setObjectName(u"label_ConfZonWidth_unit")

self.label_ConfZonWidth_unit.setGeometry(QRect(110, 80, 21, 16))

self.label_ConfZonHigh = QLabel(self.groupBox_DefConfZon)

self.label_ConfZonHigh.setObjectName(u"label_ConfZonHigh")

self.label_ConfZonHigh.setGeometry(QRect(20, 110, 16, 16))
    self.lineEdit_ConfZonHigh =
QLineEdit(self.groupBox_DefConfZon)

self.lineEdit_ConfZonHigh.setObjectName(u"lineEdit_ConfZonHigh")

self.lineEdit_ConfZonHigh.setGeometry(QRect(40, 110, 61, 22))

self.label_ConfZonHigh_unit = QLabel(self.groupBox_DefConfZon)

self.label_ConfZonHigh_unit.setObjectName(u"label_ConfZonHigh_unit")

self.label_ConfZonHigh_unit.setGeometry(QRect(110, 110, 21, 16))

self.label_CentCoorConfZon = QLabel(self.groupBox_DefConfZon)

self.label_CentCoorConfZon.setObjectName(u"label_CentCoorConfZon")

self.label_CentCoorConfZon.setGeometry(QRect(210, 100, 151, 16))

self.lineEdit_XCentCoorConfZon = QLineEdit(self.groupBox_DefConfZon)

self.lineEdit_XCentCoorConfZon.setObjectName(u"lineEdit_XCentCoorConfZon")

self.lineEdit_XCentCoorConfZon.setGeometry(QRect(360, 100, 51, 22))

self.lineEdit__YCentCoorConfZon = QLineEdit(self.groupBox_DefConfZon)

self.lineEdit__YCentCoorConfZon.setObjectName(u"lineEdit__YCentCoorConfZon")

self.lineEdit__YCentCoorConfZon.setGeometry(QRect(430, 100, 51, 22))
    self.label_X =
QLabel(self.groupBox_DefConfZon)
    self.label_X.setObjectName(u"label_X")

    self.label_X.setGeometry(QRect(380, 80, 16, 16))
    self.label_Y =
QLabel(self.groupBox_DefConfZon)
    self.label_Y.setObjectName(u"label_Y")

    self.label_Y.setGeometry(QRect(450, 80, 16, 16))
    self.btn_ConfZonHelp =
```



```
QPushButton(self.groupBox_DefConfZon)

self.btn_ConfZonHelp.setObjectName(u"btn_ConfZonHelp")

self.btn_ConfZonHelp.setGeometry(QRect(450, 170, 41, 21))
    self.btn_ProjConfZon =
QPushButton(self.groupBox_DefConfZon)

self.btn_ProjConfZon.setObjectName(u"btn_ProjConfZon")

self.btn_ProjConfZon.setGeometry(QRect(10, 170, 431, 21))
    self.btn_DefConfZon =
QPushButton(self.groupBox_DefConfZon)

self.btn_DefConfZon.setObjectName(u"btn_DefConfZon")

self.btn_DefConfZon.setGeometry(QRect(10, 140, 431, 21))

self.groupBox_DefMecProStReinf = QGroupBox(self.centralwidget)

self.groupBox_DefMecProStReinf.setObjectName(u"groupBox_DefMecProStReinf")

self.groupBox_DefMecProStReinf.setGeometry(QRect(520, 10, 521, 391))
    self.label_fyp =
QLabel(self.groupBox_DefMecProStReinf)

self.label_fyp.setObjectName(u"label_fyp")

self.label_fyp.setGeometry(QRect(20, 30, 21, 16))
    self.label_fyn =
QLabel(self.groupBox_DefMecProStReinf)

self.label_fyn.setObjectName(u"label_fyn")

self.label_fyn.setGeometry(QRect(20, 60, 21, 16))
    self.lineEdit_fyp =
QLineEdit(self.groupBox_DefMecProStReinf)

self.lineEdit_fyp.setObjectName(u"lineEdit_fyp")

self.lineEdit_fyp.setGeometry(QRect(50, 30, 61, 22))
    self.lineEdit_fyn =
QLineEdit(self.groupBox_DefMecProStReinf)

self.lineEdit_fyn.setObjectName(u"lineEdit_fyn")

self.lineEdit_fyn.setGeometry(QRect(50, 60, 61, 22))
    self.label_E0 =
QLabel(self.groupBox_DefMecProStReinf)

self.label_E0.setObjectName(u"label_E0")

self.label_E0.setGeometry(QRect(210, 30, 21, 20))
    self.lineEdit_E0 =
QLineEdit(self.groupBox_DefMecProStReinf)

self.lineEdit_E0.setObjectName(u"lineEdit_E0")

self.lineEdit_E0.setGeometry(QRect(230, 30, 81, 22))
    self.label_bp =
QLabel(self.groupBox_DefMecProStReinf)

self.label_bp.setObjectName(u"label_bp")

self.label_bp.setGeometry(QRect(400, 30, 21, 16))
    self.lineEdit_bp =
QLineEdit(self.groupBox_DefMecProStReinf)

self.lineEdit_bp.setObjectName(u"lineEdit_bp")

self.lineEdit_bp.setGeometry(QRect(430, 30, 61, 22))
    self.label_bn =
QLabel(self.groupBox_DefMecProStReinf)

self.label_bn.setObjectName(u"label_bn")
```

```

self.label_bn.setGeometry(QRect(400, 60, 21, 16))
    self.lineEdit_bn =
QLineEdit(self.groupBox_DefMecProStReinf)

self.lineEdit_bn.setObjectName(u"lineEdit_bn")

self.lineEdit_bn.setGeometry(QRect(430, 60, 61, 22))
    self.label_fyp_unit =
QLabel(self.groupBox_DefMecProStReinf)

self.label_fyp_unit.setObjectName(u"label_fyp_unit")

self.label_fyp_unit.setGeometry(QRect(120, 30, 41, 16))
    self.label_fyn_unit =
QLabel(self.groupBox_DefMecProStReinf)

self.label_fyn_unit.setObjectName(u"label_fyn_unit")

self.label_fyn_unit.setGeometry(QRect(120, 60, 41, 16))
    self.label_E0_unit =
QLabel(self.groupBox_DefMecProStReinf)

self.label_E0_unit.setObjectName(u"label_E0_unit")

self.label_E0_unit.setGeometry(QRect(320, 30, 41, 16))
    self.btn_StReinfHelp =
QPushButton(self.groupBox_DefMecProStReinf)

self.btn_StReinfHelp.setObjectName(u"btn_StReinfHelp")

self.btn_StReinfHelp.setGeometry(QRect(472, 100, 41, 21))
    self.btn_ProjStReinf =
QPushButton(self.groupBox_DefMecProStReinf)

self.btn_ProjStReinf.setObjectName(u"btn_ProjStReinf")

self.btn_ProjStReinf.setGeometry(QRect(10, 100, 451, 21))

#
=====

        self.frame_StReinf = QFrame(self.groupBox_DefMecProStReinf)

self.frame_StReinf.setObjectName(u"frame_StReinf")

self.frame_StReinf.setGeometry(QRect(10, 130, 501, 251))

self.frame_StReinf.setStyleSheet(u"background-color: rgb(255, 255, 255);")

self.frame_StReinf setFrameShape(QFrame.StyledPanel)

self.frame_StReinf setFrameShadow(QFrame.Raised)

        self.layout_StReinf =
QVBoxLayout(self.frame_StReinf)
        # Add Matplot Figure

#self.figure_StReinf = plt.figure(figsize=(5, 10))
        #self.ax =
self.figure_StReinf.add_subplot(111)
        #self.ax.grid(True, linestyle='--', alpha=0.5)

        #self.canvas = FigureCanvas(self.figure_StReinf)
        # Add Canvas

#self.layout_StReinf.addWidget(self.canvas)

#self.frame_StReinf.setLayout(self.layout_StReinf)
#
=====

        self.frame_SectGeom = QFrame(self.centralwidget)

```

```

self.frame_SectGeom.setObjectName(u"frame_SectGeom")

self.frame_SectGeom.setGeometry(QRect(10, 420, 501, 461))

self.frame_SectGeom.setStyleSheet(u"background-color: rgb(255, 255, 255);")

self.frame_SectGeom setFrameShape(QFrame.StyledPanel)

self.frame_SectGeom setFrameShadow(QFrame.Raised)

        self.layout_SectGeom =
QVBoxLayout(self.frame_SectGeom)
        # Add Matplot Figure

#self.figure_SectGeom = plt.figure(figsize=(5, 10))
#self.ax =
self.figure_SectGeom.add_subplot(111)
#self.ax.grid(True, linestyle='--', alpha=0.5)

#self.ax.axis('equal')
#self.canvas = FigureCanvas(self.figure_SectGeom)

