



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA CIVIL

**Análisis estructural de una viga continua de 2 vanos mediante
librerías de lenguaje de programación en Python, Trujillo 2023**

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero Civil

AUTORES:

Astudillo Morales, Felix Ronald (orcid.org/0000-0003-3203-8268)

Flores Adanaque, Pierina Alexandra (orcid.org/0000-0002-7449-6964)

ASESOR:

Mg. Noriega Vidal, Eduardo Manuel (orcid.org/0000-0001-7674-7125)

LÍNEA DE INVESTIGACIÓN:

Diseño Sísmico y Estructural

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

TRUJILLO – PERÚ

2023

DEDICATORIA

A mi familia, en especial a mis padres por todo el apoyo que me ha brindado y por confiar en mi a lo largo del desarrollo de mi vida y mi carrera profesional.

Felix Ronald Astudillo Morales

A mi madre y hermano, por la confianza que tienen en el desarrollo de mi futuro profesional y el apoyo brindado en el transcurso del desarrollo de mi vida universitaria.

Pierina Alexandra Flores Adanaqué

AGRADECIMIENTO

Un agradecimiento especial a todos los docentes de la Universidad Cesar Vallejo, que fueron parte de nuestro crecimiento profesional en el transcurso de los años, además, un especial agradecimiento al Mg. Eduardo Noriega, ya que sin su apoyo no hubiese sido posible el desarrollo del presente proyecto.

Índice de contenido

DEDICATORIA	ii
AGRADECIMIENTO	iii
Índice de contenido.....	iv
Índice de tablas.....	vii
Índice de figuras	viii
RESUMEN.....	ix
ABSTRACT.....	x
I. INTRODUCCIÓN.....	1
II. MARCO TEÓRICO.....	4
2.2. Bases teóricas.....	6
2.2.1. Python.....	6
2.2.2. Librerías de Python	6
2.2.2.1. NumPy.....	7
2.2.2.2. Sympy.....	8
2.2.2.3. Pandas	8
2.2.2.4. Matplotlib	9
2.2.3. Anaconda.....	9
2.2.4. Jupyter Notebook.....	10
2.2.5. OpenGL	11
2.2.6. Programación	12
2.2.6.1. Lenguaje de Programación	12
2.2.6.2. Tipos de lenguaje de programación.....	12
2.2.6.3. Código fuente	13
III. METODOLOGÍA.....	14
3.1. Tipo y diseño de investigación:.....	14

3.1.1. Tipo de investigación:	14
3.1.2. Diseño de investigación:	14
3.2. Variables y operacionalización.....	14
3.2.1. Clasificación de Variables.....	17
3.3. Población, muestra y muestreo.....	19
3.3.1. Población	19
3.3.2. Muestra	19
3.3.3. Muestreo	19
3.4. Técnicas e instrumentos de recolección de datos.....	20
3.4.1. Técnicas de recolección de datos	20
3.4.2. Instrumento de recolección de datos.....	20
3.4.3. Validación de instrumento de recolección de datos	20
3.5. Procedimientos.....	21
3.6. Método de análisis de datos	21
3.7. Aspectos éticos	22
IV. RESULTADOS	23
V. DISCUSIÓN	28
VI. CONCLUSIONES.....	29
VII. RECOMENDACIONES	32
REFERENCIAS	33
ANEXOS	39
ANEXO-1	39
ANEXO-2.....	40
ANEXO-3.....	43
ANEXO-4.....	44
ANEXO-5.....	45
ANEXO-6.....	46

ANEXO-7	51
ANEXO-8	54
ANEXO-9	63

Índice de tablas

Tabla 1 Clasificación de variables	17
Tabla 2 características de la viga continua	23
Tabla 3 Archivo CSV generado por el código de Python.....	24
Tabla 4 Diagramas de M y V a lo largo de la viga	27
Tabla 5 Matriz de consistencia	42
Tabla 6 Operacionalización de variables	45

Índice de figuras

Figura 1 Logo de Python	6
Figura 2 Logo de la librería Numpy	7
Figura 3 Logo de la librería Sympy.....	8
Figura 4 Logo de la librería Pandas	8
Figura 5 Logo de la libería Matplotlib	9
Figura 6 Logo del Software Anaconda	10
Figura 7 Logo de la aplicación Jupyter Notebook	10
Figura 8 Logo de OpenGL.....	11
Figura 9 Lenguajes de programación.....	13
Figura 10 Instrumento de validación de datos	39
Figura 11 Descripción del funcionamiento del código.....	43
Figura 12 Algoritmo del código.....	44
Figura 13 Datos iniciales	54
Figura 14 Línea 3 del código de Python.....	55
Figura 15 Se procede al cálculo de M y V en la viga	56
Figura 16 archivo CSV y se procede a graficar.....	57
Figura 17 Diagramas de M y V.....	58
Figura 18 Se encuentran los valores de Vmax y Mmax.....	59
Figura 19 Cálculo de reacciones, Vmax1 y Mmax1	60
Figura 20 Valores de Iy, E y cálculo de deformación para cada tramo.	61
Figura 21 Cálculo de deformación y procedimiento final	62
Figura 22 Inicio de prueba en OpenGL	63
Figura 23 Ingreso de datos.....	63
Figura 24 Confirmación de datos ingresados.....	64
Figura 25 Confirmación de longitud de viga ingresada.....	64
Figura 26 Tabla de Valoreses	65
Figura 27 Confirmación de datos para obtener la tabla de valores.	65
Figura 28 Tabla de valores generada	66
Figura 29 Tabla de valores en archivo CSV.....	66
Figura 30 Archivo CSV	67
Figura 31 ubicación de la opción "gráfica"	67
Figura 32 Graficas de M y V	68

RESUMEN

La presente investigación se centra en el análisis de una viga continua de 2 vanos mediante el lenguaje de programación Python. El objetivo principal del presente estudio es determinar las magnitudes de máxima de cortante, máximo momento flector y máxima deformación que experimenta una viga continua de 2 vanos a través de un código de programación diseñado mediante librerías de lenguaje de programación en Python, para el desarrollo de la presente tesis se empleó una metodología de tipo aplicada de diseño no experimental transversal descriptiva; la investigación está enfocada en inculcar la aplicación del lenguaje de programación de Python tanto en nuestra región como en nuestra casa de estudios; la población empleada en este caso serían todas las vigas continuas de 2 vanos que estén sometidas a cargas uniformemente distribuidas, la muestra está conformada por una viga continua de 2 vanos con cargas uniformemente distribuidas y parámetros definidos. Al realizar los cálculos mediante el código de programación, se obtuvieron los siguientes resultados, los cuales proporcionaron información sobre las magnitudes de máxima cortante $V_{\max} = 11.38890 \text{ KN}$, máximo momento flector $M_{\max} = 9.55560 \text{ KN}\cdot\text{m}$ y máxima deformación $u_{\max} = -11.867694 \text{ mm}$. En conclusión, el uso del lenguaje de programación Python como herramienta de análisis ha demostrado ser eficiente y preciso en el desarrollo de la presente investigación.

PALABRAS CLAVE: Python, Ingeniería civil, viga continua, lenguaje de programación, Numpy, Sympy, Pandas, Matplotlib.

ABSTRACT

This research focuses on the analysis of a continuous beam with 2 spans using the Python programming language. The main objective of this study is to determine the magnitudes of maximum shear, maximum bending moment and maximum deformation experienced by a continuous beam with 2 spans through a programming code designed using programming language libraries in Python, for the development of In this thesis, a non-experimental cross-sectional descriptive design applied methodology was used; the research is focused on instilling the application of the Python programming language both in our region and in our house of studies; The population used in this case would be all the 2-span continuous beams that are subjected to uniformly distributed loads. The sample is made up of a 2-span continuous beam with uniformly distributed loads and defined parameters. When performing the calculations using the programming code, the following results were obtained, which provided information on the magnitudes of maximum shear $V_{\max} = 11.38890$ KN, maximum bending moment $M_{\max} = 9.55560$ KN*m and maximum deformation $u_{\max} = -11.867694$ mm. In conclusion, the use of the Python programming language as an analysis tool has proven to be efficient and accurate in the development of this research.

KEYWORDS: Python, civil engineering, continuous beam, programming language, Numpy, Sympy, Pandas, Matplotlib.

I. INTRODUCCIÓN

Con el pasar del tiempo, la tecnología ha ido cambiando constantemente, el sector de la construcción no ha sido la excepción, ante ello, como ingenieros, tenemos el deber de implementar constantemente nuevas alternativas de solución ante diversos problemas que se puedan presentar en el ámbito profesional, de esta forma se contribuye al cambio y a la mejora continua de nuestro país. El progreso tecnológico es mucho más rápido que la evolución natural y esto se puede notar con el paso del tiempo, ya que han surgido increíbles tecnologías en los últimos años. La tecnología ofrece a las personas: más agilidad, comodidad y eficiencia en la ejecución de distintas labores. Con el desarrollo e implementación de la tecnología, muchos procesos en la vida humana han cambiado y se han visto simplificados; como la creación de teléfonos inteligentes, mejora de procesos en ingeniería, medicina, comunicación.

(Martínez, 2021), menciona que, con el desarrollo constante de la tecnología, diversas empresas se han adecuado al avance tecnológico, y una mejora ha sido la implementación de Python a su infraestructura con la finalidad de optimizar el control de sus labores y actividades. Esta herramienta posee gran versatilidad en su aprendizaje y su impacto se ve reflejado a través de la creación de aplicaciones y automatización en diversos ámbitos, que lo vuelve cada vez más cotizado en la industria tecnológica. (pág. 01)

Así mismo, (Puga, 2019) declara en su investigación titulada: "Automatización de procesos en redes de datos mediante programación en Python", que el lenguaje de programación Python es utilizado en la administración y configuración de dispositivos de red, como enrutadores y conmutadores, para automatizar procesos que los administradores de red pueden repetir y simplificar procesos al momento del ingreso de datos. Esta implementación está diseñada para enviar y recibir información de forma segura sin comprometer la confidencialidad de los datos. Además, Python es esencial para los administradores, ya que al ser un lenguaje simple nos permite abordar todas las necesidades de los usuarios en cada situación comercial para cualquier fabricante de equipos de red, lo que se convierte en una tarea muy compleja en una tarea más sencilla y eficaz. Con cientos de librerías existentes, este lenguaje de programación es extremadamente flexible e integrado, por lo que se pueden

diseñar scripts especialmente para resolver situaciones específicas de acuerdo a las necesidades de cada empresa. (p.163)

Según (Rodríguez, 2022), en la tesis titulada “Análisis y evaluación de datos para dosificación de mezclas de hormigón mediante el desarrollo de un software en lenguaje de programación Python aplicando el método ACI”, realizó un trabajo experimental donde creó y codificó una interfaz gráfica de software mediante el programa Python, aplicando el método ACI para obtener dosificaciones de hormigón. El software permite al usuario obtener las dosificaciones rápidamente ingresando solamente los datos de las propiedades físicas del agregado fino, agregado grueso, cemento y hasta la versatilidad de diseño, cuyas propiedades físicas se obtienen mediante ensayos de laboratorio, tales como la granulometría, el módulo de finura, las densidades, el contenido de humedad y la capacidad de absorción. (pág. 51)

A partir del análisis de la problemática presentada, se plantea la siguiente interrogante, la cual permite formular el problema general, ¿Cómo es el Análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python? Además, como problemas específicos se plantean las siguientes: ¿Cuál es la máxima cortante en el análisis estructural de una viga continua de 2 vanos a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python? ¿Cuál es el máximo momento flector a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python? y ¿Cuál es la máxima deformación a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python?

El presente trabajo de investigación plantea como justificación teórica el poder contribuir al aprendizaje y uso del lenguaje de programación Python desarrollado en la ingeniería civil, dirigido a la zona de Trujillo, ya que en esta región no existen muchos antecedentes del uso de Python aplicado en la ingeniería civil.

De la misma forma como justificación práctica, se plantea, como finalidad el poder realizar el análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, abreviando el tiempo de

búsqueda al obtener resultados referentes a las propiedades mecánicas de la viga , utilizando el lenguaje de Python como una aplicación hacia la ingeniería civil.

Y como justificación metodológica se plantea el hecho de contribuir enormemente en el desarrollo de futuras investigaciones en el mismo ámbito, ya que es una temática innovadora. Se tiene conocimiento que el programa Python es muy flexible y nos permite aplicarlo a muchos campos de investigación, uno de estos campos es la ingeniería civil donde se busca implementar y desarrollar la investigación.

Teniendo como referencia distintas fuentes de investigación, se plantea como objetivo general el siguiente: Realizar el análisis estructural de una viga continua de 2 vanos mediante librerías del lenguaje de programación Python, y como objetivos específicos: Determinar la cortante máxima en una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python; determinar el momento flector máximo en una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python y determinar la deformación máxima en una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python.

A partir de esto, se plantea la siguiente hipótesis general: sí, es posible el análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, y, por consiguiente. Como objetivos específicos de la hipótesis se plantean los siguientes: Se podría determinar el máximo cortante a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python; del mismo modo, se podría determinar el máximo momento flector a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python y por ende también se podría determinar la máxima deformación a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python.

II. MARCO TEÓRICO.

(Pascual, 2018), en su tesis de maestría titulada: “Dinámica de barras: Caracterización experimental, formulación computacional en Python y calibrado aplicado a una viga biapoyada.” Afirma que gracias al avance de la tecnología, las estructuras y los diseños de estas, son más impresionantes y complejas por ello con la intención de analizar la viabilidad de estos proyectos es imprescindible analizar y comprobar que la estructura soporte efectivamente las cargas a la que será sometida; por ello desarrollar modelos y sistemas de cargas que se asemejen, ayudará a calidad y comprobar el correcto diseño estructural diseñado para la estructura. Por ello, el diseño en Python se considera un método que permite reducir tiempo y dinero para poder descartar propuestas que no cumplan con los requerimientos necesarios para la estructura planteada, no obstante, también se debe considerar el capital humano empleado en la realización de este método. (p.08)

(Román, 2022). En su artículo científico titulado: “CESSIPy: A Python open-source module for stochastic system identification in civil engineering”, menciona que Python es una poderosa herramienta que facilita el análisis estructural de un proyecto planteado en la ingeniería civil, ya que la facilidad de manipulación de este lenguaje nos permite obtener distintos resultados, sometiendo a distintas cargas a la estructura planteada. El objetivo del empleo de Python en la ingeniería civil es poder ajustar los parámetros requeridos de la estructura en un modelo matemático que pueda plasmar el comportamiento dinámico de la estructura, identificando cuál es la mejor opción para la ejecución y desarrollo de la estructura en cuestión, ese tipo de lenguaje se aplica para poder desarrollar estructuras de una forma más automatizada que nos permite obtener varios resultados de acuerdo a los datos requeridos. (p.06)

(Herrera, 2016), analizó vigas metálicas rígidas donde utilizó el lenguaje de programación Python, el cual generó una gran ventaja al momento de analizar la estructura. Planteando un puente de tres vanos, considerando cargas actuantes del peso propio de la viga, para corroborar de una forma realista el cálculo de la deformación máxima y la sumatoria de reacciones verticales de esta. Este tipo de análisis nos permite optimizar los tiempos en obra y la cantidad de elementos de

soporte auxiliares utilizados en la realización del proyecto, ya que al realizar este tipo de análisis estructural observamos los pro y contras de la estructura frente a esfuerzo estructural. (p. 31, 37)

(Gutiérrez, y otros, 2019), en su tesis para obtener el grado de ingeniero civil titulada “Diseño integral de armaduras metálicas usando rhinoceros 3D: Modelación paramétrica, simulación estructural y optimización”. Consideró tres etapas distintas como, la modelación, simulación y optimización estructural. Con su investigación se obtuvo una armadura óptima con un peso menor en 28.35%; demostrando así que el diseño de estructuras mediante software, mejoran el proceso y lo optimizan. La construcción de estructuras de acero actualmente se encuentra en auge, ya que este tipo de estructuras nos brinda distintas ventajas, como la facilidad de montaje y la alta resistencia estructural. El sistema estructural utilizado en este tipo de edificaciones es el sistema articulado, el cual presenta esfuerzos a compresión y tracción. (pág. 212)

(Lupaca, 2022), en su tesis para obtener el título profesional de ingeniero civil. Titulada “Desarrollo del simulador web Beli para el análisis matricial de estructuras planas y espaciales”, analizó el desarrollo del simulador web Beli para el análisis matricial de estructuras planas y espaciales, donde concluyó que el simulador Beli aplicado en distintas estructuras como vigas continuas, armaduras, y pórticos, genera mejores resultados y optimiza el análisis estructural, recomendando incorporar el análisis estructural mediante softwares y simuladores para la mejora y perfeccionamiento de los cálculos estructurales; ya que con el pasar del tiempo los interfaces creados van mejorando y facilitando su aplicación. Uno de los objetivos del desarrollo estructural mediante softwares es la optimización de procesos, ya que la programación nos facilita y ahorra un tiempo considerable en la realización de desarrollo y análisis estructural. (pág. 10)

