



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA CIVIL Y ARQUITECTURA
ESCUELA PROFESIONAL DE CIENCIAS FÍSICO MATEMÁTICAS



**DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL
PARA LA SIMULACIÓN NUMÉRICA DEL ASENTAMIENTO
LINEAL EN SUELOS USANDO EL MÉTODO DE LOS
ELEMENTOS FINITOS**

TESIS

PRESENTADA POR:

DENIS RUDY LUPACA QUISPE

PARA OPTAR EL TÍTULO PROFESIONAL DE:

**LICENCIADO EN CIENCIAS FÍSICO MATEMÁTICAS:
FÍSICA**

PUNO - PERÚ

2024



DENIS RUDY LUPACA QUISPE

DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL PARA LA SIMULACIÓN NUMÉRICA DEL ASENTAMIENTO LINEAL EN

My Files

My Files

Universidad Nacional del Altiplano

Detalles del documento

Identificador de la entrega
trn:oid::8254:418842057

162 Páginas

Fecha de entrega
26 dic 2024, 11:40 a.m. GMT-5

31,712 Palabras

Fecha de descarga
26 dic 2024, 12:03 p.m. GMT-5

185,776 Caracteres

Nombre de archivo
DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL PARA LA SIMULACIÓN NUMÉRICA DEL AS....pdf

Tamaño de archivo
10.4 MB





9% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto citado
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 12 palabras)

Fuentes principales

- 7% Fuentes de Internet
- 2% Publicaciones
- 6% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

- Caracteres reemplazados**
177 caracteres sospechosos en N.º de páginas
Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.



Adelaida Otazu Conza
LIC. CIENCIAS FÍSICO - MATEMÁTICAS
Doc. MATEMÁTICA APLICADA





DEDICATORIA

A mis queridos padres, Belisario Lupaca y Gloria Quispe, por impulsarme a seguir adelante y a realizar mis sueños y aspiraciones.

Denis Lupaca



AGRADECIMIENTOS

Agradezco a Dios por darme salud y fuerzas, lo que me permite lograr un objetivo más.

A mi asesor de tesis, Dr. Ricardo Romero Loaiza, por su apoyo incondicional, su guía experta y su paciencia a lo largo de todo el proceso de realización de esta tesis.

Denis Lupaca



ÍNDICE GENERAL

| | Pág. |
|--|------------|
| DEDICATORIA | |
| AGRADECIMIENTOS | |
| ÍNDICE GENERAL | |
| ÍNDICE DE TABLAS | |
| ÍNDICE DE FIGURAS | |
| ÍNDICE DE ANEXOS | |
| ACRÓNIMOS | |
| RESUMEN. | .16 |
| ABSTRACT | .17 |
| CAPÍTULO I | |
| INTRODUCCIÓN | |
| 1.1. PLANTEAMIENTO DEL PROBLEMA | 18 |
| 1.2. FORMULACION DEL PROBLEMA | 19 |
| 1.2.1. Problema general | 19 |
| 1.2.2. Problemas específicos | 19 |
| 1.3. HIPÓTESIS DEL TRABAJO | 20 |
| 1.4. JUSTIFICACIÓN | 20 |
| 1.5. OBJETIVOS | 21 |
| 1.5.1. Objetivo general | 21 |
| 1.5.2. Objetivos específicos | 21 |
| CAPÍTULO II | |
| REVISIÓN DE LITERATURA | |
| 2.1. MATRICES | 22 |



| | | |
|-------------|--|-----------|
| 2.1.1. | Matriz fila | 22 |
| 2.1.2. | Matriz columna | 23 |
| 2.1.3. | Igualdad de matrices | 23 |
| 2.1.4. | Multiplicación de un escalar por una matriz | 23 |
| 2.1.5. | Suma de matrices | 23 |
| 2.1.6. | Producto de matrices | 24 |
| 2.1.7. | Matriz nula | 24 |
| 2.1.8. | Matriz unitaria | 24 |
| 2.1.9. | Matriz traspuesta | 25 |
| 2.1.10. | Matriz inversa | 25 |
| 2.1.11. | Cálculo de la inversa de una matriz mediante el método de Gauss-Jorda | 25 |
| 2.2. | CLASIFICACIÓN DE LOS SUELOS | 26 |
| 2.2.1. | Suelos Granulares | 27 |
| 2.2.2. | Suelos Arcillosos | 27 |
| 2.3. | CONDUCTIVIDAD HIDRÁULICA | 28 |
| 2.3.1. | Ley de Darcy | 29 |
| 2.4. | ASENTAMIENTO DE SUELOS | 29 |
| 2.4.1. | Condición drenada y no drenada | 30 |
| 2.4.2. | Prueba de consolidación de laboratorio unidimensional | 30 |
| 2.4.3. | Parámetros de los materiales para el cálculo del asentamiento elástico | 31 |
| 2.5. | TASA DE CONSOLIDACIÓN | 32 |
| 2.6. | MÉTODO DE LOS ELEMENTOS FINITOS | 33 |
| 2.6.1. | Implementación computacional del MEF | 33 |
| 2.6.2. | Importancia de la simulación numérica | 34 |
| 2.6.3. | Formulación Fuerte | 35 |
| 2.6.4. | Formulación Débil | 35 |



| | | |
|--------------|---|-----------|
| 2.6.5. | Formulación Matricial | 36 |
| 2.7. | VISUAL STUDIO Y C# COMO HERRAMIENTAS DE SIMULACIÓN . | 37 |
| 2.8. | LENGUAJE DE PROGRAMACIÓN | 37 |
| 2.9. | LENGUAJE DE PROGRAMACIÓN C# | 38 |
| 2.9.1. | Algoritmo | 38 |
| 2.9.2. | Estructura de un Programa en C# | 39 |
| 2.9.3. | Procedimientos en C# | 39 |
| 2.9.4. | Declaración de Variables y Tipos de Datos | 39 |
| 2.9.5. | Tipos de Datos Comunes en C# | 39 |
| 2.9.6. | Operadores Aritméticos: | 40 |
| 2.9.7. | Operadores de Comparación | 40 |
| 2.9.8. | Operadores Lógicos | 40 |
| 2.9.9. | Matrices | 41 |
| 2.9.9.1. | Matriz Unidimensional | 41 |
| 2.9.9.2. | Matriz Bidimensional | 41 |
| 2.10. | WINDOWS FORMS | 41 |
| 2.10.1. | Etiqueta (Label) | 42 |
| 2.10.2. | Cuadro de Texto (TextBox) | 42 |
| 2.10.3. | Botón (Button) | 43 |
| 2.10.4. | Lista (ListBox) | 43 |
| 2.10.5. | Cuadro Combinado (ComboBox) | 43 |
| 2.10.6. | Tablas (DataGridView) | 43 |
| 2.10.7. | Imágenes (PictureBox) | 44 |
| 2.10.8. | Chart | 44 |
| 2.10.9. | Control de Pestañas (TabControl) | 44 |
| 2.11. | ANTECEDENTES | 44 |



| | |
|---|----|
| 2.11.1. Antecedentes: Tesis | 44 |
| 2.11.2. Antecedentes: Artículos | 47 |

CAPÍTULO III

MATERIALES Y MÉTODOS

| | |
|--|-----------|
| 3.1. MÉTODO DE INVESTIGACIÓN | 49 |
| 3.2. TIPO DE INVESTIGACIÓN | 49 |
| 3.3. NIVEL DE INVESTIGACIÓN | 50 |
| 3.4. DISEÑO DE INVESTIGACIÓN | 50 |
| 3.5. POBLACIÓN | 51 |
| 3.6. MUESTREO | 51 |
| 3.7. MUESTRAS | 51 |
| 3.8. DESCRIPCIÓN DETALLADA DE MÉTODOS POR OBJETIVOS ESPECÍFICOS | 51 |
| 3.8.1. Definir las ecuaciones matriciales del método de los elementos finitos para determinar el asentamiento de suelos | 52 |
| 3.8.1.1. Matriz de rigidez | 52 |
| 3.8.1.2. Matriz de hidraulica | 52 |
| 3.8.1.3. Matriz que relaciona los desplazamientos verticales y presiones | 53 |
| 3.8.1.4. Vectores de fuerzas | 53 |
| 3.8.1.5. Solución del sistema matricial por equivalencia incremental . | 53 |
| 3.8.2. Desarrollo de la herramienta computacional | 54 |
| 3.8.2.1. Construcción de la interfaz gráfica en Visual Studio C# . . . | 54 |
| 3.8.2.2. Programación del algebra lineal en C# | 54 |
| 3.8.2.3. Programación de la formulación matricial del método de los elementos finitos en C# | 56 |
| 3.8.3. Validación de los valores y asentamientos y presiones | 60 |



| | |
|--|----|
| 3.8.4. Simulación con 5 elementos y 10 elementos | 61 |
| 3.8.5. Simulación con dos, cuatro y seis años | 61 |
| 3.8.6. Simulación con suelos granulares y arcillosos | 62 |
| 3.8.6.1. Parametros de suelos | 62 |
| 3.8.6.2. Discretización del suelos | 63 |
| 3.8.6.3. Definición de condición de fronteras | 63 |
| 3.8.6.4. Asignación de cargas a lo largo del tiempo | 63 |
| 3.8.6.5. Ejecución de la heramienta computacional | 64 |
| 3.8.7. Comparación del resultado de ambos modelos analizados | 64 |

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

| | |
|---|-----------|
| 4.1. DESARROLLO DE LA HERRAMIENTA COMPUTACIONAL PARA DETERMINAR LOS ASENTAMIENTO Y PRESIONES DE SUELOS | 65 |
| 4.1.1. Definición de propiedades y elementos del suelos | 66 |
| 4.1.2. Asignación de condición de fronteras y cargas | 67 |
| 4.2. VALIDACIÓN DE LOS VALORES DE ASENTAMIENTOS Y PRESIONES CALCULADOS POR LA HERRAMIENTA INFORMÁTICA | 67 |
| 4.2.1. Asentamientos | 67 |
| 4.2.2. Presiones | 69 |
| 4.3. SIMULACIÓN CON 10 Y 20 ELEMENTOS | 71 |
| 4.4. SIMULACIÓN EN DOS, CUATRO Y SEIS AÑOS | 72 |
| 4.5. SIMULACIÓN NUMÉRICA ENTRE UN SUELO GRANULAR Y ARCILLOSO | 74 |
| 4.6. DISCUSIÓN | 76 |



| | |
|--------------------------------------|------------|
| V. CONCLUSIONES | .79 |
| VI. RECOMENDACIONES | .81 |
| VII. REFERENCIAS | .82 |
| ANEXOS | .85 |

ÁREA: Física Aplicada

TEMA: Método de los Elementos Finitos

FECHA DE SUSTENTACIÓN: 27 de Diciembre de 2024



ÍNDICE DE TABLAS

| | Pág. |
|--|-------------|
| Tabla 1 Valores típicos de conductividad hidráulica para suelos saturados | 28 |
| Tabla 2 Parámetros elásticos para varios tipos de suelos | 32 |
| Tabla 3 Parámetros del suelo para comparación entre los dos métodos | 61 |
| Tabla 4 Parametros de los suelos modelados | 62 |
| Tabla 5 Discretización de los suelos modelados | 63 |



ÍNDICE DE FIGURAS

| | Pág. |
|--|-------------|
| Figura 1 Tamaño de las partículas de suelo | 26 |
| Figura 2 Modelo cilindro-resorte | 30 |
| Figura 3 Gráfica de deformación en función del tiempo | 31 |
| Figura 4 Formulario de C# | 42 |
| Figura 5 Diseño del formulario principal | 54 |
| Figura 6 Suelos analizados | 62 |
| Figura 7 Iniciando Suelo MEF en su primera versión | 66 |
| Figura 8 Creación de tipo de suelos y elementos | 66 |
| Figura 9 Comparación de asentamiento entre el método diferencia central y la herramienta informática | 68 |
| Figura 10 Comparación de presiones entre el método diferencia central y la herramienta informática | 70 |
| Figura 11 Comparación entre un análisis con 10 elementos y 20 elementos | 71 |
| Figura 12 Comparación entre un análisis con dos, cuatro y seis años | 73 |
| Figura 13 Comparación entre un un suelos granular y arcillo | 75 |



ÍNDICE DE ANEXOS

| | Pág. |
|-----------------|--|
| Anexo 1 | Definición de las matrices 85 |
| Anexo 2 | Método de la diferencia central 92 |
| Anexo 3 | Calculo de cantidad de carga 93 |
| Anexo 4 | Estructura de un Programa en C# 96 |
| Anexo 5 | Implementación de la herramienta computacional en suelos granulares y arcillosos 98 |
| Anexo 6 | Datos de asentamiento para la comparación entre el método de diferencia central y la herramienta desarrollada 104 |
| Anexo 7 | Datos de presiones para la comparación entre el método de diferencia central y la herramienta desarrollada 105 |
| Anexo 8 | Datos de asentamiento en dos, cuatro, seis años 107 |
| Anexo 9 | Datos de asentamiento 10 elementos y 20 elementos 108 |
| Anexo 10 | Comparación del asentamiento es suelos granulares y arcillosos . . . 109 |
| Anexo 11 | Código fuente 110 |
| Anexo 12 | Declaración jurada de autenticidad de tesis 161 |
| Anexo 13 | Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional 162 |



ACRÓNIMOS

| | |
|---------|---|
| P_i : | Presión en el nodo i |
| v_i : | Desplazamiento vertical en el nodo i |
| C#: | Lenguaje de programación orientado a objetos. |
| H: | Matriz hidráulica |
| JSON: | JavaScript Object Notation (Notación de Objetos JavaScript) |
| K: | Matriz de rigidez |
| MEF: | Método de los elementos finitos |
| Q: | Matriz que conectar la matriz K con H |



RESUMEN

El presente proyecto tuvo como objetivo desarrollar una herramienta computacional utilizando el lenguaje de programación C# que permitió realizar la simulación numérica de la consolidación lineal de suelos en condiciones drenadas y no drenadas, empleando el método de los elementos finitos. Esto se realiza aplicado el método de los elementos finitos para elementos lineales con 3 nodos, en un rango elástico que simuló el comportamiento del suelo. Para la construcción de la herramienta computacional, se utilizó el IDE Visual Studio C#, ya que ofreció un entorno completo para la creación de herramientas computacionales avanzadas. Esta investigación fue del tipo aplicada, ya que empleó conocimientos teóricos, con un enfoque mixto y un nivel descriptivo y exploratorio, pues buscó describir el comportamiento de distintos tipos de suelos granulares y arcillosos. La investigación integró diversos cursos del área de la física y las matemáticas, destacando la física computacional. La herramienta computacional fue validada al comparar los resultados obtenidos con el método de diferencia central llegando obtener los mismos valores al final del tiempo de análisis, se obtuvo que la cantidad elementos no varió los resultados, finalmente realizando la comparación de los asentamientos en suelos mostrando que, a lo largo de cinco años, los llegando a la conclusión que los suelos granulares tuvieron desplazamientos menores, mientras que los arcillosos fueron significativamente mayores. La diferencia acumulada alcanzó 284.5 mm. Se utilizaron condiciones saturadas y valores bajos para el módulo de elasticidad y conductividad hidráulica, ya que los suelos granulares se asientan rápidamente, a diferencia de los arcillosos, que lo hacen más lentamente.

Palabras clave: Asentamiento del suelo, Herramienta computacional, Método de los elementos finitos, Presión del suelo, Simulación numérica.



ABSTRACT

The objective of this project was to develop a computational tool using the C# programming language that allowed the numerical simulation of linear soil consolidation under drained and undrained conditions, employing the finite element method. This was done by applying the finite element method for linear elements with 3 nodes in an elastic range, simulating the soil's behavior. For the development of the computational tool, the Visual Studio C# IDE was used, as it provided a comprehensive environment for creating advanced computational tools. This research was applied in nature, as it employed theoretical knowledge with a mixed approach and a descriptive and exploratory level, aiming to describe the behavior of different types of granular and clayey soils. The research integrated various courses from the fields of physics and mathematics, particularly highlighting computational physics. The computational tool was validated by comparing the results obtained with the central difference method, yielding the same values at the end of the analysis period. It was found that the number of elements did not affect the results. Finally, the comparison of settlements in soils showed that, over a period of five years, granular soils exhibited smaller displacements, while clayey soils showed significantly greater displacements. The accumulated difference reached 284.5 mm. Saturated conditions and low values for the elasticity modulus and hydraulic conductivity were used, as granular soils settle quickly, unlike clayey soils, which settle more slowly.

Keywords: Computational tool, Finite element method, Numerical simulation, Soil pressure, Soil settlement.



CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, las simulaciones numéricas se utilizan con mayor frecuencia, destacándose programas como MATLAB, Octave y Python, que están orientados a la ciencia y las ingenierías. Estas herramientas cuentan con librerías especializadas en matemática y física, siendo MATLAB el más popular para el modelado de problemas físicos en diversas áreas científicas. Por otro lado, lenguajes como C++ y C# son ideales para el desarrollo de aplicaciones rápidas, robustas y versátiles, especialmente en entornos de producción. Estos permiten crear interfaces gráficas modernas y ofrecen un mayor control sobre el hardware y los sistemas operativos. Además, C# es más accesible económicamente, ya que es gratuito con .NET, mientras que MATLAB tiene un costo elevado.

Actualmente, existen laboratorios alojados en páginas web que permiten entender el comportamiento de problemas físicos relacionados con el movimiento y la mecánica cuántica. Sin embargo, ninguno de estos se enfoca en el análisis del comportamiento de diferentes tipos de suelos sometidos a cargas. Por ello, se plantea como solución el uso de simulaciones numéricas basadas en el método de elementos finitos.

Con el transcurso de los años, la ciencia y la tecnología han avanzado de tal manera que las simulaciones se aproximan cada vez más a los valores reales. Por esta razón, es necesario contar con un conocimiento tanto analítico como computacional del método de elementos finitos, especialmente para determinar los asentamientos de los suelos, ya que es una de las teorías más precisas y exactas. Por lo tanto, el método de elementos finitos



es el más adecuado para realizar estas simulaciones numéricas.

El rubro de la construcción va en aumento. Sin embargo, también se sabe que las viviendas construidas sin licencia están creciendo, por lo que es importante conocer el comportamiento de cada elemento de la vivienda. Una parte fundamental es el suelo sobre el cual se construye, ya que este se encarga de soportar toda la estructura. Cuando un suelo es sometido a fuerzas excesivas, puede generar hundimientos y presiones. Estas fuerzas, si se aplican de manera incorrecta, podrían llevar a fallas en la vivienda, ya sea por corte, deslizamiento, flexión o aplastamiento. Por ello, es crucial entender el comportamiento del suelo en diferentes condiciones.

Por todo lo antes mencionado se genera la necesidad de la creación de una herramienta que cumpla con las necesidades actuales de simulaciones. Por lo que se presenta el desarrollo de una herramienta computacional con la capacidad de poder simular el asentamiento de suelos usando MEF.

1.2. FORMULACION DEL PROBLEMA

1.2.1. Problema general

¿Es posible desarrollar una herramienta computacional en lenguaje de programación C# que permita realizar simulaciones precisas del asentamiento lineal de suelos, utilizando el método de los elementos finitos?

1.2.2. Problemas específicos

- ¿Es posible definir las ecuaciones matriciales del método de los elementos finitos con elementos lineales para la simulación numérica del comportamiento de suelos?
- ¿Es posible desarrollar una herramienta computacional eficiente que resuelva las ecuaciones matriciales para obtener los asentamientos en suelo?



- ¿Cómo implementar la herramienta computacional para reflejar con precisión las diferencias entre suelos granulares y arcillosos en los resultados de la simulación?

1.3. HIPÓTESIS DEL TRABAJO

Es posible el desarrollo de una herramienta computacional con el lenguaje de programación C# que permita realizar la simulación numérica del asentamiento lineal de suelos, al utilizar el método de los elementos finitos.

1.4. JUSTIFICACIÓN

Las simulaciones numéricas son fundamentales en la resolución de problemas complejos en ciencia e ingeniería, con Matlab siendo este el más utilizados gracias a su capacidad para manejar álgebra lineal. Sin embargo, lenguajes de acceso libre como Python, C++ y C# ofrecen ventajas como mayor flexibilidad y aplicabilidad a la hora de desarrollar un producto final. En particular, C# destaca por su arquitectura orientada a objetos, lo que facilita el desarrollo de herramientas computacionales avanzadas y su integración con diversas plataformas.

A pesar de contar con diversas herramientas para simular fenómenos físicos, todavía existen pocas aplicaciones dedicadas a modelar el comportamiento de los suelos sometidos a cargas a lo largo del tiempo. Esto abre una gran oportunidad para diseñar una herramienta que aplique el método de los elementos finitos (MEF) y así modelar de forma más precisa el asentamiento de los suelos, aplicando conceptos de física computacional.

Gracias a los avances tecnológicos; hoy en día, las computadoras actuales, con mayores capacidades de procesamiento, facilitan la simulación de modelos complejos con más elementos, lo que mejora la fiabilidad y eficiencia de los resultados.

Por lo que la creación de una herramienta informática en C# para determinar el



asentamiento en suelos ofrece una solución innovadora. Esto contribuiría al avance en el campo de la ingeniería geotécnica a partir de la simulación física, siendo una herramienta útil para mejorar el análisis de asentamientos y presiones en suelos, optimizando el diseño y la seguridad en la construcción de infraestructuras.

Con este trabajo se busca innovar el uso de nuevas tecnologías para comprender el comportamiento de los suelos, cuando estos son sometidos a cargas producidas por edificaciones, vehículos pesados. Al emplear el método de los elementos finitos aplicado a la consolidación de suelos nos permite aproximar el comportamiento más preciso a diferencias del uso de ábacos, por lo que se podrá obtener resultados más confiables.

1.5. OBJETIVOS

1.5.1. Objetivo general

Desarrollar una herramienta computacional con el lenguaje de programación C# que permita realizar la simulación numérica de la asentamiento lineal de suelos, usando el método de los elementos finitos.

1.5.2. Objetivos específicos

- Definir las ecuaciones matriciales del método de los elementos finitos con elementos lineales para la simulación numérica del comportamiento de suelos.
- Desarrollo de la herramienta computacional para determinar los asentamientos y presiones de suelos mediante la simulación numérica.
- Implementación de la herramienta computacional desarrollada en suelos granulares y arcillosos para obtener asentamientos.

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. MATRICES

Matriz o arreglo rectangular de número reales dispuesto en filas y columnas las cuales encerrados entre corchetes, es decir presenta la siguiente forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3j} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{ij} & \cdots & a_{im} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nm} \end{bmatrix}$$

Donde los distintos números organizados en filas y columnas se conocen como elementos o componentes de la matriz.

2.1.1. Matriz fila

Las matrices de orden $1 \times n$ se les conoce como matriz fila, es decir tiene la forma siguiente: $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \end{bmatrix}$, también se le conoce con el nombre de vector fila.

2.1.2. Matriz columna

Las matrices de orden $n \times 1$ se les conoce como matriz columna: es decir tiene de la forma: $A = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$. También se les conoce con el nombre de vector columna.

2.1.3. Igualdad de matrices

Sean dos matrices A y B serán iguales si tienen el mismo orden y todos sus elementos correspondientes son iguales. Esto es: $A = B \leftrightarrow a_{ij} = b_{ij} \forall i,j$.

2.1.4. Multiplicación de un escalar por una matriz

Si una matriz A de orden $n \times m$ y un escalar λ , al multiplicar $\lambda \cdot A$ se define como:

$$\lambda \cdot A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} = \begin{bmatrix} \lambda \cdot a_{11} & \lambda \cdot a_{12} & \cdots & \lambda \cdot a_{1m} \\ \lambda \cdot a_{21} & \lambda \cdot a_{22} & \cdots & \lambda \cdot a_{2m} \\ \vdots & \vdots & & \vdots \\ \lambda \cdot a_{n1} & \lambda \cdot a_{n2} & \cdots & \lambda \cdot a_{nm} \end{bmatrix}$$

Cada elemento de la matriz A es multiplicada por λ , el producto $\lambda \cdot A$ es por consiguiente otra matriz de orden $n \times m$.

2.1.5. Suma de matrices

Dado dos matrices de orden $n \times m$, $A = [a_{ij}]_{n \times m}$ y $B = [a_{ij}]_{n \times m}$, la suma de las matrices A y B vendra a ser otra matriz $C = [c_{ij}]_{n \times m}$ de orden $n \times m$, en cada elemento que tenga la matriz C es la suma cada elemento correspondientes considerando la fila y la columna de A y B , es decir:



$$c_{ij} = a_{ij} + b_{ij}, \forall i, j$$

Entonces:

$$A + B = [a_{ij}]_{n \times m} + [b_{ij}]_{n \times m} = [a_{ij} + b_{ij}]_{n \times m} = [c_{ij}]_{n \times m} = C$$

$$A + B = [c_{ij}]_{n \times m} = C$$

2.1.6. Producto de matrices

Sea una matriz $A = [a_{ij}]_{n \times m}$ que tiene orden $n \times m$ y otra matriz $B = [b_{ij}]_{m \times r}$ de orden $m \times r$ el producto de ambas matrices A por B vendria a ser otra matriz $C = [c_{ij}]_{n \times r}$ la cual tendra orden $n \times r$, donde c_{ij} Es el resultado de multiplicar la fila i con la columna j , es decir:

$$c_{ij} = \sum_{k=1}^m a_{ik} \cdot b_{kj}$$

Donde: $i = 1, 2, \dots, n, j = 1, 2, \dots, r$

2.1.7. Matriz nula

Una matriz en la que todos sus elementos son cero se conoce como matriz nula y se representa de la siguiente manera:

$$0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

2.1.8. Matriz unitaria

Para la matriz unitaria la diagonal principal debe estar definida por 1 es decir $a_{11} = a_{22} = a_{33} = \dots = a_{nn} = 1$, entonces denotaremos como:

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

2.1.9. Matriz traspuesta

La traspuesta de una matriz A se obtiene al intercambiar sus columnas por filas, de modo que la fila i de la matriz original se convierte en la columna i de la matriz traspuesta.

La traspuesta de la matriz A se denota como A^T , entonces:

$$\text{Si } A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \rightarrow A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{bmatrix}$$

2.1.10. Matriz inversa

Una matriz cuadrada A se considera invertible si existe otra matriz cuadrada B tal que al multiplicar A por B , y B por A , se obtiene la matriz identidad I . En este caso, la matriz B es la inversa de A y se denota como $B = A^{-1}$.

2.1.11. Cálculo de la inversa de una matriz mediante el método de Gauss-Jordan

Este procedimiento consiste en formar una matriz de tamaño $n \times 2n$ uniendo la matriz A con la matriz identidad I , es decir:

$$A : I$$

Mediante la aplicación operaciones elementales sobre las filas de la matriz construida

transformando en la forma:

$$I : B$$

Donde $B = A^{-1}$ es la matriz inversa.

2.2. CLASIFICACIÓN DE LOS SUELOS

Los suelos pueden clasificarse en granulares y cohesivos según sus propiedades físicas, químicas y mecánicas, la cual es esencial para comprender su comportamiento en aplicaciones de ingeniería civil, geotecnia y física. También los suelos con propiedades similares se agrupan en función de sus características mecánicas y su comportamiento para la ingeniería. Para ello, existen dos sistemas de clasificación comunes: el AASHTO y el Sistema Unificado de Clasificación de Suelos. Ambos utilizan la distribución granulométrica y la plasticidad de los suelos, y son utilizados principalmente en Estados Unidos, donde el sistema AASHTO es preferido por los departamentos de carreteras, mientras que los ingenieros geotécnicos optan por el Sistema Unificado (Das, 2015). En la Figura 1 se presenta los distintos tipos de suelos que son analizados en el laboratorio de mecánica de suelos.

Figura 1

Tamaño de las partículas de suelo



Nota: Tamaño de las partículas de suelos desde un suelos granular hasta un suelo arcilloso. Fuente: [CFAPE](#).



2.2.1. Suelos Granulares

Definición y Composición Los suelos granulares están compuestos predominantemente por partículas grandes como gravas y arenas. Estas partículas no poseen cohesión y su estabilidad depende principalmente de la fricción interna entre los granos. Las gravas tienen partículas mayores a 2 mm, mientras que las arenas poseen partículas entre 0.075 mm y 2 mm de diámetro, Con las siguientes propiedades mecánicas.

- Alta permeabilidad: Los suelos granulares permiten el paso fácil del agua debido a su estructura porosa, lo que los hace ideales para drenajes.
- Buena capacidad de carga: Debido a la fricción interna, soportan altas cargas sin deformarse significativamente.
- Comportamiento drenado: Responden rápidamente a las cargas, mostrando poca consolidación debido a su baja compresibilidad.

2.2.2. Suelos Arcillosos

Definición y Composición Los suelos arcillosos están compuestos principalmente por partículas finas, menores a 0.002 mm, y minerales como caolinita, illita y montmorillonita. Estas partículas tienen una alta superficie específica y propiedades de plasticidad debido a su estructura laminar. Con las siguientes propiedades mecánicas.

- Baja permeabilidad: Los suelos arcillosos limitan el flujo de agua, lo que los hace propensos a problemas de drenaje.
- Alta compresibilidad: Se deforman considerablemente bajo carga debido a su capacidad de retener agua.
- Comportamiento dependiente del tiempo: Experimentan asentamiento por consolidación en respuesta a cargas aplicadas.
- Plasticidad: Su capacidad de moldearse bajo fuerzas hace que presenten diferentes

estados (líquido, plástico y semisólido) dependiendo de su contenido de humedad.