# Add Canvas
#self.layout_SectGeom.addWidget(self.canvas)

#self.frame_SectGeom.setLayout(self.layout_SectGeom)
#
=====

self.groupBox_2 = QGroupBox(self.centralwidget)

self.groupBox_2.setObjectName(u"groupBox_2")

self.groupBox_2.setGeometry(QRect(520, 410, 521, 471))
        self.comboBox_Conc =
QComboBox(self.groupBox_2)
        self.comboBox_Conc.addItem("")

self.comboBox_Conc.addItem("")

self.comboBox_Conc.setObjectName(u"comboBox_Conc")

self.comboBox_Conc.setGeometry(QRect(10, 30, 371, 22))
        self.label_fc =
QLabel(self.groupBox_2)
        self.label_fc.setObjectName(u"label_fc")

self.label_fc.setGeometry(QRect(20, 70, 21, 16))
        self.lineEdit_fc =
QLineEdit(self.groupBox_2)
        self.lineEdit_fc.setObjectName(u"lineEdit_fc")

        self.lineEdit_fc.setGeometry(QRect(60, 70, 61, 22))
        self.label_ft =
QLabel(self.groupBox_2)
        self.label_ft.setObjectName(u"label_ft")

self.label_ft.setGeometry(QRect(20, 100, 31, 16))
        self.lineEdit_ft =
QLineEdit(self.groupBox_2)
        self.lineEdit_ft.setObjectName(u"lineEdit_ft")

        self.lineEdit_ft.setGeometry(QRect(60, 100, 61, 22))
        self.label_fc_unit =
QLabel(self.groupBox_2)
        self.label_fc_unit.setObjectName(u"label_fc_unit")

        self.label_fc_unit.setGeometry(QRect(130, 70, 41, 16))
        self.label_ft_unit =
QLabel(self.groupBox_2)
        self.label_ft_unit.setObjectName(u"label_ft_unit")

        self.label_ft_unit.setGeometry(QRect(130, 100, 41, 16))

```

```

        self.label_et =
QLabel(self.groupBox_2)
        self.label_et.setObjectName(u"label_et")

self.label_et.setGeometry(QRect(390, 100, 31, 16))
        self.lineEdit_et =
QLineEdit(self.groupBox_2)
        self.lineEdit_et.setObjectName(u"lineEdit_et")

        self.lineEdit_et.setGeometry(QRect(430, 100, 61, 22))
        self.label_ec =
QLabel(self.groupBox_2)
        self.label_ec.setObjectName(u"label_ec")

self.label_ec.setGeometry(QRect(390, 70, 31, 16))
        self.lineEdit_ec =
QLineEdit(self.groupBox_2)
        self.lineEdit_ec.setObjectName(u"lineEdit_ec")

        self.lineEdit_ec.setGeometry(QRect(430, 70, 61, 22))
        self.btn_ConcHelp =
QPushButton(self.groupBox_2)

self.btn_ConcHelp.setObjectName(u"btn_ConcHelp")

self.btn_ConcHelp.setGeometry(QRect(470, 160, 41, 21))
        self.btn_ProjConc =
QPushButton(self.groupBox_2)

self.btn_ProjConc.setObjectName(u"btn_ProjConc")

self.btn_ProjConc.setGeometry(QRect(10, 160, 451, 21))

#
=====

        self.frame_Conc = QFrame(self.groupBox_2)

self.frame_Conc.setObjectName(u"frame_Conc")

self.frame_Conc.setGeometry(QRect(10, 190, 501, 271))

self.frame_Conc.setStyleSheet(u"background-color: rgb(255, 255, 255);")

self.frame_Conc setFrameShape(QFrame.StyledPanel)

self.frame_Conc setFrameShadow(QFrame.Raised)

        self.layout_Conc=
QVBoxLayout(self.frame_Conc)
        # Add Matplot Figure
        #self.figure_Conc =
plt.figure(figsize=(5, 10))
        #self.ax = self.figure_Conc.add_subplot(111)

#self.ax.grid(True, linestyle='--', alpha=0.5)
        #self.canvas =
FigureCanvas(self.figure_Conc)
        # Add Canvas

#self.layout_Conc.addWidget(self.canvas)

#self.frame_Conc.setLayout(self.layout_Conc)

#
=====

        self.btn_CalcMander = QPushButton(self.groupBox_2)

self.btn_CalcMander.setObjectName(u"btn_CalcMander")

self.btn_CalcMander.setGeometry(QRect(390, 30, 121, 21))
        self.btn_DefConc =

```

```

QPushButton(self.groupBox_2)

self.btn_DefConc.setObjectName(u"btn_DefConc")

self.btn_DefConc.setGeometry(QRect(10, 130, 451, 21))
    self.btn_results =
QPushButton(self.centralwidget)

self.btn_results.setObjectName(u"btn_results")

self.btn_results.setGeometry(QRect(1050, 20, 101, 861))
    self.groupBox =
QGroupBox(self.centralwidget)
    self.groupBox.setObjectName(u"groupBox")

    self.groupBox.setGeometry(QRect(10, 120, 501, 71))
    self.label_NumDivX =
QLabel(self.groupBox)
    self.label_NumDivX.setObjectName(u"label_NumDivX")

    self.label_NumDivX.setGeometry(QRect(20, 30, 161, 16))
    self.lineEdit_NumDivX =
QLineEdit(self.groupBox)

self.lineEdit_NumDivX.setObjectName(u"lineEdit_NumDivX")

self.lineEdit_NumDivX.setGeometry(QRect(180, 30, 51, 22))
    self.label_NumDivY =
QLabel(self.groupBox)
    self.label_NumDivY.setObjectName(u"label_NumDivY")

    self.label_NumDivY.setGeometry(QRect(280, 30, 161, 16))
    self.lineEdit_NumDivY =
QLineEdit(self.groupBox)

self.lineEdit_NumDivY.setObjectName(u"lineEdit_NumDivY")

self.lineEdit_NumDivY.setGeometry(QRect(440, 30, 51, 22))

    #Clicked
Button
    self.btn_DefVert.clicked.connect(lambda x:
self.btn_DefVertClicked(MainWindow))
    self.btn_DefLonStReinf.clicked.connect(lambda x:
self.btn_DefLonStReinfClicked(MainWindow))

self.btn_ActConfZonList.clicked.connect(lambda x: self.btn_ActConfZonListClicked(MainWindow))

    self.btn_DefConfZon.clicked.connect(lambda x: self.btn_DefConfZonClicked(MainWindow))

    self.btn_ProjConfZon.clicked.connect(lambda x: self.btn_ProjConfZonClicked(MainWindow))

    self.btn_ProjStReinf.clicked.connect(lambda x:
self.btn_ProjStReinfClicked(MainWindow))
    self.btn_ProjConc.clicked.connect(lambda x:
self.btn_ProjConcClicked(MainWindow))
    self.btn_DefConc.clicked.connect(lambda x:
self.btn_DefConcClicked(MainWindow))
    self.btn_ConfZonHelp.clicked.connect(lambda x:
self.btn_ConfZonHelpClicked(MainWindow))
    self.btn_StReinfHelp.clicked.connect(lambda
x: self.btn_StReinfHelpClicked(MainWindow))
    self.btn_ConcHelp.clicked.connect(lambda
x: self.btn_ConcHelpClicked(MainWindow))
    self.btn_CalcMander.clicked.connect(lambda x:
self.btn_CalcManderClicked(MainWindow))
    self.btn_results.clicked.connect(lambda x:
self.btn_resultsClicked(MainWindow))

```

```

MainWindow.setCentralWidget(self.centralwidget)

        self.retranslateUi(MainWindow)

        QMetaObject.connectSlotsByName(MainWindow)
# setupUi

        def retranslateUi(self,
MainWindow):

MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
u"MainWindow", None))

self.groupBox_DefSect.setTitle(QCoreApplication.translate("MainWindow",
u"Definir Secci\u00f3n", None))

self.btn_DefVert.setText(QCoreApplication.translate("MainWindow", u"Definir
V\u00e9rtices", None))

self.groupBox_DefLonStReinf.setTitle(QCoreApplication.translate("MainWindow",
u"Definir Refuerzo de Acero Vertical", None))

self.btn_DefLonStReinf.setText(QCoreApplication.translate("MainWindow",
u"Definir Acero de Refuerzo Vertical", None))

self.groupBox_DefConfZon.setTitle(QCoreApplication.translate("MainWindow",
u"Definir Zonas Confinadas ", None))

self.label_NumConfZon.setText(QCoreApplication.translate("MainWindow",
u"Cantidad de Zonas Confindas:", None))

self.btn_ActConfZonList.setText(QCoreApplication.translate("MainWindow",
u"Actualizar Lista", None))

self.label_ConfZonWidth.setText(QCoreApplication.translate("MainWindow",
u"bc:", None))

self.label_ConfZonWidth_unit.setText(QCoreApplication.translate("MainWindow",
u"cm", None))

self.label_ConfZonHigh.setText(QCoreApplication.translate("MainWindow",
u"dc:", None))

self.label_ConfZonHigh_unit.setText(QCoreApplication.translate("MainWindow",
u"cm", None))

self.label_CentCoorConfZon.setText(QCoreApplication.translate("MainWindow",
u"Coordenadas del centro:", None))

self.label_X.setText(QCoreApplication.translate("MainWindow", u"X",
None))
        self.label_Y.setText(QCoreApplication.translate("MainWindow",
u"Y", None))

self.btn_ConfZonHelp.setText(QCoreApplication.translate("MainWindow",
u"Ayuda", None))

self.btn_ProjConfZon.setText(QCoreApplication.translate("MainWindow",
u"Proyectar Zonas Confinadas", None))

self.btn_DefConfZon.setText(QCoreApplication.translate("MainWindow", u"Definir
Zona Seleccionada", None))

self.groupBox_DefMecProStReinf.setTitle(QCoreApplication.translate("MainWindow",
u"Caracter\u00e9sticas Mec\u00e9nicas del Acero de Refuerzo ", None))

self.label_fyp.setText(QCoreApplication.translate("MainWindow", u"fyp:",
None))
        self.label_fyn.setText(QCoreApplication.translate("MainWindow",
u"fyn:", None))
        self.lineEdit_fyp.setText("")

self.label_E0.setText(QCoreApplication.translate("MainWindow", u"E0:",
None))

```