(Picha, 2021), en su tesis para obtener el título profesional de ingeniero civil titulada “Análisis y Diseño de Pórticos de Concreto Armado usando Inteligencia Artificial”, menciona que la aplicación de algoritmos en el análisis y diseño de pórticos de concreto armado, es la opción más viable para la solución del proceso y modelado estructural, ya que al poder diseñar y evaluar una estructura podemos analizar y prevenir los problemas que se pueden generar en la estructura, definiendo que problema presenta la estructura y cuál es la deformación obtenida,

teniendo en cuenta que este tipo de análisis puede reproducirse o mejorarse dependiendo de las necesidades requeridas para el proyecto. Así mismo, se recomendó implementar estudios y propuestas de aplicación de algoritmos en este tema, ya que la inteligencia artificial y estructural pueden ir de la mano facilitando, mejorando y optimizando los diseños estructurales. (pág. 126)

2.2. Bases teóricas

2.2.1. Python

Para (Masip R, 2018), Python es un lenguaje de programación de muy alto nivel con una conexión muy clara y un fuerte compromiso con la fácil legibilidad del código. Python es un lenguaje de programación versátil que va orientado a objetos de una forma imperativa que contiene características que puede convertirlo en un lenguaje de paradigma funcional, el cual nos facilita la planificación y automatización de tareas, la planificación y proceso de datos en conjunto, así mismo también facilita modelado, simulación y análisis de distintas estructuras. Python es considerado un lenguaje amplio y versátil, al ser aplicado en el ámbito de la ingeniería civil, nos ayudará a mejorar procesos en la gestión y desarrollo estructural. (pág. 6)

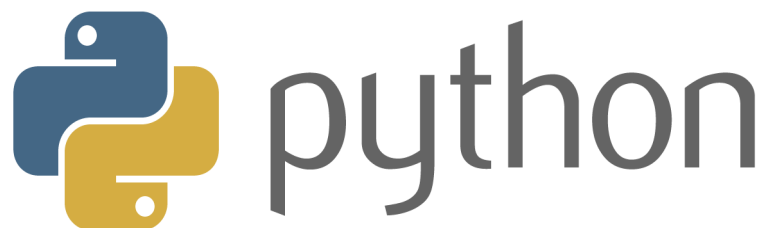


Figura 1 Logo de Python

2.2.2. Librerías de Python

El código creado mediante Python se puede combinar opcionalmente en paquetes y módulos, es muy versátil y permite a los usuarios realizar nuevas tareas que antes eran imposibles. En otras palabras, las librerías de Python

responden a un conjunto de implementaciones que permiten codificar el lenguaje para crear una interfaz autónoma que ayuda y mejora su funcionalidad, permitiéndonos mayor variedad de opciones y tareas, entre las librerías de Python tenemos a; Numpy, Pandas, Matplotlib las cuales usaremos en esta investigación.

2.2.2.1. NumPy

(Masip R, 2018), menciona que la librería NumPy permite manipular datos científicos, como análisis de datos o cálculos numéricos, ésta librería nos permite igualar el potencial que nos brinda Python, al de otros lenguajes como Octave o Matlab. NumPy, se enfoca en el tratamiento de matrices y arrays N-dimensionales (pág. 36) Python es una librería que contiene un soporte inclinado en las matemáticas de alto riesgo que facilita el desarrollo de distintas operaciones, como; algebraicas, matricial o vectorial.



Figura 2 *Logo de la librería Numpy*

2.2.2.2. Sympy

Según (Meurer, 2017), SymPy es un sistema de álgebra informática de código abierto escrito en Python puro, está construido con un enfoque en la extensibilidad y la facilidad de uso, a través de programas interactivos y aplicaciones. Estas características han llevado a SymPy a convertirse en una popular biblioteca simbólica para el ecosistema científico de Python. Esta versatilidad le permite al usuario aplicar y mejorar la capacidad en el cálculo de expresiones algebraicas. (pág. 1)

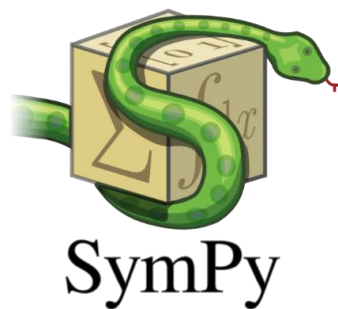


Figura 3 Logo de la librería Sympy

2.2.2.3. Pandas

Para (Wes, 2013), la librería Pandas brinda estructuras de funciones y datos diseñadas para trabajar con datos estructurados, con la finalidad de mejorar el proceso de datos, haciéndolo más rápido, fácil y expresivo. Pandas es una de las librerías fundamentales y críticas que permiten que Python sea un entorno donde se permita realizar análisis de datos poderosos y productivos, con la facilidad de poder ordenar, filtrar, agregar y transformar datos. . (pág. 4)



Figura 4 Logo de la librería Pandas

2.2.2.4. Matplotlib

Según (Yim, 2018), Matplotlib es un paquete de Python para la visualización de datos el cual permite la creación sencilla de varios gráficos, incluidos gráficos de líneas, dispersos, de barras, de caja y radiales, con gran flexibilidad para un estilo refinado y anotaciones personalizadas. Además de generar gráficos estáticos, Matplotlib también admite una interfaz interactiva que no solo ayuda a crear una amplia variedad de gráficos, sino que también es muy útil para crear aplicaciones basadas en la web. Matplotlib se integra fácilmente en entornos de desarrollo populares, como Jupyter Notebook, y es compatible con muchos paquetes de visualización de datos más avanzados. (pág. 7)



Figura 5 *Logo de la librería Matplotlib*

2.2.3. Anaconda

Según (Ross, 2020), Anaconda es un software gratuito que proporciona un grupo de herramientas las cuales han sido diseñadas para el desarrollo de la investigación y las ciencias. Al instalar el software Anaconda, nos permite acceder a distintos entornos de desarrollo los cuales permiten codificar en el lenguaje Python o R, dos de los más conocidos lenguajes de programación con desarrollo, integrado (IDE), ambas son distinguidas plataformas que facilitan y son de gran ayuda al momento de desarrollar un código de programación, gestionando los datos ingresados y facilitando la programación. (pág. 2)



Figura 6 *Logo del Software Anaconda*

2.2.4. Jupyter Notebook

Según (Wang, 2021), Jupyter Notebook es una aplicación dentro de Anaconda, la cual está compuesta por códigos, ecuaciones, texto narrativo y visualizaciones, por ello con el pasar del tiempo este sistema se ha convertido en un sistema más amplio utilizado para la programación alfabetizada interactiva, donde los científicos registran datos para realizar una fácil comprensión y análisis de datos. En septiembre de 2018, más de 2,5 millones de repositorios de Jupyter fueron almacenados en GitHub, 10 veces más que en 2015. (pág. 1)



Figura 7 *Logo de la aplicación Jupyter Notebook*

2.2.5. OpenGL

Según (Mateos Sánchez, 2021), OpenGL es una interfaz software utilizada para gráficos por hardware. Es una biblioteca de alto rendimiento que permite trazar gráficos de manera fácil, portable y con gran flexibilidad. Consta de aproximadamente 120 comandos y funciones que se utilizan para especificar objetos y operaciones en aplicaciones gráficas interactivas en 2D y 3D. Esta tecnología fue desarrollada y optimizada por Silicon Graphics. Por otro lado, PyOpenGL es una biblioteca de Python que actúa como un enlace o interfaz para utilizar las funcionalidades de OpenGL desde el código de Python. Proporciona a los desarrolladores de Python la capacidad de aprovechar todas las funciones y características de OpenGL en sus aplicaciones. OpenGL es ampliamente utilizado en la industria de los gráficos por computadora, ya que permite la creación y manipulación de gráficos en tiempo real. Con PyOpenGL, los desarrolladores de Python pueden utilizar estas capacidades para crear visualizaciones, simulaciones y aplicaciones interactivas en 2D y 3D.

En resumen, PyOpenGL es una biblioteca de Python que permite a los desarrolladores aprovechar las capacidades de OpenGL para crear gráficos 2D y 3D en aplicaciones Python. Esta biblioteca proporciona enlaces a la API de OpenGL y ofrece una interfaz conveniente para trabajar con gráficos por computadora en Python. (pág 5.)

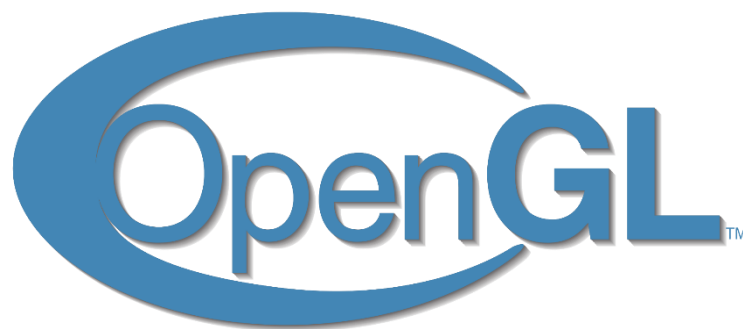


Figura 8 Logo de OpenGL

2.2.6. Programación

Según (Caceres Espinoza, 2019), en su investigación define que la programación es una “secuencia de instrucciones codificadas como un procedimiento”, es decir, se trata de un lenguaje el cual, luego es decodificado y ejecutado por un sistema informático, el cual tiene por finalidad resolver una problemática en específico planteada anteriormente. Se puede decir que el la programación es una implementación de algoritmos y lenguajes de programación para aportar soluciones a problemas. (pág 9.)

2.2.6.1. Lenguaje de Programación

Según (Caceres Espinoza, 2019), menciona que un lenguaje de programación se trata de un conjunto de comandos escritos en una secuencia los cuales respetan la sintaxis que resuelve una cierta problemática planteada en términos de información. El lenguaje de programación es representado por un conjunto de instrucciones definidas específicamente para una computadora digital. Aunque existen bastantes tipos de lenguajes de programación, y se utilizan dependiendo del problema a resolver, hay unos que son más conocidos que otros. (pág. 10)

2.2.6.2. Tipos de lenguaje de programación

Según (Manrique Rojas, 2019), en su investigación, define que existen distintos lenguajes de programación los cuales son utilizados con la finalidad de desarrollar aplicaciones de Inteligencia Artificial y Machine Learning. Cada aplicación utilizada para programar tiene sus características, y alguno podría ser mejor que otro dependiendo de la problemática para la cual sea utilizado. Los lenguajes de programación han ido mejorando a lo largo de los años, y otros han sido creados con la finalidad de satisfacer los requisitos únicos de las aplicaciones de Inteligencia Artificial. (pág.5)



Figura 9 *Lenguajes de programación*

2.2.6.3. Código fuente

Para (Kernighan, 2021) en su libro llamado “El lenguaje de programación C” nos dice que, el código fuente de programación es el conjunto de instrucciones escritas en un lenguaje de programación específico. Es un texto legible por humanos que contiene las órdenes y algoritmos que guían a un programa de computadora para realizar una determinada tarea o función. El código fuente es creado por los programadores y actúa como un plano o diseño para el software. Proporciona las instrucciones detalladas que la computadora debe seguir para ejecutar el programa correctamente. El código fuente puede ser escrito en varios lenguajes de programación, como C, C++, Java, Python, JavaScript, entre otros. Una vez que se ha escrito el código fuente, es necesario compilarlo o interpretarlo para convertirlo en un lenguaje que la máquina pueda entender y ejecutar. El resultado de este proceso es un archivo ejecutable o un código intermedio, dependiendo del lenguaje de programación utilizado. En resumen, el código fuente de programación es el texto que contiene las instrucciones detalladas escritas por los programadores para construir y ejecutar programas de computadora. Es la base del desarrollo de software y permite a los programadores trabajar en conjunto, mejorar y mantener aplicaciones de manera eficiente.

III. METODOLOGÍA

3.1. Tipo y diseño de investigación:

3.1.1. Tipo de investigación:

La presente investigación se considera como aplicada, (Lozada, 2014) afirma que una investigación aplicada busca generar una aplicación directa en el sector productivo y los problemas de la sociedad, basada principalmente en la realización de la investigación básica, la cual considera realizar el proceso de vincular el producto y la teoría; ya que tiene objetivos establecidos como la mejora constante de sus resultados basada en el conocimiento previo que se tiene de la investigación. (pág. 1)

3.1.2. Diseño de investigación:

El diseño investigación en el presente proyecto de tesis es no experimental transversal descriptivo.

Según (Alva, 2013), el estudio una investigación es considerada no experimental cuando esta es realizada sin modificar las variables propuestas la cual se enfoca en analizar, describir y comprender la información existente, por otro lado, se considera transversal cuando la data es recolectada una sola vez, es decir en un tiempo único y descriptiva porque puede especificar propiedades, características, perfiles de grupos, personas o cualquier otro ítem que pueda someterse a un análisis. (pág. 5)

3.2. Variables y operacionalización

Viga: La Norma Técnica de Edificación E.060 de concreto armado, afirma que, la viga es considerada como un elemento que trabaja principalmente a flexión y cortante. Por lo tanto, una viga se puede definir como un elemento estructural horizontal que resiste cargas, también es el elemento estructural encargado de soportar las cargas del techo o el tipo de carga asociada a los elementos que componen el techo, como por ejemplo la tabiquería. Las vigas pueden estar hechas de distintos materiales como

madera, acero, hormigón armado o distintos materiales compuestos; dichos materiales serán considerados de acuerdo al diseño, factor económico y condiciones ambientales del proyecto.

Asimismo, las vigas son diseñadas para resistir fuerzas de flexión de manera horizontal donde se deben considerar fuerzas sísmicas, fuerzas de viento y distintos factores externos que puedan afectar a las mismas.

Viga continua: Según el Diccionario Español de Ingeniería de la Real Academia de Ingeniería, se considera una viga continua cuando ésta soporta dos o más apoyos para obtener una mayor rigidez, de manera que se pueda calcular el efecto que tendría una carga sobre vigas individuales de igual luz. El diseño de las vigas continuas es enfocado de acuerdo a los factores a los que se presentan como; las características de los apoyos, luces entre los apoyos, material empleado y cargas aplicadas; estos factores no solo influirán en el diseño transversal de la viga, también influirá en la determinación de los momentos de flexión, deflexiones resultantes y ubicación de la viga.

3.2.1. Clasificación de Variables

Variables	Relación	Naturaleza	Escala de medición	Dimensión	Forma de medición
Viga continua	Independiente	Cuantitativa	Razón	Tridimensional	Indirecta

Tabla 1 *Clasificación de variables*

3.3. Población, muestra y muestreo

3.3.1. Población:

La población estimada en este caso serían todas las vigas continuas de 2 vanos que estén sometidas a cargas continuas y se puedan analizar en Python.

La población de una investigación es el grupo de objetos o individuos que comparten cierta característica en específico, la cual se investiga gracias a los datos o características que comparten al momento de realizar la investigación.

Así mismo, se debe considerar que es fundamental validar y garantizar los resultados obtenidos, considerando los objetivos que se plantearon previamente.

3.3.2. Muestra:

Está conformada por una viga continua de 2 vanos con cargas uniformemente distribuidas y parámetros definidos.

Según (Carrillo Flores, 2015), nos referimos a muestras no probabilísticas, cuando no todos los elementos del universo tienen la misma probabilidad de formar parte de la muestra, por consiguiente, sus resultados no son aplicables a la población en cuestión; considerando que los datos recolectados y analizados son enfocados en generar resultados en base a la población total. (pág. 12)

3.3.3. Muestreo:

No probabilístico. (Espinoza, 2016), asevera que un muestreo no probabilístico tiene diferentes elementos de la población del estudio realizado, considerando que no se deben generalizar los objetivos obtenidos basados en lo más conveniente para la investigación. (pág. 18)

3.4. Técnicas e instrumentos de recolección de datos

(Danae, 2020), menciona que al realizar un trabajo de investigación se debe considerar las técnicas, instrumentos y métodos que aseguren el hecho empírico de la investigación realizado, ya que con las técnicas de investigación son un conjunto de instrumentos y con el método se puede ver el camino a seguir en la investigación. (pág. 1)

3.4.1. Técnicas de recolección de datos

La técnica que se utilizará para la recolección de datos del presente proyecto es el método de inspección visual más conocido como “observación”, el cual tiene como finalidad obtener datos reales los cuales servirán para llevar a cabo el análisis estructural de una viga continua de 2 vanos mediante el código de programación realizado en Python.

3.4.2. Instrumento de recolección de datos

Se utilizó el instrumento de validación de datos “Análisis de viga continua de 2 vanos mediante Python” el cual nos permitirá obtener las propiedades requeridas de la respectiva viga continua analizada para llevar a cabo el análisis de la viga continua de 2 vanos mediante Python, el presente instrumento ha sido validado por el Ing. Junior Xavier Gavidia Valera, ver Anexo-1.

3.4.3. Validación de instrumento de recolección de datos

La validez del presente instrumento de recolección de datos el cual fue utilizado para la realización de la presente investigación, fue otorgada por el ingeniero Junior Xavier Gavidia Valera, con colegiatura número 270428, quien cuenta con 5 años de experiencia en el sector construcción.