2.3. CONDUCTIVIDAD HIDRÁULICA

La conductividad hidráulica de los suelos, que determina la velocidad de descarga del agua a través del suelo, está influenciada por varios factores, como la viscosidad del fluido, la distribución de tamaños de poros y granos, la relación de vacíos, la rugosidad de las partículas minerales y el grado de saturación del suelo. En suelos arcillosos, la estructura tiene un impacto significativo en esta propiedad, además de factores como la concentración iónica y el grosor de las capas de agua adheridas a las partículas de arcilla.

El valor de la conductividad hidráulica, representado por k , varía entre diferentes tipos de suelos, siendo mayor en suelos saturados. En suelos no saturados, la conductividad es más baja, pero aumenta rápidamente con el grado de saturación. Este estudio es crucial en la mecánica de suelos, ya que permite estimar la filtración subterránea, analizar problemas de bombeo de agua en construcción subterránea y evaluar la estabilidad de presas y estructuras de retención. Los valores típicos para suelos saturados se presenta en la Tabla 1.

Tabla 1

Valores típicos de conductividad hidráulica para suelos saturados

| Tipo de suelo | k (cm/s) |
|----------------------|------------------------------|
| Grava limpia | 100-1 |
| Arena gruesa | 1-0.01 |
| Arena fina | 0.01-0.001 |
| Arcilla limosa | 0.001-0.00001 |
| Arcilla | <0.000001 |

Nota: En donde los suelos granulares tienen valores más altos y los suelos arcillosos tienen valores menores. Fuente: Braja Das.

2.3.1. Ley de Darcy

Henri Darcy formuló una ecuación empírica para calcular la velocidad de descarga del agua en suelos saturados, basada en sus observaciones del flujo a través de arenas limpias (Das, 2015).

$$v = k \cdot i$$

Donde:

v: Velocidad de descarga, es la cantidad de agua que fluye por unidad de tiempo a través de un área transversal de suelo perpendicular al flujo.

k: Conductividad hidráulica, también conocida como coeficiente de permeabilidad.

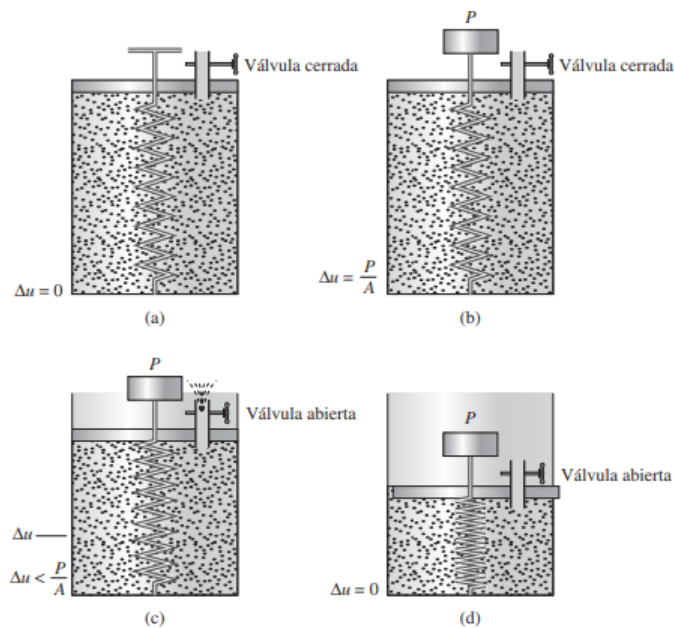
i: Gradiente hidráulico

2.4. ASENTAMIENTO DE SUELOS

El asentamiento de los suelos es un fenómeno geotécnico que describe el hundimiento gradual de una superficie debido a cargas aplicadas a lo largo del tiempo, ya sea por estructuras o procesos naturales, y puede clasificarse en asentamiento inmediato, primario (consolidación) y secundario (flujo plástico). En suelos arenosos altamente permeables, el drenaje del agua intersticial ocurre rápidamente, lo que provoca un asentamiento simultáneo por elasticidad y consolidación. Por otro lado, en arcillas de baja permeabilidad, el asentamiento elástico es inmediato, pero la consolidación se desarrolla lentamente debido a la gradual disipación del exceso de presión de poros, prolongándose en el tiempo (Das, 2015). La magnitud y velocidad del asentamiento dependen de propiedades como la capacidad de carga, compresibilidad y permeabilidad del suelo, lo que explica el comportamiento diferenciado de cada tipo de material.

Figura 2

Modelo cilindro-resorte



Nota: Δu es el exceso de presión hidrostática. Fuente: Braja Das.

2.4.1. Condición drenada y no drenada

El suelo experimenta cambios de volumen al absorber o filtrar agua, y la tasa de cambio está directamente relacionada con su permeabilidad. Existen dos condiciones principales: drenada y no drenada. En suelos como grava y arena, la carga se aplica lentamente en comparación con la tasa de drenaje, mientras que en suelos como limos y arcillas, la carga se aplica rápidamente en relación con dicha

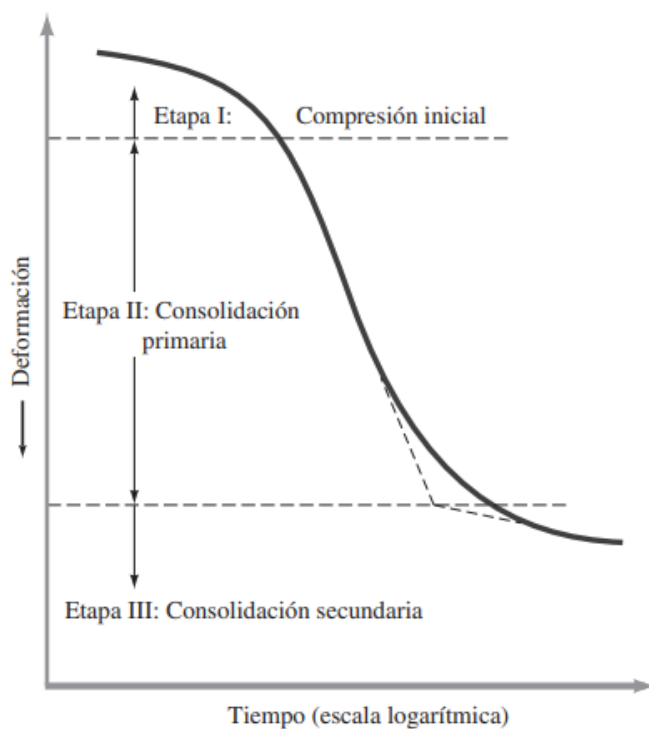
2.4.2. Prueba de consolidación de laboratorio unidimensional

El procedimiento de prueba de consolidación unidimensional, propuesto por Terzaghi, utiliza un consolidómetro. Para esta prueba es necesario una muestra de suelo de aproximadamente 63.5 mm de diámetro y 25.4 mm de espesor se coloca en un anillo metálico entre dos piedras porosas y se somete a cargas incrementales mientras se mantiene bajo agua. La compresión se mide con un micrómetro calibrado, y cada carga

se aplica durante 24 horas antes de duplicarla. Al finalizar, se determina el peso seco de la muestra. Los resultados se representan en una gráfica que relaciona la deformación de la muestra con el tiempo para cada incremento de carga (Das, 2015). En la Figura 3 se presenta la gráfica de deformación en función del tiempo durante una consolidación para un incremento particular de la carga.

Figura 3

Gráfica de deformación en función del tiempo



Nota: El asentamiento se presenta en tres etapas, el tiempo está en escala logarítmica. Fuente: Braja Das.

2.4.3. Parámetros de los materiales para el cálculo del asentamiento elástico

Se presenta la fórmula para determinar el asentamiento elástico de las cimentaciones, la cual incluye parámetros elásticos como E_s y m_s . En caso de no contar con los resultados de las pruebas de laboratorio para estos parámetros, es necesario hacer algunas suposiciones razonables. La siguiente tabla proporciona un rango aproximado de los valores de los parámetros elásticos para diferentes tipos de suelo como se muestra en

la Tabla 2.

Tabla 2

Parámetros elásticos para varios tipos de suelos

| Tipo de suelo | Módulo de elasticidad, Es [MN/m²] | Coefficiente de Poisson, ν |
|----------------------|---|--|
| Arena suelta | 10-25 | 0.20-0.40 |
| Arena semi-densa | 15-30 | 0.25-0.40 |
| Arena densa | 35-55 | 0.30-0.45 |
| Arena limosa | 10-20 | 0.20-0.40 |
| Arena y grava | 70-170 | 0.15-0.35 |
| Arcilla blanda | 4-20 | |
| Arcilla media | 20-40 | 0.20-0.50 |
| Arcilla dura | 40-100 | |

Nota: Fuente: Braja Das.

2.5. TASA DE CONSOLIDACIÓN

El asentamiento total debido a la consolidación primaria, provocado por un aumento en la presión sobre una capa de suelo. Terzaghi fue el primero en proponer una teoría para calcular la tasa de consolidación unidimensional en suelos arcillosos saturados.

Las deducciones matemáticas de esta teoría se basan en los siguientes supuestos:

- El sistema arcilla-agua es homogéneo.
- La saturación es completa.
- La compresibilidad del agua es despreciable.
- La compresibilidad de los granos del suelo es despreciable, aunque estos se reorganizan.
- El flujo de agua ocurre en una sola dirección. Es decir, en la dirección de la compresión.
- La ley de Darcy es aplicable en la siguiente formula.

$$\frac{\partial u}{\partial t} = C_v \cdot \frac{\partial^2 u}{\partial z^2}$$

Donde: C_v coeficiente de consolidación $C_v = k/(\gamma_w m_v)$

2.6. MÉTODO DE LOS ELEMENTOS FINITOS

El Método de los Elementos Finitos (MEF) es una técnica numérica ampliamente utilizada en ingeniería y física destinado a resolver problemas complejos. Su principal fortaleza radica en la capacidad de dividir un dominio complicado en elementos más pequeños y manejables; lo que facilitara los cálculos y análisis (Zienkiewicz et al., 2005). En el campo de la geotecnia, el MEF es muy fundamental para estudiar cómo se distribuyen las tensiones, deformaciones y desplazamientos en los suelos cuando están sometidos a diferentes tipos de cargas y condiciones de confinamiento. Este enfoque es especialmente útil para abordar problemas no lineales y sistemas heterogéneos, como los que surgen en el análisis de asentamientos, donde las características del terreno pueden variar significativamente. (Potts & Zdravković, 2001).

2.6.1. Implementación computacional del MEF

El desarrollo de software aplicando el Método de los Elementos Finitos (MEF) requiere elegir que el lenguaje de programación sea tanto eficiente como flexible. Con este entendimiento, C# se posiciona como una excelente opción, ya que permite manejar estructuras de datos complejas, ofreciendo bibliotecas matemáticas avanzadas y se beneficia de un entorno de desarrollo integrado (IDE) robusto como Visual Studio (Troelsen & Japikse, 2017). La implementación del MEF incluye varios pasos clave:

- Discretización del dominio: Se divide el área de análisis en pequeños elementos finitos, que pueden ser triángulos, cuadriláteros u otros polígonos.
- Formulación de matrices: Se construye la matriz de rigidez global ensamblando las matrices de cada elemento.
- Aplicación de condiciones de frontera: Se integran las restricciones y cargas al



modelo matemático.

- Resolución del sistema de ecuaciones: Se emplean métodos numéricos, como la descomposición LU o iteraciones conjugadas, para resolver las ecuaciones resultantes (Cook et al., 2002).
- Propiedades de los Suelos en el MEF
- En cuanto a las propiedades del suelo dentro del MEF, Se observa que juegan un papel fundamental en la precisión de las simulaciones. Parámetros como el módulo de elasticidad, el coeficiente de Poisson y la permeabilidad son esenciales para modelar de manera realista el comportamiento del material; Generalmente, estos valores se obtienen a partir de ensayos de laboratorio, siendo uno de los más utilizados el ensayo de consolidación unidimensional (ASTM D2435/D2435M-11, 2011).

2.6.2. Importancia de la simulación numérica

Se conoce que la simulación numérica se ha convertido en una herramienta esencial para reducir tanto los costos como los riesgos que suelen acompañar a la experimentación física. Este aplicado en el ámbito de la geotecnia, esta técnica permite predecir cómo se comportará el suelo bajo diferentes condiciones sin necesidad de realizar pruebas destructivas o económicamente exigentes. Según Chen et al. (2017), la aplicación del Método de los Elementos Finitos (MEF) en estas simulaciones no solo brinda resultados precisos; sino también un nivel de detalle que facilita la toma de decisiones en proyectos de ingeniería civil, desde el diseño de cimentaciones hasta la evaluación de estabilidad en estructuras.



2.6.3. Formulación Fuerte

En el contexto se resalta que el Método de los Elementos Finitos (MEF), la formulación fuerte se refiere a la representación directa de las ecuaciones diferenciales que expresan cómo se comporta el sistema físico. Para el caso específico de la consolidación de suelos, estas ecuaciones incluyen las leyes de equilibrio; las ecuaciones de flujo de agua (como la ecuación de Darcy) y las ecuaciones constitutivas que explican la relación entre los esfuerzos y las deformaciones del material. Con esta formulación, las ecuaciones se aplicaran de manera puntual en cada punto del dominio del problema, teniendo en cuenta las condiciones de frontera y las propiedades del suelo en todo momento. Aunque la formulación fuerte es muy precisa, puede ser menos práctica cuando se trabaja con problemas grandes o complejos. Esto se debe a que los métodos para resolver las ecuaciones de forma exacta son bastante exigentes y no siempre viables con recursos computacionales estándar. Por esta razón, se opta con mayor frecuencia por la formulación débil, que permite resolver problemas similares de manera más eficiente sin incurrir en demasiada precisión.

2.6.4. Formulación Débil

Se puede decir que la formulación débil es una aproximación que se obtiene aplicando principios como el de los trabajos virtuales o el principio de mínimos, que transforman las ecuaciones diferenciales originales en una forma integral; En lugar de resolver las ecuaciones en cada punto exacto del dominio, esta formulación busca una solución que, aunque no sea exacta en cada punto, cumpla las ecuaciones obtenidas en promedio dentro del dominio. La metodología aplicada es mucho más práctica para el análisis numérico, ya que simplifica la discretización del problema y lo convierte en un sistema de ecuaciones algebraicas que puede resolverse eficientemente con el Método de

los Elementos Finitos (MEF); Otra ventaja clave de la formulación débil es que relaja los requisitos de diferenciabilidad, por lo cual es crucial para trabajar con problemas que incluyen geometrías complejas o discontinuidades en las propiedades del material, como sucede frecuentemente en el análisis de suelos.

2.6.5. Formulación Matricial

La formulación matricial es el paso clave, donde se discretiza el problema utilizando el Método de los Elementos Finitos (MEF). En esta etapa; las ecuaciones que describen el sistema (ya sean diferenciales o débiles) se reescriben en términos de matrices que representan las propiedades del sistema, como la rigidez, la permeabilidad y las cargas externas. Cada término dado en las ecuaciones se transforma en una matriz específica, y las incógnitas del sistema, como los desplazamientos o las presiones de poro, se expresan como vectores. El proceso dado implica integrar las ecuaciones diferenciales en pequeños elementos, y luego, los resultados de esa discretización se ensamblan para formar un sistema global de ecuaciones lineales. Este sistema; que refleja el comportamiento del problema en su totalidad, se puede resolver mediante métodos numéricos. Es precisamente esta formulación la que se va a utilizar en los programas de análisis por elementos finitos, permitiendo obtener soluciones aproximadas a problemas complejos en la mecánica de suelos, lo que resulta crucial para realizar análisis geotécnicos precisos y eficientes.

$$\begin{bmatrix} 0 & 0 \\ Q^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{v} \\ \dot{P}_e \end{bmatrix} + \begin{bmatrix} K & Q \\ 0 & -H \end{bmatrix} \cdot \begin{bmatrix} v \\ P \end{bmatrix} = \begin{bmatrix} f^\sigma + f^v \\ f^w + f^P \end{bmatrix}$$

Donde:

K : Matriz de rigidez global

H : Matriz hidráulica global

Q : Matriz que relaciona los desplazamientos con desplazamientos.

v : Desplazamiento vertical



P : Presión

\dot{v} : Velocidad

\dot{P}_e : Velocidad de la presión por exceso

Como se puede apreciar se tiene dos matrices, las cuales complica el proceso de solución del sistema. Es por ellos que se adecua a un sistema equivalente incremental.

2.7. VISUAL STUDIO Y C# COMO HERRAMIENTAS DE SIMULACIÓN

Visual Studio es un entorno de desarrollo integrado (IDE) robusto que permite implementar algoritmos complejos mediante C#. Este lenguaje combina una sintaxis clara con capacidades avanzadas para la programación orientada a objetos, lo que facilita la construcción de modelos numéricos como el MEF (Albahari & Albahari, 2022). Además, la capacidad de C# para integrar bibliotecas externas, como Math.NET o Accord.NET, mejora el desempeño de los cálculos matemáticos.

2.8. LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación es un sistema formal compuesto por un conjunto de símbolos y reglas que permiten escribir programas y algoritmos para resolver los diferentes problemas computacionales. Estos programas deben ser traducidos al lenguaje máquina mediante un compilador, para que puedan ser ejecutados por el hardware de un sistema (Sebesta, 2016). Los lenguajes de programación se clasifican comúnmente en:

- Lenguajes de bajo nivel: Cercanos al lenguaje máquina y dependientes del hardware, como el ensamblador.
- Lenguajes de nivel medio: Ofrecen una combinación de control directo del hardware y abstracción, como C (Aho et al., 2006).
- Lenguajes de alto nivel: Más abstractos y enfocados en la facilidad de uso, como C#, Python o Java (Richter, 2012).



2.9. LENGUAJE DE PROGRAMACIÓN C#

C# es un lenguaje de programación de alto nivel, moderno y orientado a objetos que fue desarrollado por Microsoft como parte de su plataforma .NET. El presente lenguaje se destaca por varias características clave:

- **Abstracción y robustez:** Facilita el desarrollo de software al ofreciendo bibliotecas poderosas y manejo automático de memoria, lo que hace que el trabajo del programador sea más sencillo y seguro (Richter, 2012).
- **Portabilidad:** Gracias a la plataforma .NET, las aplicaciones desarrolladas en C# se pueden ejecutarse en diversas plataformas, lo que aumenta su versatilidad y alcance.
- **Versatilidad:** C# no solo es adecuado para aplicaciones de escritorio, sino que también se usa ampliamente con el desarrollo de aplicaciones web, móviles y videojuegos (Troelsen & Japikse, 2021).
- **Sintaxis intuitiva:** estas combinan características de lenguajes como C++ y Java, lo que hace que su sintaxis sea fácil de aprender y usar, especialmente para quienes ya tienen experiencia en programación.

Además, C# utiliza un compilador para traducir el código fuente a un lenguaje intermedio (MSIL, Microsoft Intermediate Language); que luego es interpretado y ejecutado por el entorno de ejecución común (CLR, Common Language Runtime). Este enfoque garantiza que el código sea eficiente y compatible en diferentes entornos (Troelsen & Japikse, 2021).

2.9.1. Algoritmo

Un algoritmo es un conjunto de instrucciones ordenadas y finitas que están diseñadas para resolver un problema o realizar tareas específicas en un tiempo determinado. Para crear un algoritmo, se tiene que definir claramente el problema que



se desea resolver, identificando las variables involucradas, las condiciones del problema y los resultados esperados; Además, es crucial establecer criterios precisos que permitan validar la veracidad de los resultados obtenidos, asegurando así que la solución propuesta sea correcta y efectiva. (Rodríguez, 2017, p. 12).

2.9.2. Estructura de un Programa en C#

En C#, los programas se organizan en clases, siendo el método Main el punto de entrada del programa, donde se inicia la ejecución del código principal. Este método siempre debe estar definido dentro de una clase, y tiene la capacidad de aceptar parámetros, como los argumentos que se pasan desde la línea de comandos, lo que permite personalizar la ejecución del programa en función de los datos proporcionados al iniciar el programa.

2.9.3. Procedimientos en C#

En C#, los procedimientos son bloques de código diseñados para realizar tareas específicas y lo más importante, pueden ser reutilizados en diferentes partes de un programa, esto mejora la eficiencia y organización del código. Por ende los procedimientos se implementan principalmente como métodos; los cuales tienen la opción de devolver un valor o no, dependiendo de la tarea que realicen.

2.9.4. Declaración de Variables y Tipos de Datos

En C#, cada variable debe ser declarada con un tipo de dato específico lo que asegura que las operaciones realizadas con ella sean seguras y precisas. Cuando se declara una variable; se define su tipo y es importante destacar que, una vez asignado un tipo a la variable, este no puede cambiar durante la ejecución del programa.

2.9.5. Tipos de Datos Comunes en C#

- Números enteros: int, long, short, byte



- Números decimales: float, double, decimal
- Texto: string
- Booleanos: bool
- Caracteres: char

2.9.6. Operadores Aritméticos:

Estos operadores permiten realizar operaciones matemáticas básicas.

- +: Suma
- -: Resta
- *: Multiplicación
- /: División

2.9.7. Operadores de Comparación

Permiten comparar valores.

- ==: Igualdad
- !=: Diferente
- <: Menor que
- >: Mayor que
- <=: Menor o igual
- >=: Mayor o igual

2.9.8. Operadores Lógicos

Utilizados en expresiones booleanas.

- &&: AND lógico
- ||: OR lógico



- !: NOT lógico

2.9.9. Matrices

También conocido como array es una colección de elementos con el mismo tipo de variable almacenados en ubicaciones de memoria contiguas. Por ende, Los arrays permiten organizar y manipular datos de forma estructurada. En C#, se pueden trabajar con arrays unidimensionales, bidimensionales, que son los más comunes.

2.9.9.1. Matriz Unidimensional

Una matriz unidimensional es una lista lineal de elementos del mismo tipo. En donde, cada elemento se accede mediante un índice único.

2.9.9.2. Matriz Bidimensional

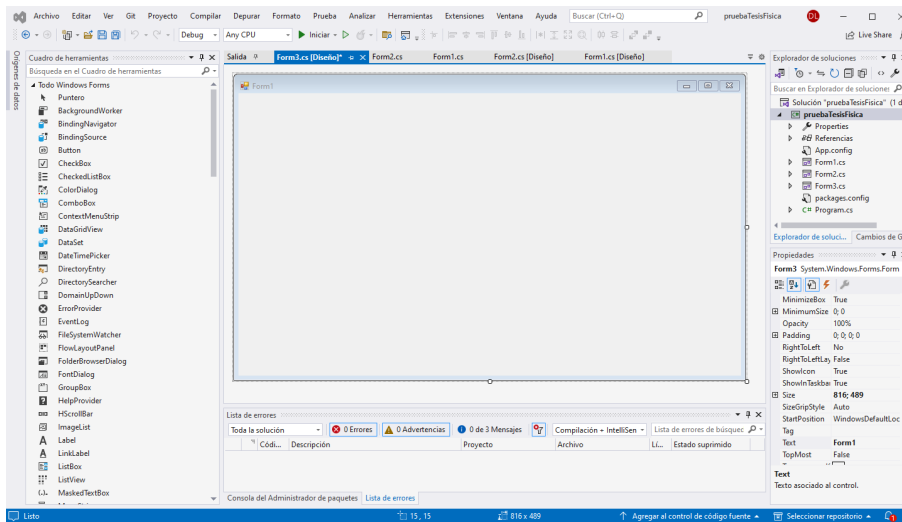
Una matriz bidimensional es una colección de datos organizados en filas y columnas, Se puede decir que es similar a una tabla.

2.10. WINDOWS FORMS

Esta es una tecnología introducida como parte del .NET Framework y se centra en la creación rápida de aplicaciones gráficas. En donde, se basa en una arquitectura orientada a objetos, donde cada ventana o formulario está representado por una clase; Esto permite a los desarrolladores crear interfaces funcionales mediante el uso de controles visuales integrados. Según González (2020), "Windows Forms sigue siendo una opción viable para aplicaciones empresariales simples debido a su equilibrio entre facilidad de uso y funcionalidad". Los componentes o controles de Windows Forms son elementos que se colocan en los formularios para construir la interfaz gráfica. A continuación; se describen algunos de los controles más comunes, junto con sus propiedades y las aplicaciones en las que suelen usarse, facilitando así la creación de interfaces intuitivas y funcionales.

Figura 4

Formulario de C#



Nota: Se muestra el entorno del programa Visual Studio 2022. En el centro, se aprecia un formulario vacío. A la izquierda se encuentra el cuadro de herramientas, que permite insertar diversos componentes en el formulario. En el lado derecho, se visualiza la ventana del explorador de archivos y, en la parte inferior, la ventana de propiedades. Fuente: Elaboración propia.

2.10.1. Etiqueta (Label)

El Label (también conocido como etiqueta) es un componente que se inserta en el proyecto desde el cuadro de herramientas de Windows Forms. Este control se utiliza para mostrar texto fijo que no puede ser modificado directamente por el usuario; Su principal función es mostrar texto estático en la interfaz gráfica, como instrucciones, descripciones o etiquetas para otros controles, facilitando la interacción del usuario con la aplicación.

2.10.2. Cuadro de Texto (TextBox)

El TextBox es un control que se agrega al proyecto desde el cuadro de herramientas de Windows Forms. Este elemento permite al usuario ingresar datos directamente en el formulario. Se utiliza para capturar texto, ya sea para que el usuario lo ingrese o para visualizar información dinámica dentro de la aplicación. Es una herramienta esencial para la interacción con el usuario, permitiendo que se introduzcan datos como nombres,



direcciones o cualquier tipo de texto.

2.10.3. Botón (Button)

El Button (comúnmente conocido como botón). Este control es fundamental para permitir que el usuario ejecute acciones de manera sencilla y rápida dentro de la aplicación. Al hacer clic en el botón, se activa una acción específica que puede ser programada, como enviar datos, abrir una nueva ventana o realizar cálculos. Es uno de los controles más utilizados para la interacción directa con el usuario

2.10.4. Lista (ListBox)

Denominada ListBox, se inserta en el proyecto utilizando el cuadro de herramientas. Este componente permite mostrar una lista de opciones o información al usuario, mejorando la interacción con la aplicación. Este nos muestra una lista de elementos, permitiendo al usuario seleccionar uno o varios.

2.10.5. Cuadro Combinado (ComboBox)

Conocido como ComboBox, se incorpora desde el cuadro de herramientas. Este componente permite al usuario seleccionar una opción de un conjunto predefinido de alternativas, ofreciendo una lista desplegable de opciones. Fuente: Elaboración propia. Ofrece una lista desplegable de opciones de las que el usuario puede seleccionar una.

2.10.6. Tablas (DataGridView)

Llamadas DataGridView; Estos se añaden al proyecto mediante el cuadro de herramientas. Este elemento permite mostrar datos en un formato tabular, organizado y visualmente claro para el usuario. Este control se utiliza para mostrar y manipular datos en formato tabular. Es ideal para trabajar con bases de datos o listas estructuradas.



2.10.7. Imágenes (PictureBox)

Conocidas como PictureBox, se insertan en el proyecto desde el cuadro de herramientas. Este componente sirve para mostrar imágenes o gráficos, lo que ayuda a mejorar el entendimiento del modelo visual para el usuario. Sirve para mostrar imágenes en la interfaz. Es común en aplicaciones que necesitan trabajar con gráficos o recursos visuales.

2.10.8. Chart

También llamados Chart, se agregan al proyecto a través del cuadro de herramientas. Este elemento permite representar y comparar datos mediante gráficos visuales. Este control permite crear gráficos para visualizar los datos estadísticos o numéricos; Es útil en aplicaciones que requieren análisis visual de la información.

2.10.9. Control de Pestañas (TabControl)

El conocido como TabControl, se integra en el proyecto desde el cuadro de herramientas. Este componente facilitara el trabajo con múltiples elementos al organizando las pestañas; optimizando el espacio y la interacción. Permite organizar la interfaz en pestañas, agrupando controles relacionados en diferentes secciones.

2.11. ANTECEDENTES

2.11.1. Antecedentes: Tesis

Bedon, A. (2022) En su tesis de pregrado titulada “Estudio sobre el comportamiento de consolidación de arcillas blandas en la construcción secuencial de un terraplén”, Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú. Realizo el estudio de suelos arcillosos con presencia de agua que ocasionan grandes



asentamientos debido a la consolidación primaria y secundaria, para esto se realizó un análisis consolidación de suelos blandos para la conformación de un terraplén, usando el método de los elementos finitos con el fin de comprobar estos resultados se realizó el análisis con la teoría de Terzaghi. Concluyendo que el método analítico brinda el asentamiento relativamente cercano a los reales.

García, J. (2019) En su tesis de maestría titulada “Diseño de un sistema simulador físico de flujo multifásico para reproducir condiciones de flujo semejantes a las reales”, En CIATEO, México. Construye un sistema con la capacidad de simular el comportamiento físico de fluidos multifásico, con la finalidad de reproducir el comportamiento más aproximado al real de un fluido. Concluyendo con la recreación de flujo dinámico a través de una herramienta virtual, la cual tiene congruencia con mapas de modelos actuales aceptados.

Montaña, D. y Villa, V. (2018) En su tesis de pregrado titulada “Simulación numérica de la licuación de arenas mediante un programa de elementos finitos”, Facultad de ingeniería de la Universidad Militar Nueva Granada, Colombia. Realizo la simulación del fenómeno de licuación de suelos arenosos mediante el método de los elementos finitos usado como software de análisis Plaxis, los resultados obtenidos fueron comparados con ensayo en el laboratorio de suelos. Concluyendo que la simulación numérica muestra un comportamiento por exceso en la presión, pero en general se llegó a que los modelos planteados no tienen la aproximación esperada entre la simulación numérica y la experimental.