```

        self.label_bp.setText(QCoreApplication.translate("MainWindow",
u"bp:", None))

self.label_bn.setText(QCoreApplication.translate("MainWindow", u"bn:",
None))
        self.label_fyp_unit.setText(QCoreApplication.translate("MainWindow",
u"kg/cm2", None))

self.label_fyn_unit.setText(QCoreApplication.translate("MainWindow",
u"kg/cm2", None))

self.label_E0_unit.setText(QCoreApplication.translate("MainWindow",
u"kg/cm2", None))

self.btn_StReinfHelp.setText(QCoreApplication.translate("MainWindow",
u"Ayuda", None))

self.btn_ProjStReinf.setText(QCoreApplication.translate("MainWindow",
u"Projectar Curva Esfuerzo-Deformaci\u00f3n", None))

self.groupBox_2.setTitle(QCoreApplication.translate("MainWindow",
u"Caracter\u00e9sticas Mec\u00e9nicas del Concreto", None))

self.comboBox_Conc.setItemText(0, QCoreApplication.translate("MainWindow",
u"Concreto Inconfinado", None))
        self.comboBox_Conc.setItemText(1,
QCoreApplication.translate("MainWindow", u"Concreto Confinado", None))

        self.label_fc.setText(QCoreApplication.translate("MainWindow",
u"fpc:", None))

self.label_ft.setText(QCoreApplication.translate("MainWindow", u"fpcu:",
None))
        self.label_fc_unit.setText(QCoreApplication.translate("MainWindow",
u"kg/cm2", None))

self.label_ft_unit.setText(QCoreApplication.translate("MainWindow",
u"kg/cm2", None))

self.label_et.setText(QCoreApplication.translate("MainWindow", u"epsc:",
None))
        self.label_ec.setText(QCoreApplication.translate("MainWindow",
u"epsu:", None))

self.btn_ConcHelp.setText(QCoreApplication.translate("MainWindow",
u"Ayuda", None))

self.btn_ProjConc.setText(QCoreApplication.translate("MainWindow", u"Projectar
Curva Esfuerzo-Deformaci\u00f3n", None))

self.btn_CalcMander.setText(QCoreApplication.translate("MainWindow",
u"C\u00e9lculadora Mander", None))

self.btn_DefConc.setText(QCoreApplication.translate("MainWindow", u"Definir
Mec\u00e9nica del Concreto", None))

self.btn_results.setText(QCoreApplication.translate("MainWindow", u"Realizar
An\u00e1lisis", None))

self.groupBox.setTitle(QCoreApplication.translate("MainWindow", u"Definir
M\u00e9todo de Fibras ", None))

self.label_NumDivX.setText(QCoreApplication.translate("MainWindow",
u"N\u00famero de divisiones en X:", None))

self.label_NumDivY.setText(QCoreApplication.translate("MainWindow",
u"N\u00famero de divisiones en Y:", None))
        # retranslateUi

        #Open
Section App
        def btn_DefVertClicked(self, MainWindow):

```

```

self.SectionScreen_MW = QMainWindow()
    self.ui = Ui_SectionWindow()

self.ui.setupUi(self.SectionScreen_MW,self,self.coord_x,self.coord_y,0,0, [], [])

self.SectionScreen_MW.show()

    def btn_DefLonStReinfClicked(self, MainWindow):

        self.ReinforcementScreen_MW = QMainWindow()
        self.ui
= Ui_ReinforcementWindow()
        self.ui.setupUi(self.ReinforcementScreen_MW,self,
self.coord_x,self.coord_y,0,0, [], self.reinforcing_steel)

self.ReinforcementScreen_MW.show()

    def
on_ProcSectionDone(self,coord_x,coord_y,matrix_1, matrix_2, matrix_3, matrix_4, matrix_5,
matrix_6,plotObj):
    self.coord_x = coord_x
    self.coord_y = coord_y

self.matrix_1 = matrix_1
    self.matrix_2 = matrix_2
    self.matrix_3 = matrix_3

    self.matrix_4 = matrix_4
    self.matrix_5 = matrix_5
    self.matrix_6 =
matrix_6

    # Add Matplot Figure
    self.figure_SectGeom = plotObj
    if
self.canvasGeom is not None:
        self.canvasGeom.deleteLater()

self.canvasGeom = FigureCanvas(self.figure_SectGeom)
    # Add Canvas

self.layout_SectGeom.addWidget(self.canvasGeom)

self.frame_SectGeom.setLayout(self.layout_SectGeom)

    def
on_ProcReinforcementDone(self,reinforcing_steel,matrix_1, matrix_2, matrix_3, matrix_4,
matrix_5, matrix_6,plotObj):
    self.reinforcing_steel = reinforcing_steel

self.matrix_1 = matrix_1
    self.matrix_2 = matrix_2
    self.matrix_3 = matrix_3

    self.matrix_4 = matrix_4
    self.matrix_5 = matrix_5
    self.matrix_6 =
matrix_6

    # Add Matplot Figure
    self.figure_SectGeom = plotObj

    if self.canvasGeom is not None:
        self.canvasGeom.deleteLater()

self.canvasGeom = FigureCanvas(self.figure_SectGeom)
    # Add Canvas

self.layout_SectGeom.addWidget(self.canvasGeom)

self.frame_SectGeom.setLayout(self.layout_SectGeom)

```



```

def
btn_ActConfZonListClicked(self, MainWindow):
    zoneConfNum = 0
    #revisa si hay
    informacion ingresada en el campo
    if self.lineEdit_NumConfZon.text() :

zoneConfNum = int(self.lineEdit_NumConfZon.text())

self.comboBox_NumConfZon.clear()
self.dicZoneConf = {}
lstZoneConf = []

    for i in range(zoneConfNum):
        strZoneConf = "Zona Confinada
" + str(i + 1)
        lstZoneConf.append(strZoneConf)

self.dicZoneConf[strZoneConf] = (0, 0, 0, 0)

self.comboBox_NumConfZon.addItem(lstZoneConf)

def btn_DefConfZonClicked(self,
MainWindow):
    strZoneConfSel = self.comboBox_NumConfZon.currentText()
    bc = 0

    dc = 0
    cent_coord_x = 0
    cent_coord_y = 0
    if
self.lineEdit_ConfZonWith.text() :
        bc =
float(self.lineEdit_ConfZonWith.text()*m
    if self.lineEdit_ConfZonHigh.text() :

        dc = float(self.lineEdit_ConfZonHigh.text()*m
            if
self.lineEdit_XCentCoorConfZon.text() :
                cent_coord_x =
float(self.lineEdit_XCentCoorConfZon.text()*m
                    if
self.lineEdit__YCentCoorConfZon.text() :
                        cent_coord_y =
float(self.lineEdit__YCentCoorConfZon.text()*m

self.dicZoneConf[strZoneConfSel] = (bc, dc, cent_coord_x, cent_coord_y)

def
btn_ProcConfZonClicked(self, MainWindow):
    self.conf_zone_info = []

npReinforcing_steel = []
    for zoneConfKey in self.dicZoneConf:

self.conf_zone_info.append(self.dicZoneConf[zoneConfKey])
    num_confine_zones =
len(self.conf_zone_info)

    self.num_x_divisions = 0
    self.num_y_divisions = 0

    if self.lineEdit_NumDivX.text() :
        self.num_x_divisions =
int(self.lineEdit_NumDivX.text())
        if self.lineEdit_NumDivY.text() :

self.num_y_divisions = int(self.lineEdit_NumDivY.text())

```

```

updateVariables(self.conf_zone_info,self.reinforcing_steel)
    self.matrix_1,
self.matrix_2, self.matrix_3, self.matrix_4, self.matrix_5, self.matrix_6, plotObj =
fiber_method(self.coord_x, self.coord_y, self.num_x_divisions,
self.num_y_divisions,num_confine_zones )

    # Add Matplot Figure

self.figure_SectGeom = plotObj
    if self.canvasGeom is not None:

self.canvasGeom .deleteLater()
    self.canvasGeom = FigureCanvas(self.figure_SectGeom)

    # Add Canvas
    self.layout_SectGeom.addWidget(self.canvasGeom )

self.frame_SectGeom.setLayout(self.layout_SectGeom)

self.figure_SectGeom.canvas.mpl_connect('scroll_event', self.on_scroll)

self.figure_SectGeom.canvas.mpl_connect('button_press_event', self.on_pan_start)

self.figure_SectGeom.canvas.mpl_connect('motion_notify_event', self.on_pan)

self.pan_start = None

    def on_scroll(self, event):
        ax =
event.inaxes
        if ax is not None:
            cur_xlim = ax.get_xlim()

cur_ylim = ax.get_ylim()

            zoom_factor = 1.2

            if event.button ==
'up':
                new_xlim = [x / zoom_factor for x in cur_xlim]
                new_ylim
= [y / zoom_factor for y in cur_ylim]
                elif event.button == 'down':

new_xlim = [x * zoom_factor for x in cur_xlim]
                new_ylim = [y * zoom_factor
for y in cur_ylim]
                else:
                    return

ax.set_xlim(new_xlim)
            ax.set_ylim(new_ylim)

self.figure_SectGeom.canvas.draw()

    def on_pan_start(self, event):
        if
event.button == 1: # Botón izquierdo del mouse
            self.pan_start = event.xdata,
event.ydata

    def on_pan(self, event):
        if self.pan_start is None:

return

            if event.button == 1: # Botón izquierdo del mouse
                dx =
event.xdata - self.pan_start[0]
                dy = event.ydata - self.pan_start[1]

```