La validación de la recolección de datos es la forma de realizar una verificación y validez de los datos que nos brinda el instrumento de validación, donde se considerará que nuestro instrumento de validación abarque adecuadamente el tema a investigar.

3.5. Procedimientos

Como trabajo previo se realizó una búsqueda de antecedentes los cuales sirvieron de punto de partida para la realización de nuestro proyecto de investigación. A partir de ello se utiliza la realización de un código mediante el lenguaje de programación de Python, el cual, se encargará de procesar la data dada en este caso vendrían a ser las medidas y cargas de la viga, luego, al correr el código nos dará las respuestas que buscamos a las interrogantes planteadas.

(Donolo, 2009), menciona que el poder organizar y adaptar los datos obtenidos al estudio realizado es una ventana para poder contribuir a la preparación teórica del tema en estudio. Donolo propone el método de la triangulación como procedimiento metodológico, teniendo en cuenta la interpretación de la investigación, la recolección de data, el análisis de los datos obtenidos; por ello debemos tener en cuenta el alcance y los límites de la triangulación para así poder incorporar estos datos obtenidos a la investigación. (pág.1)

3.6. Método de análisis de datos

En el desarrollo de la presente investigación se utilizó el método de análisis llamado estadística descriptiva.

(Suárez Ibujés, 2018), nos dice que la estadística descriptiva es la ciencia encargada de organizar, recopilar, presentar, interpretar y analizar datos de manera informativa, con la finalidad de describir fácilmente las características esenciales de los datos resultantes con el empleo de diversos métodos gráficos, tabulares o, numéricos, etc. (pág. 15)

En la presente investigación, se utilizó un método de recolección de datos mediante un cuadro de Excel, con la finalidad de analizar las fuentes obtenidas para su posterior selección; obteniendo así los mejores antecedentes para nuestra investigación, la cual consiste en realizar un código de Python aplicado a la ingeniería civil.

(Martínez, 2014), menciona que realizar un análisis de datos, es utilizado para interpretar, comprender, valorar y mejorar la investigación

realizada aportando conocimientos de distintas investigaciones aportándole valor metodológico. Así mismo, debemos tomar en cuenta que toda investigación es realizada para obtener una respuesta a un problema en específico, un problema real detectado en la práctica; interpretando así los datos obtenidos y mediante la recolección de datos realizada previamente. (pág. 20)

3.7. Aspectos éticos

En este estudio hemos obtenido resultados bajo los criterios de evaluación previamente establecidos, con la finalidad de obtener resultados reales y confiables para nuestra investigación, todo ello se presenta de forma honesta y responsable.

Así mismo, (Lilia, 2013), menciona que los aspectos éticos en una investigación convencional son aplicables a la ciencia en general, considerando que se debe tener la información clara y establecida para establecer los procedimientos, riesgos . (pág. 1)

IV. RESULTADOS

Tabla 2 *características de la viga continua*

Carga 1	q1	4.5 kN/m
Carga 2	q2	3.2 kN/m
Longitud 1	l1	4 m
Longitud 2	l2	5 m
Longitud total viga	l1+l2	9 m
Reacción A	Ra	6.8344 kN
Reacción B	Rb	20.8981 kN
Reacción C	Rc	6.2675 kN
Inercia respecto al eje y de la viga	ly	27720000 mm ⁴
Módulo Elasticidad	E	200000 MPa
Cortante máxima	V _{max}	11.38890 kN
Momento máximo	M _{max}	9.55560 kN*m
Deformación máxima	u _{max}	-11.867694 mm

Nota: Como resultados del código de Python, se obtuvieron los siguientes parámetros, el valor de cortante máximo $V_{max} = 11.38890$ kN, el momento máximo de momento $M_{max} = 9.55560$ kN*m y la máxima deformación $u_{max} = -11.867694$ mm.

Tabla 3 Archivo CSV generado por el código de Python

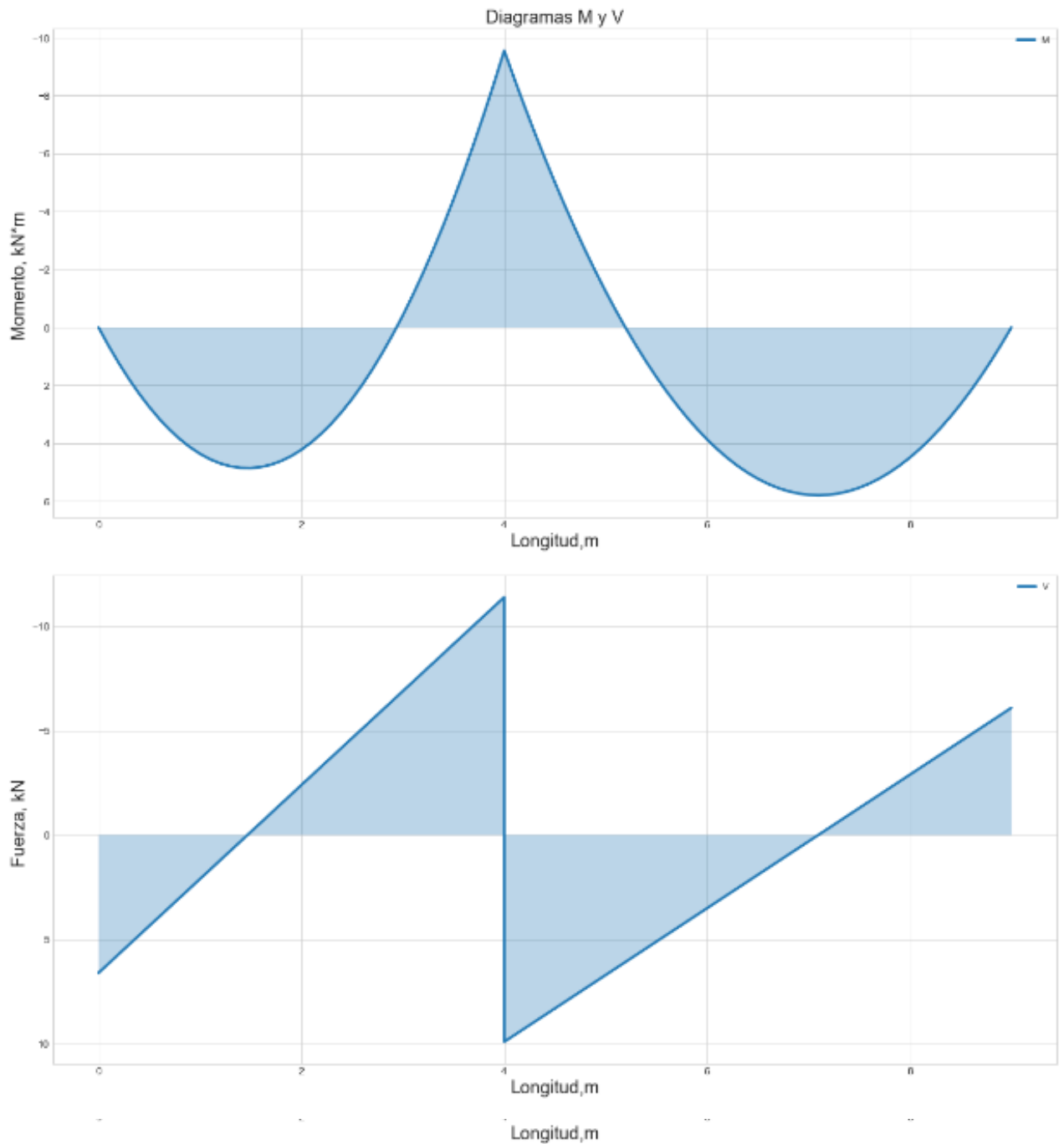
x	M	V
0	0	6.61111111
0.1	0.63861111	6.16111111
0.2	1.23222222	5.71111111
0.3	1.78083333	5.26111111
0.4	2.28444444	4.81111111
0.5	2.74305556	4.36111111
0.6	3.15666667	3.91111111
0.7	3.52527778	3.46111111
0.8	3.84888889	3.01111111
0.9	4.1275	2.56111111
1	4.36111111	2.11111111
1.1	4.54972222	1.66111111
1.2	4.69333333	1.21111111
1.3	4.79194444	0.76111111
1.4	4.84555556	0.31111111
1.5	4.85416667	-0.13888889
1.6	4.81777778	-0.58888889
1.7	4.73638889	-1.03888889
1.8	4.61	-1.48888889
1.9	4.43861111	-1.93888889
2	4.22222222	-2.38888889
2.1	3.96083333	-2.83888889
2.2	3.65444444	-3.28888889
2.3	3.30305556	-3.73888889
2.4	2.90666667	-4.18888889
2.5	2.46527778	-4.63888889
2.6	1.97888889	-5.08888889
2.7	1.4475	-5.53888889

2.8	0.87111111	-5.98888889
2.9	0.24972222	-6.43888889
3	-0.41666667	-6.88888889
3.1	-1.12805556	-7.33888889
3.2	-1.88444444	-7.78888889
3.3	-2.68583333	-8.23888889
3.4	-3.53222222	-8.68888889
3.5	-4.42361111	-9.13888889
3.6	-5.36	-9.58888889
3.7	-6.34138889	-10.0388889
3.8	-7.36777778	-10.4888889
3.9	-8.43916667	-10.9388889
4	-9.55555556	-11.3888889
4	-9.55555556	9.91111111
4.1	-8.58044444	9.59111111
4.2	-7.63733333	9.27111111
4.3	-6.72622222	8.95111111
4.4	-5.84711111	8.63111111
4.5	-5	8.31111111
4.6	-4.18488889	7.99111111
4.7	-3.40177778	7.67111111
4.8	-2.65066667	7.35111111
4.9	-1.93155556	7.03111111
5	-1.24444444	6.71111111
5.1	-0.58933333	6.39111111
5.2	0.03377778	6.07111111
5.3	0.62488889	5.75111111
5.4	1.184	5.43111111
5.5	1.71111111	5.11111111
5.6	2.20622222	4.79111111
5.7	2.66933333	4.47111111
5.8	3.10044444	4.15111111
5.9	3.49955556	3.83111111
6	3.86666667	3.51111111

6.1	4.20177778	3.19111111
6.2	4.50488889	2.87111111
6.3	4.776	2.55111111
6.4	5.01511111	2.23111111
6.5	5.22222222	1.91111111
6.6	5.39733333	1.59111111
6.7	5.54044444	1.27111111
6.8	5.65155556	0.95111111
6.9	5.73066667	0.63111111
7	5.77777778	0.31111111
7.1	5.79288889	-0.00888889
7.2	5.776	-0.32888889
7.3	5.72711111	-0.64888889
7.4	5.64622222	-0.96888889
7.5	5.53333333	-1.28888889
7.6	5.38844444	-1.60888889
7.7	5.21155556	-1.92888889
7.8	5.00266667	-2.24888889
7.9	4.76177778	-2.56888889
8	4.48888889	-2.88888889
8.1	4.184	-3.20888889
8.2	3.84711111	-3.52888889
8.3	3.47822222	-3.84888889
8.4	3.07733333	-4.16888889
8.5	2.64444444	-4.48888889
8.6	2.17955556	-4.80888889
8.7	1.68266667	-5.12888889
8.8	1.15377778	-5.44888889
8.9	0.59288889	-5.76888889
9	0	-6.08888889

Nota: La tabla 3 muestra lo que hay dentro del archivo CSV generado por el código de Python realizado para el análisis de la viga continua de 2 vanos, el cual contiene los valores de M y V a lo largo de la viga.

Tabla 4 Diagramas de M y V a lo largo de la viga



Nota: La tabla 4 muestran los diagramas de Momento y Cortante a lo largo de la viga dada por el código de Python realizado para el análisis de la viga continua de 2 vanos, estos diagramas son generados del archivo CSV previamente generado por el código.

V. DISCUSIÓN

A partir de las hipótesis general planteada anteriormente, hemos podido observar que sí es posible realizar el análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python; así mismo también se afirma que sí es posible determinar el máximo cortante a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, del mismo modo, sí es posible determinar el máximo momento flector a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, y por ende también se ha determinado que sí es posible hallar la máxima deformación a través del análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python.

(Ortiz Carvajal, 2021), en su investigación nos menciona que sí es posible obtener los resultados esperados al realizar cálculos mediante softwares como Python, ya que este tipo de programación facilita y aporta una mejora en el desarrollo de los cálculos enfocados en la ingeniería civil, sosteniendo así que como resultado de nuestra investigación sí se lograron obtener los datos requeridos tras ingresar los datos planteados.

(Arriola Carrasco, 2021), el desarrollo y aplicación de Python en la ingeniería civil no solo facilita la realización de cálculos, la implementación de estos softwares también nos permite mantenernos en constante retroalimentación ya que al investigar la aplicación de Python en la Ingeniería Civil se podrá aportar bases de datos al desarrollo de futuras investigaciones en este tema; sosteniendo así que en nuestra investigación se han generado los diagramas de momento y cortante mediante el código Python facilitando así el cálculo de una viga de 2 vanos.

(Pascual, 2018), realizó una formulación y caracterización de una viga biapoyada en el programa Python, de una forma más fácil, económica y rápida ya que nos permitirá desarrollar estructuras más complejas como puentes o pasarelas, así mismo en nuestra investigación también se pudo observar que el desarrollo de los gráficos mediante Python se puede desarrollar con mayor facilidad y precisión. (pág. 56)

VI. CONCLUSIONES

- El valor del cortante máximo ($V_{\max} = 11.38890 \text{ kN}$): Este parámetro indica la máxima fuerza cortante experimentada en la viga continua analizada. En este caso, se obtuvo un valor de V_{\max} de 11.38890 kN , lo que sugiere que la estructura puede resistir una fuerza cortante hasta ese límite.
- El valor del Momento máximo ($M_{\max} = 9.55560 \text{ kN}\cdot\text{m}$): El momento máximo obtenido representa la máxima fuerza de torsión o flexión a la que está sometida la estructura. Es esencial considerar este valor con la finalidad de garantizar que la estructura sea capaz de resistir las cargas aplicadas sobre la viga continua sin deformarse excesivamente. En este caso, el momento máximo M_{\max} de $9.55560 \text{ kN}\cdot\text{m}$ indica la máxima fuerza de torsión o flexión que la estructura puede resistir.
- Máxima deformación ($u_{\max} = -11.867694 \text{ mm}$): La deformación máxima indica la cantidad máxima de cambio o distorsión experimentada por la viga continua debido a las cargas aplicadas. En este caso, la máxima deformación registrada fue de -11.867694 mm , lo que sugiere que la viga continua experimentó una deformación negativa en esa magnitud. Es importante evaluar si esta deformación cumple con los criterios de diseño y si se mantiene dentro de los límites aceptables.
- El desarrollo de Python nos permite realizar con mayor facilidad y eficiencia el análisis estructural de una viga continua de 2 vanos, mejorando así la precisión de los cálculos; donde se pudo concluir que sí se lograron obtener los resultados requeridos, con los valores correctos los cuales son, análisis estructural de una viga continua de 2 vanos, mejorando así la precisión de los cálculos.
- Se concluye que el análisis de la viga continua de 2 vanos, nos permite obtener los valores de M y V a lo largo de la viga, de una forma más práctica y eficiente gracias al programa Python permitiéndonos así poder verificar los datos obtenidos previamente.
- Python es un lenguaje de programación versátil muy utilizado en ingeniería, esto se debe a su facilidad de uso y la gran cantidad de bibliotecas especializadas disponibles.

- El uso de Python en el análisis estructural de una viga continua de 2 vanos permite automatizar y agilizar el proceso de cálculo, lo que se traduce en un importante ahorro de tiempo y esfuerzo.
- Python ofrece una amplia gama de bibliotecas científicas y de análisis numérico, tales como NumPy y SciPy, que proporcionan herramientas para realizar cálculos complejos y resolver ecuaciones diferenciales relacionadas con la mecánica estructural.
- El lenguaje Python también cuenta con bibliotecas de gráficos, como Matplotlib, la cual facilita la visualización de resultados y la creación de gráficos para el análisis de los resultados calculados.
- Python es un lenguaje de programación de código abierto, lo que significa que es gratuito y cuenta con una gran comunidad de desarrolladores que comparten sus conocimientos y aportan nuevas bibliotecas y mejoras constantes.
- Python permite la integración con otras herramientas de software comúnmente utilizadas en ingeniería civil, como CAD y software de modelado estructural, lo que facilita la transferencia de datos y resultados entre diferentes programas.
- El uso de Python en el análisis estructural de una viga continua de 2 vanos puede mejorar la precisión de los cálculos al permitir la implementación de algoritmos más complejos y avanzados, como el método de los elementos finitos.
- El uso de Python para el análisis estructural en ingeniería civil no solo es útil para analizar vigas continuas de 2 vanos, sino que también abre la puerta a una amplia gama de aplicaciones en otros campos de la ingeniería civil, como el modelado estructural, análisis tridimensional y sísmico, entre otros.
- Python proporciona la capacidad de realizar análisis paramétricos, esto significa que es posible resolver distintas variaciones en los parámetros de diseño de la viga continua de 2 vanos de manera eficiente y automatizada, siempre y cuando se cumplan las condiciones dadas para el correcto funcionamiento del código.