Camejo, J. y Rodríguez, Y. (2016) En su tesis de pregrado titulada “Diseño de un material didáctico para la introducción al análisis estructural mediante el método de elementos finitos adaptado a la unidad curricular de estructuras avanzadas”, Facultad de



ingeniería de la Universidad de Carabobo, Venezuela. Emplea el método de los elementos finitos, mediante la incorporación de material didáctico aplicado al análisis estructural, para lo cual adapta los conceptos y fundamentos del MEF al contenido curricular. Concluyendo con la creación del material digital.

García, G. (2016) En su tesis doctoral titulada “Caracterización Adimensional y Simulación Numérica de Procesos Lineales y No Lineales de Consolidación de Suelos”, Departamento de Ingeniería de la Universidad Politécnica de Cartagena, España. Realiza la búsqueda de parámetros adimensionales que rigen la consolidación de suelos anisótropos, heterogéneos, multicapa, en diferentes geometrías con un comportamiento lineal en condiciones drenadas parcial o total, por otro lado, realiza el modelado de la consolidación no lineal 1D, encontrado como resultados el tiempo característico, grado medio de consolidación y grado medio de disipación de presiones intersticiales.

Osorio, M. (2015) En su tesis de pregrado titulada “Desarrollo de simulador interactivo de física enfocado al programa de estudio de ingeniería propuesto por la SENESCYT (desarrollo de los capítulos vectores y movimiento)”, Facultad de ciencias matemáticas y físicas de la Universidad de Guayaquil, Ecuador. Desarrolla una herramienta con la capacidad de generar simulaciones interactivas de vectores y movimientos de partículas, aplicado al programa de estudios de ingeniería, generado una interfaz interactiva, ayudado al proceso de aprendizaje, para la programación se usó JavaScript, HTML y MySQL. Obtenido una herramienta adecuada para las simulaciones de partículas en movimiento.

Di, H. (2006) En su tesis de doctoral titulada “Simulación numérica de problemas con no linealidad física y geométrica. Análisis de consolidación de suelos no saturados”, Facultad de Ingeniería de la Universidad Nacional del Nordeste, Argentina. Realiza



modelados matemáticos referidos al método de los elementos finitos, donde describe el comportamiento no lineal de suelos arcillosos no saturados, primero plantea un modelo para la consolidación basado en la combinación de diferentes estados, para después plantear el caso no lineal utilizado el principio de formas genéricas para cualquier sólido continuo. Utiliza el método de los elementos finitos para la obtención de formas algebraicas de las ecuaciones diferenciales.

Di, H. (1997) En su tesis de maestría titulada “Simulación numérica de procesos de consolidación de suelos usando el método de los elementos finitos”, Facultad de Ingeniería de la Universidad Nacional del Nordeste, Argentina. Muestra un modelo matemático bidimensional basado en la teoría Biot, la cual permite el análisis de tensiones efectivas y de poros. Para este modelado se usa el método de los elementos finitos para la interpolación de elementos que presenten ocho nodos, brindando desplazamientos y elementos de cuatro nodos brinda presiones de poros.

2.11.2. Antecedentes: Artículos

Muñoz, S., Sánchez W. y García, J. (2020) En su artículo titulado “Teoría de consolidación, una revisión de la literatura”, Universidad Señor de Sipán, Perú. El objetivo de los autores es de recabar información sobre la evolución de la teoría de consolidación unidimensional, la cual fue propuesta por Terzaghi para la terminación de asentamientos producto de la disminución de vacíos debido a la expulsión del agua. Para esto se tomó 60 referencias bibliográficas. Concluyendo que se ha logrado avances significativos al simular numéricamente el comportamiento hidromecánico mediante programas que usan el método de los elementos finitos.

Tristancho, J. (2017) En su artículo titulado “Modelamiento físico en geotecnia aplicado a la docencia”, Facultad de Ingeniería de la Universidad Distrital Francisco



José de Caldas de Bogotá, Colombia. Plantea el uso de la centrifuga como herramienta didáctica, esto aplicado a las prácticas de laboratorio en los cursos de ingeniería. Para luego describir la solución completa del modelamiento en centrifuga de un problema típico de geotecnia basado en la estructura de un túnel poco profundo en suelo blando. Finalizando con un discurso sobre las ventajas y desventajas que este tipo de laboratorios presenta frente a las técnicas de laboratorios virtuales.

Reinert, H., Duarte, J. y Silva, M. (2014) En su artículo titulado “Modelación numérica de plateas de cimentación sobre suelos compresibles de misiones, argentina”, Facultad de Ingeniería de la Universidad Nacional de Misiones, Argentina. Realizo el estudio del comportamiento de estos suelos, cuando son sometidos a cargas producidas por viviendas sociales. Para ello el método de los elementos finitos, para analizar problemas geotécnicos, la cual se plantea por etapas, siendo necesarias para reproducir el suelo y que permitan su drenaje de los suelos, los suelos analizados fueron de la provincia de Misiones, llegando a obtener las órdenes de magnitud de los asentamientos esperados, los esfuerzos de trabajo en diferentes niveles del perfil del suelo.

Falconi, R. y Mroginski, J. (2010) En su artículo titulado “Matriz de rigidez para un elemento finito de suelo saturado en dos dimensiones”, Escuela Politécnica del Ejército. El objetivo de los autores fue obtener la matriz de rigidez de un elemento finito rectangular mixto de ocho nodos, con la finalidad de modelar el comportamiento elástico de un suelo saturado en dos dimensiones. El elemento presento dieciséis grados de libertad para definir los desplazamientos y cuatro grados de libertad para el campo de las presiones, finalmente llegando a escribir un script para realizar la simulación numérica.



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. MÉTODO DE INVESTIGACIÓN

En este trabajo se utilizó la metodología de los elementos finitos aplicada a suelos para determinar asentamientos y presiones, desarrollando una herramienta computacional basada en el lenguaje de programación C#. El método de investigación adoptado fue el método analítico numérico, según Colömë y Femenia (2018), quienes definieron que este “consiste en descomponer el objeto de estudio en sus partes integrantes, estudiando cada una de ellas con asumiendo que el conocimiento de dichas partes traerá el conocimiento del objeto como un todo”. Este enfoque permitió resolver problemas complejos de la mecánica del suelo al descomponer el dominio en elementos finitos que fueron analizados individualmente para predecir el comportamiento global del sistema.

3.2. TIPO DE INVESTIGACIÓN

El tipo de investigación realizado fue aplicada, ya que se utilizaron conocimientos teóricos de la física computacional y métodos numéricos para desarrollar una herramienta computacional práctica. Según Carrasco (2005), la “ investigación aplicada se distingue por tener propósitos prácticos inmediatos bien definidos, es decir, se investiga para actuar, transformar, modificar o producir cambios en un determinado sector de la realidad”. En este contexto, el desarrollo de la herramienta computacional estuvo orientado a facilitar la simulación de fenómenos reales en suelos, lo cual tiene aplicaciones directas en la ingeniería civil y geotécnica.

3.3. NIVEL DE INVESTIGACIÓN

El nivel de investigación fue descriptivo. Por lo que se se midió la respuesta del suelo bajo la acción de cargas aplicadas a lo largo del tiempo, utilizando elementos finitos lineales. Hernández (2014, p. 92) sostiene que este nivel “se centra en la medición y análisis de las características, y fenómenos tal como se presentan en la realidad”. Además, la investigación tuvo un componente exploratorio debido a que la simulación numérica de suelos con elementos lineales es un área poco estudiada, lo que justificó la necesidad de explorar nuevas metodologías y enfoques en este campo.

3.4. DISEÑO DE INVESTIGACIÓN

El diseño de la investigación fue no experimental. Teniendo un enfoque descriptivo-comparativo. Hernández (2014, p. 93) explica que “el diseño no experimental busca recoger información de muestras sobre el mismo objeto de investigación y caracterizarlo a través de una descripción comparativa”. En este estudio, se realizaron simulaciones numéricas de suelos granulares y arcillosos bajo diversas condiciones de carga y drenaje, con el objetivo de observar y comparar el comportamiento de los asentamientos y las presiones en cada caso.

$$M_1 \rightarrow O_1$$

$$M_2 \rightarrow O_2$$

Donde:

M: Representa las muestras del estudio, que en este caso fueron suelos granulares y arcillosos.

O: Se es la observaciones de las muestras bajo diferentes condiciones simuladas, recogiendo información relevante sobre los indicadores de asentamientos.



3.5. POBLACIÓN

La población estuvo constituida por los distintos tipos de suelos que soporte cargas como granulares y arcillosos, específicamente aquellos que presentan características arcillosas, estos tipos de suelos son más comunes en aplicaciones geotécnicas y representan una variedad de comportamientos mecánicos relevantes para el análisis de asentamientos.

3.6. MUESTREO

En el estudio se empleó un muestreo no probabilístico de tipo intencional, el cual permitió al investigador seleccionar las muestras más relevantes según la conveniencia del estudio. Para ello, se seleccionaron tipos de suelos comunes, como suelos granulares y arcillosos, que fueron representativos de diferentes condiciones geotécnicas y permitieron explorar el comportamiento del suelo bajo diferentes escenarios de carga y drenaje.

3.7. MUESTRAS

De acuerdo con el muestreo planteado, se realizó la selección de suelos granulares y arcillosos como las muestras a estudiar. Estos suelos fueron elegidos debido a sus características diferenciadas, lo cual proporcionó una visión más completa sobre el comportamiento del suelo en condiciones drenadas y no drenadas, además de su respuesta a la aplicación de cargas.

3.8. DESCRIPCIÓN DETALLADA DE MÉTODOS POR OBJETIVOS ESPECÍFICOS

Esta investigación es de tipo aplicada; por lo que no requiere la realización de pruebas estadísticas para el procesamiento de los datos. La validación de la herramienta computacional denominada Suelos MEF se llevó a cabo en concordancia con los

antecedentes establecidos en la investigación. El objetivo principal es desarrollar una herramienta computacional utilizando el lenguaje de programación C#, que permita realizar la simulación numérica de la consolidación lineal de suelos, empleando el método de los elementos finitos. Esto mejora la simulación numérica al facilitar la obtención de asentamientos de suelos mediante una herramienta fácil de usar, eliminando la necesidad de realizar cálculos manuales y largos. por otro lado, las muestras utilizadas no son de tipo probabilístico.

3.8.1. Definir las ecuaciones matriciales del método de los elementos finitos para determinar el asentamiento de suelos

Plantear la formulación matricial del método de los elementos finitos, con elementos lineales, para la simulación numérica del comportamiento de suelos drenados y no drenados. Partimo de la formulacion debil del método de los elementos finitos para elemento lineales.

3.8.1.1. Matriz de rigidez

la matriz de rigidez esta definida por:

$$K^e = \frac{E}{3 \cdot He} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}$$

Donde:

E : Modulo de elasticidad del suelo, en el elemento analizado.

He : Altura del elemento.

3.8.1.2. Matriz de hidraulica

La matriz hidraulica esta definida por:

$$H^e = \frac{k}{\rho_w \cdot g \cdot H^e} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Donde:

k : Modulo de elasticidad del suelo, en el elemento analizado.

ρ_w : Constante de filtración del agua.

g : Gravedad terrestre

H^e : Altura del elemento.

3.8.1.3. Matriz que relaciona los desplazamientos verticales y presiones

la matriz esta definida por: $Q^e = \frac{1}{6} \begin{bmatrix} -5 & -1 \\ 4 & -4 \\ 1 & 5 \end{bmatrix}$

Se aprecia que la matriz Q^e es una matriz constante, donde las filas están relacionada con los desplazamientos y las columnas están relacionadas con las presiones.

3.8.1.4. Vectores de fuerzas

$$f e^v = \frac{-\gamma_s \cdot H^e}{6} \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} \quad f e^P = k \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

3.8.1.5. Solución del sistema matricial por equivalencia incremental

Forma matricial modificada para resolver el sistema.

$$\begin{bmatrix} K & \frac{1}{\theta \cdot \Delta t} \cdot Q \\ \frac{1}{\theta \cdot \Delta t} \cdot Q^T & -\frac{1}{\theta \cdot \Delta t} \cdot H \end{bmatrix} \cdot \begin{bmatrix} \Delta v \\ \theta \cdot \Delta t \cdot \Delta P \end{bmatrix} = \begin{bmatrix} \Delta f^\sigma \\ \frac{1}{\theta} \cdot Q^T \dot{v}_t \end{bmatrix}$$

Esta ecuación matricial será fundamental para obtener los asentamiento y presiones del suelo, la cual se implementará en el desarrollo de la herramienta informática.

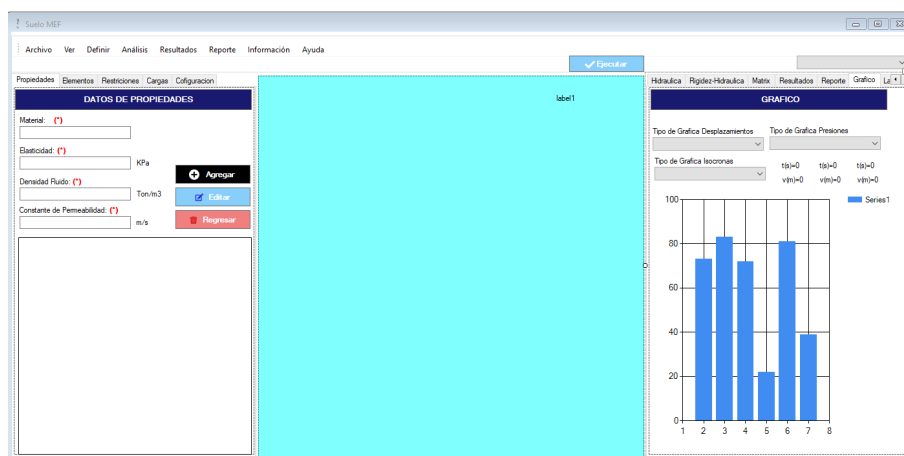
3.8.2. Desarrollo de la herramienta computacional para determinar los asentamiento y presiones de suelos mediante la simulación numérica

3.8.2.1. Construcción de la interfaz gráfica en Visual Studio C#

En el programa desarrollado para el análisis de suelos mediante el método de los elementos finitos, se utilizaron controles como Label para mostrar información estática, TextBox para capturar parámetros del usuario, ComboBox para seleccionar opciones predefinidas relacionadas con las propiedades del suelo, y PictureBox para mostrar imágenes ilustrativas o resultados visuales. También se incluyeron Chart para graficar los resultados de los cálculos y DataGridView para presentar los datos tabulares del análisis. Cada control se configuró tanto en el diseño como mediante código, y se implementaron eventos interactivos, como botones para ejecutar el análisis y actualizar los resultados dinámicamente.

Figura 5

Diseño del formulario principal



Nota: Se muestra herramientas muy usadas en el diseño de formularios.

3.8.2.2. Programación del álgebra lineal en C#

A continuación, se presentan algunas funciones importantes que permiten realizar operaciones con matrices en C#, como suma, multiplicación, transposición, solución de un

sistema lineal. Estas herramientas fueron esenciales para la programación de la herramienta informática. Estas funciones son las más usadas o llamadas en cada procedimiento.

3.8.2.2.1. Matriz Nula Este procedimiento devuelve una matriz de ceros de n filas y m columnas.

Algoritmo 1. Creación de una matriz de ceros (`zerosM`)

Input: n : Número de filas, m : Número de columnas

Output: M : Matriz de ceros de dimensiones $n \times m$

```
1 Crear una matriz  $M$  de tamaño  $n \times m$ 
2 for  $i \leftarrow 0$  to  $n - 1$  do
3   for  $j \leftarrow 0$  to  $m - 1$  do
4      $M[i][j] \leftarrow 0$ 
5 return  $M$ 
```

3.8.2.2.2. Matriz traspuesta Este procedimiento devuelve la matriz traspuesta de m filas y n columnas.

Algoritmo 2. Transposición de una matriz (`tranponerM`)

Input: $matrA$: Matriz de dimensiones $n \times m$

Output: $matrM$: Matriz traspuesta de dimensiones $m \times n$

```
1  $matrM \leftarrow$  Matriz de ceros de tamaño  $m \times n$  (usando la función zerosM)
2 for  $i \leftarrow 0$  to  $m - 1$  do
3   for  $j \leftarrow 0$  to  $n - 1$  do
4      $matrM[i][j] \leftarrow matrA[j][i]$ 
5 return  $matrM$ 
```

3.8.2.2.3. Multiplicación de matrices Este procedimiento devuelve la multiplicación entre dos matrices.

Algoritmo 3. Multiplicación de matrices (multM)

Input: *matrA*: Matriz de dimensiones $n \times m$, *matrB*: Matriz de dimensiones $m \times p$

Output: *matrC*: Matriz resultado de dimensiones $n \times p$

```
1 matrC ← Matriz de ceros de tamaño  $n \times p$  (usando la función zerosM)
2 for  $i \leftarrow 0$  to  $n - 1$  do
3   for  $j \leftarrow 0$  to  $p - 1$  do
4      $sum \leftarrow 0$ 
5     for  $k \leftarrow 0$  to  $m - 1$  do
6        $sum \leftarrow sum + matrA[i][k] \cdot matrB[k][j]$ 
7      $matrC[i][j] \leftarrow sum$ 
8 return matrC
```

3.8.2.3. Programación de la formulación matricial del método de los elementos finitos en C#

3.8.2.3.1. FUNCION MATRIZ DE RIGIDEZ DEL ELEMENTO Ke RIGIDEZ

Para obtener la matriz global de rigidez se necesita tres procedimientos, la primera se encarga de generar la matriz de rigidez por elemento, el segundo procedimiento se encarga de crear la matriz global con un elemento, y finalmente la tercera matriz se encarga de generar la matriz global.

Algoritmo 4. Cálculo de la matriz de rigidez elemental (Ke)

Input: E : Módulo de elasticidad, L : Longitud

Output: Ke : Matriz de rigidez elemental

```
1  $ke \leftarrow \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}$ 
2  $r \leftarrow \frac{E}{3L}$ 
3  $Ke \leftarrow ke$  multiplicada por el escalar  $r$ 
4 return  $Ke$ 
```

Algoritmo 5. Ensamblado de la matriz global (KEnsamblado)

Input: K : Matriz global, Ke : Matriz elemental, gdl : Grados de libertad del elemento

Output: K : Matriz global actualizada

```
1 for  $i \leftarrow 0$  to 2 do
2    $ifil \leftarrow gdl[i]$ 
3   for  $j \leftarrow 0$  to 2 do
4      $jcol \leftarrow gdl[j]$ 
5      $K[ifil - 1, jcol - 1] \leftarrow K[ifil - 1, jcol - 1] + Ke[i, j]$ 
6 return  $K$ 
```

Algoritmo 6. Ensamblado completo de la matriz global (KKEnsamblado)

Input: d_L : Longitudes de elementos, d_E : Módulos de elasticidad, d_{gdl} : Matriz de grados de libertad

Output: K : Matriz global ensamblada, $KeAcumulado$: Matriz elemental acumulada

```
1  $gdl \leftarrow$  Vector de ceros de tamaño 3
2 for  $m \leftarrow 0$  to  $nel - 1$  do
3    $L \leftarrow d_L[m]$ 
4    $E \leftarrow d_E[m]$ 
5    $gdl[0] \leftarrow d_{gdl}[m, 0]$ 
6    $gdl[1] \leftarrow d_{gdl}[m, 1]$ 
7    $gdl[2] \leftarrow d_{gdl}[m, 2]$ 
8    $kke \leftarrow Ke(E, L)$ 
9    $KeAcumulado \leftarrow$  apilar ( $KeAcumulado, kke$ )
10   $K \leftarrow KEnsamblado(K, kke, gdl)$ 
11 Imprimir  $K$  en  $dgvK$ 
```

3.8.2.3.2. FUNCION MATRIZ HIDRAULICA Para obtener la matriz global de hidráulica se necesita tres procedimientos, la primera se encarga de generar la matriz hidráulica por elemento, el segundo procedimiento se encarga de crear la matriz global con un elemento, y finalmente la tercera matriz se encarga de generar la matriz global.

Algoritmo 7. Cálculo de la matriz de rigidez elemental de conductividad (H_e)

Input: L : Longitud, k : Coeficiente de conductividad térmica, ρ : Densidad, g : Gravedad

Output: H_e : Matriz de rigidez elemental de conductividad

```
1  $he \leftarrow \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ 
2  $r \leftarrow \frac{k}{L \cdot \rho \cdot g}$ 
3  $He \leftarrow he$  multiplicada por el escalar  $r$ 
4 return  $He$ 
```

Algoritmo 8. Ensamblado de la matriz global (HEnsamblado)

Input: H : Matriz global, H_e : Matriz elemental, gdl : Grados de libertad del elemento

Output: H : Matriz global actualizada

```
1  $gdlp \leftarrow \{gdl[0], gdl[2]\}$ 
2 for  $i \leftarrow 0$  to 1 do
3    $ifil \leftarrow \left( \frac{gdlp[i]}{2} + 0.5 \right)$ 
4   for  $j \leftarrow 0$  to 1 do
5      $jcol \leftarrow \left( \frac{gdlp[j]}{2} + 0.5 \right)$ 
6      $H[ifil - 1, jcol - 1] \leftarrow H[ifil - 1, jcol - 1] + He[i, j]$ 
7 return  $H$ 
```

Algoritmo 9. Ensamblado completo de la matriz global de conductividad (HHEnsamblado)

Input: d_L : Longitudes de los elementos, d_k : Coeficientes de conductividad térmica, d_ρ : Densidades, d_{gdl} : Matriz de grados de libertad

Output: H : Matriz global ensamblada, $HeAcumulado$: Matriz elemental acumulada

```
1  $gdl \leftarrow$  Vector de ceros de tamaño 3
2 for  $i \leftarrow 0$  to  $nel - 1$  do
3    $L \leftarrow d_L[i]$ 
4    $k \leftarrow d_k[i]$ 
5    $E \leftarrow d_E[i]$ 
6    $rho \leftarrow d_\rho[i]$ 
7    $gdl[0] \leftarrow d_{gdl}[i, 0]$ 
8    $gdl[1] \leftarrow d_{gdl}[i, 1]$ 
9    $gdl[2] \leftarrow d_{gdl}[i, 2]$ 
10   $hhe \leftarrow$  He(L, k, rho, g)
11   $HeAcumulado \leftarrow$  apilar ( $HeAcumulado$ ,  $hhe$ )
12   $H \leftarrow$  HEnsamblado(H,  $hhe$ ,  $gdl$ )
13 Imprimir  $H$  en  $dgvH$ 
```

3.8.2.3.3. FUNCION MATRIZ DE K-H Para obtener la matriz global de Q se necesita tres procedimientos, la primera se encarga de generar la matriz hidráulica por elemento, el segundo procedimiento se encarga de crear la matriz global con un elemento, y finalmente la tercera matriz se encarga de generar la matriz global.

Algoritmo 10. Cálculo de la matriz de rigidez de carga elemental (Qe)

Output: Qe : Matriz de rigidez de carga elemental

```
1  $qe \leftarrow \begin{bmatrix} -5 & -1 \\ 4 & -4 \\ 1 & 5 \end{bmatrix}$ 
2  $r \leftarrow \left| \frac{1}{6} \right|$ 
3  $Qe \leftarrow$   $qe$  multiplicada por el escalar  $r$ 
4 return  $Qe$ 
```

Algoritmo 11. Ensamblado de la matriz global de carga (QEnsamblado)

Input: Q : Matriz global, Q_e : Matriz elemental, gdl : Grados de libertad del elemento

Output: Q : Matriz global actualizada

```
1  $gdlp \leftarrow \{gdl[0], gdl[2]\}$ 
2 for  $i \leftarrow 0$  to 2 do
3    $ifil \leftarrow gdl[i]$ 
4   for  $j \leftarrow 0$  to 1 do
5      $jcol \leftarrow \left(\frac{gdlp[j]}{2} + 0.5\right)$ 
6      $Q[ifil - 1, jcol - 1] \leftarrow Q[ifil - 1, jcol - 1] + Q_e[i, j]$ 
7 return  $Q$ 
```

Algoritmo 12. Ensamblado completo de la matriz global de carga (QQEnsamblado)

Input: d_gdl : Matriz de grados de libertad

Output: Q : Matriz global ensamblada, $Q_eAcumulado$: Matriz elemental acumulada

```
1  $gdl \leftarrow$  Vector de ceros de tamaño 3
2 for  $m \leftarrow 0$  to  $nel - 1$  do
3    $gdl[0] \leftarrow d_{gd}[m, 0]$ 
4    $gdl[1] \leftarrow d_{gd}[m, 1]$ 
5    $gdl[2] \leftarrow d_{gd}[m, 2]$ 
6    $qqe \leftarrow Q_e()$ 
7    $Q_eAcumulado \leftarrow$  apilar( $Q_eAcumulado$ ,  $qqe$ )
8    $Q \leftarrow QEnsamblado(Q, qqe, gdl)$ 
9 Imprimir  $Q$  en  $dgvQ$ 
```

3.8.3. Validación de los valores y asentamientos y presiones

La herramienta informática desarrollada en C# que utiliza como método de cálculo el método de los elementos finitos, fue validada al comparar los resultados con el método de diferencia central, mediante una hoja de cálculo de Excel desarrollada por el Doc. Osmar Carrillo de la Universidad de Wisconsin-Madison, esta hoja fue modificada en la cantidad en la carga y el tiempo de análisis. Los datos utilizados para esta simulación se encuentran especificados en la parte metodológica. Estos mismos datos son utilizados en la herramienta informática para simular el comportamiento del suelo en 2 años. Se

obtuvieron los siguientes resultados. tomando como datos

Tabla 3

Parámetros del suelo para comparación entre los dos métodos

| Parámetro | Valor |
|------------------------------------|-----------------------------|
| Conductividad hidráulica k [m/s] | $5 \cdot 10^{-9}$ |
| Módulo de elasticidad [KPa] | 750 |
| Carga aplicada [KPa] | 10 |
| Tiempo de análisis [s] | 63072000 (2 años) |
| Condiciones de frontera | Los extremos son permeables |
| Cantidad de elementos | 5 |
| Altura [m] | 5 |

Nota: Estos datos son tomados de la hoja de cálculo del Doc. Osmar Carrillo.

3.8.4. Simulación con 5 elementos y 10 elementos

En esta parte se simula el asentamiento a lo largo del tiempo utilizando la herramienta desarrollada con dos modelos: uno con 10 elementos y otro con 20 elementos. Los resultados permiten comparar la evolución de los asentamientos y analizar las diferencias entre ambos modelos, los parametros del suelos son los mismos que la anteriores simulacion. Los datos tomados son los mismo que se usan para la validación, con la diferencia de la cantidad de elementos, el tiempo de análisis es de 4 años, y la carga es de 63.38KPa .

3.8.5. Simulación con dos, cuatro y seis años

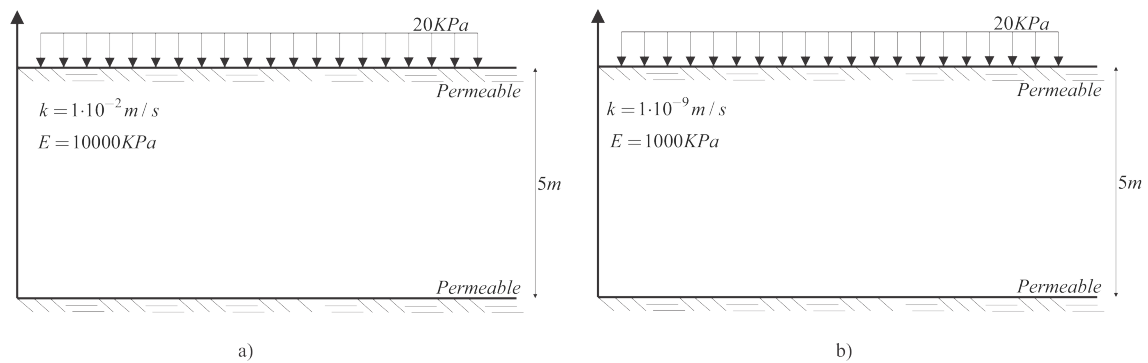
En esta parte se realiza el análisis del asentamiento del suelo debido a una carga de 63.38KPa se realizó considerando períodos de tiempo de dos, cuatro y seis años. Los resultados permiten comparar la evolución de los asentamientos y analizar las diferencias entre ambos modelos, los parametros del suelos son los mismos que la anteriores simulacion. Los datos tomados son los mismo que se usan para la validación, con la diferencia de la cantidad de elementos, el tiempo de análisis es de 4 años, y la carga es de 63.38KPa .

3.8.6. Simulación con suelos granulares y arcillosos

Para la implementación de la herramienta computacional desarrollada, se utilizaron dos tipos de suelos. Ambos suelos estarán sometidos a las mismas cargas externas y se tomará la misma cantidad de elementos en cada uno.

Figura 6

Suelos analizados



Nota: En figura a) suelo granular, b) suelo arcilloso.

3.8.6.1. Parametros de suelos

En tabla muestra las propiedades en los de dos tipos de suelos analizados: granular y arcilloso. Los parámetros incluyen elasticidad (módulo de elasticidad en Pa) y permeabilidad (coeficiente de permeabilidad en m/s). Estas propiedades de los suelos son obtenidos de libros de mecánica de suelos.