```

pan_speed = 0.5 # Ajusta este valor para controlar la velocidad de la panorámica

dx *= pan_speed
dy *= pan_speed

    ax = event.inaxes
    if
ax is not None:
    cur_xlim = ax.get_xlim()
    cur_ylim =
ax.get_ylim()

        new_xlim = cur_xlim[0] - dx, cur_xlim[1] - dx
new_ylim = cur_ylim[0] - dy, cur_ylim[1] - dy

        ax.set_xlim(new_xlim)

ax.set_ylim(new_ylim)

        self.figure_SectGeom.canvas.draw()

self.pan_start = event.xdata, event.ydata

def updateMomentCurvature(self,
MainWindow):
    updateVarAceroRefuerzo(self.fyp,self.fyn,self.E0,self.bp,self.bn)

    updateVarMatrix(self.matrix_1,self.matrix_2,self.matrix_3,self.matrix_4,self.matrix_5,self.mat
rix_6, self.reinforcing_steel)

def btn_ProjStReinfClicked(self,
MainWindow):
    self.fyp = 0
    self.fyn = 0
    self.E0 = 0
    self.bp
= 0
    self.bn = 0
    if self.lineEdit_fyp.text() :
        self.fyp =
float(self.lineEdit_fyp.text()*kgfcm2
    if self.lineEdit_fyn.text() :

self.fyn = float(self.lineEdit_fyn.text()*kgfcm2
    if self.lineEdit_E0.text() :

        self.E0 = float(self.lineEdit_E0.text()*kgfcm2
    if self.lineEdit_bp.text() :

        self.bp = float(self.lineEdit_bp.text())
    if self.lineEdit_bn.text()

:
        self.bn = float(self.lineEdit_bn.text())

self.updateMomentCurvature(MainWindow)
plotObj = showReinforcement()

# Add
Matplot Figure
self.figure_StReinf = plotObj
if self.canvasReinf is not
None:
    self.canvasReinf.deleteLater()
    self.canvasReinf =
FigureCanvas(self.figure_StReinf)
# Add Canvas

self.layout_StReinf.addWidget(self.canvasReinf)

```

```
self.frame_StReinf.setLayout(self.layout_StReinf)
```

```
def
```

```
btn_ProjConcClicked(self, MainWindow):
```

```
    lstConf =
```

```
self.dicMecanicCon["Concreto Confinado"]
```

```
    fpcc = lstConf[0]
```

```
    epcc =
```

```
lstConf[1]
```

```
    fpcu = lstConf[2]
```

```
    epsu = lstConf[3]
```

```
updateVarConcConfinado(fpcc, epcc, fpcu, epsu)
```

```
    lstInconf =
```

```
self.dicMecanicCon["Concreto Inconfinado"]
```

```
    fpcc = lstInconf[0]
```

```
epcc = lstInconf[1]
```

```
    fpcu = lstInconf[2]
```

```
    epsu = lstInconf[3]
```

```
updateVarConcInconfinado(fpcc, epcc, fpcu, epsu)
```

```
self.updateMomentCurvature(MainWindow)
```

```
    plotObj = showConcrete()
```

```
    # Add
```

```
Matplot Figure
```

```
    self.figure_Conc = plotObj
```

```
    if self.canvasConc is not None:
```

```
        self.canvasConc.deleteLater()
```

```
        self.canvasConc =
```

```
FigureCanvas(self.figure_Conc)
```

```
    # Add Canvas
```

```
self.layout_Conc.addWidget(self.canvasConc)
```

```
self.frame_Conc.setLayout(self.layout_Conc)
```

```
def btn_DefConcClicked(self,
```

```
MainWindow):
```

```
    strConcSel = self.comboBox_Conc.currentText()
```

```
    fpcc = 0
```

```
    epcc = 0
```

```
    fpcu = 0
```

```
    epsu = 0
```

```
    if self.lineEdit_fc.text() :
```

```
        fpcc = -float(self.lineEdit_fc.text()*kgfcm2
```

```
        if self.lineEdit_ec.text() :
```

```
            epcc = -float(self.lineEdit_ec.text())
```

```
            if self.lineEdit_ft.text() :
```

```
                fpcu = -float(self.lineEdit_ft.text()*kgfcm2
```

```
                if self.lineEdit_et.text() :
```

```
                    epsu = -float(self.lineEdit_et.text())
```

```
                self.dicMecanicCon[strConcSel] =
```

```
[fpcc, epcc, fpcu, epsu]
```

```
def btn_ConfZonHelpClicked(self, MainWindow):
```

```
self.ConfZoneHelpScreen_MW = QMainWindow()
```

```
    self.ui = Ui_ConfZoneHelpWindow()
```

```

self.ui.setupUi(self.ConfZoneHelpScreen_MW)
self.ConfZoneHelpScreen_MW.show()

def btn_StReinfHelpClicked(self, MainWindow):
    self.ReinforcementHelpScreen_MW =
QMainWindow()
    self.ui = Ui_ReinforcementHelpWindow()

self.ui.setupUi(self.ReinforcementHelpScreen_MW)

self.ReinforcementHelpScreen_MW.show()

def btn_ConcHelpClicked(self, MainWindow):

self.ConcreteHelpScreen_MW = QMainWindow()
    self.ui = Ui_ConcreteHelpWindow()

self.ui.setupUi(self.ConcreteHelpScreen_MW)
    self.ConcreteHelpScreen_MW.show()

def btn_CalcManderClicked(self, MainWindow):
    self.CalculadoraScreen_MW =
QMainWindow()
    self.ui = Ui_CalculadoraWindow()

self.ui.setupUi(self.CalculadoraScreen_MW)
    self.CalculadoraScreen_MW.show()

def
btn_resultsClicked(self, MainWindow):
    self.ResultsScreen_MW = QMainWindow()

self.ui = Ui_ResultsWindow()
    self.ui.setupUi(self.ResultsScreen_MW, self)

self.ResultsScreen_MW.show()

def showResultsCurvature(self,P):
    self.P =
P
    updatePResults(self.P)
    self.updateMomentCurvature(self)

defineCharacteristics()
    defineModel()

    coord_xSorted =
sorted(self.coord_x)
    coordXMax = coord_xSorted[len(coord_xSorted) -1]

coordXMin = coord_xSorted[0]
    maxCurv = 0.12/(coordXMax-coordXMin)

MomentCurvature(1, P, maxCurv, numIncr)
    return showMomentCurvature()

```

Anexo 3

UI_SectionApp.py

```
# -*- coding: utf-8
```

```
 -*-
```

```
#####
```

```
##  
Form generated from reading UI file 'Section App.ui'
```

```
##  
## Created by: Qt User Interface  
Compiler version 6.6.0
```

```
##  
## WARNING! All changes made in this file will be lost when  
recompiling UI  
file!
```

```
#####
```

```
from  
PyQt5.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,  
    QMetaObject, QObject,  
    QPoint, QRect,  
    QSize, QTime, QUrl, Qt, Signal)  
from PyQt5.QtGui import (QBrush, QColor,  
    QConicalGradient, QCursor,  
    QFont, QFontDatabase, QGradient, QIcon,  
    QImage,  
    QKeySequence, QLinearGradient, QPainter,  
    QPalette, QPixmap, QRadialGradient,  
    QTransform)  
from PyQt5.QtWidgets import (QApplication, QHeaderView, QMainWindow,  
    QPushButton,  
    QSizePolicy, QTableWidget, QTableWidgetItem, QWidget)
```

```
from Units import
```

```
*
```

```
from Section_Designer import *
```

```
class Ui_SectionWindow(object):
```

```
    def setupUi(self,  
SectionWindow,MainWindow,coord_x,coord_y, num_x_divisions, num_y_divisions, conf_zone_info,  
reinforcing_steel):
```

```
        if not SectionWindow.setObjectName():
```

```
SectionWindow.setObjectName(u"SectionWindow")
```

```
SectionWindow.resize(482,
```

```
274)
```

```
        #parameters
```

```
        self.MainWindow = MainWindow
```

```
        self.coord_x
```

```
= coord_x
```

```
        self.coord_y = coord_y
```

```
        self.num_x_divisions = num_x_divisions
```

```
        self.num_y_divisions = num_y_divisions
```

```
        self.conf_zone_info = conf_zone_info
```

```
        self.num_confine_zones = len(conf_zone_info)
```

```
        self.reinforcing_steel =
```

```
reinforcing_steel
```

```
        self.centralwidget = QWidget(SectionWindow)
```

```
self.centralwidget.setObjectName(u"centralwidget")
```

```
self.table_DefSectVert =
```

```
QTableWidget(self.centralwidget)
```

```
if (self.table_DefSectVert.columnCount() < 2):
```

```
self.table_DefSectVert.setColumnCount(2)
```

```
__qtablewidgetitem =
```

```
QTableWidgetItem()
```

```
self.table_DefSectVert.setHorizontalHeaderItem(0,
```

```
__qtablewidgetitem)
```

```
__qtablewidgetitem1 = QTableWidgetItem()
```

```

self.table_DefSectVert.setHorizontalHeaderItem(1, __qtablewidgetitem1)
    if
(self.table_DefSectVert.rowCount() < 4):

self.table_DefSectVert.setRowCount(4)
    __qtablewidgetitem2 = QTableWidgetItem()

    self.table_DefSectVert.setVerticalHeaderItem(0, __qtablewidgetitem2)
__qtablewidgetitem3 = QTableWidgetItem()

self.table_DefSectVert.setVerticalHeaderItem(1, __qtablewidgetitem3)
__qtablewidgetitem4 = QTableWidgetItem()

self.table_DefSectVert.setVerticalHeaderItem(2, __qtablewidgetitem4)
__qtablewidgetitem5 = QTableWidgetItem()

self.table_DefSectVert.setVerticalHeaderItem(3, __qtablewidgetitem5)
    font = QFont()

        font.setUnderline(True)
        __qtablewidgetitem6 = QTableWidgetItem()

__qtablewidgetitem6.setFont(font);
        self.table_DefSectVert.setItem(0, 0,
__qtablewidgetitem6)

self.table_DefSectVert.setObjectName(u"table_DefSectVert")

self.table_DefSectVert.setGeometry(QRect(10, 10, 341, 251))
        self.btn_SectAddRow =
QPushButton(self.centralwidget)

self.btn_SectAddRow.setObjectName(u"btn_SectAddRow")

self.btn_SectAddRow.setGeometry(QRect(370, 100, 93, 21))
        self.btn_ProjSect =
QPushButton(self.centralwidget)

self.btn_ProjSect.setObjectName(u"btn_ProjSect")

self.btn_ProjSect.setGeometry(QRect(360, 230, 111, 28))