- Python ofrece una herramienta poderosa y versátil para la ingeniería civil al permitir el desarrollo de soluciones personalizadas y flexibles. El código desarrollado para analizar una viga continua de 2 vanos puede ser modificado y adaptado para aplicarse a otras estructuras de ingeniería civil, lo que amplía su utilidad y eficiencia.
- La flexibilidad de Python como lenguaje de programación, junto con su amplia gama de bibliotecas especializadas, facilita la modificación y personalización del código existente. Esto significa que los ingenieros civiles pueden ajustar y adaptar el código de la presente investigación a diferentes geometrías, materiales, condiciones de carga y otros parámetros específicos de una estructura en particular.

VII. RECOMENDACIONES

- Se recomienda buscar la teoría a utilizar antes de empezar con el desarrollo del código, de esta forma se agiliza el proceso de a realización del código, empleando los principios fundamentales del análisis estructural tales como carga, deformación, tensión y desplazamiento; para así poder desarrollar y plasmar el código en Python.
- Al obtener los datos y gráficos dados en el programa Python, se recomienda verificar su fiabilidad con otros programas, ya que al momento de tener estos datos verificados, entonces el código realizado tendrá más veracidad en sus resultados.
- Se recomienda para el desarrollo de futuros códigos de programación en base a la presente investigación, realizar el código en Python de una forma correcta y ordenada, para poder realizar modificaciones o mejoras en un futuro y de tal manera de que el código en cuestión se mantenga actualizado.
- Utilizar técnicas analíticas o programas de referencia, esto sirve para validar los resultados de sus cálculos, con la finalidad de garantizar la precisión y confiabilidad de sus cálculos, compare sus resultados con los cálculos de otros programas para verificar la fiabilidad de su código.
- Grafique los resultados de su análisis estructural de vigas continuas de 2 tramos para comprender y transmitir mejor los hallazgos, puede usar bibliotecas de visualización como Matplotlib o Plotly para crear gráficos y visualizaciones para este propósito.
- Asegúrese de documentar y comentar correctamente su código para que pueda entenderse y mantenerse en el futuro. Para hacer que su código sea simple para que otros ingenieros lo lean y lo comprendan, use comentarios claros y una estructura lógica.
- Participar y ser parte de la comunidad de Python en internet, ya que actualmente, por ejemplo, en la plataforma Reddit, existe un foro con más de 1 millón de miembros, dispuestos a resolver dudas o a dar opiniones acerca del funcionamiento de un código o biblioteca de Python en específico.

REFERENCIAS

- **Aguilar Nevado, José y Exebio Tantarico, José. 2017.** *Sistema de Análisis Estructural en Plataforma Java para Vigas Continuas por el Método Elementos Finitos.* Perú, UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO. 2017.
- **Arriola Carrasco, Guillermo Gustavo. 2021.** *USO DEL SOFTWARE LIBRE COMO APORTE AL SÍLABO HIDROLOGÍA DE LA UNIVERSIDAD SEÑOR DE SIPÁN EN TIEMPOS DE COVID-19.* 2021.
- **Bacuy, Christian y Romero, Esteven. 2022.** Optimización de edificios con disipadores de energía pasivos utilizando algoritmos genéticos desarrollados en python 3 adaptados a etabs. *El Repositorio Digital Institucional de la Escuela Politécnica Naciona.* 2022, pág. 30.
- **Cabanillas Alcívar, Frank. 2020.** *Influencia de las paredes de mampostería en estructuras de pórticos de hormigón armado.* Universidad Católica de Santiago de Guayaquil. Ecuador : s.n., 2020.
- **Cabrera Jara, Pablo. 2020.** *Análisis estático lineal y no lineal de un edificio de hormigón armado de 3 pisos incorporando la influencia de la mampostería, utilizando el método de diagonales equivalentes, mediante SAP2000.* Ecuador : Universidad del Azuay, 2020.
- **Caceres Espinoza, Liliana. 2019.** [En línea] 15 de mayo de 2019. <https://repositorio.une.edu.pe/handle/20.500.14039/4362>.
- **Carini, Matheus Roman y Rocha, Marcelo Maia. 2022.** *CESSIPy: un módulo de código abierto de Python para la identificación de sistemas estocásticos en ingeniería civil.* [ed.] SOFTWARE X. BRASIL : s.n., 2022.
- **Carrillo Flores, Ana Lilia. 2015.** <http://ri.uaemex.mx/>. [En línea] 2015. <http://ri.uaemex.mx/oca/view/20.500.11799/35134/1/secme-21544.pdf>.

- **Cisneros. 2020.** Optimización de estructuras de acero utilizando diseño paramétrico y algoritmos genéticos en un entorno BIM. *Universidad Politécnica de Cataluña - BarcelonaTech.* 2020, pág. 50.
- **Contreras Landa, Luis. 2017.** *Análisis del comportamiento de vigas de concreto reforzadas con fibra de carbono.* México : Repositorio del Tecnológico de Monterrey, 2017.
- **Cruz, Dr. Pedro Ponce. 2010.** *INTELIGENCIA ARTIFICIAL CON APLICACIONES A LA INGENIERÍA.* México D.F. : Alfaomega Grupo Editor, S.A. de C.V., México, 2010.
- **Danae, Duana. 2020.** *Técnicas e instrumentos de recolección de datos.* México : Boletín Científico de las Ciencias Económico Administrativas del ICEA, 2020.
- **David, Masip Rodó. 2018.** *El lenguaje Python.* Universidad Oberta de Catalunya. s.l. : España, 2018.
- *Diseño integral de armaduras metálicas usando rhinoceros 3D: Modelación paramétrica, simulación estructural y optimización.*
Gonzales Ramirez, Jhoseph Alejandro. 2018. 2018, Universidad Nacional de Cajamarca.
- **Donolo, Danilo Silvio. 2009.** *Triangulación: Procedimiento incorporado a nuevas metodologías de investigación.* México : Revista UNAM, 2009.
- **Espinoza, Iván. 2016.** *Tipos de muestreo.* México : Universidad Nacional Autónoma de México, 2016. pág. 18.
- **Estela Urbina, Ronald Omar. 2022.** *Métodos numéricos aplicados al cálculo hidráulico en canales de regadío de Bagua.* Revista Unibagua. Perú : s.n., 2022.
- **Grajales G., Teni. 2000.** TIPOS DE INVESTIGACION. 2000, pág. 2.
- **Gutiérrez, David, Moreno, Dannyris y Sevilla, Carlos. 2019.** Software para diseño de vigas aperaltadas de concreto reforzado aplicando el modelo puntal-tensor según ACI 318-14. *Universidad Nacional de Ingeniería.* 2019, pág. 22.

- **Herrera, Juan. 2016.** Análisis de vigas metálicas rigidizadas transversalmente durante el lanzamiento de puentes. *UPC Barcelonatech*. 2016, págs. 31-37.
- **Kernighan, Brian W. 2021.** *El lenguaje de programación C*. Naucalpan de Juarez, México : AT & T, 2021. Vol. Segunda edición.
- **Lilia, Martha. 2013.** *Aspectos éticos en la investigación cualitativa*. México : Enfermería Neurológica, 2013.
- **Lozada, José. 2014.** *Investigación Aplicada, Definición, Propiedad Intelectual e Industria*. Quito, Ecuador : s.n., 2014.
- **Lupaca, Denis. 2022.** *Desarrollo del simulador web Beli para el análisis matricial de estructuras planas y espaciales*. Perú : Universidad César Vallejo, 2022.
- **Lutz, Mark. 2013.** *Learning Python*. 2013.
- **Manrique Rojas, Esperanza. 2019.** [En línea] 10 de Noviembre de 2019.
<https://www.proquest.com/openview/c7e24c997199215aa26a39107dd2fe98/1?pq-origsite=gscholar&cbl=1006393>.
- **Manzano. 2015.** Optimización estructural mediante el uso acoplado de algoritmos genéticos y elementos finitos. *Universidad Carlos III de Madrid*. 2015, pág. 96.
- **Martínez, Catalina. 2014.** *Técnicas e instrumentos de recogida y Análisis de datos*. [ed.] Universidad Nacional de Educación a Distancia. España : s.n., 2014.
- **Martínez, César. 2021.** ¿Qué es Python y por qué es tan popular dentro del mundo de la programación? *Galileo Universidad*. [En línea] 2021. <https://www.galileo.edu/esec/historias-de-exito/que-es-python-y-por-que-es-tan-popular-dentro-del-mundo-de-la-programacion/>.
- **Masip R, David. 2018.** Universitat Orbeta de Catalunya. [En línea] 2018. <https://d1wqtxts1xzle7.cloudfront.net/32652932/Inteligencia-Artificial-Lenguaje-Python-with-cover-page-v2.pdf?Expires=1666583419&Signature=FRsLu-INvTBjUMc1FGU7OUvVVxauTR1UVNmzZ80KUVKaBrLsD~xGGVU>

VXkCxd3hYWYy3A0vqv304wIPPo5QzZHP0f4dLbcKcNqSqudAHEPI
OtHegunVm.

- **McKinney, Wes. 2013.** *Python for Data Analysis*. 2013.
- **Medina, Jorge y Moscoso, Álvaro. 2022.** Análisis de unión viga-columna mediante el método de los elementos finitos para definir un armado eficiente. *Universidad Privada Boliviana*. Bolivia : s.n., 2022, pág. 81.
- **Meurer, Aaron. 2017.** *SymPy: symbolic computing in Python*. 2017.
- **Ministerio de Vivienda, Construcción y Saneamiento. 2009.** Norma Técnica de Edificación de Concreto Armado. 08 de Mayo de 2009.
- **Moreno, Lucía. 2017.** Herramientas avanzadas de análisis de datos de aplicación en ingeniería civil : Machine Learning using Python. *Universidad de Cantabria*. España : s.n., 2017, pág. 17.
- **Ortiz Carvajal, Diana Elizabeth. 2021.** *Interfaz de Consulta con Lenguaje Python para el Catastro de Redes de Acueducto*. 2021.
- **Pascual, Álvaro. 2018.** Dinámica de barras: Caracterización experimental, formulación computacional en Python y calibrado aplicado a una viga biapoyada. *Universidad de Valladolid*. España : s.n., 2018, pág. 08.
- **Pérez, Julio. 2016.** Componente para el modelado de vigas para el Sistema CAD 2D. *REPXOS*. s.l.: Universidad de las Ciencias Informáticas, 2016, pág. 13.
- **Picha, Gleddynuri. 2021.** Análisis y diseño de pórticos de concreto armado usando inteligencia artificial. Perú : s.n., 2021.
- **Puente, Adrián Polo. 2016.** Estudio elastoplástico de vigas sometidas a flexión : cálculo de estructuras y/o elemento estructural por procedimiento analítico con alguna herramienta informática. *Universidad de Salamanca*. España : s.n., 2016, pág. 35.
- **Puga. 2019.** Desarrollo de un código de programación en lenguaje Python para el estudio de zonas forestales. *ESCUELA POLITÉCNICA NACIONAL*. 2019, pág. 20.

- **Real Academia de Ingeniería. 2022.** [diccionario.raing.es/. Diccionario Español de Ingeniería.](https://diccionario.raing.es/es/lema/viga-continua) [En línea] 2022. <https://diccionario.raing.es/es/lema/viga-continua>.
- **Rodríguez, William. 2022.** Análisis y evaluación de datos para dosificación de mezclas de hormigón mediante el desarrollo de un software en lenguaje de programación Python aplicando el método ACI. *Universidad Técnica de Ambato.* 2022, pág. 51.
- **Román, Matheus. 2022.** CESSIPy: A Python open-source module for stochastic system identification in civil engineering. *Federal University of Rio Grande do Sul.* s.l. : Federal University of Rio Grande do Sul, 2022, pág. 06.
- **Ross, Matt. 2020.** Introduction to Anaconda and Python: Installation and setup. 2020.
- **Salcedo Malaver, Christian Gonzalo. 2015.** *Análisis estructural por el método de elementos finitos asistido por computadora (vigas-pórticos, placas, sólidos de revolución).* Cajamarca. Cajamarca : s.n., 2015. Tesis.
- **San José, Alonso. 2015.** Aplicación del método de elementos finitos extendido a la propagación de fisuras en vigas cilíndricas. [aut. libro] Universidad Carlos III de Madrid. 2015, pág. 24.
- **Sepúlveda, Néstor. 2022.** Rehabilitación sísmica de una estructura industrial existente en Chile mediante la incorporación de diagonales con pandeo restringido. *Univesidad de Chile.* 2022, pág. 71.
- **Sotelo Ojeda, Harold y Acuña Torres, Juan. 2015.** *Software Libre para el Diseño de Conexiones Metálicas de acuerdo a la NSR-10.* Pontificia Universidad Javeriana. Colombia : s.n., 2015.
- **Suárez Ibujés, Mario Orlando. 2018.** Repositorio Digital Universidad Técnica del Norte. [En línea] 13 de Diciembre de 2018. <http://repositorio.utn.edu.ec/handle/123456789/8696>.
- *Técnicas de inteligencia artificial aplicadas a problemas de ingeniería civil.* **Morales, Lic. Daril Alemán. 2017.** 2017, Revista de Arquitectura e Ingeniería. 2017, Vol.11 No.3 ISSN 1990-8830 / RNPS 2125.

- **Vargas Añamaco, Luis Alberto. 2021.** *Cuadernos interactivos, jupyter python notebook y el nivel de aprendizaje del.* Perú : Universidad Andina del Cuzco, 2021.
- **Vargas, Edwart y Ramírez, Juan. 2020.** Implementación de una plataforma web para determinar la vulnerabilidad sísmica en viviendas informales. *Universidad Distrital Francisco José de Caldas.* 2020, pág. 40.
- **Wang, Jiawei. 2021.** ACM Digital Library. [En línea] 2021. <https://dl.acm.org/doi/abs/10.1145/3324884.3416585>.
- **Wes, McKinney. 2013.** *Python for data analysis.* s.l. : OREILLY, 2013.
- **Yim, Aldrin. 2018.** *Matplotlib for Python Developers second edition.* Mumbai : s.n., 2018.

ANEXOS

ANEXO-1



Ficha de validación del instrumento

1. Datos generales


1.1 Título del trabajo de investigación

"Análisis estructural de una viga continua de 2 vanos mediante el programa Python, Trujillo 2023"

Investigadores: Felix Ronald Astudillo Morales; Pierina Alexandra Flores Adanaque.

ANÁLISIS DE VIGA CONTINUA DE 2 VANOS MEDIANTE PYTHON

VIGA CONTINUA DE 2 VANOS	und	
LONGITUD 1	m	
LONGITUD 2	m	
CARGA Q1	kN/m	
CARGA Q2	kN/m	
REACCIONES	KN	
MODULO DE ELASTICIDAD	MPa	
INERCIA	mm ⁴	
MOMENTO	KN*m	
CORTANTE	kN	
DEFORMACIÓN	mm	



JUNIOR XAVIER
GAVIDIA VALERA
Ingeniero Civil
CIP N° 270428

Figura 10 Instrumento de validación de datos

ANEXO-2

Matriz de consistencia “Análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, Trujillo 2023”

Problema general	Objetivos	Marco Teórico	Hipótesis	Variables	Metodología
¿Cómo es el Análisis estructural de una viga continua de 2 vanos mediante el programa Python?	<p>O. General:</p> <ul style="list-style-type: none"> – Realizar el análisis estructural de una viga continua de 2 vanos mediante librerías del lenguaje de programación Python. <p>O. Específicos:</p> <ul style="list-style-type: none"> – Determinar la cortante máxima en una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python. – Determinar el momento flector máximo en una viga continua de 2 vanos mediante librerías de 	<p>Pascual, Á. (2018). En su tesis de maestría titulada: “Dinámica de barras: Caracterización experimental, formulación computacional en Python y calibrado aplicado a una viga biapoyada.”</p> <p>Román, M. (2022). En su artículo científico titulado “CESSIPy: A Python open-source module for stochastic system identification in civil engineering”.</p> <p>Herrera, J. (2016). En su tesis de maestría titulada: “Análisis de vigas metálicas rigidizadas transversalmente durante el lanzamiento de puentes”</p>	<p>H. General:</p> <p>Sí es posible el análisis estructural de una viga continua de 2 vanos mediante el programa Python.</p>	<p>VARIABLE INDEPENDIENTE:</p> <p>Viga continua de 2 vanos.</p> <p>Según la norma técnica NTE E.060 de Concreto Armado, la viga es un elemento estructural que trabaja fundamentalmente a flexión y cortante.</p>	<p>Tipo de investigación:</p> <p>Propósito: Aplicada</p> <p>Por el diseño:</p> <p>No Experimental</p> <p>Diseño de Investigación:</p> <p>No experimental transversal descriptivo</p> <p>Unidad de Estudio:</p> <p>Vigas continuas de 2 vanos.</p>

lenguaje de programación en Python.
– Determinar la deformación máxima en una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python.