Tabla 4

Parametros de los suelos modelados

| Nro | Material | Elasticidad [KN/m2] | Densidad [Kg/cm3] | Permeabilidad [m/s2] |
|-----|-----------|---------------------|-------------------|----------------------|
| 1 | granular | 10000 | 1 | 1E-09 |
| 1 | arcilloso | 1000 | 1 | 1E-09 |

Nota: La tabla muestra las propiedades en los de dos tipos de suelos analizados: granular y arcilloso.

3.8.6.2. Discretización del suelos

En tabla muestra la discretización de suelos con una profundidad de 5 m, divididos en 10 elementos de 0.5 m cada uno. Esta discretización fue idéntica para ambos suelos analizados y se realizó siguiendo las consideraciones del tesista

Tabla 5

Discretización de los suelos modelados

| Elemento | Longitud [m] | Material |
|----------|--------------|----------|
| 1 | 0.5 | suelo |
| 2 | 0.5 | suelo |
| 3 | 0.5 | suelo |
| 4 | 0.5 | suelo |
| 5 | 0.5 | suelo |
| 6 | 0.5 | suelo |
| 7 | 0.5 | suelo |
| 8 | 0.5 | suelo |
| 9 | 0.5 | suelo |
| 10 | 0.5 | suelo |

Nota: La tabla muestra los 10 elementos tomados para cada modelo analizado.

3.8.6.3. Definición de condición de fronteras

Las condiciones de frontera están definidas por los valores conocidos de desplazamiento y presión. Se asignará un valor de 1 cuando el desplazamiento en un nodo sea cero. En todos los nodos se calcularán los desplazamientos, y en los nodos pares se determinarán las presiones

3.8.6.4. Asignación de cargas a lo largo del tiempo

A las cargas asignadas está definida de forma progresiva cada intervalo de tiempo, para ambos modelos se les asigna el mismo intervalo de tiempo y las mismas cargas en ambos modelos analizados. Se tomó en consideración 1,822 intervalos de integración, avanzando en incrementos de dos días. Además, se asignó una carga progresiva que simuló



las diferentes etapas de construcción de una vivienda.

3.8.6.5. Ejecución de la herramienta computacional

Para obtener los resultados, se hizo clic en el botón "Ejecutar", que realizó el análisis del suelo. Los resultados mostraron los asentamientos en cada nodo y, en los nodos pares, también se presentaron las presiones correspondientes en esos puntos.

3.8.7. Comparación del resultado de ambos modelos analizados

Finalmente mediante tablas y gráficos se compararon los resultados obtenidos entre ambos modelos analizados. Para esta medición se utilizó la razón como instrumento de comparación.



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

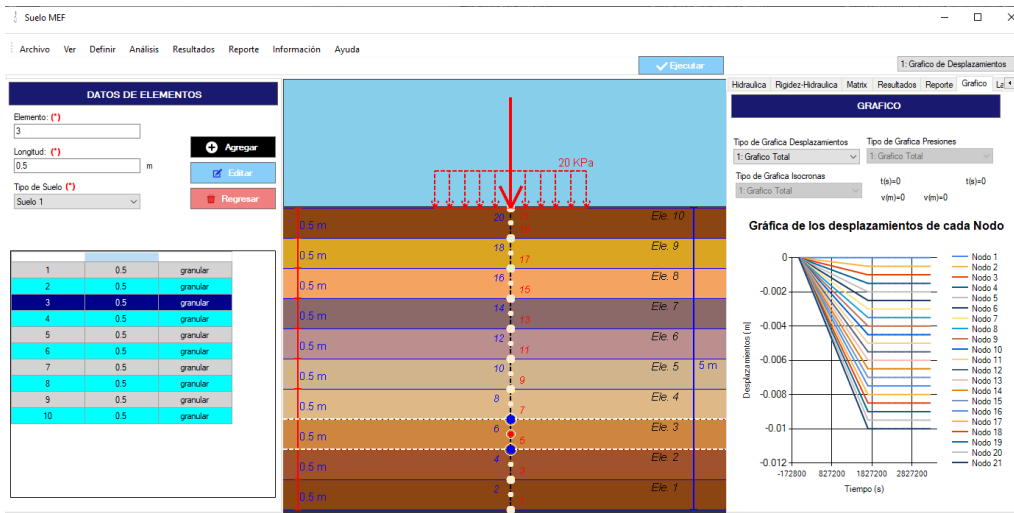
En el siguiente capítulo se presentan los resultados del desarrollo de la herramienta computacional para la simulación numérica del asentamiento de suelos. En la primera parte, se muestra la ejecución de la herramienta desarrollada en el lenguaje de programación C# para calcular los asentamientos y presiones del suelo a través de simulaciones numéricas. En otra sección, se expone la aplicación de la herramienta computacional a suelos granulares y arcillosos para determinar los asentamientos y presiones correspondientes. Finalmente, se realiza la comparación de estos resultados entre ambos modelos analizados.

4.1. DESARROLLO DE LA HERRAMIENTA COMPUTACIONAL PARA DETERMINAR LOS ASENTAMIENTO Y PRESIONES DE SUELOS

Para ingresar a la herramienta informática denominada Suelo MEF, primero se debe realizar la descarga directamente desde la página oficial de Suelo MEF (<https://suelosmef.web.app>). Luego, se procede a instalar el programa. Al iniciar el programa se muestra un modelo de ejemplo.

Figura 7

Iniciando Suelo MEF en su primera versión



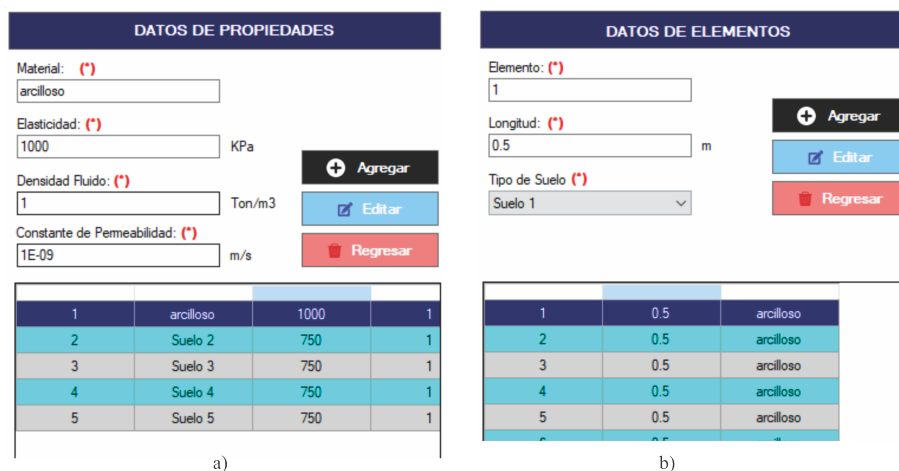
Nota: En la interfaz se muestra un modelo de ejemplo junto con las opciones principales disponibles para realizar simulaciones de asentamientos y presiones del suelo.

4.1.1. Definición de propiedades y elementos del suelos

Aquí se define el tipo de suelos, especificando el módulo de elasticidad, la constante de permeabilidad y la densidad del agua. También se establece la discretización de los elementos.

Figura 8

Creacion de tipo de suelos y elementos



Nota: En figura a) se muestra la interfaz para el ingreso de datos respecto a tipo de suelos y en la figura b) se muestra la interfaz para la discretización.



4.1.2. Asignación de condición de fronteras y cargas

El programa generó automáticamente las condiciones de frontera, aunque estas pueden ser editadas si es necesario. Sin embargo, no se recomendó realizar modificaciones como añadir o eliminar condiciones, ya que esto podría comprometer la precisión del análisis.

4.2. VALIDACIÓN DE LOS VALORES DE ASENTAMIENTOS Y PRESIONES CALCULADOS POR LA HERRAMIENTA INFORMÁTICA

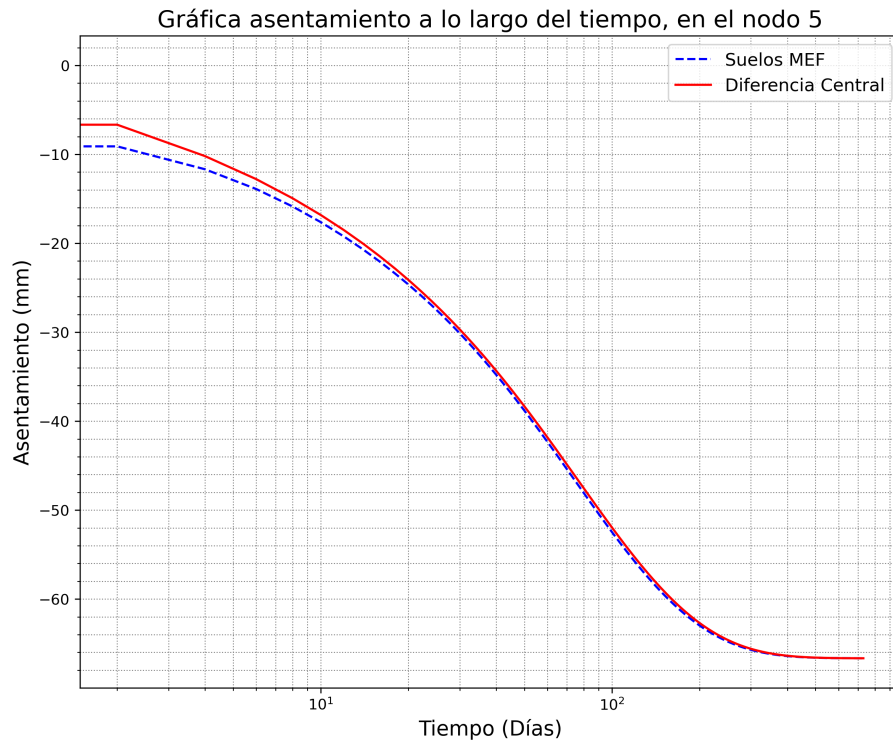
La herramienta informática desarrollada en C# que utiliza como método de cálculo el método de los elementos finitos, fue validada al ser comparado con método de diferencia central, mediante una hoja de cálculo de Excel desarrollada por el Doc. Osmar Carrillo de la Universidad de Wisconsin-Madison, esta hoja fue modificada en la cantidad en la carga y el tiempo de análisis. Los datos utilizados para esta simulación se encuentran especificado en la parte metodológica. Estos mismos datos son utilizados en la herramienta informática para simular el comportamiento del suelo en 2 años. Se obtuvieron los siguientes resultados.

4.2.1. Asentamientos

Ambos métodos, el método de diferencia central y Suelos MEF, muestran una tendencia decreciente de los asentamientos a medida que el tiempo avanza. Este comportamiento es consistente con la teoría de consolidación de suelos, donde los asentamientos son más rápidos al inicio y luego disminuyen gradualmente su ritmo conforme se estabilizan. Esto refleja que ambos métodos capturan adecuadamente el fenómeno esperado en un suelo sometido a carga.

Figura 9

Comparación de asentamiento entre el método diferencia central y la herramienta informática



Nota: En línea entre cortadas se precia los resultados obtenidos con la herramienta informática y en líneas continuas el método de diferencia central. (tiempo es escala logarítmica)

La diferencia entre los valores del método de diferencia central y los de Suelos MEF es mayor al inicio, específicamente en 2 días, con una diferencia de 2.435 mm. A medida que transcurre el tiempo, la diferencia disminuye progresivamente, alcanzando valores mínimos de 0.002 mm al día 646. Este comportamiento sugiere que Los métodos se acercan más conforme los asentamientos se estabilizan. Al inicio, los efectos numéricos y la aproximación del método de diferencia central podrían influir más en la discrepancia.

Al inicio del proceso (día 2), la diferencia entre los resultados de los métodos es más notable, alcanzando un valor máximo de 2.435 mm. Sin embargo, a medida que el tiempo transcurre, la diferencia entre los métodos disminuye progresivamente, llegando a valores muy pequeños al final del análisis, como 0.002 mm en el día 646. Esto sugiere que



las discrepancias son más significativas durante los primeros días, probablemente debido a los efectos numéricos o las aproximaciones inherentes al método de diferencia central, mientras que Suelos MEF logra representar mejor el comportamiento inicial.

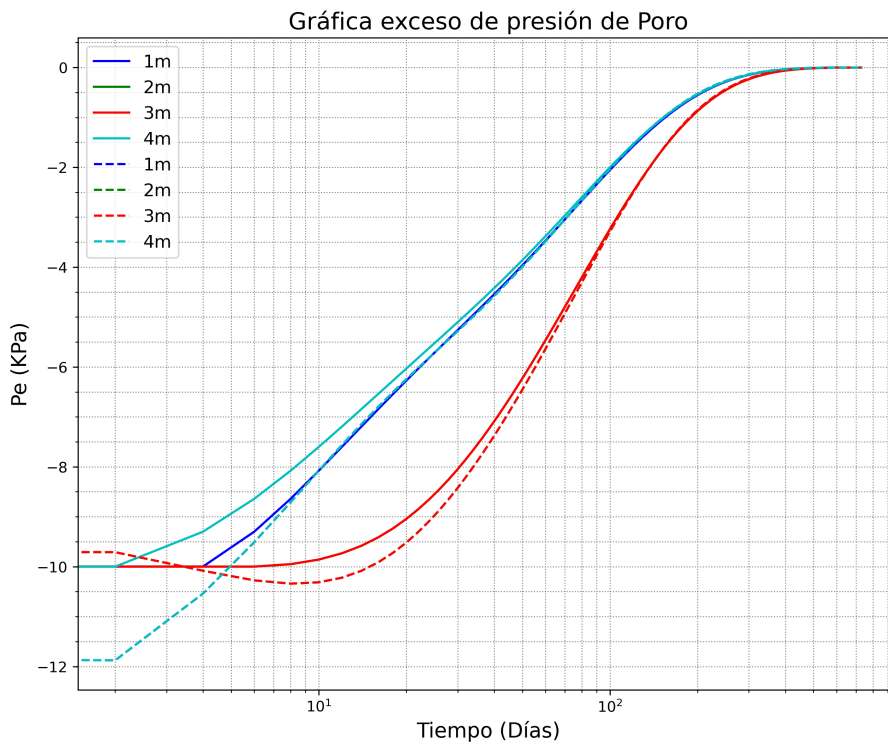
La discrepancia máxima observada entre los métodos ocurre en los primeros días, mientras que, a partir del día 206, las diferencias son menores a 0.3 mm. Esto indica que, a largo plazo, ambos métodos son prácticamente equivalentes en términos de resultados. Sin embargo, el programa Suelos MEF demuestra una mayor precisión, especialmente en los tiempos iniciales, donde los asentamientos son más pronunciados y el comportamiento del suelo es más sensible a las aproximaciones numéricas.

4.2.2. Presiones

A medida que transcurre el tiempo, ambos métodos muestran una disminución progresiva de las presiones (Kpa) a diferentes profundidades (1m, 2m, 3m, 4m). Las presiones iniciales son iguales, con valores de -10 Kpa para todas las profundidades en el tiempo 0. Sin embargo, conforme avanza el tiempo, las presiones tienden a acercarse a valores muy pequeños, cercanos a cero.

Figura 10

Comparación de presiones entre el método diferencia central y la herramienta informática



Nota: En línea entre cortadas se precia los resultados obtenidos con la herramienta informática y en líneas continuas el método de diferencia central. (tiempo es escala logarítmica)

En los tiempos finales (300-696 días), las diferencias entre ambos métodos se vuelven casi despreciables, con valores del orden de 10^{-3} o menores. Tanto el método de diferencia central como el MEF coinciden en valores cercanos a cero al final del análisis.

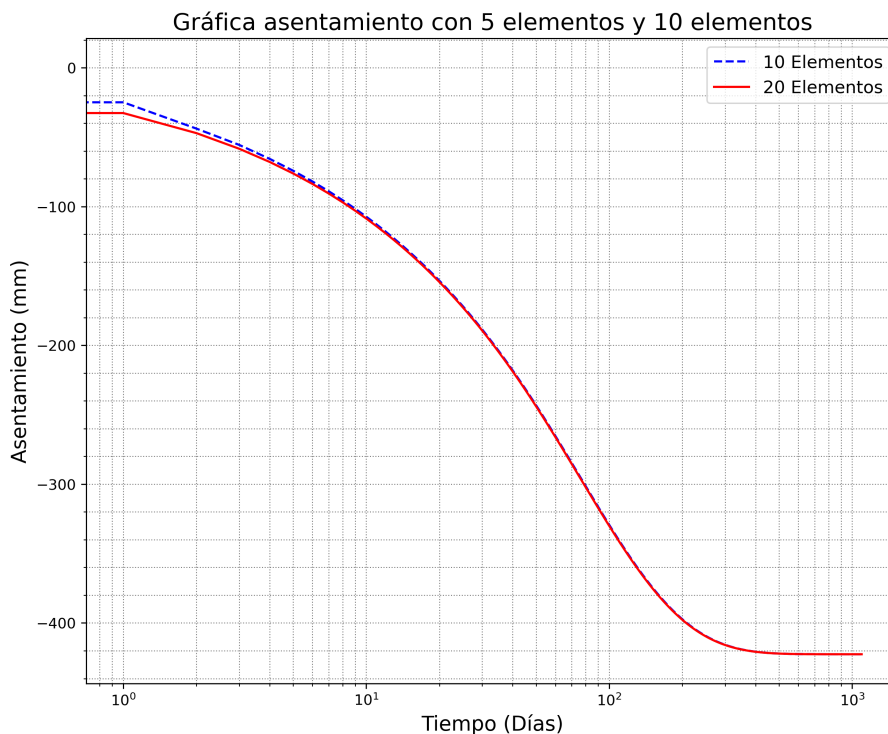
En todas las profundidades analizadas (1m, 2m, 3m, 4m), el comportamiento general es similar. La presión disminuye de forma exponencial con el tiempo, mostrando un patrón claro de pérdida de presión. Sin embargo, las diferencias entre los métodos son ligeramente mayores a profundidades intermedias, como 2m y 3m. A mayores profundidades, los valores tienden a ser un poco más atenuados, lo que sugiere una distribución más uniforme.

4.3. SIMULACIÓN CON 10 Y 20 ELEMENTOS

La simulación de asentamientos a lo largo del tiempo utilizando el método de los elementos finitos (MEF) en Suelos MEF se realizó con dos modelos: uno con 10 elementos y otro con 20 elementos. Los resultados permiten comparar la evolución de los asentamientos y analizar las diferencias entre ambos modelos.

Figura 11

Comparación entre un análisis con 10 elementos y 20 elementos



Nota: En línea entre cortadas representa el modelo con 10 elementos, líneas continuas 20 elementos. (tiempo es escala logarítmica)

En general, los asentamientos muestran una tendencia decreciente conforme avanza el tiempo. Durante los primeros días, las diferencias entre los modelos son más pronunciadas debido a las altas tasas de deformación que ocurren en la etapa de consolidación primaria. En este período inicial, el modelo con 20 elementos presenta mayor precisión en la captura de los asentamientos respecto al modelo con 10 elementos. Por ejemplo, en el día 1, la diferencia entre ambos modelos es de -7.75 mm, siendo el valor



más alto de todo el proceso.

A medida que transcurre el tiempo, las diferencias entre los modelos se reducen significativamente. En el día 4, la diferencia disminuye a -2.22 mm, y para el día 26, se reduce aún más a -0.82 mm. Esto indica que, aunque al inicio las discrepancias son considerables, la evolución del proceso tiende hacia una estabilización donde ambas aproximaciones se acercan gradualmente.

En tiempos más prolongados, las diferencias se vuelven prácticamente insignificantes. Por ejemplo, en el día 463, la diferencia entre ambos modelos es de -0.03 mm, mientras que para el día 1033, es apenas de -0.000037 mm. Esto refleja que, a largo plazo, los modelos con 10 elementos y 20 elementos convergen hacia resultados similares, capturando correctamente el proceso de consolidación secundaria y estabilización de los asentamientos.

El análisis muestra que la discretización juega un papel fundamental en la precisión de los resultados, especialmente en los tiempos iniciales donde las variaciones son más significativas. El modelo con 20 elementos proporciona una mayor resolución y precisión en estas etapas críticas, mientras que el modelo con 10 elementos sigue siendo una aproximación aceptable, especialmente en tiempos más avanzados donde las diferencias son mínimas.

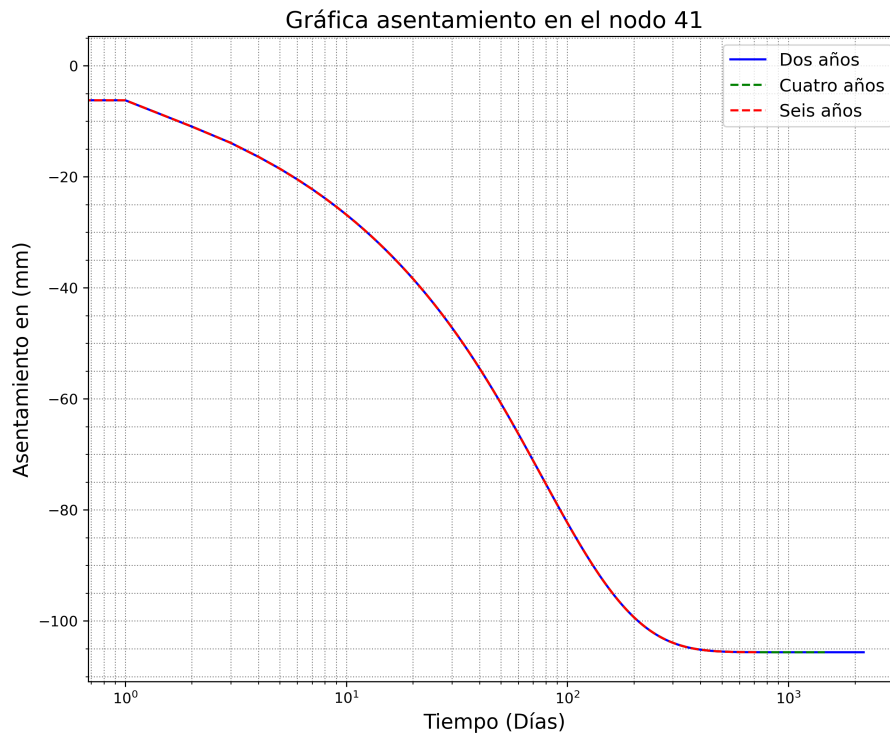
4.4. SIMULACIÓN EN DOS, CUATRO Y SEIS AÑOS

El análisis del asentamiento del suelo debido a una carga de 63.38 KPa se ha realizado considerando períodos de tiempo de dos, cuatro y seis años. Los resultados muestran que, al inicio (día 0), no se observa deformación alguna en el suelo. Sin embargo, con el paso de los días, el asentamiento comienza a manifestarse y presenta una disminución considerable de la superficie, alcanzando los valores más significativos

durante las primeras etapas del análisis.

Figura 12

Comparación entre un análisis con dos, cuatro y seis años



Nota: (tiempo es escala logarítmica)

En los primeros días tras la aplicación de la carga, el asentamiento del suelo es notable, alcanzando -6.229 mm al día 1 y aumentando de forma pronunciada hasta los -56.462 mm al día 43. A partir del día 208, la tasa de cambio disminuye y el asentamiento comienza a estabilizarse, alcanzando aproximadamente -105.196 mm al día 404. Para los períodos de cuatro y seis años, el asentamiento máximo de -105.633 mm se alcanza alrededor de los días 1376 y 2014, respectivamente, mostrando que la deformación inicial es significativa y se reduce progresivamente hasta estabilizarse.

La comparación entre los asentamientos a los 2, 4 y 6 años muestra que el comportamiento es prácticamente idéntico a lo largo del tiempo. En las primeras etapas (0-4 días), no hay diferencia significativa en los valores de asentamiento entre los tres períodos, con un asentamiento inicial de -6.23 mm y -16.41 mm a los 4 días. A medida



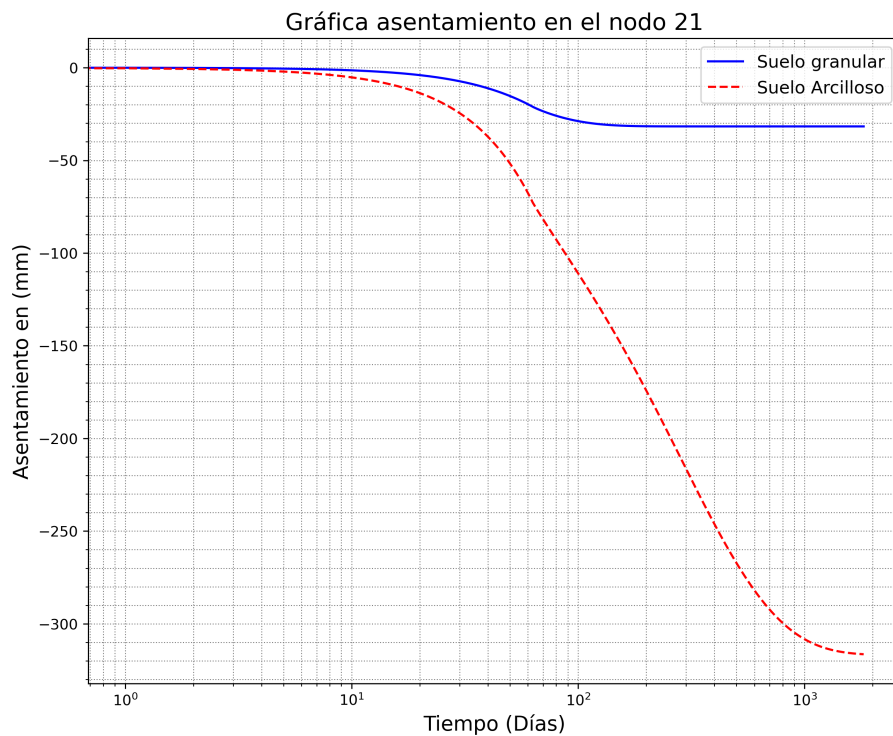
que pasa el tiempo, hasta los 200 días, el asentamiento sigue una tendencia similar, con un valor de aproximadamente -99.96 mm a los 200 días. Después de este período, el asentamiento se estabiliza alrededor de -105.63 mm, y este valor se mantiene constante a lo largo del tiempo. Esto indica que la duración de la carga (2, 4 o 6 años) no afecta de manera significativa el comportamiento del asentamiento, que se estabiliza rápidamente después de los primeros 200 días y permanece constante.

4.5. SIMULACIÓN NUMÉRICA ENTRE UN SUELO GRANULAR Y ARCILLOSO

Se aplicó una herramienta computacional desarrollada para analizar asentamientos en suelos granulares y arcillosos, utilizando el método de elementos finitos. Se modelaron dos tipos de suelo con 10 elementos cada uno, manteniendo la misma altura y cargas, pero variando parámetros como el módulo de elasticidad. Ambos modelos fueron sometidos a una carga creciente hasta un valor máximo de 63.38 kN/m^2 durante un período de 5 años (1825 días), con iteraciones diarias para evaluar su comportamiento. Finalmente, se realizó una comparación entre los modelos para analizar las diferencias en los asentamientos.

Figura 13

Comparación entre un un suelos granular y arcillo



Nota: (tiempo es escala logarítmica)

En los primeros días del análisis, el suelo granular mostró desplazamientos muy pequeños, con un valor inicial de -0.051 mm; lo cual es típico de su estructura, que permite que las partículas grandes y resistentes distribuyan de manera eficiente la carga aplicada. Por otro lado, el suelo arcilloso presentó un comportamiento distinto, con desplazamientos más significativos desde el inicio (-0.318 mm), debido a su mayor compresibilidad y la estructura cohesiva que lo hace más susceptible a deformarse bajo pequeñas cargas.

A medida que avanzaba el tiempo, el suelo granular experimentó un aumento moderado en los desplazamientos, estabilizándose en -31.69 mm al final del análisis. Esto demuestra que después el suelo alcanzó un equilibrio, limitando deformaciones adicionales. por otro lado, el suelo arcilloso continuó con un incremento pronunciado de desplazamientos, alcanzando un valor de -316.19 mm al final, lo cual es característico del proceso de consolidación donde la expulsión de agua de los poros provoca deformaciones



acumulativas, la diferencia en los desplazamientos entre ambos tipos de suelo se amplió constantemente, pasando de 0 mm al principio a 284.5 mm al final del estudio.

Es importante mencionar que para la simulación de los suelos arcillosos y granulares, especialmente los granulares, se realizaron varias pruebas. Por esta razón, se seleccionaron los valores más bajos para el módulo de elasticidad y el coeficiente de conductividad hidráulica, y se analizaron los suelos en condiciones completamente saturadas. Esto se debe a que los suelos granulares, que son considerados adecuados para la construcción de estructuras, presentan un asentamiento instantáneo en comparación con los suelos arcillosos, presentado asentamiento más lento y dependiente del tiempo.

4.6. DISCUSIÓN

La simulación numérica de la consolidación de suelos constituye un componente principal en la ingeniería geotécnica para el diseño, análisis y evaluación de cimentaciones y terraplenes. El desarrollo de una herramienta computacional en C# que emplea el método de los elementos finitos (FEM) para modelar o simular la consolidación lineal de suelos, tanto en condiciones drenadas y no drenadas, representa un avance importante en la capacidad de realizar simulaciones precisas y eficientes. En el estudio realizado por Bedon (2022) y García (2016) se muestra y destaca que el método de los elementos finitos ha sido exitosamente utilizado y desarrollado para modelar procesos de consolidación en suelos, en especial en suelos arcillosos. En este sentido, la herramienta desarrollada en este estudio sirve como una solución fiable para la simulación de procesos de consolidación, con la capacidad de simular diferentes tipos de suelos y condiciones de drenaje.