        #Clicked Button

self.btn_ProjSect.clicked.connect(lambda x: self.btn_ProjSectClicked(SectionWindow))
self.btn_SectAddRow.clicked.connect(lambda x: self.btn_SectAddRowClicked(SectionWindow))

        SectionWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(SectionWindow)

        QMetaObject.connectSlotsByName(SectionWindow)

#Update table
    if len(self.coord_x) <= self.table_DefSectVert.rowCount():

        for i in range(len(self.coord_x)):
            tempQtablewidgetitemX =
QTableWidgetItem()
                tempQtablewidgetitemX.setText(str(self.coord_x[i]))

            self.table_DefSectVert.setItem(i,0,tempQtablewidgetitemX)

tempQtablewidgetitemY = QTableWidgetItem()

tempQtablewidgetitemY.setText(str(self.coord_y[i]))

```



```

self.table_DefSectVert.setItem(i,1,tempQtablewidgetitemY)
    if len(self.coord_x) >
self.table_DefSectVert.rowCount():
    for i in range(len(self.coord_x) -
self.table_DefSectVert.rowCount()):

self.btn_SectAddRowClicked(SectionWindow)

    # setupUi

    def
retranslateUi(self, SectionWindow):

SectionWindow.setWindowTitle(QCoreApplication.translate("SectionWindow",
u"MainWindow", None))
    __qtablewidgetitem =
self.table_DefSectVert.horizontalHeaderItem(0)

__qtablewidgetitem.setText(QCoreApplication.translate("SectionWindow", u"X
(m)", None));
    __qtablewidgetitem1 =
self.table_DefSectVert.horizontalHeaderItem(1)

__qtablewidgetitem1.setText(QCoreApplication.translate("SectionWindow", u"Y
(m)", None));
    __qtablewidgetitem2 =
self.table_DefSectVert.verticalHeaderItem(0)

__qtablewidgetitem2.setText(QCoreApplication.translate("SectionWindow",
u"V\u00e9rtice 01", None));
    __qtablewidgetitem3 =
self.table_DefSectVert.verticalHeaderItem(1)

__qtablewidgetitem3.setText(QCoreApplication.translate("SectionWindow",
u"V\u00e9rtice 02", None));
    __qtablewidgetitem4 =
self.table_DefSectVert.verticalHeaderItem(2)

__qtablewidgetitem4.setText(QCoreApplication.translate("SectionWindow",
u"V\u00e9rtice 03", None));
    __qtablewidgetitem5 =
self.table_DefSectVert.verticalHeaderItem(3)

__qtablewidgetitem5.setText(QCoreApplication.translate("SectionWindow",
u"V\u00e9rtice 04", None));

    __sortingEnabled =
self.table_DefSectVert.isSortingEnabled()

self.table_DefSectVert.setSortingEnabled(False)

self.table_DefSectVert.setSortingEnabled(__sortingEnabled)

self.btn_SectAddRow.setText(QCoreApplication.translate("SectionWindow",
u"A\u00f1adir campo", None))

self.btn_ProjSect.setText(QCoreApplication.translate("SectionWindow",
u"Proyectar Secci\u00f3n", None))

    # retranslateUi

    def
btn_ProjSectClicked(self, SectionWindow):

        self.coord_x = []
        self.coord_y =
[]
        for i in range(self.table_DefSectVert.rowCount()):
            cellX =
self.table_DefSectVert.item(i,0)
            cellY = self.table_DefSectVert.item(i,1)

            self.coord_x.append(float(cellX.text())*m)

```

```

self.coord_y.append(float(cellyY.text())*m)

updateVariables(self.conf_zone_info,self.reinforcing_steel)
    matrix_1, matrix_2,
matrix_3, matrix_4, matrix_5, matrix_6,plotObj = fiber_method(self.coord_x, self.coord_y,
self.num_x_divisions, self.num_y_divisions, self.num_confine_zones)

self.MainWindow.on_ProcSectionDone(self.coord_x,self.coord_y,matrix_1, matrix_2, matrix_3,
matrix_4, matrix_5, matrix_6,plotObj)

    def btn_SectAddRowClicked(self, SectionWindow):

        newRowInt = self.table_DefSectVert.rowCount()

self.table_DefSectVert.insertRow(newRowInt)

        tempQtablewidgetitemX =
QTableWidgetItem()
        tempQtablewidgetitemX.setText(str(0))

self.table_DefSectVert.setItem(newRowInt,0,tempQtablewidgetitemX)

tempQtablewidgetitemY = QTableWidgetItem()
        tempQtablewidgetitemY.setText(str(0))

        self.table_DefSectVert.setItem(newRowInt,1,tempQtablewidgetitemY)

        newRowStr =
str(newRowInt + 1)
        newRowStr = newRowStr.zfill(2)

        newRowHeader =
QTableWidgetItem()
        newRowHeader.setText("Vertice " + newRowStr);

        self.table_DefSectVert.setVerticalHeaderItem(newRowInt,  newRowHeader)

```

Anexo 4

UI_ReinforcementApp.py

```
# -*- coding: utf-8
```

```
 -*-
```

```
#####
```

```
##
```

```
Form generated from reading UI file 'Reinforcement App.ui'
```

```
##
```

```
## Created by: Qt User
```

```
Interface Compiler version 6.6.0
```

```
##
```

```
## WARNING! All changes made in this file will be lost
```

```
when recompiling UI
```

```
file!
```

```
#####
```

```
from
PyQt5.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,
    QMetaObject, QObject,
QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PyQt5.QtGui import (QBrush, QColor,
    QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage,
QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient,
QTransform)
from PyQt5.QtWidgets import (QApplication, QHeaderView, QMainWindow,
    QPushButton,
    QSizePolicy, QTableWidget, QTableWidgetItem, QWidget)
```

```
from Units import
```

```
*
```

```
from Section_Designer import *
```

```
class Ui_ReinforcementWindow(object):
```

```
    def
```

```
    setupUi(self, ReinforcementWindow,MainWindow, coord_x,coord_y, num_x_divisions,
num_y_divisions, conf_zone_info, reinforcing_steel):
```

```
        if not
```

```
ReinforcementWindow.setObjectName():
```

```
ReinforcementWindow.setObjectName(u"ReinforcementWindow")
```

```
        #Parameters
```

```
        self.MainWindow = MainWindow
```

```
        self.coord_x = coord_x
```

```
        self.coord_y =
```

```
coord_y
```

```
        self.num_x_divisions = num_x_divisions
```

```
        self.num_y_divisions =
```

```
num_y_divisions
```

```
        self.conf_zone_info = conf_zone_info
```

```
        self.num_confine_zones =
```

```
len(conf_zone_info)
```

```
        self.reinforcing_steel = reinforcing_steel
```

```
ReinforcementWindow.resize(575, 274)
```

```
        self.centralwidget =
```

```
QWidget(ReinforcementWindow)
```

```
self.centralwidget.setObjectName(u"centralwidget")
```

```
        self.table_DefLonStReinf
```

```
= QTableWidgetItem(self.centralwidget)
```

```
        if (self.table_DefLonStReinf.columnCount() <
```

```
3):
```

```
            self.table_DefLonStReinf.setColumnCount(3)
```

```
            __qtablewidgetitem =
```

```
QTableWidgetItem()
```

```
            self.table_DefLonStReinf.setHorizontalHeaderItem(0,
```

```
__qtablewidgetitem)
```

```
            __qtablewidgetitem1 = QTableWidgetItem()
```

```

self.table_DefLonStReinf.setHorizontalHeaderItem(1, __qtablewidgetitem1)
__qtablewidgetitem2 = QTableWidgetItem()
self.table_DefLonStReinf.setHorizontalHeaderItem(2, __qtablewidgetitem2)
    if
(self.table_DefLonStReinf.rowCount() < 1):
self.table_DefLonStReinf.setRowCount(1)
    __qtablewidgetitem3 = QTableWidgetItem()
    self.table_DefLonStReinf.setVerticalHeaderItem(0, __qtablewidgetitem3)
    font =
QFont()
    font.setUnderline(True)
    __qtablewidgetitem4 = QTableWidgetItem()
    __qtablewidgetitem4.setFont(font);
    self.table_DefLonStReinf.setItem(0, 0,
__qtablewidgetitem4)
self.table_DefLonStReinf.setObjectName(u"table_DefLonStReinf")
self.table_DefLonStReinf.setGeometry(QRect(10, 10, 431, 261))
self.btn_LonStReinfAddRow = QPushButton(self.centralwidget)
self.btn_LonStReinfAddRow.setObjectName(u"btn_LonStReinfAddRow")
self.btn_LonStReinfAddRow.setGeometry(QRect(460, 130, 101, 21))
self.btn_ProjLonStReinf = QPushButton(self.centralwidget)
self.btn_ProjLonStReinf.setObjectName(u"btn_ProjLonStReinf")
self.btn_ProjLonStReinf.setGeometry(QRect(450, 240, 121, 28))
ReinforcementWindow.setCentralWidget(self.centralwidget)

    #Clicked button

self.btn_ProjLonStReinf.clicked.connect(lambda x:
self.btn_ProjLonStReinfClicked(ReinforcementWindow))

self.btn_LonStReinfAddRow.clicked.connect(lambda x:
self.btn_LonStReinfAddRowClicked(ReinforcementWindow))