Gonzales, J. (2018) en su tesis para obtener el grado de ingeniero civil titulada “Diseño integral de armaduras metálicas usando rhinoceros 3D: Modelación paramétrica, simulación estructural y optimización.

Lupaca, D. (2022) en su tesis para obtener el título profesional de ingeniero civil titulada “Desarrollo del simulador web Beli para el análisis matricial de estructuras planas y espaciales”.

Picha, G. (2021) en su tesis para obtener el título profesional de ingeniero civil titulada “Análisis y Diseño de Pórticos de Concreto Armado usando Inteligencia Artificial”

Población:

La población estimada en este caso serían todas las vigas continuas de 2 vanos que estén sometidas a cargas continuas y se puedan analizar en Python.

Muestra: Está conformada por dos vigas continua de 2 vanos con medidas previamente definidas.

Muestreo: No probabilístico.

Técnicas, instrumentos y procedimientos de recolección de datos:

Para recolectar los datos, se utilizará la guía de recolección de datos, ver ANEXOS

ANEXO-1

Técnica:

Observación

Instrumento:

Vigas continuas de 2 vanos mediante Python.

Análisis de datos:

Estadística descriptiva

Tabla 5 *Matriz de consistencia*

ANEXO-3

Funcionamiento del código de Python realizado.

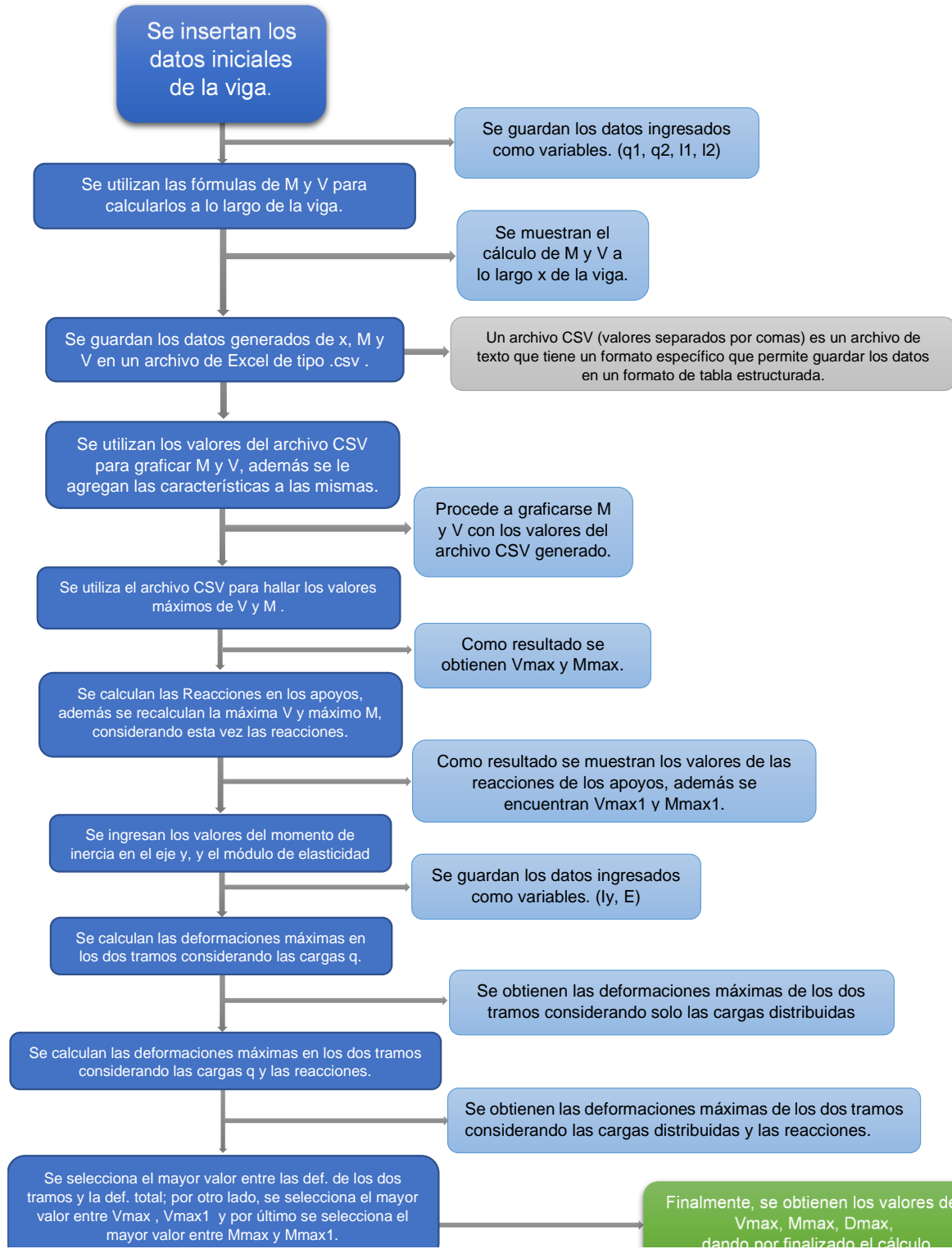


Figura 11 Descripción del funcionamiento del código

ANEXO-4

Algoritmo del código realizado en Python.

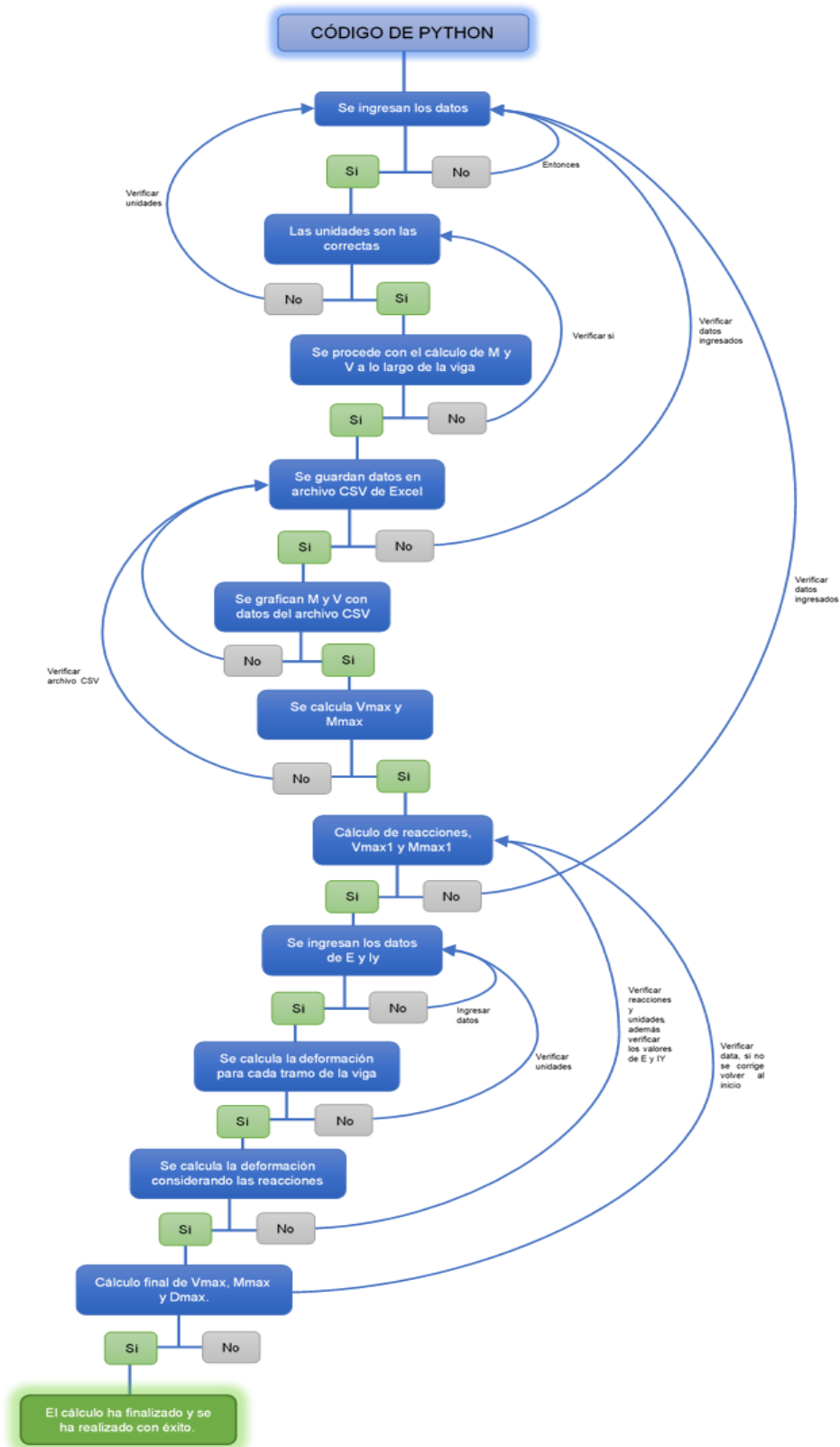


Figura 12 Algoritmo del código

ANEXO-5

Operacionalización de variables

Tabla 6 Operacionalización de variables

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES					
VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ESCALA DE MEDICION
VARIABLE INDEPENDIENTE: Viga continua de 2 vanos	La Norma Técnica de Edificación E.060 de concreto armado, afirma que, la viga es considerada como un elemento que trabaja principalmente a flexión y cortante.	Elemento de concreto armado que recibe las cargas de la losa.		$F_{yA} = \frac{ql}{2} F_{yB} = \frac{ql}{2}$	RAZÓN
			Cortante	$V = F_{yA} - qx = q\left(\frac{l}{2} - x\right)$	
			Momento	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$	
			Deformación	$y = -\frac{5qL^4}{384EI}$	

ANEXO-6

**DOCUMENTOS PARA VALIDAR LOS INSTRUMENTOS DE MEDICIÓN
A TRAVÉS DE JUICIO DE EXPERTOS**

CARTA DE PRESENTACIÓN

Señor(a):

Presente

Asunto: VALIDACIÓN DE INSTRUMENTOS A TRAVÉS DE JUICIO DE EXPERTO.

Nos es muy grato comunicarnos con usted para expresarle nuestros saludos y así mismo, hacer de su conocimiento que, siendo estudiante de la Escuela Profesional de Ingeniería civil de la UCV, en la sede Trujillo, promoción 2023-1, requerimos validar los instrumentos con los cuales recogeremos la información necesaria para poder desarrollar nuestra investigación y con la cual optaremos el grado de Ingeniero.

El título nombre de nuestro proyecto de investigación es: "Análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, Trujillo 2023" y siendo imprescindible contar con la aprobación de ingenieros civiles colegiados, para poder aplicar los instrumentos en mención, hemos considerado conveniente recurrir a usted, ante su connotada experiencia profesional.

El expediente de validación, que le hacemos llegar contiene:

- Carta de presentación.
- Definiciones conceptuales de las variables y dimensiones.
- Matriz de operacionalización de las variables.
- Certificado de validez de contenido de los instrumentos.

Expresándole nuestros sentimientos de respeto y consideración nos despedimos de usted, no sin antes agradecerle por la atención que dispense a la presente.

Atentamente.



Astudillo Morales, Felix Ronald

DNI: 71374713



Flores Adanaqué, Pierina Alexandra

DNI: 72232014

DEFINICIÓN CONCEPTUAL DE LAS VARIABLES Y DIMENSIONES

Variables y operacionalización

Rojas y Gisbert (2017), la herramienta está bajo un enfoque de mejora continua y optimización para una producción o servicio, puesto que esta da un valor al mejorar la eficiencia y eficacia para aumentar la productividad.

Variable Independiente: Viga continua de 2 vanos.

Definición conceptual:

La Norma Técnica de Edificación E.060 de concreto armado, afirma que, la viga es considerada como un elemento que trabaja principalmente a flexión y cortante.

Definición operacional:

Elemento de concreto armado que recibe las cargas de la losa.

MATRIZ DE OPERACIONALIZACIÓN DE LAS VARIABLES

VARIABLE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES	ESCALA DE MEDICIÓN
VIGA CONTINUA DE 2 VANOS	La Norma Técnica de Edificación E.060 de concreto armado, afirma que, la viga es considerada como un elemento que trabaja principalmente a flexión y cortante.	Elemento de concreto armado que recibe las cargas de la losa.	Cortante	$F_{yA} = \frac{ql}{2} F_{yB} = \frac{ql}{2}$ $V = F_{yA} - qx = q\left(\frac{l}{2} - x\right)$	Razón
			Momento	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$	
			Deformación	$y = -\frac{5qL^4}{384EI}$	

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO

Nº	VARIABLES-DIMENSIONES-INDICADORES	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	VARIABLE INDEPENDIENTE: VIGA CONTINUA DE 2 VANOS							
	DIMENSIÓN 1: CORTANTE	Si	No	Si	No	Si	No	
1	$F_{yA} = \frac{ql}{2} \quad F_{yB} = \frac{ql}{2}$							
2	$V = F_{yA} - qx = q\left(\frac{l}{2} - x\right)$							
	DIMENSIÓN 2: MOMENTO	Si	No	Si	No	Si	No	
3	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$							
	DIMENSIÓN 3: DEFORMACIÓN	Si	No	Si	No	Si	No	
4	$y = -\frac{5qL^4}{384EI}$							

Opinión de aplicabilidad: **Aplicable** [] **Aplicable después de corregir** [] **No aplicable** []

Apellidos y nombres del juez validador. :

DNI:

Especialidad del validador:

..... de.....del 2023

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Firma del Experto Informante.

ANEXO-7

Certificados de validez de contenido del instrumento firmados por expertos



CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO

N°	VARIABLES-DIMENSIONES-INDICADORES VARIABLE INDEPENDIENTE: VIGA CONTINUA DE 2 VANOS DIMENSIÓN 1: CORTANTE	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
1	$F_{yA} = \frac{ql}{2}$ $F_{yB} = \frac{ql}{2}$	X		X		X		
2	$V = F_{yA} - qx = q \left(\frac{l}{2} - x \right)$ DIMENSIÓN 2: MOMENTO	X		X		X		
3	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$ DIMENSIÓN 3: DEFORMACIÓN	X		X		X		
4	$y = -\frac{5qL^4}{384EI}$	X		X		X		

Opinión de aplicabilidad: Aplicable [] No aplicable [X] Aplicable después de corregir [] No aplicable [X]

Apellidos y nombres del juez validador.: *Ing. Gaudia Tolera Jumar Xavier* DNI: *70 23 53 80*

Especialidad del validador: *Ingeniería Civil*
21 de *JUNIO* del 2023

¹Pertinencia: El ítem corresponde al concepto teórico formulado.
²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo
³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

[Firma]
 GAUDIA VALERA
 INGENIERA CIVIL
 N° 27043

Firma del Experto Informante.

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO

N°	VARIABLES-DIMENSIONES-INDICADORES VARIABLE INDEPENDIENTE: VIGA CONTINUA DE 2 VANOS DIMENSIÓN 1: CORTANTE $F_{yA} = \frac{ql}{2}$ $F_{yB} = \frac{ql}{2}$	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
1		X		X		X		
2	$V = F_{yA} - qx = q \left(\frac{l}{2} - x \right)$ DIMENSIÓN 2: MOMENTO	X		X		X		
3	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$ DIMENSIÓN 3: DEFORMACIÓN $y = -\frac{5qL^4}{384EI}$	X		X		X		
4		X		X		X		

Opinión de aplicabilidad: **Aplicable [x]** **Aplicable después de corregir []** **No aplicable []**

Apellidos y nombres del juez validador.: *Ingg. Huaman Floriz Pablo Edderson* DNI: *47098111*

Especialidad del validador.: *Ingeniería Civil* *21* de *JUNIO* del 2023

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo



PABLO EDDERSON
HUAMAN FLORES
Ingeniero Civil
CIP N° 288391

Firma del Experto Informante.

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO

N°	VARIABLES-DIMENSIONES-INDICADORES VARIABLE INDEPENDIENTE: VIGA CONTINUA DE 2 VANOS DIMENSIÓN 1: CORTANTE $F_{yA} = \frac{ql}{2}$ $F_{yB} = \frac{ql}{2}$	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
1		X		X		X		
2	$V = F_{yA} - qx = q \left(\frac{l}{2} - x \right)$ DIMENSIÓN 2: MOMENTO	X		X		X		
3	$M = F_{yA}x - \frac{qx^2}{2} = \frac{q}{2}(lx - x^2)$ DIMENSIÓN 3: DEFORMACIÓN $y = -\frac{5qL^4}{384EI}$	X		X		X		
4		X		X		X		

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: : *Ings. Germán Guayada Carlos Anderson* DNI: *719.964.45*

Especialidad del validador: *Ingeniería Civil* *21* de *JUNIO* del 2023

¹Pertinencia: El ítem corresponde al concepto técnico formulado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo


CARLOS AGOSTI
INGENIERO CIVIL
REG. CIP 207348

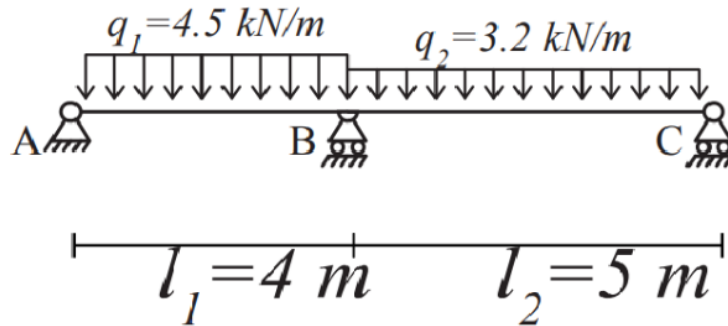
Firma del Experto Informante.