El uso de FEM en la herramienta es particularmente relevante debido a su capacidad para manejar configuraciones complejas, como suelos heterogéneos y multicapa. En el estudio de de García (2016), analizó suelos anisotrópicos utilizando



FEM. De este modo, el objetivo principal se alcanzó exitosamente, ya que la herramienta simula con precisión la consolidación de suelos bajo condiciones drenadas y no drenadas. Esta funcionalidad proporciona a los ingenieros una herramienta robusta para predecir asentamientos y presiones en una amplia variedad de escenarios geotécnicos, mejorando la exactitud de los análisis y el diseño.

La formulación matricial del método de los elementos finitos, representan un avance esencial para modelar el comportamiento de suelos en condiciones o procesos de consolidación. La herramienta desarrollada en el estudio integra la formulación matricial basada en elementos lineales en concordancia con las metodologías comúnmente empleadas en la simulación de consolidación de suelos. En estudios previamente realizados, como el trabajo de Falconi y Mroginski (2010), han abordado la formulación de matrices de rigidez para elementos finitos, demostrando su efectividad para modelar el comportamiento de suelos saturados en dos dimensiones. Asemejándose, a la herramienta implementada la cual usa una formulación matricial eficiente que proporciona resultados confiables para la simulación de la consolidación bajo condiciones drenadas como no drenadas, alineándose con las formulaciones planteadas por Di (2006) y otros estudios los cuales emplean y destacan la aplicabilidad de FEM en la geotecnia.

El diseño de la herramienta computacional en C# para determinar evaluar asentamientos y presiones en suelos mediante simulación numérica se alinea o en sintonía con los avances realizados por otros autores en este campo. En el estudio realizado por Reinert et al. (2014) utilizaron el método de los elementos finitos (FEM) para estudiar el comportamiento de suelos compresibles bajo cargas, simulando el comportamiento de asentamientos y presiones. De igual manera la herramienta desarrollada en este estudio no solo simula la consolidación de suelos, sino que también permite calcular de manera precisa los asentamientos y las presiones intersticiales, lo que la convierte en una



herramienta valiosa para la ingeniería geotécnica en el análisis de suelos bajo cargas.

En cuanto, a la simulación de asentamientos y presiones forma parte de un aspecto es crucial para el diseño de cimentaciones, la herramienta cumple con esta finalidad al proporcionar resultados fiables que faciliten a los ingenieros en cuanto a la evaluación del comportamiento de los suelos ante diferentes condiciones de carga. Así también, la implementación en C# garantiza un procesamiento rápido y una interfaz de usuario accesible, evidenciando la facilidad de su uso en aplicaciones prácticas.

La capacidad de la herramienta para analizar suelos granulares y arcillosos resulta crucial para validar su efectividad en la simulación de consolidación en diferentes tipos de suelo. Es así, que en el estudio desarrollado por Montaña y Villa (2018) sobre la licuación de arenas utilizando FEM demostraron que al realizar las simulaciones numéricas se puede proporcionar información útil relacionada a el comportamiento de suelos arenosos bajo cargas. De manera similar, la herramienta desarrollada en este trabajo ha sido aplicada exitosamente a suelos granulares y arcillosos, mostrando resultados consistentes con las predicciones teóricas en términos de asentamientos y presiones.

En lo que respecta, a la validación de la herramienta con suelos de diferentes características, como suelos granulares y arcillosos, resulta relevante puesto que es un paso significativo para garantizar su versatilidad y aplicabilidad en una amplia gama de escenarios geotécnicos. Los resultados obtenidos mediante la simulación numérica evidenciaron una buena concordancia con las teorías y modelos existentes, evidenciando un respaldo en cuanto a la fiabilidad de la herramienta desarrollada en este estudio.



V. CONCLUSIONES

El desarrollada de la herramienta computacional en el lenguaje C# ha sido óptimo para realizar simular los asentamientos de los suelos mediante el método de elementos finitos ha logrado cumplir sus objetivos propuestos. La definición de las ecuaciones matricial de los elementos finitos, combinada con la capacidad para calcular los asentamientos y las presiones, por lo tanto, se cumplió con los objetivos propuestos. Además, su aplicación de la herramienta computacional en los suelos granulares y arcillosos ha demostrado su utilidad y precisión en el análisis. Este trabajo contribuye al avance de las herramientas disponibles para los ingenieros, ofreciendo una solución computacional avanzada para simular los procesos de consolidación de suelos.

PRIMERA: Ha sido posible la definición de las ecuaciones matricial del método de los elementos finitos (MEF) con elementos lineales de 3 nodos proporciona una base sólida para la simulación numérica del comportamiento de los suelos. Así como el algebra matricial, de esta forma se facilita bastante la programación de los cálculos necesarios para representar el comportamiento del suelo bajo distintas condiciones de carga.

SEGUNDA: Con el desarrollo de la herramienta computacional en C# se ha logrado cumplir con el objetivo planteado, demostrado que se puede calcular de forma eficiente los asentamientos en suelos mediante simulaciones numéricas. Esto facilita el análisis de problemas geotécnicos complejos, y ofrece una solución accesible tanto para ingenieros como para estudiantes. La validación de la herramienta computacional realizó mediante el método de diferencia central, realizando simulaciones con 10 y 20 elementos y considerando períodos de 2, 4 y 6 años, así como diversas propiedades del material.



TERCERA: Los suelos más granulares mostraron desplazamientos iniciales mínimos (-0.051 mm) que se estabilizaron en -31.69 mm, mientras que los suelos arcillosos presentaron desplazamientos más grandes (-0.318 mm) que alcanzaron -316.19 mm, debido a su mayor compresibilidad. La diferencia en los desplazamientos fue de 0 mm a 284.5 mm, lo que muestra una mayor deformación en los suelos arcillosos. Para la simulación se utilizaron valores bajos para el módulo de elasticidad y el coeficiente de conductividad hidráulica, considerando que los suelos estaban saturados, ya que los suelos granulares tienen un asentamiento más rápido que los arcillosos.



VI. RECOMENDACIONES

- La herramienta desarrollada es de fácil uso, pero aun así se recomienda tener cuidado con el ingreso de datos a la herramienta computacional. Esto garantizará una óptima simulación del modelo. Datos coherentes permitirán interpretar adecuadamente los resultados obtenidos.
- La herramienta presenta limitaciones en el análisis de elementos bidimensionales; porque se recomienda que en futura investigación se optimice la capacidad de cálculo mediante el método de elementos finitos, además de mejorar la interfaz gráfica para el diseño de elementos. También sería beneficioso incorporar la posibilidad de realizar los cálculos directamente desde un navegador Web.
- Por otro lado, se recomienda aplicar herramientas computacionales en la ingeniería de suelos para determinar asentamientos y presiones, permitiendo análisis más precisos y detallados. Es fundamental usar software especializado que contemple métodos reconocidos y combinar los resultados con datos de campo para garantizar y validar este método.
- Se recomienda que los profesionales relacionados con la mecánica de suelos realicen investigaciones utilizando esta herramienta computacional y comparen esos resultados con otros métodos. Y vea más a profundidad los asentamientos en suelos granulares y arcillosos.
- Finalmente, cabe mencionar que C# puede conectarse a controladores como Arduino, ESP8266 y ESP32, a los cuales se pueden conectar todo tipo de sensores. Dicho esto, se recomienda implementar un sistema de monitoreo de estructuras, suelos, meteorología, entre otros, y comparar los resultados obtenidos con los proyectados.



VII. REFERENCIAS

- Bedon, A. (2022). Estudio sobre el comportamiento de consolidación de arcillas blandas en la construcción secuencial de un terraplén [Tesis de pregrado, Pontificia Universidad Católica del Perú]. <https://bit.ly/3RB4oUM>
- Albahari, J., & Albahari, B. (2022). C# 10 in a nutshell: The definitive reference. O'Reilly Media.
- ASTM. (2011). Standard test methods for one-dimensional consolidation properties of soils using incremental loading (ASTM D2435/D2435M-11). ASTM International.
- Atterberg, A. (1911). Die Plastizität der Tone. Internationale Mitteilungen für Bodenkunde, 1(4), 10-43.
- Camejo, J. y Rodríguez, Y. (2016). Diseño de un material didáctico para la introducción al análisis estructural mediante el método de elementos finitos adaptado a la unidad curricular de estructuras avanzadas [Tesis de pregrado, Universidad de Carabobo]. <https://renati.sunedu.gob.pe/bitstream/sunedu/744906/1/RodriguezTrilloYuletsy.pdf>
- Carrasco, S. (2005). Metodología de la investigación científica. San Marcos, 4° ed
- Chen, Z., Wei, Y., & Li, X. (2017). Simulation of soil-structure interaction using finite element methods. Geotechnical Journal, 34(2), 145-158.
- Colömë, G. y Femenia, P. (2018). Metodología de investigación para cursos de posgrado en ingeniería. Universidad Nacional de San Juan.
- Cook, R. D., Malkus, D. S., Plesha, M. E., & Witt, R. J. (2002). Concepts and applications of finite element analysis (4th ed.). John Wiley & Sons.
- Das, B. M. (2015). Principles of geotechnical engineering (8th ed.). Cengage Learning.
- Di, H. (1997). Simulación numérica de procesos de consolidación de suelos usado el método de los elementos finitos [Tesis de Maestría, Universidad Nacional del Nordeste]. <https://bit.ly/3BotMX9>
- Di, H. (2006). Simulación numérica de problemas con no linealidad física y geométrica. Análisis de consolidación de suelos no saturados. [Tesis Doctoral, Universidad



Nacional del Nordeste]. <https://bit.ly/3BudIUQ>

Falconi, R. y Mroginski, J. (2010). Matriz de rigidez para un elemento finito de suelo saturado en dos dimensiones. *Revista Ciencia*, 13(2), 119-140. <https://bit.ly/3XtDgKU>

García, G. (2016). Caracterización Adimensional y Simulación Numérica de Procesos Lineales y No Lineales de Consolidación de Suelos [Tesis Doctoral, Universidad Politécnica de Cartagena]. <https://bit.ly/3qKcoHr>

García, J. (2019). Diseño de un sistema simulador físico de flujo multifásico para reproducir condiciones de flujo semejantes a las reales [Tesis de Maestría, CIATEO]. <https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/389/1/GarciaMontejoJoseL>

Hernández, A. y etl (2018). Metodología de la investigación científica. Área de innovación y desarrollo. Hernández, R., Fernández, C. y Baptista, P. (2006). Metodología de la investigación. McGraw - Hill Interamericana, 4° ed. Montaña, D. y Villa, V. (2018). Simulación numérica de la licuación de arenas mediante un programa de elementos finitos [Tesis de pregrado, Universidad Militar Nueva Granada]. <https://bit.ly/3RNGpkY>

Muñoz, S., Sánchez W. y García, J. (2020). Teoría de consolidación, una revisión de la literatura. *Revista Ingeniería: Ciencia, Tecnología e Innovación*, 7(2), 121-135. <https://bit.ly/3TWB7V3>

Osorio, M. (2015). Desarrollo de simulador interactivo de física enfocado al programa de estudio de ingeniería propuesto por la SENESCYT (desarrollo de los capítulos vectores y movimiento) [Tesis de pregrado, Universidad de Guayaquil]. <http://repositorio.ug.edu.ec/bitstream/redug/10188/1/PTG-582-Osorio>

Potts, D. M., & Zdravković, L. (2001). Finite element analysis in geotechnical engineering: Theory. Thomas Telford.

Reinert, H., Duarte, J. y Silva, M. (2014). Modelación numérica de plateas de cimentación sobre suelos compresibles de misiones, argentina. *Asociación Argentina de Mecánica Computacional*, 33(16), 1085-1099. <https://bit.ly/3OzOREc>

Tristancho, J. (2017). Modelamiento físico en geotecnia aplicado a la docencia. *Revista*



Educación en Ingeniería, 12(23), 109-114. <https://bit.ly/3VpDW1J>

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2005). *The finite element method: Its basis and fundamentals* (6th ed.). Butterworth-Heinemann.

ANEXOS

ANEXO 1: Definición de las matrices

Definición de las matrices

Definir las ecuaciones matriciales del método de los elementos finitos para determinar el asentamiento de suelos

Plantear la formulación matricial del método de los elementos finitos, con elementos lineales, para la simulación numérica del comportamiento de suelos drenados y no drenados. Partimo de la formulacion debil del método de los elementos finitos para elemento lineales.

Matriz de rigidez

Se establece por la formulación débil que la matriz de rigidez:

$$K_e = E \int_0^{He} B^{vT} \cdot B^v \cdot dy$$

También se sabe que la matriz de normalización de los desplazamientos:

$$N^v = \frac{2}{He^2} \begin{bmatrix} (y - y_2)(y - y_3) & -2(y - y_1)(y - y_3) & (y - y_2)(y - y_1) \end{bmatrix}$$

De la formulación débil podemos afirmar que, B^v es la derivada de N^v con respecto a y .

$$B^v = \frac{2}{He^2} \begin{bmatrix} 2y - y_2 - y_3 & -4y + 2y_1 + 2y_3 & 2y - y_1 - y_2 \end{bmatrix}$$

Remplazando B^v en K^e se tendrá:

$$K^e = E \int_0^{He} \frac{2}{He^2} \begin{bmatrix} 2y - y_2 - y_3 \\ -4y + 2y_1 + 2y_3 \\ 2y - y_1 - y_2 \end{bmatrix} \frac{2}{He^2} \cdot \begin{bmatrix} 2y - y_2 - y_3 & -4y + 2y_1 + 2y_3 & 2y - y_1 - y_2 \end{bmatrix} dy$$

Se conoce los valores de $y_1 = 0$, $y_2 = \frac{He}{2}$ y $y_3 = He$ para el elemento analizado, entonces se obtiene:

$$K^e = E \int_0^{\frac{He}{2}} \frac{2}{He^2} \begin{bmatrix} 2y - \frac{3 \cdot He}{2} \\ -4y + 2He \\ 2y - \frac{He}{2} \end{bmatrix} \frac{2}{He^2} \cdot \begin{bmatrix} 2y - \frac{3 \cdot He}{2} & -4y + 2He & 2y - \frac{He}{2} \end{bmatrix} dy$$

Multiplicado las matrices

$$K^e = E \int_0^{\frac{He}{2}} \frac{2}{He^2} \begin{bmatrix} (2y - \frac{3 \cdot He}{2})^2 & (2y - \frac{3 \cdot He}{2})(-4y + 2He) & (2y - \frac{3 \cdot He}{2})(2y - \frac{He}{2}) \\ (2y - \frac{3 \cdot He}{2})(-4y + 2He) & (-4y + 2He)^2 & (-4y + 2He)(2y - \frac{He}{2}) \\ (2y - \frac{3 \cdot He}{2})(2y - \frac{He}{2}) & (-4y + 2He)(2y - \frac{He}{2}) & (2y - \frac{He}{2})^2 \end{bmatrix} dy$$

Resolviendo la integral y evaluando desde $y = 0$ hasta $y = He$.

$$K^e = \frac{4 \cdot E}{12 \cdot He^4} \begin{bmatrix} \frac{7 \cdot He^3}{12} & \frac{-8 \cdot He^3}{12} & \frac{He^3}{12} \\ \frac{-8 \cdot He^3}{12} & \frac{16 \cdot He^3}{12} & \frac{-8 \cdot He^3}{12} \\ \frac{He^3}{12} & \frac{-8 \cdot He^3}{12} & \frac{7 \cdot He^3}{12} \end{bmatrix} K^e = \frac{4 \cdot E}{12 \cdot He} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}$$

Finalmente se obtiene la matriz de rigidez del elemento.

$$K^e = \frac{E}{3 \cdot He} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}$$

Donde:

E : Modulo de elasticidad del suelo, en el elemento analizado.

He : Altura del elemento.

Matriz de hidraulica

Análogamente a la matriz de rigidez, se establece por la formulación débil que la matriz de presiones:



$$H^e = \frac{k}{\rho_w \cdot g} \int_0^{He} B^{PT} \cdot B^P dy$$

También se sabe que la matriz de normalización de presiones es:

$$N^P = \frac{1}{He} \begin{bmatrix} y_2 - y & y - y_1 \end{bmatrix}$$

De la formulación débil podemos afirmar que, B^P es la derivada de N^P con respecto a y .

$$B^P = \frac{1}{He} \begin{bmatrix} -1 & 1 \end{bmatrix}$$

Remplazando B^P en H^e se tendrá:

$$H^e = \frac{k}{\rho_w \cdot g} \int_0^{He} \frac{1}{He} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \cdot \frac{1}{He} \begin{bmatrix} -1 & 1 \end{bmatrix} dy$$

Multiplicado las matrices

$$H^e = \frac{k}{\rho_w \cdot g \cdot He^2} \int_0^{He} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} dy$$

Resolviendo la integral y evaluando desde $y = 0$ hasta $y = He$.

$$H^e = \frac{k}{\rho_w \cdot g \cdot He^2} \int_0^{He} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} dy$$

Finalmente se obtiene la matriz de rigidez del elemento.

$$H^e = \frac{k}{\rho_w \cdot g \cdot He} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Donde:

k : Modulo de elasticidad del suelo, en el elemento analizado.

ρ_w : Constante de filtración del agua.

g : Gravedad terrestre

He : Altura del elemento.

Matriz que relaciona los desplazamientos verticales y presiones

Análogamente a la matriz de rigidez, se establece por la formulación débil que la matriz que relaciona los desplazamientos y presiones:

$$Q^e = \int_0^{He} B^{vT} \cdot N^P dy$$

También se sabe que la matriz de normalización de los desplazamientos es:

$$N^v = \frac{2}{He^2} \begin{bmatrix} (y - y_2)(y - y_3) & -2(y - y_1)(y - y_3) & (y - y_2)(y - y_1) \end{bmatrix}$$

Así como también la matriz de normalización de las presiones:

$$N^P = \frac{1}{He} \begin{bmatrix} y_2 - y & y - y_1 \end{bmatrix}$$

De la formulación débil podemos afirmar que, B^v es la derivada de N^v con respecto a y .

$$B^v = \frac{2}{He^2} \begin{bmatrix} 2y - y_2 - y_3 & -4y + 2y_1 + 2y_3 & 2y - y_1 - y_2 \end{bmatrix}$$

Remplazando B^P y N^P en Q^e se tendrá que:

$$Q^e = \int_0^{He} \frac{2}{He^2} \begin{bmatrix} 2y - \frac{3 \cdot He}{2} \\ -4y + 2He \\ 2y - \frac{He}{2} \end{bmatrix} \cdot \frac{1}{He} \begin{bmatrix} He - y & y \end{bmatrix} dy$$

Multiplicado las matrices

$$Q^e = \frac{2}{He^3} \int_0^{He} \begin{bmatrix} (2y - \frac{3 \cdot He}{2})(He - y) & (2y - \frac{3 \cdot He}{2}) \cdot y \\ (-4y + 2He)(He - y) & (-4y + 2He) \cdot y \\ (2y - \frac{He}{2})(He - y) & (2y - \frac{He}{2}) \cdot y \end{bmatrix} dy$$

Resolviendo la integral y evaluando desde $y = 0$ hasta $y = He$. $Q^e = \frac{1}{6} \begin{bmatrix} -5 & -1 \\ 4 & -4 \\ 1 & 5 \end{bmatrix}$ Se

aprecia que la matriz Q^e es una matriz constante, donde las filas están relacionada con los

desplazamientos y las columnas están relacionadas con las presiones.

Vectores de fuerzas

$$f e^v = \frac{-\gamma_s \cdot H e}{6} \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} \quad f e^P = k \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Solución del sistema matricial por equivalencia incremental

Partimos de la ecuación matricial

$$\begin{bmatrix} 0 & 0 \\ Q^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{v} \\ \dot{P}_e \end{bmatrix} + \begin{bmatrix} K & Q \\ 0 & -H \end{bmatrix} \cdot \begin{bmatrix} v \\ P \end{bmatrix} = \begin{bmatrix} f^\sigma + f^v \\ f^w + f^P \end{bmatrix}$$

Donde:

K : Matriz de rigidez global

H : Matriz hidráulica global

Q : Matriz que relaciona los desplazamientos con desplazamientos.

v : Desplazamiento vertical

P : Presión

\dot{v} : Velocidad

\dot{P}_e : Velocidad de la presión por exceso

Como se puede apreciar se tiene dos matrices, las cuales complica el proceso de solución del sistema. Es por ellos que se adecua a un sistema equivalente incremental.

$$K \cdot v + Q \cdot P = f^\sigma + f^v \quad Q^T \cdot \dot{v} - H \cdot P = f^w + f^P$$

Se sabe que la variación de la velocidad esta denotada por:

$$\Delta \dot{v} = \dot{v}_{t+\Delta t} - \dot{v}_t \quad v_{t+\Delta t} = v_t + \Delta v \quad v_{t+\Delta t} = v_t + \dot{v}_{prom} \cdot \Delta t$$

Ponderamos la velocidad promedio por θ , tomaremos el valor de $\theta = \frac{1}{2}$.



$$\dot{v}_{prom} = \theta \cdot \dot{v}_{t+\Delta t} + (1 - \theta) \cdot \dot{v}_t$$

Por otro lado, tenemos que la velocidad promedio es:

$$\dot{v}_{prom} = \frac{v_{t+\Delta} - v_t}{\Delta t}$$

Igualando la velocidad promedio.

$$\theta \cdot \dot{v}_{t+\Delta t} + (1 - \theta) \cdot \dot{v}_t = \frac{\Delta v}{\Delta t} \quad \theta \cdot \dot{v}_{t+\Delta t} = \frac{\Delta v}{\Delta t} - \dot{v}_t + \theta \cdot \dot{v}_t \quad \dot{v}_{t+\Delta t} = \frac{1}{\theta} \cdot \frac{\Delta v}{\Delta t} - \frac{1}{\theta} \cdot \dot{v}_t + \dot{v}_t$$

Remplazamos $\dot{v}_{t+\Delta t}$ en $\Delta \dot{v} = \dot{v}_{t+\Delta t} - \dot{v}_t$

$$\Delta \dot{v} = \left(\frac{1}{\theta} \cdot \frac{\Delta v}{\Delta t} - \frac{1}{\theta} \cdot \dot{v}_t + \dot{v}_t \right) - \dot{v}_t \quad \Delta \dot{v} = \frac{1}{\theta} \cdot \frac{\Delta v}{\Delta t} - \frac{1}{\theta} \dot{v}_t$$

Partimos de la ecuación lineal

$$K \cdot v + Q \cdot P = f^\sigma + f^v$$

Evaluaremos para $t + \Delta t$ y t

$$K \cdot v_t + Q \cdot P_t = f^\sigma_t + f^v_t \quad K \cdot \Delta v + Q \cdot \Delta P = \Delta f^\sigma$$

De la diferencia entre ambas podemos decir que.

$$K \cdot \Delta v + Q \cdot \Delta P = \Delta f^\sigma$$

Por otro lado, usando la otra ecuación lineal.

$$Q^T \cdot \Delta \dot{v} - H \cdot \Delta P = \Delta f^P$$

Remplazamos $\Delta \dot{v}$ en la ecuación

$$Q^T \cdot \left(\frac{1}{\theta} \cdot \frac{\Delta v}{\Delta t} - \frac{1}{\theta} \dot{v}_t \right) - H \cdot \Delta P = 0 \quad Q^T \cdot \left(\frac{1}{\theta} \cdot \frac{\Delta v}{\Delta t} - \frac{1}{\theta} \dot{v}_t \right) - H \cdot \Delta P = 0$$

$$\frac{1}{\theta \cdot \Delta t} \cdot Q^T \cdot \Delta v - H \cdot \Delta P = \frac{1}{\theta} \cdot Q^T \cdot \dot{v}_t$$

Finalmente, de las dos ecuaciones obtenidas podemos expresarla de forma matricial:



$$K \cdot \Delta v + Q \cdot \Delta P = \Delta f^\sigma \frac{1}{\theta \cdot \Delta t} \cdot Q^T \cdot \Delta v - H \cdot \Delta P = \frac{1}{\theta} \cdot Q^T \cdot \dot{v}_t$$

Forma matricial modificada para resolver el sistema.

$$\begin{bmatrix} K & \frac{1}{\theta \cdot \Delta t} \cdot Q \\ \frac{1}{\theta \cdot \Delta t} \cdot Q^T & -\frac{1}{\theta \cdot \Delta t} \cdot H \end{bmatrix} \cdot \begin{bmatrix} \Delta v \\ \theta \cdot \Delta t \cdot \Delta P \end{bmatrix} = \begin{bmatrix} \Delta f^\sigma \\ \frac{1}{\theta} \cdot Q^T \dot{v}_t \end{bmatrix}$$

ANEXO 2: Método de la diferencia central

Método de la diferencia central

Usando las series de Taylor, para aproximar la derivada de segundo orden:

$$f(x + \Delta x) = f(x) + \Delta x \cdot \frac{\partial f(x)}{\partial x} + \frac{\Delta x^2}{2!} \cdot \frac{\partial^2 f(x)}{\partial x^2} + \dots$$

$$f(x - \Delta x) = f(x) - \Delta x \cdot \frac{\partial f(x)}{\partial x} + \frac{\Delta x^2}{2!} \cdot \frac{\partial^2 f(x)}{\partial x^2} + \dots$$

De la expansión de la serie de Taylor tomaremos tres términos:

$$f(x + \Delta x) = f(x) + \Delta x \cdot \frac{\partial f(x)}{\partial x} + \frac{\Delta x^2}{2!} \cdot \frac{\partial^2 f(x)}{\partial x^2}$$

$$f(x - \Delta x) = f(x) - \Delta x \cdot \frac{\partial f(x)}{\partial x} + \frac{\Delta x^2}{2!} \cdot \frac{\partial^2 f(x)}{\partial x^2}$$

De la adición de ambas ecuaciones se mostrará:

$$f(x + \Delta x) + f(x - \Delta x) = 2 \cdot f(x) + \Delta x^2 \cdot \frac{\partial^2 f(x)}{\partial x^2}$$

Entonces, despejando la ecuación diferencial de segundo orden:

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{f(x+\Delta x) - 2 \cdot f(x) + f(x-\Delta x)}{\Delta x^2}$$

Esta ecuación permite obtener la ecuación diferencial de segundo orden.

ANEXO 3: Calculo de cantidad de carga

Calculo de cantidad de carga

Para estima la carga que produce una edificación de cuatro niveles sobre un suelo, expresada en kN/m^2 , utilizando parámetros típicos de la región de Puno y contemplado las disposiciones del Reglamento Nacional de Edificaciones (RNE). Se considera una altura de 2.65 m por nivel.