    #Update table
    if
len(self.reinforcing_steel) <= self.table_DefLonStReinf.rowCount():
        for i in
range(len(self.reinforcing_steel)):
            tempQtablewidgetitemX =
QTableWidgetItem()
tempQtablewidgetitemX.setText(str(self.reinforcing_steel[i][0]))
self.table_DefLonStReinf.setItem(i,0,tempQtablewidgetitemX)

tempQtablewidgetitemY = QTableWidgetItem()
tempQtablewidgetitemY.setText(str(self.reinforcing_steel[i][1]))
self.table_DefLonStReinf.setItem(i,1,tempQtablewidgetitemY)

tempQtablewidgetitemArea = QTableWidgetItem()
tempQtablewidgetitemArea.setText(str(self.reinforcing_steel[i][2]))
self.table_DefLonStReinf.setItem(i,2,tempQtablewidgetitemArea)

```

```

        if
len(self.reinforcing_steel) > self.table_DefLonStReinf.rowCount():
            for i in
range(len(self.reinforcing_steel) - self.table_DefLonStReinf.rowCount()):

self.btn_LonStReinfAddRowClicked(ReinforcementWindow)

self.retranslateUi(ReinforcementWindow)

QMetaObject.connectSlotsByName(ReinforcementWindow)
    # setupUi

    def
retranslateUi(self, ReinforcementWindow):

ReinforcementWindow.setWindowTitle(QCoreApplication.translate("ReinforcementWindow",
u"MainWindow", None))
        __qtablewidgetitem =
self.table_DefLonStReinf.horizontalHeaderItem(0)

__qtablewidgetitem.setText(QCoreApplication.translate("ReinforcementWindow",
u"X (m)", None));
        __qtablewidgetitem1 =
self.table_DefLonStReinf.horizontalHeaderItem(1)

__qtablewidgetitem1.setText(QCoreApplication.translate("ReinforcementWindow",
u"Y (m)", None));
        __qtablewidgetitem2 =
self.table_DefLonStReinf.horizontalHeaderItem(2)

__qtablewidgetitem2.setText(QCoreApplication.translate("ReinforcementWindow",
u"Area (cm2)", None));
        __qtablewidgetitem3 =
self.table_DefLonStReinf.verticalHeaderItem(0)

__qtablewidgetitem3.setText(QCoreApplication.translate("ReinforcementWindow",
u"Varilla 01", None));

        __sortingEnabled =
self.table_DefLonStReinf.isSortingEnabled()

self.table_DefLonStReinf.setSortingEnabled(False)

self.table_DefLonStReinf.setSortingEnabled(__sortingEnabled)

self.btn_LonStReinfAddRow.setText(QCoreApplication.translate("ReinforcementWindow",
u"A\u00f1adir campo", None))

self.btn_ProjLonStReinf.setText(QCoreApplication.translate("ReinforcementWindow",
u"Proyectar Refuerzo", None))
    # retranslateUi

    def
btn_ProjLonStReinfClicked(self, ReinforcementWindow):
        self.reinforcing_steel= []

        for i in range(self.table_DefLonStReinf.rowCount()):
            cellX =
self.table_DefLonStReinf.item(i,0)
            cellY = self.table_DefLonStReinf.item(i,1)

            cellArea = self.table_DefLonStReinf.item(i,2)

self.reinforcing_steel.append([float(cellX.text())*m,float(cellY.text())*m,float(cellArea.text(
))*m])

        npReinforcing_steel = np.array(self.reinforcing_steel)

updateVariables(self.conf_zone_info,npReinforcing_steel)
        matrix_1, matrix_2, matrix_3,
matrix_4, matrix_5, matrix_6, plotObj = fiber_method(self.coord_x, self.coord_y,

```

```

self.num_x_divisions, self.num_y_divisions, self.num_confine_zones)

self.MainWindow.on_ProcReinforcementDone(self.reinforcing_steel,matrix_1, matrix_2, matrix_3,
matrix_4, matrix_5, matrix_6, plotObj)

    def btn_LonStReinfAddRowfClicked(self,
ReinforcementWindow):
        newRowInt = self.table_DefLonStReinf.rowCount()

self.table_DefLonStReinf.insertRow(newRowInt)

        tempQtablewidgetitemX =
QTableWidgetItem()
        tempQtablewidgetitemX.setText(str(0))

self.table_DefLonStReinf.setItem(newRowInt,0,tempQtablewidgetitemX)

tempQtablewidgetitemY = QTableWidgetItem()
        tempQtablewidgetitemY.setText(str(0))

        self.table_DefLonStReinf.setItem(newRowInt,1,tempQtablewidgetitemY)

tempQtablewidgetitemArea = QTableWidgetItem()

tempQtablewidgetitemArea.setText(str(0))

self.table_DefLonStReinf.setItem(newRowInt,2,tempQtablewidgetitemArea)

newRowStr = str(newRowInt + 1)
        newRowStr = newRowStr.zfill(2)

        newRowHeader
= QTableWidgetItem()
        newRowHeader.setText("Varilla " + newRowStr);

        self.table_DefLonStReinf.setVerticalHeaderItem(newRowInt,  newRowHeader)

```

Anexo 5

Section_Designer.py


```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Mon Sep 4 20:31:19 2023
```

```
@author:
```

```
USER
```

```
"""
```

```
#
```

```
=====
```

```
#
```

```
//////////////////////////////////// Nomenclature //////////////////////////////////////
```

```
#
```

```
=====
```

```
# matrix_1:
```

```
List of coordinates X..Y of de bootom left corner and the right top corner of each fiber in de confine zone
```

```
# matrix_2: Area of each fiber in de the confine zone
```

```
# matrix_3: List of
```

```
coordinates X..Y of de bootom left corner and the right top corner of each fiber in de unconfine zone
```

```
# matrix_4: Area of each fiber in de the unconfine zone
```

```
# matrix_5: Center
```

```
coordinates (X , Y) of each fiber in de confine zone
```

```
# matrix_6: Center coordinates (X , Y) of
```

```
each fiber in de unconfine zone
```

```
#
```

```
=====
```

```
#
```

```
////////////////////////////////////
```

```
#
```

```
=====
```

```
import
```

```
matplotlib.pyplot as plt
```

```
from shapely.geometry import Polygon
```

```
import numpy as np
```

```
from Units
```

```
import *
```

```
conf_zone_info = []
```

```
reinforcing_steel = []
```

```
def
```

```
updateVariables(conf_zone_infoPar, reinforcing_steelPar):
```

```
    global conf_zone_info,
```

```
    reinforcing_steel
```

```
    conf_zone_info = conf_zone_infoPar
```

```
    reinforcing_steel =
```

```
    reinforcing_steelPar
```

```
def fiber_method(coord_x, coord_y, num_x_divisions,
```

```
num_y_divisions, num_confine_zones):
```

```
    # Create a representation of the section as a
```

```
    polygon
```

```
    section_polygon = Polygon(zip(coord_x, coord_y))
```

```
    # List for fiber
```

```
    coordenates
```

```
    coords_conf_fibers = []
```

```
    areas_conf_fibers = []
```

```
    coords_unconf_fibers
```

```
= []
```

```
    areas_unconf_fibers = []
```

```
    cent_coords_conf_fibers = []
```

```
    cent_coords_unconf_fibers = []
```

```

# Confine zones info
print(conf_zone_info)

for i in range(num_confine_zones):
    bc, dc, cent_coord_x, cent_coord_y =
conf_zone_info[i]

    # Fiber Method
    for i in range(num_x_divisions):
        for j in
range(num_y_divisions):
            x_left = min(coord_x) + i * (max(coord_x) -
min(coord_x)) / num_x_divisions
            x_right = min(coord_x) + (i + 1) * (max(coord_x)
- min(coord_x)) / num_x_divisions
            y_bottom = min(coord_y) + j * (max(coord_y) -
min(coord_y)) / num_y_divisions
            y_top = min(coord_y) + (j + 1) * (max(coord_y)
- min(coord_y)) / num_y_divisions

            # Center of each fiber
            x_center =
(x_left + x_right) / 2
            y_center = (y_bottom + y_top) / 2

            # Create
fibers
            fibers_polygon = Polygon([(x_left, y_bottom), (x_right, y_bottom),
(x_right, y_top), (x_left, y_top)])

            # Verify if the confine zone is inside de
section
            if section_polygon.contains(fibers_polygon):

fiber_inside_section = False
            for bc, dc, cent_coord_x, cent_coord_y in
conf_zone_info:
                x_rect = cent_coord_x - bc / 2
                y_rect
= cent_coord_y - dc / 2
                rect_extra_polygon = Polygon([(x_rect, y_rect),
(x_rect + bc, y_rect), (x_rect + bc, y_rect + dc), (x_rect, y_rect + dc)])

                # Verificar if each fiber is inside de confine zone

if rect_extra_polygon.contains(fibers_polygon):
                    fiber_inside_section =
True
                    break

            # Store coordenates and
areas of each fiber
            if fiber_inside_section:

coords_conf_fibers.append((x_left, y_bottom, x_right, y_top))

cent_coords_conf_fibers.append((x_center, y_center))

areas_conf_fibers.append(fibers_polygon.area)
            else:

coords_unconf_fibers.append((x_left, y_bottom, x_right, y_top))

cent_coords_unconf_fibers.append((x_center, y_center))

areas_unconf_fibers.append(fibers_polygon.area)

    # Convert lists in Numpy arrays

coords_conf_fibers = np.array(coords_conf_fibers).flatten()
areas_conf_fibers =
np.array(areas_conf_fibers)
coords_unconf_fibers =

```

```

np.array(coords_unconf_fibers).flatten()
    areas_unconf_fibers =
np.array(areas_unconf_fibers)

    cent_coords_conf_fibers =
np.array(cent_coords_conf_fibers)
    cent_coords_unconf_fibers =
np.array(cent_coords_unconf_fibers)

    plotObj = plot_section(coord_x, coord_y,
coords_unconf_fibers, coords_conf_fibers)

    return coords_conf_fibers,
areas_conf_fibers, coords_unconf_fibers, areas_unconf_fibers, cent_coords_conf_fibers,
cent_coords_unconf_fibers, plotObj

def plot_section(coord_x, coord_y,
coords_unconf_fibers, coords_conf_fibers):

    figure_SectGeom = plt.figure()