ANEXO-8

Código de programación

- Para el desarrollo del proyecto del Análisis estructural de una viga continua de 2 vanos mediante el programa Python, Trujillo 2023, se tomaron los siguientes datos iniciales mostrados en la Figura 11.

Datos Iniciales



$$q_1 = 4.5 \frac{\text{kN}}{\text{m}}$$
$$q_2 = 3.2 \frac{\text{kN}}{\text{m}}$$
$$l_1 = 4 \text{ m}$$
$$l_2 = 5 \text{ m}$$

Figura 13 Datos iniciales

Nota: Para el primer tramo se toma una carga de $q_1=4.5 \text{ kN/m}$ y una longitud de $l_1= 4 \text{ m}$, para el segundo tramo se toma una carga de $q_2= 3.2 \text{ kN/m}$ y longitud de 5 m .

- En la Figura 12, se muestra la línea 3, la cual muestra el inicio del código, las líneas de código presentadas a continuación, permiten ingresar los valores de la viga como variables para ser utilizadas durante el desarrollo del procedimiento.

```
In [3]: print("ingresa los datos de la viga de 2 vanos: ")

q1 = input("ingresa el valor de q1 en KN/m = ")
print(" q1 es " + q1 + " " + "KN/m ")
q1 = float(q1)

q2 = input("ingresa el valor de q2 en KN/m = ")
print(" q2 es " + q2 + " " + "KN/m ")
q2 = float(q2)

l1 = input("ingresa el valor de l1 en m = ")
print(" l1 es " + l1 + " " + "m ")
l1 = float(l1)

l2 = input("ingresa el valor de l2 en m = ")
print(" l2 es " + l2 + " " + "m ")
l2 = float(l2)

l = l1 + l2

print(f"la longitud total de la viga es = {l1 + l2} m")
```

```
ingresa los datos de la viga de 2 vanos:
ingresa el valor de q1 en KN/m = 4.5
  q1 es 4.5 KN/m
ingresa el valor de q2 en KN/m = 3.2
  q2 es 3.2 KN/m
ingresa el valor de l1 en m = 4
  l1 es 4 m
ingresa el valor de l2 en m = 5
  l2 es 5 m
la longitud total de la viga es = 9.0 m
```

Figura 14 Línea 3 del código de Python

Nota: Se ingresan los parámetros de la viga continua de 2 vanos en el código de Python como variables.

- La Figura 13, muestra la línea 4 del código, la cual nos permitirá calcular el momento y la cortante a lo largo de la viga continua de 2 vanos, utilizando las librerías de Python numpy, sympy y pandas.

```
In [4]: import numpy as np
import sympy as sp
import pandas as pd

def solve_beam(l1, l2, q1, q2):
    l = l1 + l2 # Longitud total
    Mx = sp.symbols('Mx') # creando Mx
    Mx = sp.solve(Mx * l1 / 3 + q1 * l1 ** 3 / 24 + Mx * l2 / 3 + q2 * l2 ** 3 / 24, Mx).args[0]

    # resolviendo las ecuaciones de equilibrio
    Va, Vb1, Vb2, Vc = sp.symbols('Va, Vb1, Vb2, Vc')
    Va, Vb1 = sp.linsolve([Va + Vb1 - q1 * l1, Vb1 * l1 + Mx - (q1 * l1 ** 2) / 2], (Va, Vb1)).args[0]
    Vc, Vb2 = sp.linsolve([Vb2 + Vc - q2 * l2, Vb2 * l2 + Mx - (q2 * l2 ** 2) / 2], (Vc, Vb2)).args[0]

    Vb = Vb1 + Vb2

    x1 = np.arange(0, l1 + 0.1, 0.1) # creando eje x1
    x2 = np.arange(0, l2 + 0.1, 0.1) # creando eje x2

    beam1 = pd.DataFrame({"x": x1})
    beam2 = pd.DataFrame({"x": x2})

    beam1["M"] = Va * beam1.x - (q1 * beam1.x ** 2) / 2
    beam2["M"] = Mx - (q2 * beam2.x ** 2) / 2 + Vb2 * beam2.x

    beam1["V"] = Va - q1 * beam1.x
    beam2["V"] = Vb2 - q2 * beam2.x

    beam2.x = beam2.x + l1

    beam = pd.concat([beam1, beam2])

    return (beam)

header=pd.MultiIndex.from_tuples([("viga 1", "M"), ("viga 1", "V")])
combos=pd.DataFrame(columns=header)
combo1_df = solve_beam(l1, l2, q1, q2)

combos = pd.concat({"viga 1": pd.DataFrame(combo1_df)},axis=1, )

combos=combos.astype("float")
combos.head()
```

```
Out[4]:
```

	viga 1		
	x	M	V
0	0.0	0.000000	6.611111
1	0.1	0.638611	6.161111
2	0.2	1.232222	5.711111
3	0.3	1.780833	5.261111
4	0.4	2.284444	4.811111

Figura 15 Se procede al cálculo de M y V en la viga

Nota: El resultado de esta parte del código, muestra el momento y la cortante a lo largo de la viga continua de 2 vanos, sin embargo solo muestra una parte del mismo.

- En la Figura 14, se muestran las líneas 5 y 6 del código, la línea 5 sirve para crear un archivo CSV, (Comma Separated Values) con la data de la línea 4; la línea 6 sirve para crear un gráfico con esta base de datos generados del archivo de excel CSV.

```
In [5]: combo1_df.to_csv('vigal.csv', header=True, index=False)
```

```
In [6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data=pd.read_csv("vigal.csv")
x=np.array(data.x)
M=np.array(data.M)
V=np.array(data.V)

plt.style.use("seaborn-whitegrid")
fig, ax=plt.subplots(figsize=(20,10))
ax.invert_yaxis()
ax.plot(x,M, label="M", linewidth=3)
plt.title('Diagramas M y V', fontsize=20) #Agregamos el título de La gráfica
plt.legend() #Activamos La leyenda en La gráfica
plt.xlabel('Longitud,m', fontsize=20) #Agregamos La etiqueta X
plt.ylabel('Momento, kN*m ', fontsize=20) #Agregamos La etiqueta Y
ax.fill_between(x, M, alpha=0.3)
plt.tick_params(axis='both', which='major', labelsize=12)

plt.style.use("seaborn-whitegrid")
fig, ax=plt.subplots(figsize=(20,10))
ax.invert_yaxis()
ax.plot(x,V, label="V", linewidth=3)
plt.legend() #Activamos La leyenda en La gráfica
plt.xlabel('Longitud,m', fontsize=20) #Agregamos La etiqueta X
plt.ylabel('Fuerza, kN ', fontsize=20) #Agregamos La etiqueta Y
ax.fill_between(x, V, alpha=0.3)
plt.tick_params(axis='both', which='major', labelsize=12)

plt.show()
```

Figura 16 archivo CSV y se procede a graficar

Nota: Esta parte es muy importante ya que sin el archivo CSV generado no se podría graficar los diagramas de momento y cortante a lo largo de la viga continua de 2 vanos.

- En la Figura 15, se muestran los diagramas M y V a lo largo de la viga continua de 2 vanos obtenidos del paso anterior.

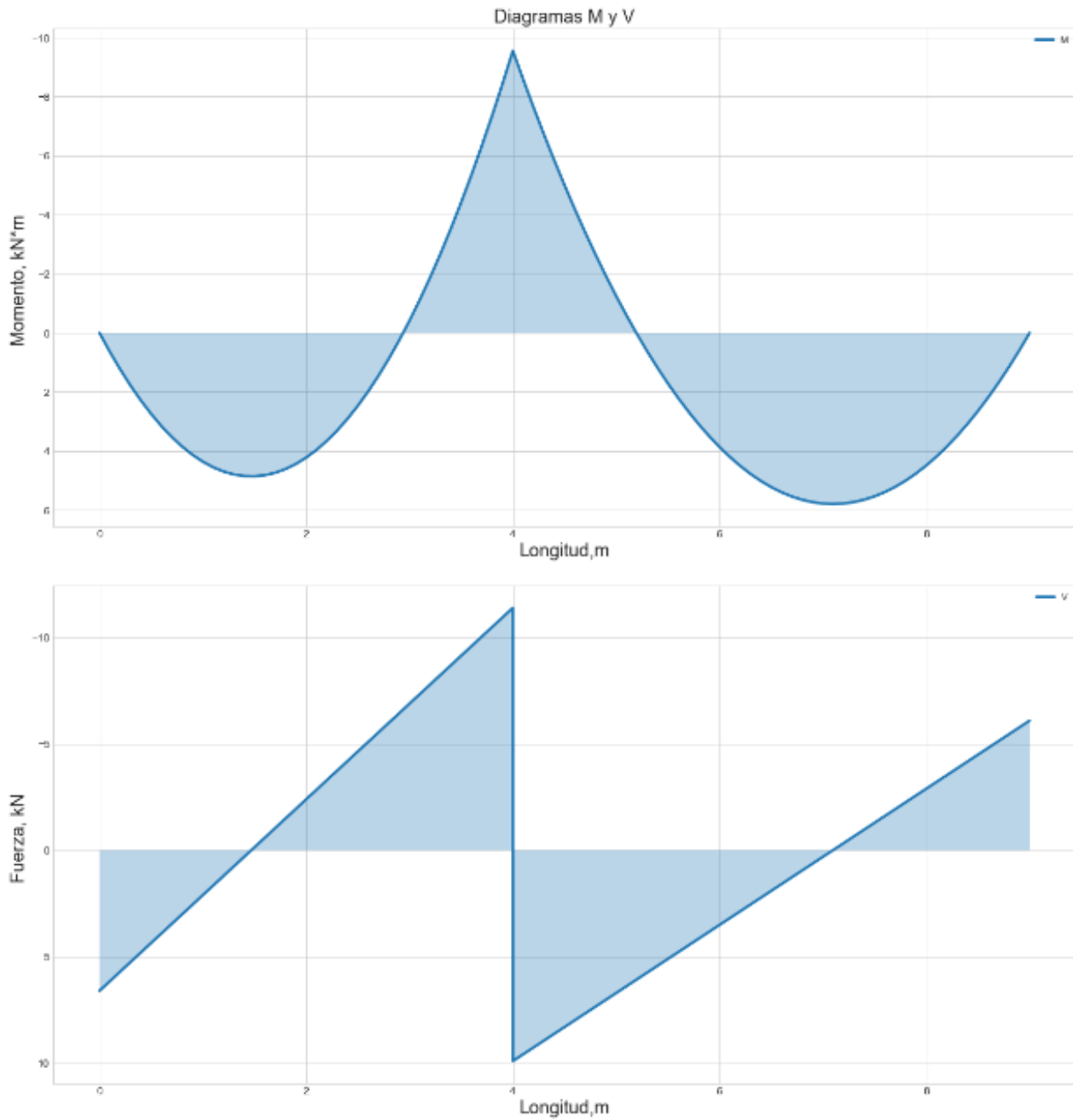


Figura 17 Diagramas de M y V

Nota: En esta parte se generan las graficas empleando la librería de Python Matplotlib, utilizando la data del archivo CSV.

- En la Figura 16, se emplea la librería de pandas para encontrar los máximos valores de V y M dentro del archivo de Excel CSV, generado en la línea 5.

```
In [7]: import pandas as pd

# Lee los datos desde el archivo CSV
datos = pd.read_csv('viga1.csv')

# Calcula el valor máximo en la columna V y M
max_valor = datos['V'].max()
min_valor = datos['V'].min()
mayor = max(abs(max_valor), abs(min_valor))
mayor_round = round(mayor,4)
Vmax = mayor_round

max_valor1 = datos['M'].max()
min_valor1 = datos['M'].min()
mayor1 = max(abs(max_valor1), abs(min_valor1))
mayor_round1 = round(mayor1,4)
Mmax = mayor_round1

# Imprime el valor máximo
print("El valor máximo de V es:", mayor_round, "KN")

print("El valor máximo de M es:", mayor_round1, "KN*m")

El valor máximo de V es: 11.3889 KN
El valor máximo de M es: 9.5556 KN*m
```

Figura 18 Se encuentran los valores de Vmax y Mmax

Nota: Se encuentran los máximos valores de V y M, utilizando el archivo CSV generado por excel, se encontraron los valores de $V_{\max} = 11.3889$ KN y $M_{\max} = 9.5556$ KN*m.

- En la Figura 17, se muestra la línea 8 del código la cual representa el cálculo de las reacciones de la viga, además se encuentran V_{max1} y M_{max1} , los cuales son la cortante máximo y momento máximo considerando las reacciones de la viga.

In [8]:

```
import numpy as np

# Cálculo de Las reacciones en Los apoyos

M1= (((q1+q2)/2)*((l1+l2)/2)**2)/9 #Momento max en B
M2=(q1*l1**2)/11 #Momento max de AB
M3=(q2*l2**2)/11 #Momento max de BC

R1=(q1*l1)/2
R2=(q2*l2)/2
R3=M1/l1
R4=M1/l2

Ra=R1 - R3 #Reacción en A
Ra= round(Ra, 4)

Rc=R2 - R4 #Reacción en C
Rc=round(Rc, 4)

Rb= R1+R2+R3+R4 #por equilibrio se puede hallar Rb
Rb= round(Rb, 4)

# Imprimir las reacciones en Los apoyos
print("Las reacciones en los apoyos son:")
print("Ra =", Ra, "KN")
print("Rb =", Rb, "KN")
print("Rc =", Rc, "KN")

Va = Ra
Vb1= - (R1 + R3)
Vb2= (R2 + R4)
Vc = -(Rc)

print(" ")

print("La Vmax1 entonces será: ")
print("Va es =", Va, "KN")
print("Vb1 es =", Vb1, "KN")
print("Vb2 es =", Vb2, "KN")
print("Vc es =", Vc, "KN")

Vmax1= max(abs(Va),abs(Vb1),abs(Vb2),abs(Vc),)
Vmax1=round(Vmax1,4)
print(" ")
print("Vmax1 es =", Vmax1, "KN")
print(" ")
print("El Mmax1 entonces será: ")
print("M1 es =", M1, "KN*m") #Momento max en B
print("M2 es =", M2, "KN*m") #Momento max tramo de A a B
print("M3 es =", M3, "KN*m") #Momento max tramo de B a C

Mmax1= max(abs(M1),abs(M2),abs(M3))
Mmax1=round(Mmax1, 4)
print(" ")
print("Mmax1 es =", Mmax1, "KN*m")
```

Las reacciones en los apoyos son:

Ra = 6.8344 KN
Rb = 20.8981 KN
Rc = 6.2675 KN

La Vmax1 entonces será:

Va es = 6.8344 KN
Vb1 es = -11.165625 KN
Vb2 es = 9.7325 KN
Vc es = -6.2675 KN

Vmax1 es = 11.1656 KN

El Mmax1 entonces será:

M1 es = 8.662500000000001 KN*m
M2 es = 6.545454545454546 KN*m
M3 es = 7.2727272727272725 KN*m

Mmax1 es = 8.6625 KN*m

Figura 19 Cálculo de reacciones, V_{max1} y M_{max1}

Nota: Se encontraron las reacciones $R_A= 6.8344$ kN, $R_B= 20.8981$ kN, $R_C= 6.2675$ kN; $V_{max1}= 11.1656$ kN y $M_{max1}= 8.6625$ kN*m.