1. Determinación de Cargas por Nivel

a) Carga muerta (w_m)

La carga muerta incluye el peso propio de los elementos estructurales y no estructurales:

- **Peso de la losa:**

$$w_{\text{losa}} = \text{espesor de la losa} \times \gamma_{\text{concreto}}$$

$$w_{\text{losa}} = 0.15 \text{ m} \times 25 \text{ kN/m}^3 = 3.75 \text{ kN/m}^2$$

- **Peso de las paredes:** Suponemos paredes perimetrales de 0.15 m de espesor y 2.65 m de altura:

$$w_{\text{pared}} = \text{espesor} \times \gamma_{\text{albañilería}} \times \text{altura}$$

$$w_{\text{pared}} = 0.15 \text{ m} \times 18 \text{ kN/m}^3 \times 2.65 \text{ m} = 7.155 \text{ kN/m}^2$$

- **Peso de los acabados:** Se considera:

$$w_{\text{acabado}} = 1.5 \text{ kN/m}^2$$



b) Carga viva (w_v)

La carga viva depende del uso de la edificación. Para un uso residencial, el RNE E.020 indica:

$$w_v = 2.0 \text{ kN/m}^2$$

c) Carga total por nivel

Sumamos todas las cargas por nivel:

$$w_{\text{nivel}} = w_{\text{losa}} + w_{\text{pared}} + w_{\text{acabado}} + w_v$$

$$w_{\text{nivel}} = 3.75 + 7.155 + 1.5 + 2.0 = 14.405 \text{ kN/m}^2$$

2. Carga Total Generada por los Cuatro Niveles

a) Sin considerar cimentación

La carga generada por los cuatro niveles es:

$$w_{\text{total}} = w_{\text{nivel}} \times \text{número de niveles}$$

$$w_{\text{total}} = 14.405 \text{ kN/m}^2 \times 4 = 57.62 \text{ kN/m}^2$$

b) Considerando un 10 % adicional para cimentación

Para incluir un 10 % adicional por cimentación:

$$w_{\text{total, final}} = w_{\text{total}} \times 1.1$$



$$w_{\text{total, final}} = 57.62 \times 1.1 = 63.38 \text{ kN/m}^2$$

3. Resultado Final

La carga aproximada generada por la edificación de cuatro niveles, con una altura de 2.65 m por nivel, es:

$$63.38 \text{ kN/m}^2$$



ANEXO 4: Estructura de un Programa en C#

Estructura de un Programa en C#

Procedimientos en C#

Codigo A.1

Procedimientos en C#

```
1 public int ValorNumerico (int numA, int numB) {  
2     Int c = numA * numB;  
3     return c;  
4 }
```

Tipos de Datos Comunes en C#

Codigo A.2

Declaración de Variables y Tipos de Datos

```
1 int MiEdad = 29;  
2 double MiAltura = 1.71;  
3 string MiNombre = "Denis";  
4 bool SoyEstudiante = false;  
5 char Miinicial = 'D';
```

Condicional if-else

Codigo A.3

Condicional if-else

```
1 int Miedad = 20;  
2 if (Miedad >= 18) {  
3     Console.WriteLine("Eres mayor de Edad :::");  
4 } else {  
5     Console.WriteLine("Es menor de Edad :::");  
6 }
```

Estructura switch

Codigo A.4

Estructura switch

```
1 int opcion = 2;  
2 switch (opcion){  
3     case 1:  
4         Console.WriteLine("Opcion 1 seleccionada");  
5         break;
```




```
6     case 2:  
7     Console.WriteLine("Opcion 2 seleccionada");  
8     break;  
9     default:  
10    Console.WriteLine("Opcion no valida");  
11    break;  
12 }
```

Bucles for

Codigo A.5

Bucles for

```
1     for (int i = 0; i < 5; i++) {  
2         Console.WriteLine("Valor de i: " + i);  
3     }
```

Bucle foreach

Codigo A.6

Bucle foreach

```
1     string[] frutas = { "Manzana", "Banana", "Cereza" };  
2     foreach (string fruta in frutas) {  
3         Console.WriteLine(fruta);  
4     }
```

Matriz Unidimensional

Codigo A.7

Matriz Unidimensional

```
1     int[] numeros = { 1, 2, 3, 4, 5 };  
2     numeros[0] = 10; // Asigna 10 al primer elemento
```

Matriz Bidimensional

Codigo A.8

Matriz Bidimensional

```
1     int[,] matriz = {  
2         { 1, 2, 3 },  
3         { 4, 5, 6 },  
4         { 7, 8, 9 }  
5     };
```



ANEXO 5: Implementación de la herramienta computacional en suelos granulares y arcillosos

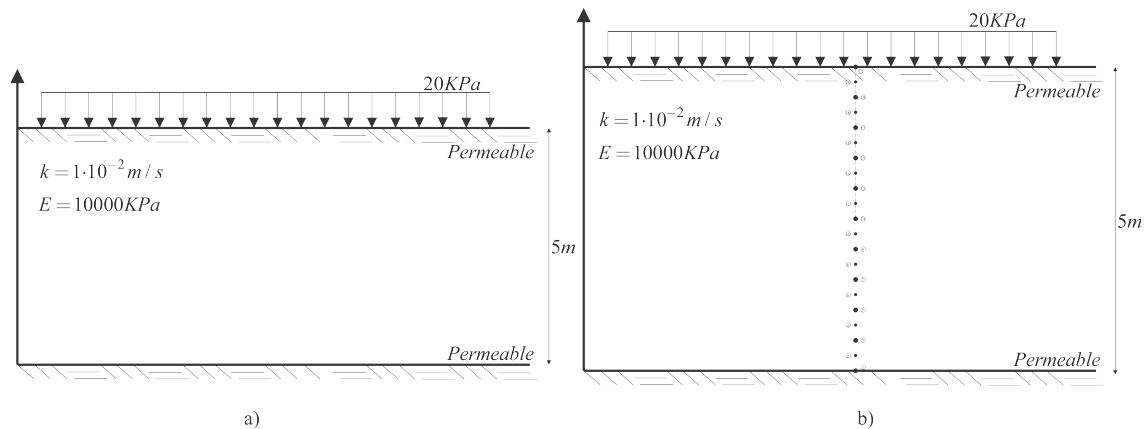
Ejemplos del Uso de la herramienta computacional

Aplicar la herramienta computacional desarrollada en suelos granulares y arcillosos para obtener asentamientos y presiones. Para la aplicación de la herramienta computacional, se utilizó dos modelos con 10 elementos cada uno, manteniendo la misma altura y las mismas cargas. La única variación estará en los parámetros del suelo, como el módulo de elasticidad y la constante de permeabilidad o conductividad hidráulica. Finalmente, realizaremos una comparación entre ambos modelos. El tiempo de análisis será para 5 años.

Simulación numérica de suelo granular

Se utilizaron dos modelos con 10 elementos cada uno, manteniendo la misma altura y cargas, pero variando los parámetros del suelo, como el módulo de elasticidad y la permeabilidad. Para los suelos granulares, el módulo de elasticidad se fijó en 10,000 kPa y la permeabilidad en $10^{-3} m/s$. El análisis se realizará durante 5 años, y se compararán los efectos del asentamiento y consolidación en ambos modelos. Se espera que los suelos granulares experimenten un asentamiento rápido debido a su alta permeabilidad.

Primer Modelos, suelo granular



Nota: En figura a) suelo granular, b) suelo granular discretizado en 10 elementos

Definición de tipo de suelo y elementos

Aquí se definió un nuevo tipo de suelo con el nombre de “granular” tenido como módulo de elasticidad 10000KPa , constante de permeabilidad 10^{-3}m/s y con densidad del agua $1\text{ton}/\text{m}^3$, como se muestra. Se tomo una discretización de 10 elementos de 0.5m cada elemento.

Definición de tipo de suelo y elementos - Modelo 1

PROPIEDADES DEL SUELO

Material: (*)

Elasticidad: (*) KPa

Densidad Fluido: (*) Ton/m³

Constante de Permeabilidad: (*) m/s

Agregar **Editar** **Regresar**

| Nro | Nombre del Material | Elasticidad [KN/m ²] | Densidad [Ton/m ³] | Constatae Rigidez [m/s] |
|-----|---------------------|----------------------------------|--------------------------------|-------------------------|
| 1 | granular | 10000 | 1 | 1E-03 |
| 2 | Suelo 2 | 750 | 1 | 5E-09 |

a)

DISCRETIZACIÓN DE ELEMENTOS

Elemento: (*)

Longitud: (*) m

Tipo de Suelo (*)

Suelo 1

Agregar **Editar** **Regresar**

| Elemento | Longitud [m] | Material |
|----------|--------------|----------|
| 1 | 0.5 | granular |
| 2 | 0.5 | granular |
| 3 | 0.5 | granular |
| 4 | 0.5 | granular |

b)

Nota: En la figura a) se muestra la creación del tipo del suelo granular, b) se muestra la discretización del suelo en 10 elementos.

Asignación de condiciones de frontera y cargas

El programa generó automáticamente las condiciones de frontera, aunque estas pueden ser editadas si es necesario. Sin embargo, no se recomendó realizar modificaciones como añadir o eliminar condiciones, ya que esto podría comprometer la precisión del análisis. Para este análisis se consideraron 1,822 intervalos de integración, avanzando en incrementos de dos días. Además, se asignó una carga progresiva que simuló las diferentes etapas de construcción de una vivienda.

Asignación de condiciones y cargas - Modelo 1

CONDICIÓN DE FRONTERA

Nodo: (*)

Condicion Frontera: (*)

Presión: (*) [Editar]

| Nro | UY | UZ |
|-----|----|----|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 2 | 3 |
| 4 | 2 | 2 |

CARGA APLICADA

Parametros de carga: [+ Agregar] [Editar] [Regresar]

| Nodo | Tiempo 1 | Tiempo 2 | Carga [KN] |
|------|----------|----------|------------|
| 1 | 1 | 0 | 0 |
| 2 | 2 | 172800 | 2 |
| 3 | 3 | 345600 | 4 |
| 4 | 4 | 518400 | 6 |
| 5 | 5 | 691200 | 8 |
| 6 | 6 | 864000 | 10 |
| 7 | 7 | 1036800 | 12 |

a)

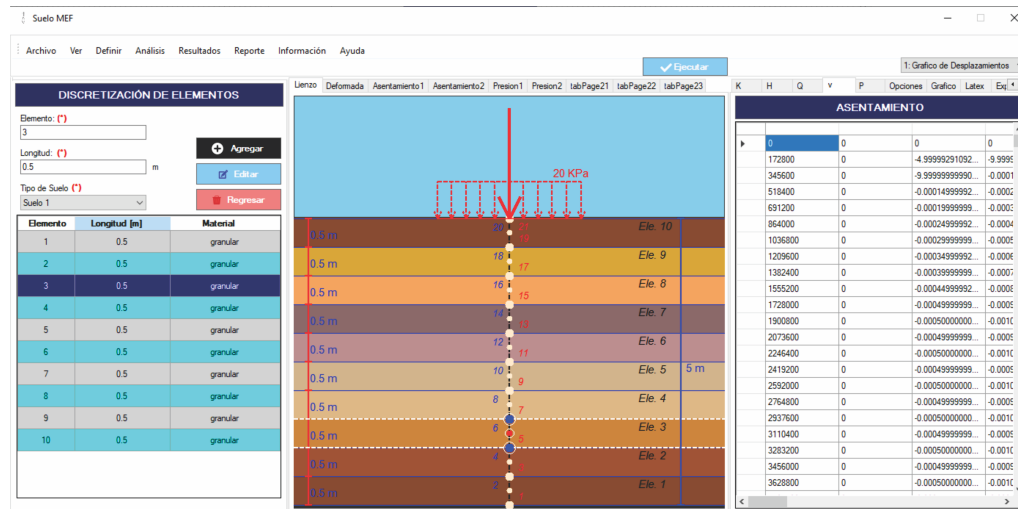
b)

Nota: En la figura a) se muestra la asignación de condiciones de frontera del suelo granular, b) se muestra las cargas a lo largo de 5 años generadas por el programa Suelo MEF.

Ejecución del cálculo matricial y resultados

Para obtener los resultados, se hizo clic en el botón "Ejecutar", que realizó el análisis del suelo. Los resultados mostraron los asentamientos en cada nodo y, en los nodos pares, también se presentaron las presiones correspondientes en esos puntos.

Ejecución del cálculo matricial y resultados

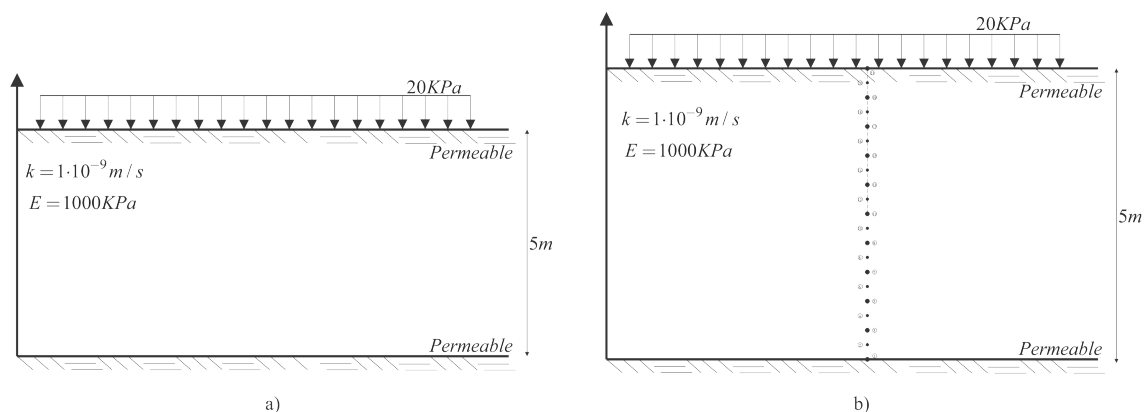


Nota: Se muestra la pestaña de asentamientos y presiones generado por el programa Suelo MEF.

Simulación numérica de suelo arcilloso.

Se utilizaron dos modelos con 10 elementos cada uno, manteniendo la misma altura y cargas, pero variando los parámetros del suelo, como el módulo de elasticidad y la permeabilidad. Para los suelos arcillosos, el módulo de elasticidad se fijó en $1,000kPa$ y la permeabilidad en $10^{-9} m/s$. El análisis se realizará durante 5 años, y se compararán los efectos del asentamiento y consolidación en ambos modelos. En suelos arcillosos, se espera un asentamiento más lento debido a su baja permeabilidad, con una consolidación gradual a lo largo del tiempo.

Primer Modelos, suelo arcilloso



Nota: En figura a) suelo arcilloso, b) suelo arcilloso discretizado en 10 elementos

Definición de tipo de suelo y elementos

Aquí se definió un nuevo material con el nombre de “arcilloso” tenido como módulo de elasticidad 1000KPa , constante de permeabilidad 10^{-9}m/s y con densidad del agua $1\text{ton}/\text{m}^3$, como se muestra. Se tomo una discretización de 10 elementos de 0.5m cada elemento.

Definición de tipo de suelo y elementos - Modelo 2

PROPIEDADES DEL SUELO

Material: (*)

Elasticidad: (*)
 KPa

Densidad Fluido: (*)
 Ton/m³

Constante de Permeabilidad: (*)
 m/s

+ Agregar
✎ Editar
🗑 Regresar

| Nro | Nombre del Material | Elasticidad [KN/m ²] | Densidad [Ton/m ³] | Constata Rigidez [m/s] |
|-----|---------------------|----------------------------------|--------------------------------|------------------------|
| 1 | arcilloso | 1000 | 1 | 1E-09 |
| 2 | Suelo 2 | 750 | 1 | 5E-09 |

DISCRETIZACIÓN DE ELEMENTOS

Elemento: (*)

Longitud: (*)
 m

Tipo de Suelo (*)

+ Agregar
✎ Editar
🗑 Regresar

| Elemento | Longitud [m] | Material |
|----------|--------------|-----------|
| 1 | 0.5 | arcilloso |
| 2 | 0.5 | arcilloso |
| 3 | 0.5 | arcilloso |
| 4 | 0.5 | arcilloso |

Nota: En la figura a) se muestra la creación del tipo del suelo granular, b) se muestra la discretización del suelo en 10 elementos.

Asignación de condiciones de frontera y cargas

El programa generó automáticamente las condiciones de frontera, aunque estas pueden ser editadas si es necesario. Sin embargo, no se recomendó realizar modificaciones como añadir o eliminar condiciones, ya que esto podría comprometer la precisión del análisis. Para este análisis se consideraron 1,822 intervalos de integración, avanzando en incrementos de dos días. Además, se asignó una carga progresiva que simuló las diferentes etapas de construcción de una vivienda.

Asignación de condiciones y cargas - Modelo 2

CONDICIÓN DE FRONTERA

Nodo: (*)

Condicion Frontera: (*)

Presión: (*)

[Editar](#)

| Nro | UY | UZ |
|-----|----|----|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 2 | 3 |
| 4 | 2 | 2 |

a)

CARGA APLICADA

Parametros de carga:

[Agregar](#) [Editar](#) [Regresar](#)

| Nodo | Tiempo 1 | Tiempo 2 | Carga [KN] |
|------|----------|----------|------------|
| 1 | 1 | 0 | 0 |
| 2 | 2 | 172800 | 2 |
| 3 | 3 | 345600 | 4 |
| 4 | 4 | 518400 | 6 |
| 5 | 5 | 691200 | 8 |
| 6 | 6 | 864000 | 10 |
| 7 | 7 | 1036800 | 12 |

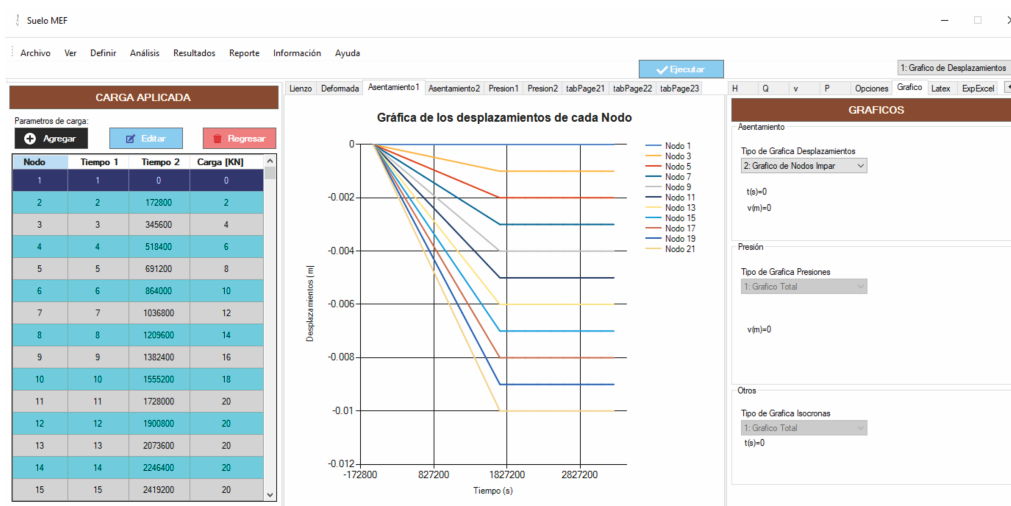
b)

Nota: En la figura a) se muestra la asignación de condiciones de frontera del suelo granular, b) se muestra las cargas a lo largo de 5 años generadas por el programa Suelo MEF.

Ejecución del cálculo matricial y resultados

Para obtener los resultados, se hizo clic en el botón "Ejecutar", que realizó el análisis del suelo. Los resultados mostraron los asentamientos en cada nodo y, en los nodos pares, también se presentaron las presiones correspondientes en esos puntos.

Ejecución del cálculo matricial y resultados



Nota: Se muestra la pestaña de asentamientos y presiones generado por el programa Suelo MEF.



ANEXO 6: Datos de asentamiento para la comparación entre el método de diferencia central y la herramienta desarrollada

Estos datos han sido filtrados de forma progresiva, la data real ocuparía más de cincuenta hojas.

| Tiempo (días) | Método de diferencia central (mm) | Suelos MEF (mm) | Diferencia (mm) |
|----------------------|--|------------------------|------------------------|
| 0 | 0.00 | 0.00 | 0.00 |
| 2 | -6.67 | -9.10 | 2.44 |
| 8 | -14.94 | -15.86 | 0.92 |
| 16 | -21.50 | -22.10 | 0.60 |
| 26 | -27.59 | -28.05 | 0.46 |
| 38 | -33.45 | -33.89 | 0.45 |
| 52 | -39.06 | -39.54 | 0.49 |
| 68 | -44.29 | -44.81 | 0.52 |
| 86 | -48.99 | -49.52 | 0.53 |
| 106 | -53.07 | -53.57 | 0.51 |
| 128 | -56.47 | -56.94 | 0.46 |
| 152 | -59.22 | -59.63 | 0.40 |
| 178 | -61.37 | -61.71 | 0.34 |
| 206 | -63.00 | -63.27 | 0.27 |
| 236 | -64.19 | -64.40 | 0.21 |
| 268 | -65.04 | -65.19 | 0.15 |
| 302 | -65.62 | -65.74 | 0.11 |
| 338 | -66.02 | -66.09 | 0.08 |
| 376 | -66.27 | -66.32 | 0.05 |
| 416 | -66.43 | -66.47 | 0.03 |
| 458 | -66.53 | -66.55 | 0.02 |
| 502 | -66.59 | -66.60 | 0.01 |
| 548 | -66.63 | -66.63 | 0.01 |
| 596 | -66.64 | -66.65 | 0.00 |
| 646 | -66.66 | -66.66 | 0.00 |



ANEXO 7: Datos de presiones para la comparación entre el método de diferencia central
y la herramienta desarrollada

Estos datos han sido filtrados de forma progresiva, la data real ocuparía más de cincuenta hojas.

Método de diferencia central

| Tiempo (días) | 1m | 2m | 3m | 4m |
|----------------------|--------------|--------------|--------------|--------------|
| 0 | -10.000000 | -10.000000 | -10.000000 | -10.000000 |
| 6 | -9.301449764 | -10.000000 | -10.000000 | -8.642859422 |
| 14 | -7.196469375 | -9.583273635 | -9.583273635 | -6.846235016 |
| 24 | -5.813084751 | -8.647505289 | -8.647505289 | -5.614680949 |
| 36 | -4.810698391 | -7.464644002 | -7.464644002 | -4.67524997 |
| 50 | -3.966866742 | -6.231373107 | -6.231373107 | -3.862270667 |
| 66 | -3.209581765 | -5.055902453 | -5.055902453 | -3.126269333 |
| 84 | -2.534119182 | -3.993886448 | -3.993886448 | -2.468527256 |
| 104 | -1.949687769 | -3.073017183 | -3.073017183 | -1.899243598 |
| 126 | -1.461303676 | -2.303265283 | -2.303265283 | -1.423497203 |
| 150 | -1.06692664 | -1.681660675 | -1.681660675 | -1.039323506 |
| 176 | -0.758831003 | -1.196048798 | -1.196048798 | -0.739198812 |
| 204 | -0.525740924 | -0.828658556 | -0.828658556 | -0.512139153 |
| 234 | -0.354825397 | -0.559266148 | -0.559266148 | -0.345645488 |
| 266 | -0.233278012 | -0.367686462 | -0.367686462 | -0.227242731 |
| 300 | -0.149399487 | -0.235479412 | -0.235479412 | -0.14553428 |
| 336 | -0.093205299 | -0.146907659 | -0.146907659 | -0.090793926 |
| 374 | -0.056643267 | -0.089279578 | -0.089279578 | -0.055177814 |
| 414 | -0.033532982 | -0.052853775 | -0.052853775 | -0.03266543 |
| 456 | -0.019338031 | -0.030480079 | -0.030480079 | -0.018837725 |
| 500 | -0.010863469 | -0.017122705 | -0.017122705 | -0.010582413 |
| 546 | -0.005944851 | -0.009370113 | -0.009370113 | -0.005791048 |
| 594 | -0.003169054 | -0.004994977 | -0.004994977 | -0.003087066 |
| 644 | -0.001645639 | -0.002593811 | -0.002593811 | -0.001603063 |
| 696 | -0.000832445 | -0.001312077 | -0.001312077 | -0.000810908 |



Herramienta desarrollada

| Tiempo (días) | 1m | 2m | 3m | 4m |
|---------------|--------------|--------------|--------------|--------------|
| 0 | 0 | 0 | 0 | 0 |
| 6 | -9.513385384 | -10.2753661 | -10.2753661 | -9.513385384 |
| 14 | -7.149197578 | -10.08412293 | -10.08412293 | -7.149197578 |
| 24 | -5.808519803 | -9.082826193 | -9.082826193 | -5.808519803 |
| 36 | -4.842528486 | -7.786124541 | -7.786124541 | -4.842528486 |
| 50 | -3.993558543 | -6.456073027 | -6.456073027 | -3.993558543 |
| 66 | -3.21656517 | -5.204037786 | -5.204037786 | -3.21656517 |
| 84 | -2.523067244 | -4.08237933 | -4.08237933 | -2.523067244 |
| 104 | -1.926523499 | -3.117179179 | -3.117179179 | -1.926523499 |
| 126 | -1.431879744 | -2.316830049 | -2.316830049 | -1.431879744 |
| 150 | -1.035913903 | -1.676143904 | -1.676143904 | -1.035913903 |
| 176 | -0.729500416 | -1.180356468 | -1.180356468 | -0.729500416 |
| 204 | -0.500048567 | -0.809095577 | -0.809095577 | -0.500048567 |
| 234 | -0.333644208 | -0.539847668 | -0.539847668 | -0.333644208 |
| 266 | -0.216690441 | -0.350612498 | -0.350612498 | -0.216690441 |
| 300 | -0.136987417 | -0.221650296 | -0.221650296 | -0.136987417 |
| 336 | -0.084295882 | -0.136393602 | -0.136393602 | -0.084295882 |
| 374 | -0.050491329 | -0.081696687 | -0.081696687 | -0.050491329 |
| 414 | -0.029438251 | -0.047632091 | -0.047632091 | -0.029438251 |
| 456 | -0.01670675 | -0.027032089 | -0.027032089 | -0.01670675 |
| 500 | -0.009229044 | -0.014932906 | -0.014932906 | -0.009229044 |
| 546 | -0.004962565 | -0.008029598 | -0.008029598 | -0.004962565 |
| 594 | -0.002597409 | -0.004202697 | -0.004202697 | -0.002597409 |
| 644 | -0.001323303 | -0.00214115 | -0.00214115 | -0.001323303 |
| 696 | -0.000656241 | -0.00106182 | -0.00106182 | -0.000656241 |



ANEXO 8: Datos de asentamiento en dos, cuatro, seis años

| Tiempo (días) | Dos años (mm) | Cuatro años (mm) | Seis años (mm) |
|---------------|---------------|------------------|----------------|
| 0 | 0 | 0 | 0 |
| 1 | -0.050956831 | -0.317751262 | 0.266794431 |
| 4 | -0.372428655 | -1.585421596 | 1.212992941 |
| 8 | -1.032644416 | -3.874500186 | 2.84185577 |
| 13 | -2.121909092 | -7.501410926 | 5.379501833 |
| 19 | -3.731077245 | -12.77234 | 9.041262759 |
| 26 | -5.939654752 | -19.98920681 | 14.04955206 |
| 34 | -8.804585698 | -29.45253449 | 20.6479488 |
| 43 | -12.35619137 | -41.46240378 | 29.10621242 |
| 53 | -16.60045805 | -56.31871121 | 39.71825316 |
| 64 | -21.50632747 | -74.14529673 | 52.63896926 |
| 76 | -25.07683843 | -88.57740196 | 63.50056353 |
| 89 | -27.50063577 | -101.3986727 | 73.89803691 |
| 103 | -29.12695404 | -113.4153023 | 84.28834829 |
| 118 | -30.17600415 | -124.9270819 | 94.75107777 |
| 134 | -30.82652276 | -136.0835617 | 105.2570389 |
| 151 | -31.214517 | -146.9645243 | 115.7500073 |
| 169 | -31.43720005 | -157.6067401 | 126.1695401 |
| 188 | -31.56022924 | -168.017107 | 136.4568778 |
| 208 | -31.62568169 | -178.182219 | 146.5565373 |
| 229 | -31.6592213 | -188.0762932 | 156.4170719 |
| 251 | -31.67577921 | -197.6675038 | 165.9917246 |
| 274 | -31.68365612 | -206.9226727 | 175.2390166 |
| 298 | -31.6872676 | -215.8104694 | 184.1232018 |
| 323 | -31.6888637 | -224.3034093 | 192.6145456 |
| 349 | -31.68954376 | -232.3789631 | 200.6894193 |
| 376 | -31.68982313 | -240.0200501 | 208.330227 |
| 404 | -31.6899338 | -247.2151265 | 215.5251927 |
| 433 | -31.68997607 | -253.9580136 | 222.2680376 |
| 463 | -31.68999165 | -260.2475667 | 228.557575 |
| 494 | -31.68999719 | -266.0872446 | 234.3972474 |
| 526 | -31.68999909 | -271.4846242 | 239.7946251 |
| 559 | -31.68999971 | -276.4508833 | 244.7608836 |
| 593 | -31.68999991 | -281.0002749 | 249.310275 |
| 628 | -31.68999997 | -285.1496038 | 253.4596038 |
| 664 | -31.68999999 | -288.9177185 | 257.2277185 |
| 701 | -31.69 | -292.3250277 | 260.6350277 |
| 739 | -31.69 | -295.3930465 | 263.7030465 |
| 778 | -31.69 | -298.1439794 | 266.4539794 |
| 818 | -31.69 | -300.6003437 | 268.9103437 |
| 859 | -31.69 | -302.7846342 | 271.0946342 |
| 901 | -31.69 | -304.7190328 | 273.0290328 |
| 944 | -31.69 | -306.4251611 | 274.7351611 |
| 988 | -31.69 | -307.9238754 | 276.2338754 |
| 1033 | -31.69 | -309.2351027 | 277.5451027 |
| 1079 | -31.69 | -310.3777154 | 278.6877154 |
| 1126 | -31.69 | -311.3694401 | 279.6794401 |
| 1174 | -31.69 | -312.2267997 | 280.5367997 |
| 1223 | -31.69 | -312.9650825 | 281.2750825 |
| 1273 | -31.69 | -313.5983364 | 281.9083364 |
| 1324 | -31.69 | -314.1393843 | 282.4493843 |
| 1376 | -31.69 | -314.599856 | 282.909856 |
| 1651 | -31.69 | -316.0237268 | 284.3337268 |
| 1709 | -31.69 | -316.1851017 | 284.4951017 |



ANEXO 9: Datos de asentamiento 10 elementos y 20 elementos

| Tiempo (días) | Tomando 10 elementos (mm) | Tomando 20 elementos (mm) | Diferencia (mm) |
|---------------|---------------------------|---------------------------|-----------------|
| 0 | 0 | 0 | 0 |
| 1 | -32.66986437 | -24.91955053 | -7.750313841 |
| 4 | -67.85397148 | -65.63175921 | -2.222212271 |
| 8 | -96.9803697 | -95.46574154 | -1.514628158 |
| 13 | -124.1546956 | -122.9867419 | -1.167953674 |
| 19 | -150.4244467 | -149.4669591 | -0.957487532 |
| 26 | -176.1949445 | -175.3701908 | -0.824753677 |
| 34 | -201.6161962 | -200.8548886 | -0.761307552 |
| 43 | -226.6009947 | -225.8498049 | -0.751189812 |
| 53 | -250.8647037 | -250.0978332 | -0.766870456 |
| 64 | -274.0307343 | -273.2464419 | -0.78429237 |
| 76 | -295.731944 | -294.9424142 | -0.789529793 |
| 89 | -315.6725879 | -314.8952473 | -0.77734064 |
| 103 | -333.6536499 | -332.9059935 | -0.747656361 |
| 118 | -349.5753483 | -348.8724119 | -0.702936394 |
| 134 | -363.4277973 | -362.7810986 | -0.646698687 |
| 151 | -375.2761039 | -374.6933418 | -0.58276218 |
| 169 | -385.2432741 | -384.7284519 | -0.514822165 |
| 188 | -393.4929326 | -393.0467474 | -0.446185242 |
| 208 | -400.2131967 | -399.8335911 | -0.379605608 |
| 229 | -405.6025987 | -405.2853954 | -0.317203247 |
| 251 | -409.8585813 | -409.5981301 | -0.260451222 |
| 274 | -413.1687631 | -412.9585456 | -0.210217517 |
| 298 | -415.704902 | -415.538057 | -0.166845062 |
| 323 | -417.6192926 | -417.489039 | -0.130253629 |
| 349 | -419.0432061 | -418.943157 | -0.100049077 |
| 376 | -420.0869276 | -420.0112992 | -0.0756284 |
| 404 | -420.8409434 | -420.7846709 | -0.056272473 |
| 433 | -421.3778676 | -421.3366457 | -0.041221855 |
| 463 | -421.7547619 | -421.725028 | -0.029733939 |
| 494 | -422.0155781 | -421.994456 | -0.021122073 |
| 526 | -422.193524 | -422.1787452 | -0.014778805 |
| 559 | -422.3132279 | -422.3030416 | -0.010186263 |
| 593 | -422.3926278 | -422.3857109 | -0.006916916 |
| 628 | -422.4445613 | -422.4399335 | -0.004627827 |
| 664 | -422.4780587 | -422.4750077 | -0.003051054 |
| 701 | -422.4993661 | -422.4973838 | -0.001982297 |
| 739 | -422.5127328 | -422.5114635 | -0.001269312 |
| 988 | -422.5325557 | -422.532491 | -6.47003E-05 |
| 1033 | -422.5329032 | -422.5328657 | -3.74844E-05 |