# Plot section as a polygon
    ax = figure_SectGeom.add_subplot(111)
    ax.plot(coord_x +
[coord_x[0]], coord_y + [coord_y[0]], linestyle='-', color='black', linewidth=1)

    # Plot
reinforcing steel position in section plane
    for YZ in reinforcing_steel:

ax.scatter(YZ[0], YZ[1], s=YZ[2]*4.0352, color='blue')

    # Plot confine fiber
divisions
    for i in range(0, len(coords_conf_fibers), 4):
        x1, y1, x2, y2 =
coords_conf_fibers[i:i+4]
        ax.fill([x1, x2, x2, x1, x1], [y1, y1, y2, y2, y1],
fill=True, edgecolor='black', facecolor='gray', linewidth=0.5, alpha=0.5)

    # Plot
unconfine fiber divisions
    for i in range(0, len(coords_unconf_fibers), 4):
        x1,
y1, x2, y2 = coords_unconf_fibers[i:i+4]
        ax.fill([x1, x2, x2, x1, x1], [y1, y1, y2,
y2, y1], fill=False, edgecolor='black', linewidth=0.5, alpha=0.5)

    # Show figure

ax.axis('equal')
    ax.grid(True, linestyle='--', alpha=0.5)
    ax.set_xlabel('metros')

    ax.set_ylabel('metros')
    #ax.show()
    return figure_SectGeom

```

Anexo 6

UI_ResultsApp.py

```
# -*- coding: utf-8
 -*-
```

```
#####
##
Form generated from reading UI file 'Results App.ui'
##
## Created by: Qt User Interface
Compiler version 6.6.0
##
## WARNING! All changes made in this file will be lost when
recompiling UI
file!
#####
```

```
from
PyQt5.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,
    QMetaObject, QObject,
QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PyQt5.QtGui import (QBrush, QColor,
    QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage,
QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient,
QTransform)
from PyQt5.QtWidgets import (QApplication, QFrame, QGroupBox, QLabel,
    QLineEdit, QMainWindow, QPushButton, QSizePolicy,
    QWidget, QVBoxLayout)
```

```
import
matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAff as
FigureCanvas
```

```
from Units import *
```

```
class Ui_ResultsWindow(object):
    def setupUi(self,
ResultsWindow, MainWindow):
        if not ResultsWindow.setObjectName():
```

```
ResultsWindow.setObjectName(u"ResultsWindow")
        ResultsWindow.resize(829,
728)
```

```
        #parameters
        self.MainWindow = MainWindow
```

```
self.centralwidget = QWidget(ResultsWindow)
```

```
self.centralwidget.setObjectName(u"centralwidget")
        self.groupBox_AxialLoad =
QGroupBox(self.centralwidget)
```

```
self.groupBox_AxialLoad.setObjectName(u"groupBox_AxialLoad")
```

```
self.groupBox_AxialLoad.setGeometry(QRect(10, 10, 811, 101))
        self.label_P =
QLabel(self.groupBox_AxialLoad)
        self.label_P.setObjectName(u"label_P")
```

```
        self.label_P.setGeometry(QRect(350, 30, 21, 16))
        self.lineEdit_P =
QLineEdit(self.groupBox_AxialLoad)
```

```
self.lineEdit_P.setObjectName(u"lineEdit_P")
```

```
self.lineEdit_P.setGeometry(QRect(370, 30, 71, 22))
        self.label_P_unit =
```

```

QLabel(self.groupBox_AxialLoad)

self.label_P_unit.setObjectName(u"label_P_unit")

self.label_P_unit.setGeometry(QRect(450, 30, 31, 16))
    self.btn_GenDiag =
QPushButton(self.groupBox_AxialLoad)

self.btn_GenDiag.setObjectName(u"btn_GenDiag")

self.btn_GenDiag.setGeometry(QRect(10, 70, 791, 28))

#
=====

    self.frame_MCDiagram = QFrame(self.centralwidget)

self.frame_MCDiagram.setObjectName(u"frame_MCDiagram")

self.frame_MCDiagram.setGeometry(QRect(10, 120, 811, 601))

self.frame_MCDiagram.setStyleSheet(u"background-color: rgb(255, 255, 255);")

self.frame_MCDiagram setFrameShape(QFrame.StyledPanel)

self.frame_MCDiagram setFrameShadow(QFrame.Raised)

    self.layout_MCDiagram=
QVBoxLayout(self.frame_MCDiagram)
    self.canvas = None
    # Add Matplot Figure

    #self.figure_MCDiagram = plt.figure(figsize=(5, 10))
    #self.ax =
self.figure_MCDiagram.add_subplot(111)
    #self.ax.grid(True, linestyle='--',
alpha=0.5)
    #self.canvas = FigureCanvas(self.figure_MCDiagram)
    # Add Canvas

    #self.layout_MCDiagram.addWidget(self.canvas)

#self.frame_MCDiagram.setLayout(self.layout_MCDiagram)

#
=====

#Clicked Button
    self.btn_GenDiag.clicked.connect(lambda x:
self.btn_GenDiagClicked(ResultsWindow))

ResultsWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(ResultsWindow)

    QMetaObject.connectSlotsByName(ResultsWindow)

# setupUi

    def retranslateUi(self, ResultsWindow):

ResultsWindow.setWindowTitle(QCoreApplication.translate("ResultsWindow",
u"MainWindow", None))

self.groupBox_AxialLoad.setTitle(QCoreApplication.translate("ResultsWindow",
u"Carga Axial", None))

self.label_P.setText(QCoreApplication.translate("ResultsWindow", u"P:",
None))
    self.label_P_unit.setText(QCoreApplication.translate("ResultsWindow",

```

```
u"tonf", None))

self.btn_GenDiag.setText(QCoreApplication.translate("ResultsWindow", u"Generar
Diagrama Momento-Curvatura", None))
    # retranslateUi

    def
btn_GenDiagClicked(self, ResultsWindow):
    P = 0
    if self.lineEdit_P.text():

        P = -float(self.lineEdit_P.text())*tonf

        plotObj =
self.MainWindow.showResultsCurvature(P)

        # Add Matplot Figure

self.figure_MCDiagram = plotObj
    if self.canvas is not None:

self.canvas.deleteLater()
        self.canvas = FigureCanvas(self.figure_MCDiagram)
        #
Add Canvas
        self.layout_MCDiagram.addWidget(self.canvas)

self.frame_MCDiagram.setLayout(self.layout_MCDiagram)
```

Anexo 7

OpenSeesPy_Analysis.py


```
# -*- coding: utf-8 -*-  
"""
```

```
Created on II Semester, 2023
```

```
@author: Jose  
Pablo Sánchez Soto  
"""
```

```
import matplotlib.pyplot as plt  
import  
openseespy.opensees as op  
from Section_Designer import *  
from Units import *
```

```
fpcc_1 =  
0  
epcc_1 = 0  
fpcu_1 = 0  
epsu_1 = 0
```

```
fpcc_2 = 0  
epcc_2 = 0  
fpcu_2 = 0  
epsu_2 =  
0
```

```
fyp = 0  
fyn = 0  
E0 = 0  
bp = 0  
bn = 0
```

```
matrix_1 = []  
matrix_2 = []  
matrix_3 =  
[]  
matrix_4 = []  
matrix_5 = []  
matrix_6 = []
```

```
reinforcing_steel = []
```

```
P = 0
```

```
def  
updatePResults(PPar):  
    global P  
    P = PPar
```

```
def updateVarMatrix(matrix_1Par,  
matrix_2Par, matrix_3Par, matrix_4Par, matrix_5Par, matrix_6Par, reinforcing_steelPar):
```

```
global matrix_1, matrix_2, matrix_3, matrix_4, matrix_5, matrix_6, reinforcing_steel
```

```
matrix_1 = matrix_1Par  
matrix_2 = matrix_2Par  
matrix_3 = matrix_3Par  
matrix_4 =  
matrix_4Par  
matrix_5 = matrix_5Par  
matrix_6 = matrix_6Par  
reinforcing_steel =  
reinforcing_steelPar
```

```
def updateVarConcConfinado(fpcc,epcc,fpcu,epsu):  
    global  
fpcc_1,epcc_1,fpcu_1,epsu_1  
    fpcc_1 = fpcc  
    epcc_1 = epcc  
    fpcu_1 = fpcu
```

```
epsu_1 = epsu
```

```
print(fpcc_1)
```

```

def
updateVarConcInconfinado(fpcc,epcc,fpcu,epsu):
    global fpcc_2,epcc_2,fpcu_2,epsu_2

fpcc_2 = fpcc
epcc_2 = epcc
fpcu_2 = fpcu
epsu_2 = epsu

    print(fpcc_2)

def updateVarAceroRefuerzo(fypPar,fynPar,E0Par,bpPar,bnPar):
    global
fyp,fyn,E0,bp,bn
fyp = fypPar
fyn = fynPar
E0 = E0Par
bp = bpPar
bn =
bnPar

    print(fyp)

def showReinforcement():
    return plot_Reinforcing_SS(fyp,
E0, bp)

def showConcrete():
    return plot_Concrete_SS(epcc_1, epsu_1, fpcc_1, fpcu_1,
epcc_2, epsu_2, fpcc_2, fpcu_2)

def showMomentCurvature():
    # Call de function to plot
Moment Curve Diagram
    return plot_MomentCurvature_diagram('AMomentCurvature.txt')

def MomentCurvature(secTag, axialLoad, maxK, numIncr=100):