- En la Figura 18, se muestra la línea 9 y 10 del código, en la línea 9 se ingresa la inercia y el módulo de elasticidad, en la línea 10 se calcula la deformación de la viga de 2 vanos para cada tramo.

```
In [9]: print("ingresa los datos de la viga para encontrar la deformación de la viga: ")

Iy_str = input("ingresa el valor de Iy en mm^4 = ") #mm^4 Iy = 2772*10**4
Iy_num= float(eval(Iy_str))
print(" Iy es " + Iy_str + " " + "mm^4 ")

E = input("ingresa el valor de E en MPa = ") #E= 200000
print(" E es " + E + " " + "MPa ")
```

```
ingresa los datos de la viga para encontrar la deformación de la viga:
ingresa el valor de Iy en mm^4 = 2772*10**4
 Iy es 2772*10**4 mm^4
ingresa el valor de E en MPa = 200000
 E es 200000 MPa
```

```
In [10]: # Datos de entrada
q1 = q1 # Carga distribuida del primer tramo en KN/m
q2 = q2 # Carga distribuida del segundo tramo en KN/m
l1 = l1 # Longitud del primer tramo en m
l2 =l2 # Longitud del segundo tramo en m
E = float(E) # Módulo de elasticidad del material de la viga en MPa
Iy = Iy_num # Momento de inercia de la sección transversal de la viga en mm^4

# Cálculo de las deformaciones máximas en el centro de los tramos
Delta1 = (5/384)*((q1*l1**4))/((E*1000))*(Iy*10**(-12)) #resultado en m
Delta1= Delta1*1000 #resultado en mm

print("La deformación máxima del primer tramo es = ", Delta1, " mm")

Delta2 = (5/384)*((q2*l2**4))/((E*1000))*(Iy*10**(-12))
Delta2= Delta2*1000 #resultado en mm

print("La deformación máxima del segundo tramo tramo es = ", Delta2, " mm")
```

```
La deformación máxima del primer tramo es = 2.079e-09 mm
La deformación máxima del segundo tramo tramo es = 3.609375e-09 mm
```

Figura 20 Valores de Iy, E y cálculo de deformación para cada tramo.

Nota: Se ingresa el valor de Iy es $2772 \cdot 10^4 \text{ mm}^4$, y $E = 200000 \text{ MPa}$, posteriormente se calcula la deformación de la viga continua para cada tramo.

- En la Figura 19, se muestra la línea 11 y 12 del código de Python, la línea 11 muestra el cálculo de la deformación de la viga para cada tramo y para el total de la viga, considerando las reacciones, la línea 12 muestra el procedimiento final para mostrar los resultados obtenidos de acuerdo con los objetivos deseados.

```
In [11]:
I = Iy_num # mm^4

# Cálculo de reacciones
RA = Ra
RC = Rc
RB = Rb

print(f"Las reacciones en los apoyos son: RA = {RA:.2f} kN, RB = {RB:.2f} kN y RC = {RC:.2f} kN.")

# Cálculo de la deformación
a1 = l1

a2 = l-11

delta1 = -Delta1 + ((RA * l2**3) / (48 * ((E*1000)*(Iy*10**(-12)))) - (RB * l1**3) / (16 * ((E*1000)*(Iy*10**(-12)))) #resu
delta1a=delta1*1000 #resultado en mm

delta2 = -Delta2 + ((RC * l1**3) / (48 * ((E*1000)*(Iy*10**(-12)))) - (RA * l2**3) / (16 * ((E*1000)*(Iy*10**(-12)))) #resu
delta2a=delta2*1000 #resultado en mm

delta = delta1 * (l1 + l2 - a1) / (l1 + l2) + delta2 * a2 / (l1 + l2) #resultado en m
delta= delta*1000

print(f"La deformación máx. del tramo 1 es {delta1a:.5f} mm.")
print(f"La deformación máx. del tramo 2 es {delta2a:.5f} mm.")

print(f"La deformación total de la viga es {delta:.10f} mm.")

Las reacciones en los apoyos son: RA = 6.83 kN, RB = 20.90 kN y RC = 6.27 kN.
La deformación máx. del tramo 1 es -11.86769 mm.
La deformación máx. del tramo 2 es -8.12358 mm.
La deformación total de la viga es -11.1062608429 mm.
```

```
In [12]:
mayor_defo= max(abs(delta1a), abs(delta2a), abs(delta))
mayor_defo=-float(mayor_defo)

Vmax= float(Vmax)
Vmax=max(abs(Vmax), abs(Vmax1))

Mmax=float(Mmax)
Mmax=max(abs(Mmax),abs(Mmax1))

print("Se ha realizado el cálculo con éxito y se han obtenido los siguientes resultados para la viga continua de 2 vanos: ")
print(" ")
print(f"El máximo valor de V es = {Vmax: .5f} KN")
print(f"El máximo valor de M es = {Mmax: .5f} KN*m ")
print(f"El máximo valor de la deformación de la viga es = {mayor_defo: .10f} mm")

Se ha realizado el cálculo con éxito y se han obtenido los siguientes resultados para la viga continua de 2 vanos:

El máximo valor de V es = 11.38890 KN
El máximo valor de M es = 9.55560 KN*m
El máximo valor de la deformación de la viga es = -11.8676938779 mm
```

Figura 21 Cálculo de deformación y procedimiento final

Nota: Se obtiene la deformación máxima para cada tramo considerando las reacciones, posteriormente se obtienen los valores requeridos por los objetivos, el valor máximo de $V_{max} = 11.38890$ kN, $M_{max} = 9.55560$ kN*m y $u_{max} = -11.867694$ mm

ANEXO-9

- En el paso 01 se muestra simulación de la viga utilizando la plataforma OpenGL, buscaremos la opción **prueba**, donde colocaremos los datos de la viga.

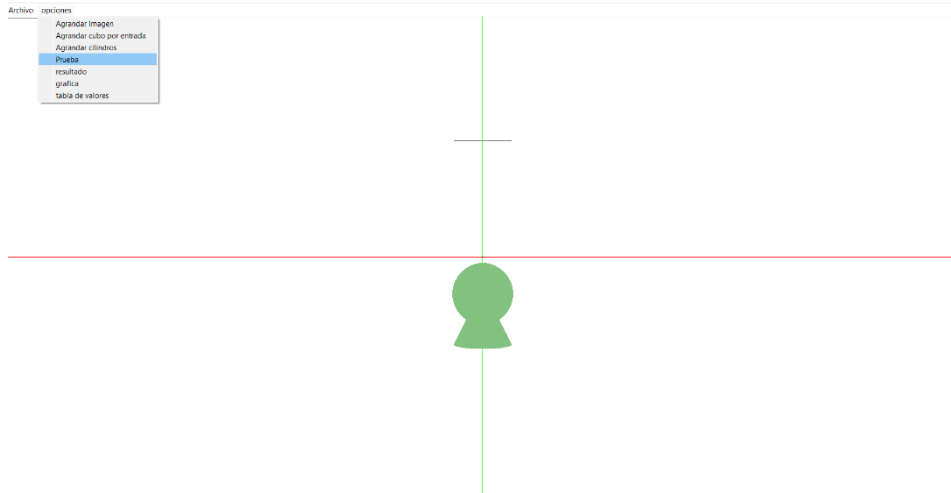


Figura 22 Inicio de prueba en OpenGL

- En el paso 02, se abre una ventana donde ingresaremos los datos de la viga continua de 2 vanos, y de esta manera proceder con la simulación en base al código de Python realizado, se utiliza la plataforma OpenGL para realizar la simulación.

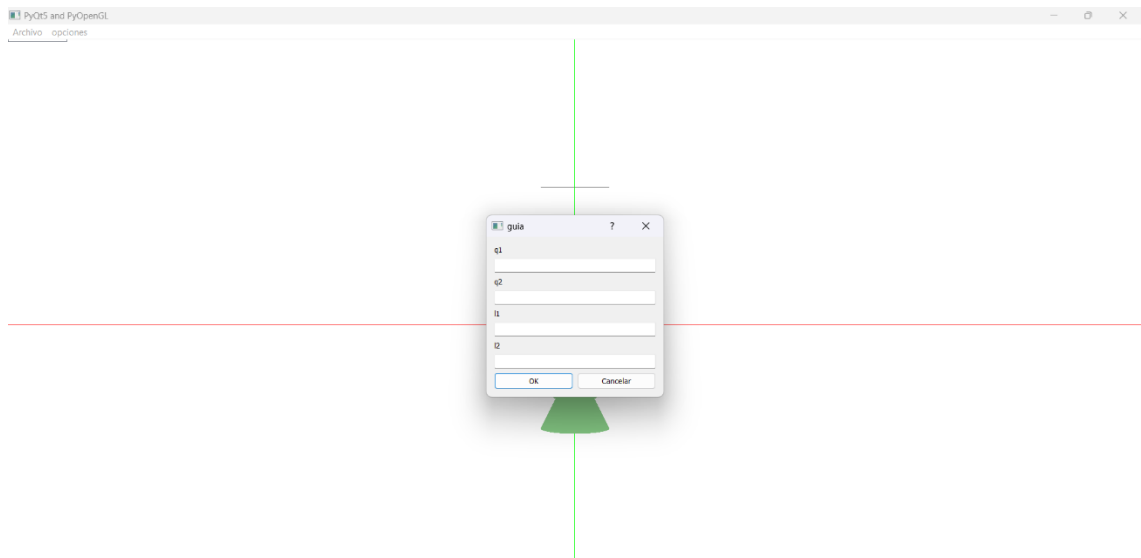


Figura 23 Ingreso de datos

- En el paso 03, se procede a aceptar los datos ingresados, verificando con cautela si los datos ingresados son los correctos.

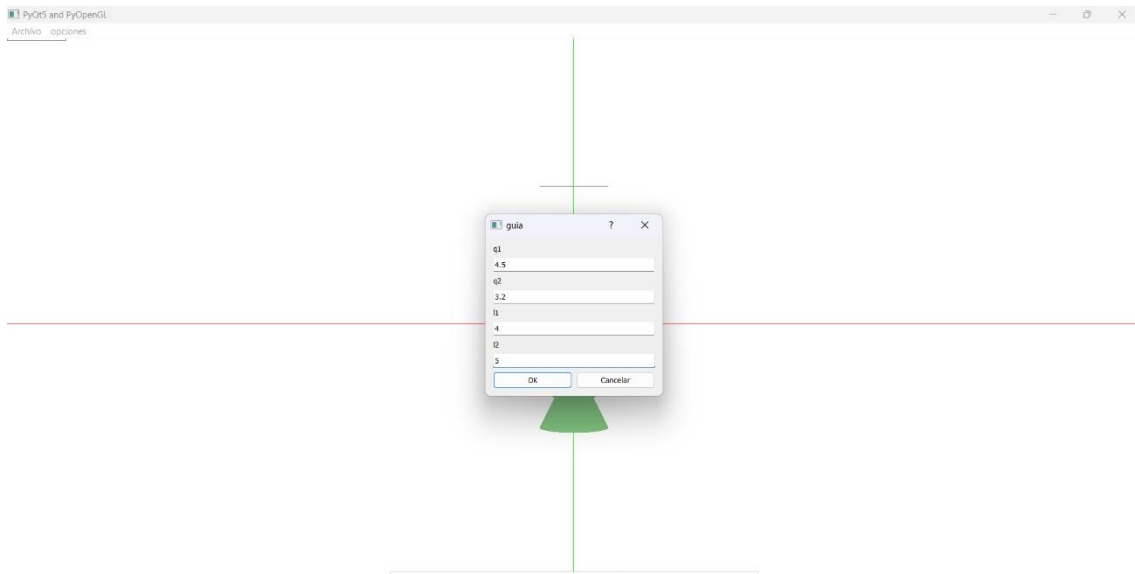


Figura 24 Confirmación de datos ingresados

- En el paso 04, después de confirmar el paso 03, se puede observar la viga generada y aparece un mensaje de confirmación donde se aprecia la distancia de la viga continua de 2 vanos ingresada.

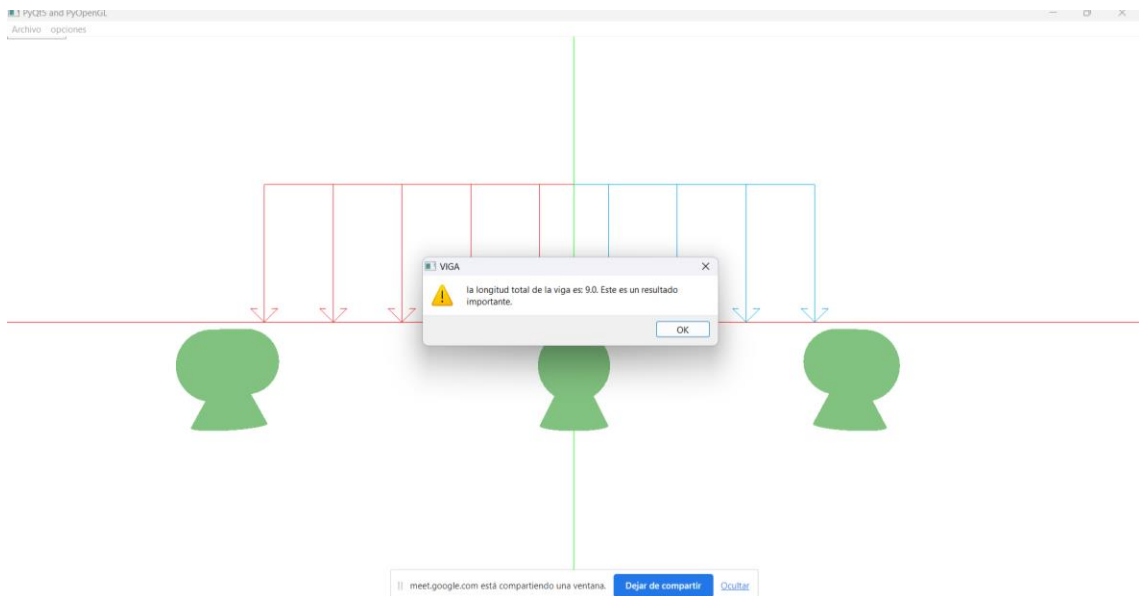


Figura 25 Confirmación de longitud de viga ingresada

- En el paso 05, se procede a obtener los datos, en la opción "tabla de valores".

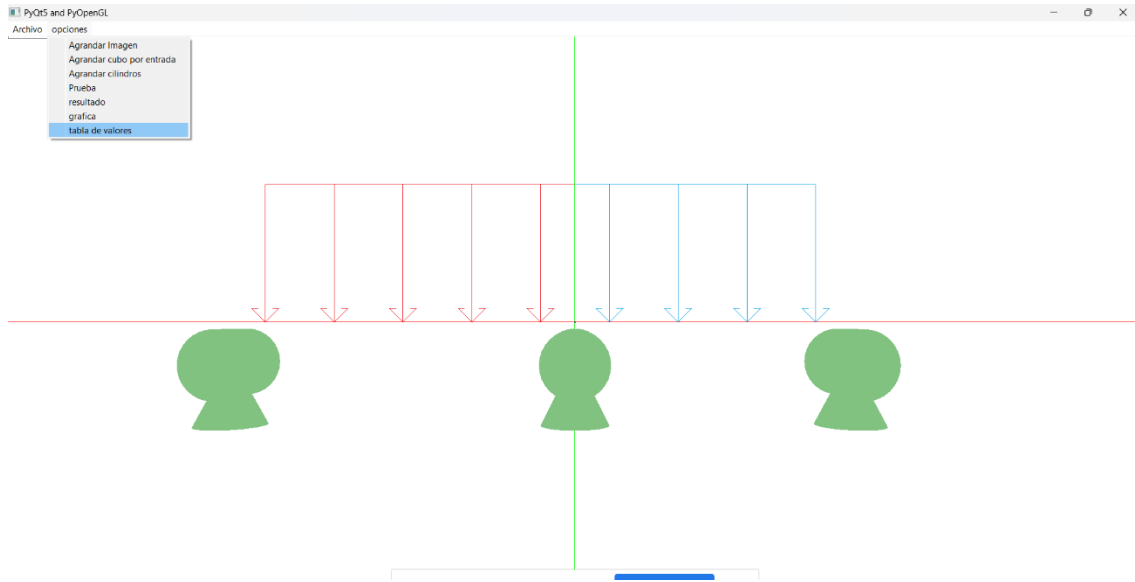


Figura 26 Tabla de Valoreses

- En el paso 06, se procede a verificar los datos ingresados de la viga, para proceder a generar la tabla de valores.