ANEXO 10: Comparación del asentamiento es suelos granulares y arcillosos

| Tiempo (días) | Suelo granular (mm) | Suelo arcilloso (mm) | Diferencia (mm) |
|---------------|---------------------|----------------------|-----------------|
| 0 | 0 | 0 | 0 |
| 1 | -0.050956831 | -0.317751262 | 0.266794431 |
| 4 | -0.372428655 | -1.585421596 | 1.212992941 |
| 8 | -1.032644416 | -3.874500186 | 2.84185577 |
| 13 | -2.121909092 | -7.501410926 | 5.379501833 |
| 19 | -3.731077245 | -12.77234 | 9.041262759 |
| 26 | -5.939654752 | -19.98920681 | 14.04955206 |
| 34 | -8.804585698 | -29.45253449 | 20.6479488 |
| 43 | -12.35619137 | -41.46240378 | 29.10621242 |
| 53 | -16.60045805 | -56.31871121 | 39.71825316 |
| 64 | -21.50632747 | -74.14529673 | 52.63896926 |
| 76 | -25.07683843 | -88.57740196 | 63.50056353 |
| 89 | -27.50063577 | -101.3986727 | 73.89803691 |
| 103 | -29.12695404 | -113.4153023 | 84.28834829 |
| 118 | -30.17600415 | -124.9270819 | 94.75107777 |
| 134 | -30.82652276 | -136.0835617 | 105.2570389 |
| 151 | -31.214517 | -146.9645243 | 115.7500073 |
| 169 | -31.43720005 | -157.6067401 | 126.1695401 |
| 188 | -31.56022924 | -168.017107 | 136.4568778 |
| 208 | -31.62568169 | -178.182219 | 146.5565373 |
| 229 | -31.6592213 | -188.0762932 | 156.4170719 |
| 251 | -31.67577921 | -197.6675038 | 165.9917246 |
| 274 | -31.68365612 | -206.9226727 | 175.2390166 |
| 298 | -31.6872676 | -215.8104694 | 184.1232018 |
| 323 | -31.6888637 | -224.3034093 | 192.6145456 |
| 349 | -31.68954376 | -232.3789631 | 200.6894193 |
| 376 | -31.68982313 | -240.0200501 | 208.330227 |
| 404 | -31.6899338 | -247.2151265 | 215.5251927 |
| 433 | -31.68997607 | -253.9580136 | 222.2680376 |
| 463 | -31.68999165 | -260.2475667 | 228.557575 |
| 494 | -31.68999719 | -266.0872446 | 234.3972474 |
| 526 | -31.68999909 | -271.4846242 | 239.7946251 |
| 559 | -31.68999971 | -276.4508833 | 244.7608836 |
| 593 | -31.68999991 | -281.0002749 | 249.310275 |
| 628 | -31.68999997 | -285.1496038 | 253.4596038 |
| 778 | -31.69 | -298.1439794 | 266.4539794 |
| 818 | -31.69 | -300.6003437 | 268.9103437 |
| 1651 | -31.69 | -316.0237268 | 284.3337268 |
| 1709 | -31.69 | -316.1851017 | 284.4951017 |