#
=====
# Define
element
#=====

# Define nodes
# node(nodeTag, *crds)
op.node(1, 0.0, 0.0)
op.node(2,
0.0, 0.0)

# Define constriants in nodes
# fix(nodeTag, *constrValues)

op.fix(1, 1, 1, 1)
op.fix(2, 0, 1, 0)

# Define element of 0 length
#
element('zeroLengthSection', eleTag, *eleNodes, secTag)
op.element('zeroLengthSection',
1, 1, 2, secTag)

#
=====
# Define
constant axial load
#
=====

```

```

#
timeSeries('Constant', tag)
  op.timeSeries('Constant', 1)
  # pattern('Plain',
patternTag, tsTag)
  op.pattern('Plain', 1, 1)
  # load(nodeTag, *loadValues)

op.load(2, axialLoad, 0, 0)

#
=====
# Define
analysis
# =====

# integrator('LoadControl', incr)
  op.integrator('LoadControl', 0)
  #
system('SparseGeneral', '-piv') ; perform parial-pivoting
  op.system('SparseGeneral',
'-piv')
  # test('EnergyIncr', tol, iter)
  op.test('EnergyIncr', 1e-9, 100)
  #
numberer(numbererType)
  op.numberer('Plain')
  # constraints(constraintType)

op.constraints('Plain')
  # algorithm(algoType)
  op.algorithm('Newton')
  #
analysis(analysisType)
  op.analysis('Static')

  # One analysis for constant axial load

  # analyze(numIncr=1)
  op.analyze(1)

#
=====
# Define moment
reference
# =====

# timeSeries('Linear', tag)
  op.timeSeries('Linear', 2)
  # pattern('Plain',
patternTag, tsTag)
  op.pattern('Plain', 2, 2)
  # load(nodeTag, *loadValues)

op.load(2, 0, 0, 1)

#
=====
# Section
Analysis
# =====

# Curvature increment
dK = maxK / numIncr

# Dispmacement control
#
integrator('DisplacementControl', nodeTag, dof, incr)
  op.integrator('DisplacementControl',
2, 3, dK)

# Recorder for moment curvature diagram

```

```

    op.recorder('Node', '-file',
'AMomentCurvature.txt', '-time', '-closeOnWrite', '-node', *[2], '-dof', *[3], 'disp')

#
Recorder for confine concrete stress strain
#row1_confine = matrix_5[0]

#op.recorder('Element', '-ele', 1, '-file', 'AConfineConcrete.txt', 'section', 'fiber',
row1_confine[0], row1_confine[1], 1, 'stressStrain')

# Recorder for unconfine
concrete stress strain
#row1_unconfine = matrix_6[0]
#op.recorder('Element', '-ele',
1, '-file', 'AUnconfineConcrete.txt', 'section', 'fiber', row1_unconfine[0], row1_unconfine[1],
2, 'stressStrain')

# Recorder for steel reinforcement stress strain

#row1_reinfsteel = reinforcing_steel[6]
#op.recorder('Element', '-ele', 1, '-file',
'AReinforcingSteel.txt', 'section', 'fiber', row1_reinfsteel[0], row1_reinfsteel[1], 3,
'stressStrain')

# Section Analysis
# analyze(numIncr)
ok =
op.analyze(numIncr)

#plot_Concrete_SS('AConfineConcrete.txt',
'AUnconfineConcrete.txt')

#
=====
# Loop for
convergence failure
#
=====
    IDctrlNode
= 2
    IDctrlDOF = 3
    Dmax = maxK
    Dincr = dK
    TolStatic = 1.e-9

testTypeStatic = 'EnergyIncr'
maxNumIterStatic = 6
algorithmTypeStatic = 'Newton'

if ok != 0:
    Dstep = 0.0
    ok = 0
    while Dstep <= 1.0 and ok ==
0:
        controlDisp = op.nodeDisp(IDctrlNode, IDctrlDOF)
        Dstep =
controlDisp / Dmax
        ok = op.analyze(1)
        if ok != 0:
            Nk
= 4
            DincrReduced = Dincr / Nk

op.integrator('DisplacementControl', IDctrlNode, IDctrlDOF, DincrReduced)
for
ik in range(1, Nk + 1):
    ok = op.analyze(1)
    if ok !=
0:
        print("Trying Newton with Initial Tangent ..")

```

```

        op.test('NormDispIncr', TolStatic, 2000, 0)

op.algorithm('Newton', '-initial')
        ok = op.analyze(1)

        op.test(testTypeStatic, TolStatic, maxNumIterStatic, 0)

op.algorithm(algorithmTypeStatic)
        if ok != 0:

print("Trying Broyden ..")
        op.algorithm('Broyden', 8)

        ok = op.analyze(1)

op.algorithm(algorithmTypeStatic)
        if ok != 0:

print("Trying NewtonWithLineSearch ..")

op.algorithm('NewtonLineSearch', 0.8)
        ok = op.analyze(1)

        op.algorithm(algorithmTypeStatic)
        if ok != 0:

            print("PROBLEM: CtrlNode %i, dof %i, Curv=%.4f /Length" % (IDctrlNode,
IDctrlDOF, op.nodeDisp(IDctrlNode, IDctrlDOF)))
                return -1

        op.integrator('DisplacementControl', IDctrlNode, IDctrlDOF, Dincr)
        LunitTXT =
"Length"
        fmt1 = "%s Pushover analysis: CtrlNode %.3i, dof %.1i, Curv=%.4f
/%s"
        u = op.nodeDisp(IDctrlNode, IDctrlDOF)
        print (u)
        if ok != 0:

print(fmt1 % ("PROBLEM", IDctrlNode, IDctrlDOF, op.nodeDisp(IDctrlNode, IDctrlDOF),
LunitTXT))
        else:
            print(fmt1 % ("DONE", IDctrlNode, IDctrlDOF,
op.nodeDisp(IDctrlNode, IDctrlDOF),
LunitTXT))

op.wipe()
#=====
# Concrete Characteristics
#=====

def
defineCharacteristics():
    op.wipe()
    #Confine Concrete

op.uniaxialMaterial('Concrete01', 1, fpcc_1, epcc_1, fpcu_1, epsu_1)

#op.uniaxialMaterial('Concrete01', 1, -452*kgfcm2, -0.0021, -452*kgfcm2, -0.008)

#Unconfine Concrete
    op.uniaxialMaterial('Concrete01', 2, fpcc_2, epcc_2, fpcu_1,
epsu_1)
    #op.uniaxialMaterial('Concrete01', 2, -280*kgfcm2, -0.0021, -0*kgfcm2,
-0.008)

#=====
#
Steel Characteristics
#=====

```

```
op.uniaxialMaterial('SteelMPF', 3, fyp, fyn, E0, bp, bn, 20, 0.925,
0.15)
```

```
#=====
#
Define
model
#=====
```

```
d
def defineModel():
    # Define model
    op.model('basic', '-ndm', 2, '-ndf', 3)

    # Define
    fiber section
    op.section('Fiber', 1)

    for XY, A in zip(matrix_5, matrix_2):
op.fiber(*XY, A, 1)

    for XY, A in zip(matrix_6, matrix_4):
        op.fiber(*XY, A,
2)

    for YX in reinforcing_steel:
        op.fiber(YX[0], YX[1], YX[2]*cm2,
3)
```

```
numIncr = 100
```

```
#
=====
# Plot Moment
Curvature diagram
#
=====
```

```
def
plot_MomentCurvature_diagram(filename):
    figure_MCDiagram = plt.figure()

    with
open(filename, 'r') as f:
        lines = f.readlines()
        y1 = [float(line.split()[0])
for line in lines]
        x1 = [float(line.split()[1]) for line in lines]

    ax =
figure_MCDiagram.add_subplot(111)

    ax.plot(x1, y1)
    #plt.xticks(rotation=70)

    ax.set_xlabel('Curvatura (1/m)')
    ax.set_ylabel('Momento (tonf-m)')

ax.set_title('Diagrama Momento-Curvatura')
ax.grid(True)
plt.show()
return
figure_MCDiagram
```

```
#
=====
```

```

# Plot Concrete
Stress Strain
#
=====

def
plot_Concrete_SS(epcc_1, epsu_1, fpcc_1, fpcu_1, epcc_2, epsu_2, fpcc_2, fpcu_2):

figure_Conc = plt.figure()

    xCC = [0, -epcc_1, -epsu_1]
    xUN = [0,
-epcc_2, -epsu_2]

    yCC= [0, -fpcc_1, -fpcu_1]
    yUN= [0, -fpcc_2, -fpcu_2]

    ax = figure_Conc.add_subplot(111)
    ax.plot(xCC, yCC, label='Curva 1',
linestyle='-')
    ax.plot(xUN, yUN, label='Curva 2', linestyle='-')

#ax.set_xticks(rotation=70)
    ax.set_xlabel('Deformación (m/m)')

ax.set_ylabel('Esfuerzo (tonf/m2)')

    ax.grid(True, linestyle='--', alpha=0.5)

#plt.show()
    return figure_Conc

#
=====
# Plot Steel
Reinforcement Stress Strain
#
=====

def
plot_Reinforcing_SS(fyp, E0, bp):
    figure_StReinf = plt.figure()

    ax =
figure_StReinf.add_subplot(111)
    x= [0, fyp/E0, ((fyp/E0)*10)]
    y= [0, fyp,
(fyp+(fyp/E0-((fyp/E0)*10))*(E0*-bn))]

    ax.plot(x, y, label='Curva 1',
linestyle='-')

    #ax.set_xticklabels(ax.get_xticks(), rotation=70)

ax.set_xlabel('Deformación (m/m)')
    ax.set_ylabel('Esfuerzo (tonf/m2)')
    ax.grid(True,
linestyle='--', alpha=0.5)
    #plt.show()
    return figure_StReinf

```