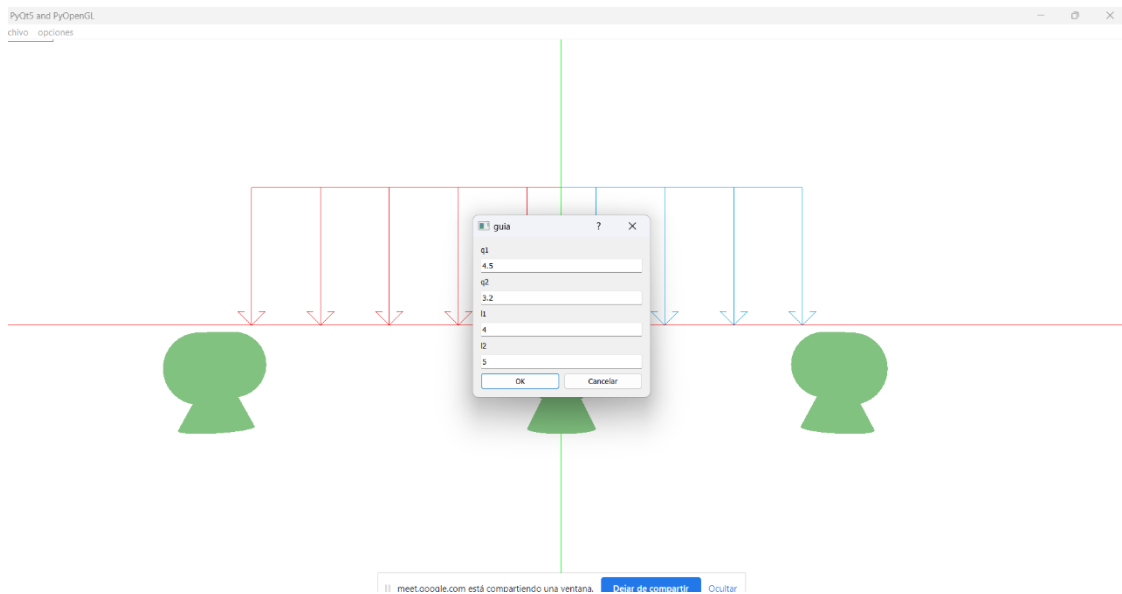
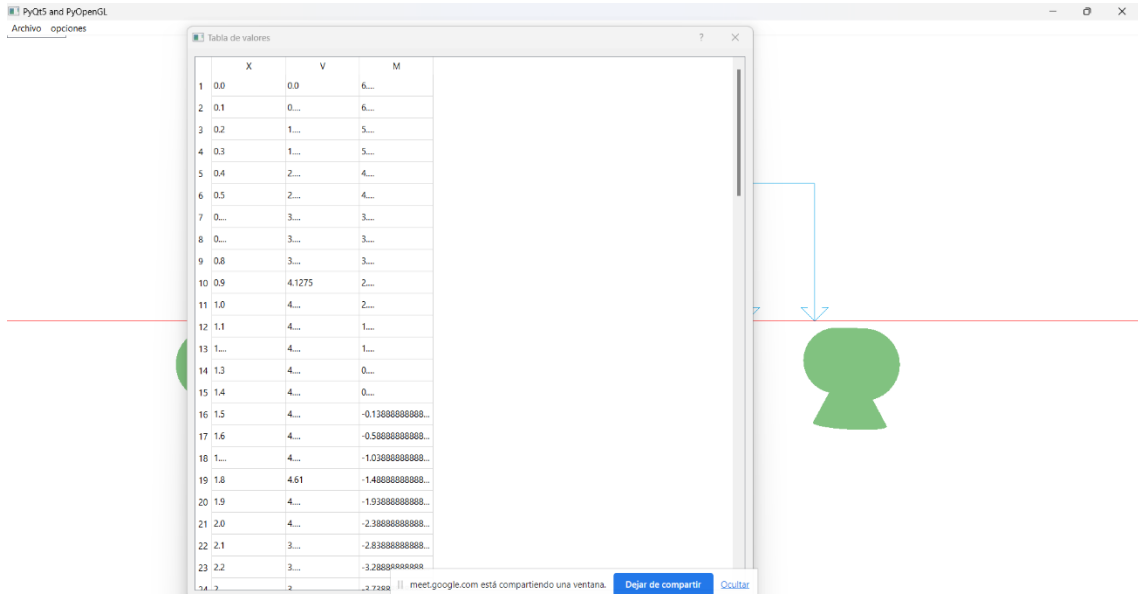


Figura 27 Confirmación de datos para obtener la tabla de valores.

- En el paso 07, se puede observar el cuadro con la tabla de valores generados de M y V.



	X	V	M
1	0.0	0.0	6...
2	0.1	0...	6...
3	0.2	1...	5...
4	0.3	1...	5...
5	0.4	2...	4...
6	0.5	2...	4...
7	0...	3...	3...
8	0...	3...	3...
9	0.8	3...	3...
10	0.9	4.1275	2...
11	1.0	4...	2...
12	1.1	4...	1...
13	1...	4...	1...
14	1.3	4...	0...
15	1.4	4...	0...
16	1.5	4...	-0.5888888888...
17	1.6	4...	-0.5888888888...
18	1...	4...	-1.0388888888...
19	1.8	4.61	-1.4888888888...
20	1.9	4...	-1.9388888888...
21	2.0	4...	-2.3888888888...
22	2.1	3...	-2.8388888888...
23	2.2	3...	-3.2888888888...
24	2...	2...	-3.7388...

Figura 28 Tabla de valores generada

- En el paso 08, se debe proceder a buscar en la carpeta de instalación del aplicativo para revisar el archivo generado el cual contiene los valores en Excel con el formato .csv (Comma Separated Values).

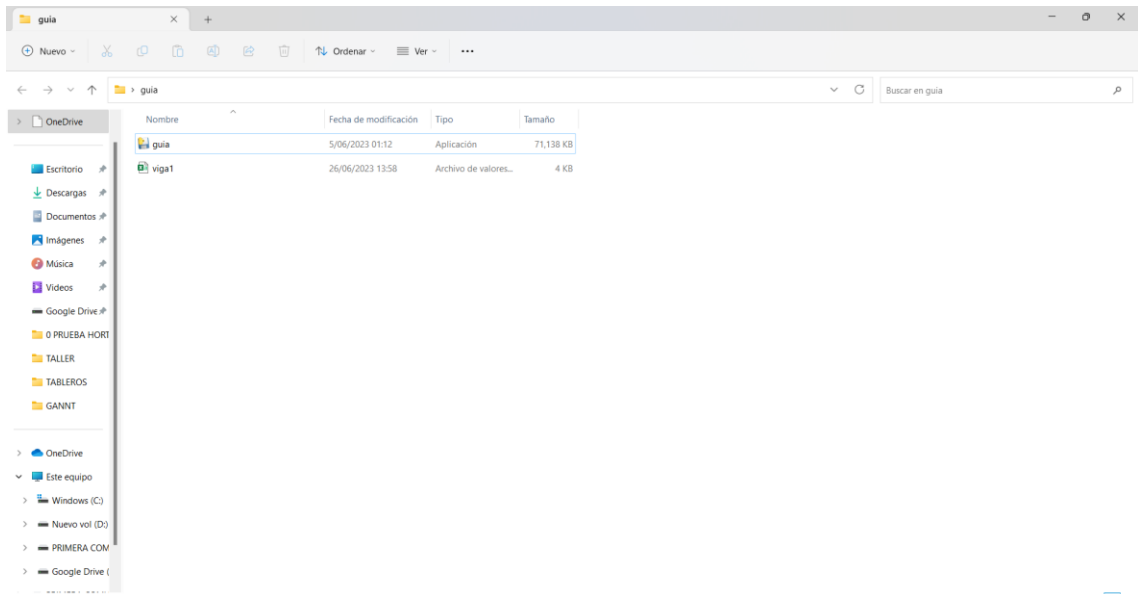


Figura 29 Tabla de valores en archivo CSV.

- En el paso 09, se observa el archivo en Excel con los valores de M y V, generados a lo largo de la viga continua de 2 vanos por el código de programación realizado en Python.

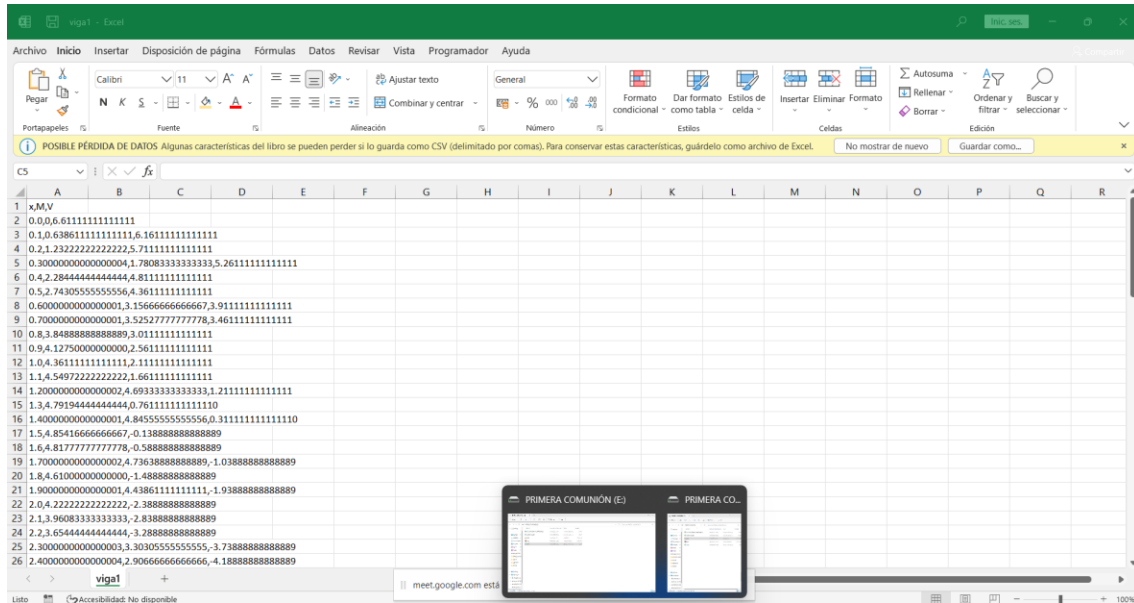


Figura 30 Archivo CSV

- En el paso 10, luego se procede a buscar en el tablero de opciones, la opción de “**gráfica**”, para obtener los datos requeridos.

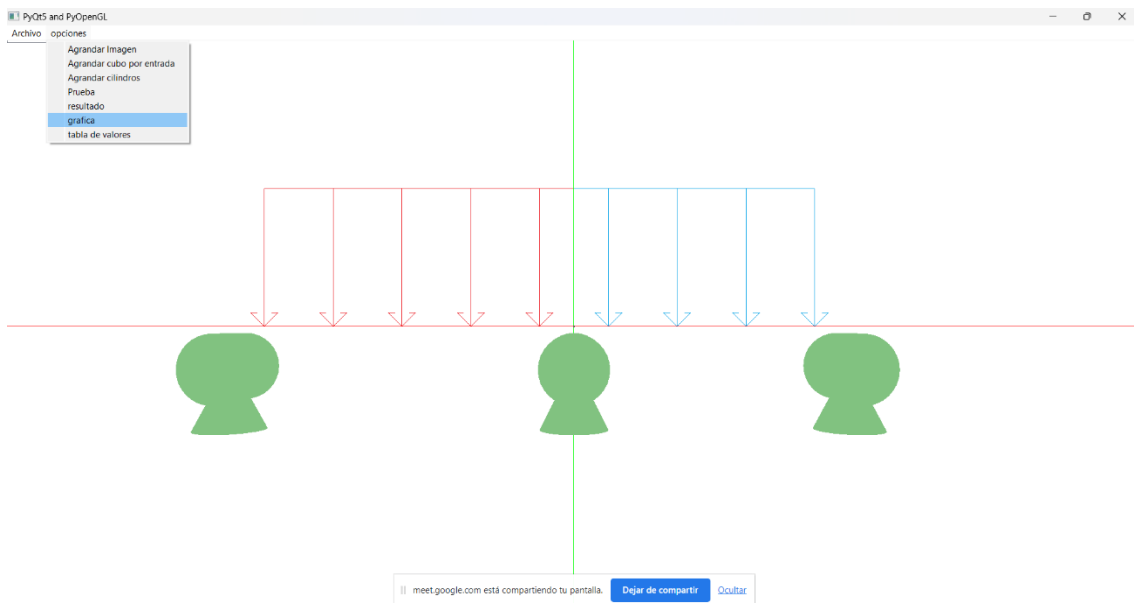


Figura 31 ubicación de la opción "gráfica"

- Paso 11: En el paso 11, se pueden observar las gráficas obtenidas después de haber procesado los datos ingresados de la viga continua de 2 vanos mediante el código de programación realizado en Python y simulado a través de la plataforma OpenGL.

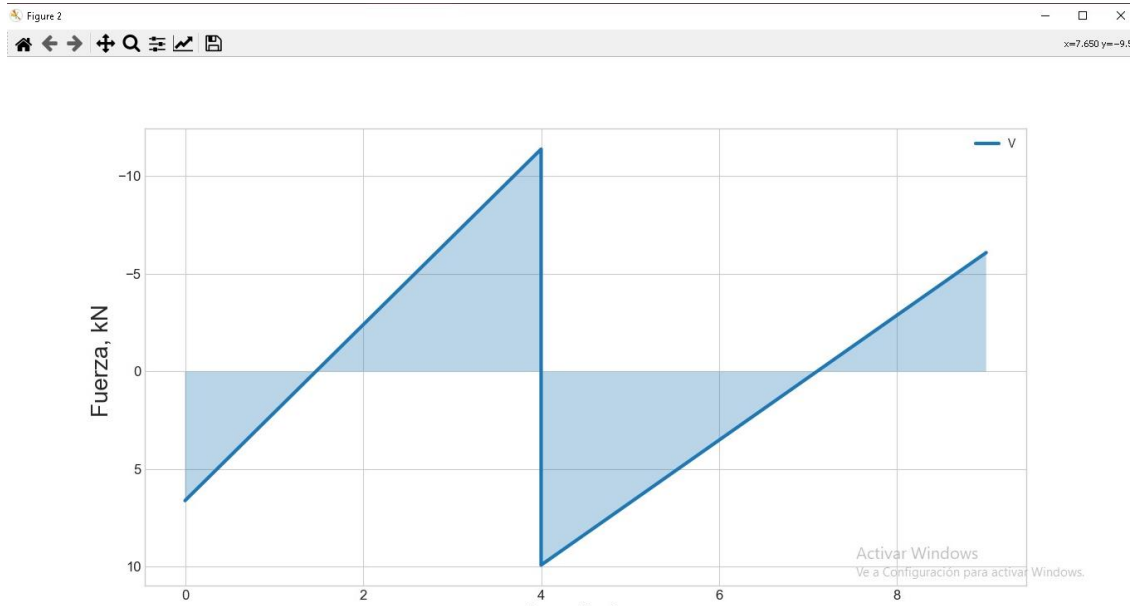


Figura 32 Gráfica de Cortante a lo largo de la viga.

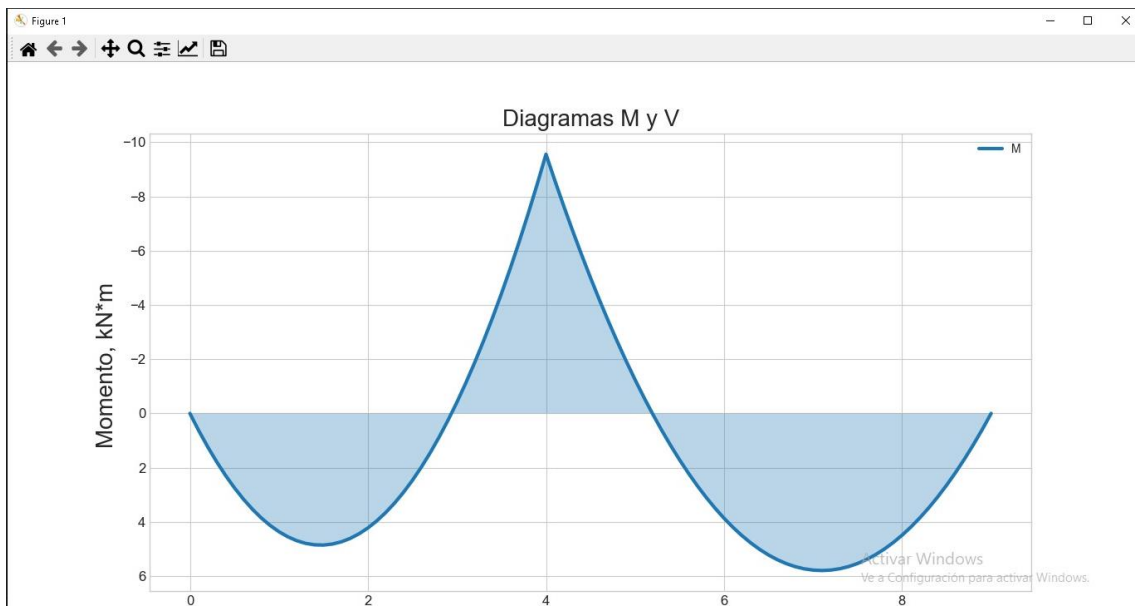


Figura 33 : Gráfica de Momento a lo largo de la viga.



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA CIVIL**

Declaratoria de Autenticidad del Asesor

Yo, EDUARDO MANUEL NORIEGA VIDAL, docente de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA CIVIL de la UNIVERSIDAD CÉSAR VALLEJO SAC - TRUJILLO, asesor de Tesis titulada: "Análisis estructural de una viga continua de 2 vanos mediante librerías de lenguaje de programación en Python, Trujillo 2023", cuyos autores son FLORES ADANAQUE PIERINA ALEXANDRA, ASTUDILLO MORALES FELIX RONALD, constato que la investigación tiene un índice de similitud de 12.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

TRUJILLO, 24 de Julio del 2023

Apellidos y Nombres del Asesor:	Firma
EDUARDO MANUEL NORIEGA VIDAL DNI: 43236142 ORCID: 0000-0001-7674-7125	Firmado electrónicamente por: ENORIEGAVI el 24- 07-2023 10:33:32

Código documento Trilce: TRI - 0613733