ANEXO 11: Código fuente

```
namespace TesisFisicalupaca {
    public partial class Form1 : Form {

        #region VARIABLES GLOBALES
        int numElementos;
        int numNodos;
        int numCondi;
        int numCantCargas;

        double[,] K;
        double[,] C;
        double[,] H;
        double[,] Q;

        double g;
        double ft;
        double theta;

        double[,] carga;
        double[,] nodinf2;

        int numMateriales;
        int numElementos;
        int numRestriciones;
        int numCargas;

        int [] idMaterial;
        string [] vMateriales;
        double [] vElasticidad;
        double [] vDensidad;
        double [] vRigidez;

        int indiceTablaMaterial;
        int indiceTablaElemento;

        // variables para imprimir
        double[,] KKensablando;
        double[,] HHensablando;
        double[,] QQensablando;

        // Variables Grafica
        double[,] mtiempo;
        double[,] mDesplazamientos;
        double[,] mPresiones;

        // Condicion de analisis
        bool analisis;

        // Salir del programa
        bool salir = false;

        #endregion

        Bitmap drawingBitmap;
```



```
public Form1() {
    InitializeComponent();
}

#region CARGAR DATOS EN LA TABLA
private void ActualizarEncabezadosTablaDatos() {
    string[] encabezadoPropiedades = { "Nro", "Nombre del Material", "P
    int[] anchosPropiedades = { 30,100,70,70 };
    DarEstiloTablaDatos(this.dgvTablaMateriales, encabezadoPropiedades

    string[] encabezadoElementos = { "Elemento", "Longitud [m]", "Mater
    int[] anchosElementos = { 90,150,150 };
    DarEstiloTablaDatos(this.dgvTablaElementos, encabezadoElementos, ar

    string[] encabezadoCondiciones = { "Nro", "UY", "UZ" };
    int[] anchosCondiciones = { 90,150,150 };
    DarEstiloTablaDatos(this.dgvTablaRestriciones, encabezadoCondicion

    string[] encabezadoCargas = { "Nodo", "Tiempo 1", "Tiempo 2", "Carg
    int[] anchosCargas = { 90,100,100,100 };
    DarEstiloTablaDatos(this.dgvTablaCargas, encabezadoCargas, anchosCa
}

private void DarEstiloTablaDatos(DataGridView Tabla, string[] encabezados
    // CanumCantCargasdad Columnas
    Tabla.ColumnCount = encabezados.GetLength(0);

    // Ancho de cada Columana
    for ( int i = 0; i < anchos.GetLength(0); i++ ) {
        Tabla.Columns[i].Width = anchos[i];
    }
    // Nombres de las columana
    for ( int i=0;i< encabezados.GetLength(0); i++ ) {
        Tabla.Columns[i].HeaderText = encabezados[i];
    }

    // Centrar el contenido
    Tabla.DefaultCellStyle.Alignment = DataGridViewContentAlignment.M
    Tabla.ColumnHeadersDefaultCellStyle.Alignment = DataGridViewConte
    Tabla.ColumnHeadersDefaultCellStyle.Font = new System.Drawing.For

    // Puedes cambiar el color de fondo, el color de texto, el tipo d
    Tabla.DefaultCellStyle.BackColor = Color.LightGray;
    Tabla.DefaultCellStyle.ForeColor = Color.Black;
    //dataGridView1.DefaultCellStyle.Font = new Font("Arial",10);

    Tabla.AlternatingRowsDefaultCellStyle.BackColor = Color.Aqua;

    Tabla.AllowUserToOrderColumns = true;
    Tabla.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

    Tabla.BorderStyle = BorderStyle.FixedSingle;
    Tabla.DefaultCellStyle.SelectionBackColor = Color.DarkBlue;
    Tabla.DefaultCellStyle.SelectionForeColor = Color.White;

    // Desactivar el alto de las filas
    Tabla.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.None;
    // Desactivar el ajuste automático de envoltura del contenido de
```



```
Tabla.DefaultCellStyle.WrapMode = DataGridViewTriState.False;

Tabla.AllowUserToAddRows = false; // Deshabilitar la fila adicional

// Establecer el alto deseado de las filas (en este ejemplo, 30 p
foreach ( DataGridViewRow row in Tabla.Rows ) {
    row.Height = 30;
}

// Desactivar la primera columna -1
Tabla.RowHeadersVisible = false;
}

private void IniciarTablasResultados() {

    ActualizarTabla(this.dgvKe, 3, 3, 100);
    ActualizarTabla(this.dgvK, 3, 3, 100);

    ActualizarTabla(this.dgvHe, 2, 2, 100);
    ActualizarTabla(this.dgvH, 2, 2, 100);

    ActualizarTabla(this.dgvQe, 3, 2, 100);
    ActualizarTabla(this.dgvQ, 3, 2, 100);

    ActualizarTabla(this.dgvMatrizEnsablada, 30, 30, 100);
    ActualizarTabla(this.dgvMatrizEnsablada2, 30, 30, 100);
    ActualizarTabla(this.dgvTabla, 50, 150, 100);
}

#region PROCEDIMIENTOS PARA TABLAS
// PROCEDIMIENTOS PARA TABLAS
private void ActualizarTabla(DataGridView tabla, int num_filas, int n
    tabla.ColumnCount = num_columnas;
    tabla.RowCount = num_filas;
    for ( int j = 0; j < num_columnas; j++ ) {
        tabla.Columns[j].Width = ancho_columnas;
    }
    tabla.AllowUserToAddRows = false; // Deshabilitar la fila adicional
    // Establecer el alto deseado de las filas (en este ejemplo, 30 p
    foreach ( DataGridViewRow row in tabla.Rows ) {
        row.Height = 30;
    }
}

#endregion

private void IniciarDatosIniciales() {

    // I. DATOS - PROPIEDADES DEL SUELO
    string[] dMateriales = { "Suelo 1", "Suelo 2", "Suelo 3", "Suelo
    double[] dElasticidad = { 750, 750, 750, 750, 750 };
    double[] dDensidad = { 1, 1, 1, 1, 1 };
    double[] dRigidez = { 0.000000005, 0.000000005, 0.000000005, 0.00
    numMateriales = dMateriales.Length;
}
}
```




```
// Cargamos los tipos de suelos en el cuadro desplegable
this.cboTipoSuelo.Items.Clear();
for (int i = 0; i < dMateriales.Length; i++) {
    this.cboTipoSuelo.Items.Add(dMateriales[i]);
}
this.cboTipoSuelo.SelectedIndex = 0;

// II. DATOS - ELEMENTOS

// Ejemplo 1 - Con Dos elementos
int[] idElemento = { 1, 2 };
double[] dDistancia = { 2.5, 2.5 };

// Ejemplo 2 - Con Cinco elementos
//int[] idElemento = { 1, 2, 3, 4, 5 };
//double[] dDistancia = { 1, 1, 1, 1, 1 };

numElementos = idElemento.Length;
this.dgvTablaElementos.RowCount = numElementos;
for (int i = 0; i < numElementos; i++) {
    this.dgvTablaElementos.Rows[i].Cells[0].Value = idElemento[i];
    this.dgvTablaElementos.Rows[i].Cells[1].Value = dDistancia[i];
    this.dgvTablaElementos.Rows[i].Cells[2].Value = dMateriales[i];
}

// III. DATOS - CONDICION DE FRONTERA

// Ejemplo 1 - Con Dos elementos
int[] idRestriciones = { 1, 2, 3, 4, 5 };
int[] dRestri1 = { 1, 2, 2, 2, 2 };
int[] dRestri2 = { 1, 2, 3, 2, 1 };

// Ejemplo 2 - Con Cinco elementos
//int[] idRestriciones = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };
//int[] dRestri1 = { 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 };
//int[] dRestri2 = { 1, 2, 3, 2, 3, 2, 3, 2, 3, 2, 1 };

numRestriciones = idRestriciones.Length;
this.dgvTablaRestriciones.RowCount = numRestriciones;
for (int i = 0; i < numRestriciones; i++) {
    this.dgvTablaRestriciones.Rows[i].Cells[0].Value = idRestriciones[i];
    this.dgvTablaRestriciones.Rows[i].Cells[1].Value = dRestri1[i];
    this.dgvTablaRestriciones.Rows[i].Cells[2].Value = dRestri2[i];
}

// IV. DATOS - CARGA EXTERNA

// Ejemplo 1 - Con Dos elementos

// Ejemplo 2 - Con Cinco elementos
int[] idCargas = { 1, 2 };
double[] dCarga1 = { 0.5, 0.5 };
double[] dCarga2 = { 0.5, 0.5 };

numCargas = idCargas.Length;
```



```
this.dgvTablaCargas.RowCount = numCargas;
for (int i = 0; i < numCargas; i++) {
    this.dgvTablaCargas.Rows[i].Cells[0].Value = idCargas[i];
    this.dgvTablaCargas.Rows[i].Cells[1].Value = dCarga1[i];
    this.dgvTablaCargas.Rows[i].Cells[2].Value = dCarga2[i];
}

// COMBOBOX DE SELECCION DE ELEMENTOS
// Desplazamientos
this.cboTipoGraficaDesplazamiento.Items.Clear();
this.cboTipoGraficaDesplazamiento.Items.Add("1: Grafico Total");
this.cboTipoGraficaDesplazamiento.Items.Add("2: Grafico de Nodos Impar");
this.cboTipoGraficaDesplazamiento.Items.Add("3: Grafico de Nodos Pares");
this.cboTipoGraficaDesplazamiento.Items.Add("4: Grafico de un solo");
this.cboTipoGraficaDesplazamiento.SelectedIndex = 0;
//1: Grafica Total      2: Grafica de Nodos Impar      3: Grafica

// Presiones
this.cboTipoGraficaPresiones.Items.Clear();
this.cboTipoGraficaPresiones.Items.Add("1: Grafico Total");
this.cboTipoGraficaPresiones.Items.Add("2: Grafico de un solo");
this.cboTipoGraficaPresiones.SelectedIndex = 0;
//1: Grafica Total      2: Grafica de Nodos Impar      3: Grafica

// Isocronas
this.cboTipoGraficaIsocronas.Items.Clear();
this.cboTipoGraficaIsocronas.Items.Add("1: Grafico Total");
this.cboTipoGraficaIsocronas.Items.Add("2: Grafico de un solo");
this.cboTipoGraficaIsocronas.SelectedIndex = 0;

// Graficos
this.cboGrafico.Items.Clear();
this.cboGrafico.Items.Add("1: Grafico de Desplazamientos");
this.cboGrafico.Items.Add("2: Grafico de Presiones");
this.cboGrafico.Items.Add("3: Grafico de Isocronas"); ;
this.cboGrafico.SelectedIndex = 0;

}

#endregion

#region AGREGRA, EDITAR, ELIMINAR VALORES DE LA TABLA

// DATOS - PROPIEDADES DEL SUELO
private void dgvTablaMateriales_CellClick(object sender, DataGridViewCell
    // Verificar si se hizo clic en una celda válida
    if (e.RowIndex >= 0) {
        // Obtener el índice de la fila seleccionada
        indiceTablaMaterial = e.RowIndex;

        // Mostrar el contenido en la txt
```



```
        this.txtProMaterial.Text = this.dgvTablaMateriales.Rows[indiceTablaMateriales].Text;
        this.txtProElasticidad.Text = this.dgvTablaMateriales.Rows[indiceTablaMateriales].Text;
        this.txtProDensidad.Text = this.dgvTablaMateriales.Rows[indiceTablaMateriales].Text;
        this.txtConstanteRigidez.Text = this.dgvTablaMateriales.Rows[indiceTablaMateriales].Text;
    }
}
private void btnMaterialAgregar_Click(object sender, EventArgs e) {
    int conT = 0;
    double result;
    bool duplicidad = false;

    this.dgvTablaMateriales.Rows.Add("", "", "", "", "");
    numMateriales = this.dgvTablaMateriales.RowCount;
    if (this.txtProMaterial.Text != "") {
        for (int i = 0; i < numMateriales; i++) {
            if (this.txtProMaterial.Text == this.dgvTablaMateriales.Rows[i].Text) {
                duplicidad = true;
                break;
            }
        }
        if (duplicidad) {
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[1].Value = this.txtProMaterial.Text;
        } else {
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[1].Value = this.txtProMaterial.Text;
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[0].Value = conT;
            this.txtProMaterial.BackColor = Color.FromArgb(255, 255, 255);
            conT += 1;
        }
    } else {
        this.txtProMaterial.BackColor = Color.FromArgb(250, 200, 100);
        this.txtProMaterial.Focus();
    }

    if (this.txtProElasticidad.Text != "") {
        if (double.TryParse(this.txtProElasticidad.Text, out result)) {
            this.txtProElasticidad.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[2].Value = result;
            conT += 1;
        } else {
            this.txtProElasticidad.BackColor = Color.FromArgb(250, 200, 100);
            this.txtProElasticidad.Focus();
        }
    } else {
        this.txtProElasticidad.BackColor = Color.FromArgb(250, 200, 100);
        this.txtProElasticidad.Focus();
    }

    if (this.txtProDensidad.Text != "") {
        if (double.TryParse(this.txtProDensidad.Text, out result)) {
            this.txtProDensidad.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[3].Value = result;
            conT += 1;
        } else {
            this.txtProDensidad.BackColor = Color.FromArgb(250, 200, 100);
            this.txtProDensidad.Focus();
        }
    } else {
        this.txtProDensidad.BackColor = Color.FromArgb(250, 200, 100);
    }
}
```



```
        this.txtProDensidad.Focus();
    }

    if (this.txtConstanteRigidez.Text != "") {
        if (double.TryParse(this.txtConstanteRigidez.Text, out result)) {
            this.txtConstanteRigidez.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaMateriales.Rows[numMateriales - 1].Cells[4].Value = result;
            conT += 1;
        } else {
            this.txtConstanteRigidez.BackColor = Color.FromArgb(250, 200, 100);
            this.txtConstanteRigidez.Focus();
        }
    }
} else {
    this.txtConstanteRigidez.BackColor = Color.FromArgb(250, 200, 100);
    this.txtConstanteRigidez.Focus();
}

if (conT == 4) {
    this.txtProMaterial.Text = "";
    this.txtProElasticidad.Text = "";
    this.txtProDensidad.Text = "";
    this.txtConstanteRigidez.Text = "";
    // Establecer el alto deseado de las filas (en este ejemplo, 30 p
    foreach (DataGridViewRow row in this.dgvTablaMateriales.Rows) {
        row.Height = 30;
    }
} else {
    this.dgvTablaMateriales.Rows.RemoveAt(numMateriales - 1);
}

private void btnMaterialEditar_Click(object sender, EventArgs e) {
    int conT = 0;
    double result;
    int duplicidad = 0;
    numMateriales = this.dgvTablaMateriales.RowCount;
    if (this.txtProMaterial.Text != "") {
        for (int i = 0; i < numMateriales; i++) {
            if (this.txtProMaterial.Text == this.dgvTablaMateriales.Rows[i].Cells[0].Value) {
                duplicidad = duplicidad + 1;
            }
        }
        if (duplicidad > 0) {
            this.dgvTablaMateriales.Rows[indiceTablaMaterial].Cells[1].Value = this.txtProMaterial.Text;
        } else {
            this.dgvTablaMateriales.Rows[indiceTablaMaterial].Cells[1].Value = this.txtProMaterial.Text;
        }
        this.txtProMaterial.BackColor = Color.FromArgb(255, 255, 255);
        conT += 1;
    } else {
        this.txtProMaterial.BackColor = Color.FromArgb(250, 200, 100);
        this.txtProMaterial.Focus();
    }

    if (this.txtProElasticidad.Text != "") {
        if (double.TryParse(this.txtProElasticidad.Text, out result)) {
            this.txtProElasticidad.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaMateriales.Rows[indiceTablaMaterial].Cells[2].Value = result;
            conT += 1;
        }
    }
}
```



```
        } else {
            this.txtProElasticidad.BackColor = Color.FromArgb(250, 200, 100);
            this.txtProElasticidad.Focus();
        }
    } else {
        this.txtProElasticidad.BackColor = Color.FromArgb(250, 200, 100);
        this.txtProElasticidad.Focus();
    }
}

if (this.txtProDensidad.Text != "") {
    if (double.TryParse(this.txtProDensidad.Text, out result)) {
        this.txtProDensidad.BackColor = Color.FromArgb(255, 255, 255);
        this.dgvTablaMateriales.Rows[indiceTablaMaterial].Cells[3].Value = result;
        conT += 1;
    } else {
        this.txtProDensidad.BackColor = Color.FromArgb(250, 200, 100);
        this.txtProDensidad.Focus();
    }
} else {
    this.txtProDensidad.BackColor = Color.FromArgb(250, 200, 100);
    this.txtProDensidad.Focus();
}

if (this.txtConstanteRigidez.Text != "") {
    if (double.TryParse(this.txtConstanteRigidez.Text, out result)) {
        this.txtConstanteRigidez.BackColor = Color.FromArgb(255, 255, 255);
        this.dgvTablaMateriales.Rows[indiceTablaMaterial].Cells[4].Value = result;
        conT += 1;
    } else {
        this.txtConstanteRigidez.BackColor = Color.FromArgb(250, 200, 100);
        this.txtConstanteRigidez.Focus();
    }
} else {
    this.txtConstanteRigidez.BackColor = Color.FromArgb(250, 200, 100);
    this.txtConstanteRigidez.Focus();
}

if (conT == 4) {
    this.txtProMaterial.Text = "";
    this.txtProElasticidad.Text = "";
    this.txtProDensidad.Text = "";
    this.txtConstanteRigidez.Text = "";
}
}

private void btnMaterialEliminar_Click(object sender, EventArgs e) {
    if (this.dgvTablaMateriales.SelectedRows.Count > 0) {
        this.dgvTablaMateriales.Rows.Remove(this.dgvTablaMateriales.CurrentRow);
        numMateriales = this.dgvTablaMateriales.RowCount;
        for (int i = 0; i < numMateriales; i++) {
            this.dgvTablaMateriales.Rows[i].Cells[0].Value = i + 1;
        }
        this.txtProMaterial.Text = "";
        this.txtProElasticidad.Text = "";
        this.txtProDensidad.Text = "";
        this.txtConstanteRigidez.Text = "";
    }
}
}
```



```
// DATOS - ELEMENTOS
private void dgvTablaElementos_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Verificar si se hizo clic en una celda válida
    if (e.RowIndex >= 0) {
        // Obtener el índice de la fila seleccionada
        indiceTablaElemento = e.RowIndex;

        // Mostrar el contenido en la txt
        this.txtElemento.Text = this.dgvTablaElementos.Rows[indiceTablaElemento].Text;
        this.txtLongitud.Text = this.dgvTablaElementos.Rows[indiceTablaElemento].Text;

        // Obtén el índice del elemento que conumCantCargasene el texto de
        int indice = this.cboTipoSuelo.FindStringExact(this.dgvTablaElementos.Rows[indiceTablaElemento].Text);
        // Verifica si se encontró el texto dentro del ComboBox
        if (indice != -1) {
            // Selecciona el texto dentro del ComboBox
            this.cboTipoSuelo.SelectedIndex = indice;
        }

        // Llamamos a dibujar El elemento seleccionado
        DibujoSuelosNodosElementos(indiceTablaElemento); // empieza de
    }
}

private void btnnumElementosementoAgregar_Click(object sender, EventArgs e)
{
    int conT = 0;
    double result;

    this.dgvTablaElementos.Rows.Add("", "", "", "");
    numElementos = this.dgvTablaElementos.RowCount;

    if (this.txtElemento.Text != "") {
        this.dgvTablaElementos.Rows[numElementos - 1].Cells[0].Value = numElementos;
        this.txtElemento.BackColor = Color.FromArgb(255, 255, 255);
        conT += 1;
    } else {
        this.txtElemento.BackColor = Color.FromArgb(250, 200, 100);
        this.txtElemento.Focus();
    }

    if (this.txtLongitud.Text != "") {
        if (double.TryParse(this.txtLongitud.Text, out result)) {
            this.txtLongitud.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaElementos.Rows[numElementos - 1].Cells[1].Value = result;
            conT += 1;
        } else {
            this.txtLongitud.BackColor = Color.FromArgb(250, 200, 100);
            this.txtLongitud.Focus();
        }
    } else {
        this.txtLongitud.BackColor = Color.FromArgb(250, 200, 100);
        this.txtLongitud.Focus();
    }

    if (this.cboTipoSuelo.Text != "") {
        this.dgvTablaElementos.Rows[numElementos - 1].Cells[2].Value = this.cboTipoSuelo.Text;
    } else {
        this.cboTipoSuelo.SelectedIndex = 0;
    }
}
```



```
        if (conT == 2) {
            //this.txtElemento.Text = "";
            //this.txtLongitud.Text = "";
            // Establecer el alto deseado de las filas (en este ejemplo, 30 p
            foreach (DataGridViewRow row in this.dgvTablaElementos.Rows) {
                row.Height = 30;
            }
        } else {
            this.dgvTablaElementos.Rows.RemoveAt(numElementos - 1);
        }

        GenerarGDL();
        DibujoSuelosNodosElementos(0);
    }
private void btnnumElementosementoEditar_Click(object sender, EventArgs e)
{
    int conT = 0;
    double result;
    numElementos = this.dgvTablaElementos.RowCount;
    if (this.txtElemento.Text != "") {
        this.dgvTablaElementos.Rows[indiceTablaElemento].Cells[0].Value =
        conT += 1;
    } else {
        this.txtElemento.BackColor = Color.FromArgb(250, 200, 100);
        this.txtElemento.Focus();
    }

    if (this.txtLongitud.Text != "") {
        if (double.TryParse(this.txtLongitud.Text, out result)) {
            this.txtLongitud.BackColor = Color.FromArgb(255, 255, 255);
            this.dgvTablaElementos.Rows[indiceTablaElemento].Cells[1].Val
            conT += 1;
        } else {
            this.txtLongitud.BackColor = Color.FromArgb(250, 200, 100);
            this.txtLongitud.Focus();
        }
    } else {
        this.txtLongitud.BackColor = Color.FromArgb(250, 200, 100);
        this.txtLongitud.Focus();
    }

    if (this.cboTipoSuelo.Text != "") {
        this.dgvTablaElementos.Rows[indiceTablaElemento].Cells[2].Value =
    } else {
        this.cboTipoSuelo.SelectedIndex = 0;
    }

    if (conT == 2) {
        this.txtElemento.Text = "";
        this.txtLongitud.Text = "";
    }
    GenerarGDL();
    DibujoSuelosNodosElementos(0);
}
private void btnnumElementosementoEliminar_Click(object sender, EventArgs
{
    if (this.dgvTablaElementos.SelectedRows.Count > 0) {
        this.dgvTablaElementos.Rows.Remove(this.dgvTablaElementos.Current
        numElementos = this.dgvTablaElementos.RowCount;
    }
}
```



```
        for (int i = 0; i < numElementos; i++) {
            this.dgvTablaElementos.Rows[i].Cells[0].Value = i + 1;
        }
        this.txtElemento.Text = "";
        this.txtLongitud.Text = "";
    }
    GenerarGDL();
    DibujoSuelosNodosElementos(0);
}

private void btnCargaAgregar_Click(object sender, EventArgs e) {

    // Instancia Form2
    Form2 form2 = new Form2();
    // Enviar información a Form2 a través de la propiedad
    form2.Form1AForm2AA = "tab-AsignacionCargas";
    form2.Form1AForm2BB = "";

    if ( form2.ShowDialog() == DialogResult.OK ) {
        // Recibe la información de Form2
        string returnedData = form2.Form2AForm1AA;

        // Imprime el valor de la carga en una tabla
        imprimirM(ConvertStringToMatrix2(returnedData),this.dgvTablaCarga
    }
}

#endregion

#region EJECUCION DE CALCULO

// EJECUTAR CALCULOS
private void btnEjecutarCalculo_Click(object sender, EventArgs e) {

    // Datos globales
    g = Convert.ToDouble(this.txtGravedad.Text);           // g: Gravedad 9.
    theta = Convert.ToDouble(this.txtTheta.Text);         // theta tomarem

    ft = 1;                                               // ??????????

    // Datos canumCantCargasdad de Elementos, Nodos, Restriciones, fronte
    numElementos = this.dgvTablaElementos.Rows.Count;    // numElementos:
    numElementos = numElementos;                          // numEL
    numNodos = 2 * numElementos + 1;                       //
    numCondi = numNodos - numElementos;

    //numNodos=this.dgvTablaRestriciones.Rows.Count

    // Matriz de Cargas
    carga = zerosM(this.dgvTablaCargas.RowCount, 2);
    for ( int i = 0; i <this.dgvTablaCargas.RowCount; i++ ) {
        carga[i,0] = Convert.ToDouble(this.dgvTablaCargas.Rows[i].Cells[2]
        carga[i,1] = Convert.ToDouble(this.dgvTablaCargas.Rows[i].Cells[3]
    }
    numCantCargas = rows(carga);                            // numC
```




```
MessageBox.Show(numCantCargas.ToString());

//double[,] nodinf = { { 1, 1, 1 }, { 2, 2, 2 }, { 3, 2, 3 }, { 4, 2, 2 }, { 5, 2, 2 } };
//double[,] nodinf = { { 1,1,1 },{ 2,2,2 },{ 3,2,3 },{ 4,2,2 },{ 5,2,2 } };
//double[] vector = { 1,1,1,2,2,2,3,5 };

K = zerosM(numNodos, numNodos); // Matriz de rigidez
H = zerosM(numCondi,numCondi); // Matriz de restricciones
Q = zerosM(numNodos,numCondi); // Matriz de cargas

C = zerosM(numCantCargas, 2 + numNodos + numCondi);

// MATRIZ DE RIGIDEZ DEL ELEMENTO Ke [RIGIDEZ]
// Elemento E, L, gdl
double[] L = zerosM(numElementos);
double[] E = zerosM(numElementos);
double[] rho = zerosM(numElementos);
double[] kel = zerosM(numElementos);

KKensablado = zerosM(1, 3); // En total tendra 3 elementos
HHensablado = zerosM(1, 2);
QQensablado = zerosM(1, 2);

string tsueloElemento;
string tsueloMaterial;

numMateriales = this.dgvTablaMateriales.RowCount;

// Elemento E, L, gdl
for (int i = 0; i < numElementos; i++) {
    L[i] = Convert.ToDouble(this.dgvTablaElementos.Rows[i].Cells[1].Value);
    tsueloElemento = this.dgvTablaElementos.Rows[i].Cells[2].Value.ToString();
    for (int j = 0; j < numMateriales; j++) {
        tsueloMaterial = this.dgvTablaMateriales.Rows[j].Cells[1].Value.ToString();
        if (tsueloElemento == tsueloMaterial) {
            E[i] = Convert.ToDouble(this.dgvTablaMateriales.Rows[j].Cells[2].Value);
            rho[i] = Convert.ToDouble(this.dgvTablaMateriales.Rows[j].Cells[3].Value);
            kel[i] = Convert.ToDouble(this.dgvTablaMateriales.Rows[j].Cells[4].Value);
        }
    }
}

// dgvTablaRestriciones
double[,] nodinf = zerosM(numNodos, 3);
for (int i = 0; i < numNodos; i++) {
    nodinf[i, 0] = i + 1;
    nodinf[i, 1] = Convert.ToDouble(this.dgvTablaRestriciones.Rows[i].Cells[1].Value);
    nodinf[i, 2] = Convert.ToDouble(this.dgvTablaRestriciones.Rows[i].Cells[2].Value);
}
nodinf2 = nodinf;
imprimirM(nodinf, this.dgvTabla);

// Nodos Nodo-i Nodo-ij Nodo-j
int[,] gdl = new int[numElementos, 3];
for (int i = 0; i < numElementos; i++) {
    gdl[i, 0] = 2 * i + 1;
    gdl[i, 1] = 2 * i + 2;
```



```
        gdl[i, 2] = 2 * i + 3;
    }

    // CALCULO DE LAS MATRICES

    KKEnsamblado(L, E, gdl); // Se obtiene K global
    HHEnsamblado(L, E, rho, kel, gdl); // Se obtiene H global
    QQEnsamblado(gdl); // Se obtiene Q global

    // Carga Numero de Elemento en los combobox de Ke, He, He.
    CargarComboboxPos();

    // Imprimimos las matrices ensambladas
    double[,] A1 = zerosM(numNodos, numNodos);
    double[,] A2 = zerosM(numNodos, numCondi);
    double[,] B2 = zerosM(numCondi, numCondi);

    imprimirM(stack(augment(A1, A2), augment(tranponerM(Q), B2)), this.dg
    imprimirM(stack(augment(K, Q), augment(zerosM(numCondi, numNodos), H

    // ANALISIS INTERATIVO
    Bkf();

    // CONDICION DEL ANALISIS
    analisis = true;

    // IMPRIMIR RESULTADOS
    mtiempo = submatrix(carga, 0, numCantCargas - 1, 0, 0);
    mDesplazamientos = submatrix(C, 0, numCantCargas - 1, 1, numNodos);
    mPresiones = submatrix(C, 0, numCantCargas - 1, numNodos + 1, numNodo

    imprimirM(augment(mtiempo, mDesplazamientos), this.dgvMatrizEnsablada
    imprimirM(augment(mtiempo, mPresiones), this.dgvMatrizEnsablada2);

    // GRAFICA DE RESULTADOS
    GraficaDesplazamiento((int)this.cboTipoGraficaDesplazamieto.SelectedI
    chart1.MouseMove += Chart1_MouseMove;
}

private void GenerarGDL() {
    int numElementos = this.dgvTablaElementos.RowCount;
    this.dgvTablaRestriciones.RowCount = 2 * numElementos + 1;
    int tt = 1;
    for (int i=0; i < 2*numElementos+1; i++) {
        this.dgvTablaRestriciones.Rows[i].Cells[0].Value = i + 1;
        this.dgvTablaRestriciones.Rows[i].Cells[1].Value = 2;
        if (tt == 1) {
            this.dgvTablaRestriciones.Rows[i].Cells[2].Value = 3;
            tt= 2;
        } else {
            this.dgvTablaRestriciones.Rows[i].Cells[2].Value = 2;
            tt = 1;
        }
    }
}

this.dgvTablaRestriciones.Rows[0].Cells[1].Value = 1;
```



```
        this.dgvTablaRestriciones.Rows[0].Cells[2].Value = 1;
        this.dgvTablaRestriciones.Rows[2*numElementos].Cells[2].Value = 1;
    }

    #endregion EJECUCION DE CALCULO

    #region CONTRUCCION DE LAS MATRICES K, H, Q

    // 1.1 .- FUNCION MATRIZ DE RIGUIDEZ DEL ELEMENTO Ke [RIGIDEZ]
    private static double[,] Ke(double E, double L) {
        double[,] ke = { { 7, -8, 1 }, { -8, 16, -8 }, { 1, -8, 7 } };
        double r = E / (3 * L);
        return multScalarM(ke, r);
    }

    // 1.2 .- FUNCION MATRIZ DE RIGUIDEZ GLOBAL K
    private static double[,] KEnsamblado(double[,] K, double[,] Ke, double[]
        for (int i = 0; i < 3; i++) {
            int ifil = (int)gdl[i];
            for (int j = 0; j < 3; j++) {
                int jcol = (int)gdl[j];
                K[ifil - 1, jcol - 1] = K[ifil - 1, jcol - 1] + Ke[i, j];
            }
        }
        return K;
    }

    // 1.3 .- PROCEDIMIENTO MATRIZ DE RIGUIDEZ GLOBAL KK
    private void KKEnsamblado(double[] d_L, double[] d_E, int[,] d_gdl) {
        double[] gdl = zerosM(3);
        double L, E;
        for (int m = 0; m < numElementos; m++) {
            L = d_L[m];
            E = d_E[m];
            gdl[0] = d_gdl[m, 0];
            gdl[1] = d_gdl[m, 1];
            gdl[2] = d_gdl[m, 2];
            double[,] kke = Ke(E, L);
            KKensablado = stack(KKensablado, kke);
            K = KEnsamblado(K, kke, gdl);
        }
        imprimirM(K, this.dgvK);
    }

    // 2.1 .- FUNCION MATRIZ DE RIGUIDEZ DEL ELEMENTO He
    private static double[,] He(double L, double k, double rho, double g) {
        double[,] he = { { 1, -1 }, { -1, 1 } };
        double r = k / (L * rho * g);
        return multScalarM(he, r);
    }

    // 2.2 .- FUNCION MATRIZ DE RIGUIDEZ GLOBAL H
    private static double[,] HEnsamblado(double[,] H, double[,] He, double[]
        double[] gd1p = { gd1[0], gd1[2] };
        for (int i = 0; i < 2; i++) {
            int ifil = (int)((gd1p[i] / 2) + 0.5);
            for (int j = 0; j < 2; j++) {
                int jcol = (int)((gd1p[j] / 2) + 0.5);
```



```
        H[ifil - 1, jcol - 1] = H[ifil - 1, jcol - 1] + He[i, j];
    }
}
return H;
}
// 2.3 .- PROCEDIMIENTO MATRIZ DE RIGUIDEZ GLOBAL HH
private void HHensablado(double[] d_L, double[] d_E, double[] d_rho, do
double[] gdl = zerosM(3);
double L, k, E, rho;
for (int i = 0; i < numElementos; i++) {
    int m = i;
    L = d_L[m];
    k = d_k[m];
    E = d_E[m];
    rho = d_rho[m];
    gdl[0] = d_gdl[m, 0];
    gdl[1] = d_gdl[m, 1];
    gdl[2] = d_gdl[m, 2];
    double[,] hhe = He(L, k, rho, g);
    HHensablado = stack(HHensablado, hhe);
    H = Hensablado(H, hhe, gdl);
}
imprimirM(H, this.dgvH);
}

// 3.1 .- FUNCION MATRIZ DE RIGUIDEZ DEL ELEMENTO Qe
private static double[,] Qe() {
    double[,] qe = { { -5, -1 }, { 4, -4 }, { 1, 5 } };
    double r = Math.Abs(1.0 / 6.0);
    return multScalarM(qe, r);
}

// 3.2 .- FUNCION MATRIZ DE RIGUIDEZ GLOBAL QQ
private static double[,] QEnsablado(double[,] Q, double[,] Qe, double[]
double[] gdlp = { gdl[0], gdl[2] };
for (int i = 0; i < 3; i++) {
    int ifil = (int)gdl[i];
    for (int j = 0; j < 2; j++) {
        int jcol = (int)((gdlp[j] / 2) + 0.5);
        Q[ifil - 1, jcol - 1] = Q[ifil - 1, jcol - 1] + Qe[i, j];
    }
}
return Q;
}

// 3.3 .- FUNCION MATRIZ DE RIGUIDEZ GLOBAL QQ
private void QQEnsablado(int[,] d_gdl) {
    double[] gdl = zerosM(3);
    for (int m = 0; m < numElementos; m++) {
        gdl[0] = d_gdl[m, 0];
        gdl[1] = d_gdl[m, 1];
        gdl[2] = d_gdl[m, 2];
        double[,] qqe = Qe();
        QQensablado = stack(QQensablado, qqe);
        Q = QEnsablado(Q, qqe, gdl);
    }
    imprimirM(Q, this.dgvQ);
}
}
```



```
#endregion

#region PROCESO DE CALCULO
// 9.- FUNCION CONSOLIDACION1D - FUNCION PARA MODIFICAR BK CONSIDERANDO I
private void BKf() {
    double ft = 86400;
    double theta = 0.5;

    double[] fsigma = zerosM(numNodos);
    double[] vel = zerosM(numNodos);

    double[,] BK1;
    double[,] BK2;

    C[1, 1] = (1 / ft) * carga[1, 1];

    for (int n = 1; n < numCantCargas; n++) {

        C[n, 0] = (1 / ft) * (carga[n, 0]);
        //MessageBox.Show(C[n,0].ToString());
        double dt = carga[n, 0] - carga[n - 1, 0];

        fsigma[numNodos - 1] = -(carga[n, 1] - carga[n - 1, 1]);

        //imprimirME(fsigma, this.dgvMatrizEnsablada2);
        double dtheta = 1 / (dt * theta);

        BK1 = stack(K, multScalarM(tranponerM(Q), dtheta));
        BK2 = stack(multScalarM(Q, dtheta), multScalarM(H, (-1) * dtheta));
        double[,] BK = augment(BK1, BK2);
        //imprimirME(BK, this.dgvMatrizEnsablada);
        double[] Bf = stack(fsigma, multMV(multScalarM(tranponerM(Q), (1

        double[,] Bkm = BKMod(nodinf2, BK, numNodos, numCondi);
        //imprimirME(Bkm, this.dgvMatrizEnsablada2);
        double[] Bfm = BfMod(nodinf2, Bf, numNodos); //*****
        //imprimirM(Bfm, this.dgvMatrizEnsablada2);
        double[] d = xxM(Bkm, Bfm);

        for (int e = 0; e < numCondi; e++) {
            d[numNodos + e] = dtheta * d[numNodos + e];
        }
        for (int e = 0; e < numNodos + numCondi; e++) {
            C[n, e + 1] = C[n - 1, e + 1] + d[e];
        }

        vel = SumaM(multScalarV(vel, (-1) * (1 - theta) / (theta)), multS

    }

    // Porcentaje de consolidacion
```

```
// Calcular vmax
double vmax = -1*C[numCantCargas - 1, numNodos]; // numCantCargas - 1

// Bucle
for (int n = 1; n < numCantCargas; n++) // n = 2 en MATLAB equivale a
{
    C[n, 1 + numNodos + numCondi] = 100 * (-1*C[n, numNodos] / vmax);
}
imprimirM(C, this.dgvTabla);
}

// 7.- FUNCION PARA MODIFICAR BT CONSIDERANDO LAS CONDICIONES DE FRONTERA
private static double[] BfMod(double[,] nodinf, double[] Bf, int numNodos)
double[] Bfmod = Bf;
for (int i = 0; i < numNodos; i++) {
    if (nodinf[i, 1] == 1) {
        Bfmod[i] = 0;
    }
    if (nodinf[i, 2] == 1) {
        int k = (int)(numNodos + 0.5 * i + 0.5);
        Bfmod[k] = 0;
    }
}
return Bfmod;
}

// 8.- FUNCION PARA MODIFICAR BK CONSIDERANDO LAS CONDICIONES DE FRONTERA
private static double[,] BKMod(double[,] nodinf, double[,] BK, int numNodos)
int sbk = numNodos + numCondi;
double[,] BKmod = BK;
for (int i = 0; i < numNodos; i++) {
    if (nodinf[i, 1] == 1) {
        for (int j = 0; j < sbk; j++) {
            BKmod[i, j] = 0;
            BKmod[j, i] = 0;
        }
        BKmod[i, i] = 1;
    }
    if (nodinf[i, 2] == 1) {
        int k = (int)(numNodos + 0.5 * i + 0.5);
        for (int j = 0; j < sbk; j++) {
            BKmod[k, j] = 0;
            BKmod[j, k] = 0;
        }
        BKmod[k, k] = 1;
    }
}
return BKmod;
}
#endregion

#region CARGAR AL INICIAR FORMULARIO

private void Form1_Load(object sender, EventArgs e) {
    ActualizarEncabezadosTablaDatos();
}
```



```
IniciarTablasResultados();
IniciarDatosIniciales();
DesactivarAnalisis();
InicarcboEstilos();
analisis = false;
}

private void cboElementoKe_SelectedIndexChanged(object sender, EventArgs
// Obtener el índice de la fila seleccionada
int selectedIndex = this.cboElementoKe.SelectedIndex;

int inicioFila = selectedIndex * 3 + 1;
int finalFila = inicioFila + 2;

imprimirM(submatrix(KKensablando, inicioFila, finalFila, 0, 2), dgvKe
}
private void cboElementoHe_SelectedIndexChanged(object sender, EventArgs
// Obtener el índice de la fila seleccionada
int selectedIndex = this.cboElementoHe.SelectedIndex;

int inicioFila = selectedIndex * 2 + 1;
int finalFila = inicioFila + 1;

imprimirM(submatrix(HHensablando, inicioFila, finalFila, 0, 1), dgvHe
}
private void cboElementoQe_SelectedIndexChanged(object sender, EventArgs
// Obtener el índice de la fila seleccionada
int selectedIndex = this.cboElementoQe.SelectedIndex;

int inicioFila = selectedIndex * 3 + 1;
int finalFila = inicioFila + 2;

imprimirM(submatrix(QQensablando, inicioFila, finalFila, 0, 1), dgvQe
}
#endregion CARGAR AL INICIAR FORMULARIO

#region ALEBRA MATRICIAL
// Funcion Cracion de cargas
private static double[,] condicionCarga(int n) {
double[,] q = zerosM(n, 2);
for (int i = 0; i < n; i++) {
q[i, 0] = 300000 * (i);
q[i, 1] = 10;
}
q[0, 1] = 0;
return q;
}
private static double[] SumaM(double[] A, double[] B) {
double[] C = zerosM(A.GetLength(0));
for (var i = 0; i < A.GetLength(0); i++) {
C[i] = A[i] + B[i];
}
return C;
}
private static double[,] multM(double[,] A, double[,] B) {
double sum;
```



```
double[,] C = zerosM(A.GetLength(0), B.GetLength(1));
for (var i = 0; i < A.GetLength(0); i++) {
    for (var j = 0; j < B.GetLength(1); j++) {
        sum = 0;
        for (var k = 0; k < B.GetLength(0); k++) {
            sum = sum + A[i, k] * B[k, j];
        }
        C[i, j] = sum;
    }
}
return C;
}
private static double[] xxM(double[,] A, double[] B) {
    int n = A.GetLength(0);
    double[,] C = zerosM(n, n + 1);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            C[i, j] = A[i, j];
        }
        C[i, n] = B[i];
    }
    for (int e = 0; e < n; e++) {
        double t = C[e, e];

        for (int j = 0; j < n + 1; j++) {
            C[e, j] = (C[e, j]) / t;
        }

        for (int i = 0; i < n; i++) {
            if (i != e) {
                t = C[i, e];
                for (var j = 0; j < n + 1; j++) {
                    C[i, j] = C[i, j] - t * C[e, j];
                }
            }
        }

        double[] x = zerosM(n);
        for (var i = 0; i < n; i++) {
            x[i] = C[i, n];
        }
        return x;
    }
}
private static double[,] multScalarM(double[,] A, double scalar) {
    double[,] M = new double[A.GetLength(0), A.GetLength(1)];
    for (int i = 0; i < A.GetLength(0); i++) {
        for (int j = 0; j < A.GetLength(1); j++) {
            M[i, j] = scalar * A[i, j];
        }
    }
    return M;
}
private static double[] multScalarV(double[] A, double scalar) {
    double[] M = new double[A.GetLength(0)];
    for (int i = 0; i < A.GetLength(0); i++) {
        M[i] = scalar * A[i];
    }
}
```




```
        return M;
    }
    private static double[,] submatrix(double[,] A, int ir, int jr, int ic, int jc) {
        double[,] M = zerosM(jr - ir + 1, jc - ic + 1);
        int ik = 0;
        int jk = 0;
        for (int i = ir; i <= jr; i++) {
            jk = 0;
            for (int j = ic; j <= jc; j++) {
                M[ik, jk] = A[i, j];
                jk = jk + 1;
            }
            ik = ik + 1;
        }
        return M;
    }
    private static double[] submatrix(double[] A, int ir, int jr) {
        double[] M = zerosM(jr - ir + 1);
        int ik = 0;
        for (int i = ir; i <= jr; i++) {
            M[ik] = A[i];
            ik = ik + 1;
        }
        return M;
    }
    // PROCEDIMIENTOS DE MATRICES BASICAS
    private static int rows(double[,] A) {
        return A.GetLength(0);
    }
    private static int cols(double[,] A) {
        return A.GetLength(1);
    }
    private static double[] zerosM(int n) {
        double[] V = new double[n];
        for (int i = 0; i < n; i++) {
            V[i] = 0;
        }
        return V;
    }
    private static double[,] zerosM(int n, int m) {
        double[,] M = new double[n, m];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                M[i, j] = 0;
            }
        }
        return M;
    }
    private void imprimirM(double[,] M, DataGridView Tabla) {
        // Limpiamos la tabla
        Tabla.Rows.Clear();
        Tabla.Columns.Clear();
        // CanumCantCargasdad Filas y Columnas
        Tabla.ColumnCount = M.GetLength(1);
        Tabla.RowCount = M.GetLength(0);
        for (int i = 0; i < M.GetLength(0); i++) {
            for (int j = 0; j < M.GetLength(1); j++) {
                Tabla.Rows[i].Cells[j].Value = M[i, j];
            }
        }
    }
}
```



```
    }  
  }  
}  
private void imprimirM(double[] V, DataGridView Tabla) {  
    // Limpiamos la tabla  
    Tabla.Rows.Clear();  
    Tabla.Columns.Clear();  
    // CanumCantCargasdad Filas y Columnas  
    Tabla.ColumnCount = 1;  
    Tabla.RowCount = V.Length;  
    for (int i = 0; i < V.Length; i++) {  
        Tabla.Rows[i].Cells[0].Value = V[i];  
    }  
}  
  
private void limpiarTabla(DataGridView Tabla, int numRows, int numColumns)  
    // Limpiamos la tabla  
    Tabla.Rows.Clear();  
    Tabla.Columns.Clear();  
    // CanumCantCargasdad Filas y Columnas  
    Tabla.RowCount = numRows;  
    Tabla.ColumnCount = numColumns;  
}  
  
#endregion  
  
#region DIBUJAR ESTRATOS, ELEMENTOS, NODOS  
  
private void DibujoSuelosNodosElementos() {  
    double wLienzo; double hLienzo; // Dimensiones del Lienzo  
    double scala = 10; // multiplicado por 10 la  
  
    // Limpiamos el Lienzo de trabajo  
    DibujarEliminarLienzo();  
  
    // Creacion del Lienzo de trabajo  
    Graphics Lienzo = this.ptbGrafico1.CreateGraphics();  
  
    // Dimensiones del Lienzo de trabajo  
    wLienzo = this.ptbGrafico1.Size.Width;  
    hLienzo = this.ptbGrafico1.Size.Height-5;  
  
    // Dimensiones basicas  
    double XX = 0.5 * wLienzo;  
    double YY = 0.3 * hLienzo;  
  
    // Informacion del lienzo  
    this.lblTamanoWH.Text = "Ancho=" + wLienzo + "\nAlto=" + hLienzo;  
  
    // Fuente de trabajo  
    Font fuente = new Font("Arial", 11);  
  
    // Colores Hexadecimales  
    Color colorMarco = ColorTranslator.FromHtml("#333333");  
    Color colorCielo = ColorTranslator.FromHtml("#87CEEB");  
    Color colorTierra = ColorTranslator.FromHtml("#D2B48C");  
    Color colorSuperficie = ColorTranslator.FromHtml("#333333");
```



```
Color colorBase = ColorTranslator.FromHtml("#333333");
Color colorCota = ColorTranslator.FromHtml("#0000FF");
Color colorTextoCota = ColorTranslator.FromHtml("#0000FF");
Color colorCarga = ColorTranslator.FromHtml("#FF0000");
Color colorNodo = ColorTranslator.FromHtml("#87CEEB");
Color colorLinea = ColorTranslator.FromHtml("#87CEEB");
Color colorElemento = ColorTranslator.FromHtml("#87CEEB");

// Fodo del trabajo color cielo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, YY, 1, colorCielo, true);

// Terreno
DibujarRectangulo(Lienzo, 0, YY, wLienzo, hLienzo-YY, 0, colorTierra);

// Superficie y base del terreno
DibujarLinea(Lienzo, 0, YY, wLienzo, YY, 5, colorSuperficie);
DibujarLinea(Lienzo, 0, hLienzo, wLienzo, hLienzo, 10, colorBase);

// Marco Lienzo de trabajo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, hLienzo, 2, colorMarco, false);

// Generacion Elementos
// Datos canumCantCargasdad de Elementos, Nodos, Restriciones, fronte
numElementos = this.dgvTablaElementos.Rows.Count; // numElementos
numNodos = 2 * numElementos + 1; // numNodos
numCondi = numNodos - numElementos;

double[] hElemento = zerosM(numElementos);
double HElemento = 0;
for(int i=0; i<numElementos; i++) {
    hElemento[i]=Convert.ToDouble(this.dgvTablaElementos.Rows[i].Cell
    HElemento = HElemento + hElemento[i];
}

// Dibujamos los Nodos, Elementos, Medidas
DibujarEstratos(Lienzo, hElemento, 0, YY, wLienzo, hLienzo, 2, fuente);

// Medir la profundidad
DibujarAcotar(Lienzo, wLienzo - 50, YY, wLienzo - 50, hLienzo, 2, colorCota);
InsertarTexto(Lienzo, HElemento.ToString()+" m", wLienzo - 45, 0.65 * hLienzo);

// Cargas Distribuidas y Puntuales
DibujarCargaPuntual(Lienzo, XX, YY, XX, YY - 150, 15, 4, colorCarga);
InsertarTexto(Lienzo, "", XX + 5, YY - 150, fuente, colorCarga);
DibujarCargaDistribuida(Lienzo, wLienzo / 3, YY, 2 * wLienzo / 3, YY, 10, colorCarga);
InsertarTexto(Lienzo, "20 KPa", 2 * wLienzo / 3 - 40, YY - 70, fuente, colorCarga);
}
private void DibujoSuelosNodosElementos(int Elemento_Seleccionado) {
    double wLienzo; double hLienzo; // Dimensiones del Lienzo
    double scala = 10; // multiplicado por 10 la
}

// Limpiarnos el Lienzo de trabajo
```



```
DibujarEliminarLienzo();

// Creacion del Lienzo de trabajo
Graphics Lienzo = this.ptbGrafico1.CreateGraphics();

// Dimensiones del Lienzo de trabajo
wLienzo = this.ptbGrafico1.Size.Width;
hLienzo = this.ptbGrafico1.Size.Height - 5;

// Dimensiones basicas
double XX = 0.5 * wLienzo;
double YY = 0.3 * hLienzo;

// Informacion del lienzo
this.lblTamanoWH.Text = "Ancho=" + wLienzo + "\nAlto=" + hLienzo;

// Fuente de trabajo
Font fuente = new Font("Arial", 11);

// Colores Hexadecimales
Color colorMarco = ColorTranslator.FromHtml("#333333");
Color colorCielo = ColorTranslator.FromHtml("#87CEEB");
Color colorTierra = ColorTranslator.FromHtml("#D2B48C");
Color colorSuperficie = ColorTranslator.FromHtml("#333333");
Color colorBase = ColorTranslator.FromHtml("#333333");
Color colorCota = ColorTranslator.FromHtml("#0000FF");
Color colorTextoCota = ColorTranslator.FromHtml("#0000FF");
Color colorCarga = ColorTranslator.FromHtml("#FF0000");
Color colorNodo = ColorTranslator.FromHtml("#87CEEB");
Color colorLinea = ColorTranslator.FromHtml("#87CEEB");
Color colorElemento = ColorTranslator.FromHtml("#87CEEB");

// Fondo del trabajo color cielo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, YY, 1, colorCielo, true);

// Terreno
DibujarRectangulo(Lienzo, 0, YY, wLienzo, hLienzo - YY, 0, colorTierra);

// Superficie y base del terreno
DibujarLinea(Lienzo, 0, YY, wLienzo, YY, 5, colorSuperficie);
DibujarLinea(Lienzo, 0, hLienzo, wLienzo, hLienzo, 10, colorBase);

// Marco Lienzo de trabajo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, hLienzo, 2, colorMarco, false);

// Generacion Elementos
// Datos canumCantCargasdad de Elementos, Nodos, Restricciones, fronte
numElementos = this.dgvTablaElementos.Rows.Count; // numElementos
numNodos = 2 * numElementos + 1; // numNodos
numCondi = numNodos - numElementos;

double[] hElemento = zerosM(numElementos);
double HElemento = 0;
for (int i = 0; i < numElementos; i++) {
```



```
        hElemento[i] = Convert.ToDouble(this.dgvTablaElementos.Rows[i].Cells[1].Text);
        HElemento = HElemento + hElemento[i];
    }

    // Dibujamos los Nodos, Elementos, Medidas
    //DibujarEstratos(Lienzo, hElemento, 0, YY, wLienzo, hLienzo, 2, fuente);

    DibujarEstratosSeleccionado(Lienzo, hElemento, 0, YY, wLienzo, hLienzo, fuente);

    // Medir la profundidad
    DibujarAcotar(Lienzo, wLienzo - 60, YY, wLienzo - 60, hLienzo, 2, fuente);
    InsertarTexto(Lienzo, HElemento.ToString() + " m", wLienzo - 55, 0.65 * hLienzo, fuente);

    // Cargas Distribuidas y Puntuales
    DibujarCargaPuntual(Lienzo, XX, YY, XX, YY - 150, 15, 4, colorCarga);
    InsertarTexto(Lienzo, "", XX + 5, YY - 150, fuente, colorCarga);
    DibujarCargaDistribuida(Lienzo, wLienzo / 3, YY, 2 * wLienzo / 3, YY, 2, fuente);
    InsertarTexto(Lienzo, "20 KPa", 2 * wLienzo / 3 - 40, YY - 70, fuente);
}

private void DibujarEstratosSeleccionado(Graphics Lienzo, double[] Estratos, double x1, double y1, double x2, double y2)
{
    // X1, Y1, X2, Y2: son las cordenedados donde se presentan todos los estratos
    Font fuenteEstrato = new Font("Arial", 11, FontStyle.Italic);
    Font fuenteNodos = new Font("Arial", 10, FontStyle.Italic);
    Font fuenteCota = fuente;

    string[] coloresTierra = {
        "#D2B48C", // Tan
        "#BC8F8F", // Rosy Brown
        "#8B6969", // Rosy Brown
        "#F4A460", // Sandy Brown
        "#DAA520", // Goldenrod
    }; // 9 colores de tierra

    double espacio = 0;
    double Y0;
    double altura = 0;

    for (int i = 0; i < Estratos.Length; i++) {
        altura += Estratos[i];
    }

    double xCentro = x2 / 2;
    double proporcioAltura = Math.Abs(y2 - y1) / altura;
    Y0 = y1 + altura * proporcioAltura;
    double qq = 0;

    // PARTE ELEMENTOS
    for (int i = 0; i < Estratos.Length; i++) {
        espacio = Estratos[i] * proporcioAltura;

        Color color = ColorTranslator.FromHtml(coloresTierra[i]);

        DibujarRectangulo(Lienzo, x1, Y0 - espacio, x2, Y0 + espacio, color, grosor);
    }
}
```



```
DibujarLinea(Lienzo, x1, Y0, x2, Y0, 1, Color.Blue);
DibujarLinea(Lienzo, x1, Y0 - espacio, x2, Y0 - espacio, 1, Color

InsertarTexto(Lienzo, "Ele. " + (i + 1), x2 - 120, Y0 - espacio +

DibujarAcotar(Lienzo, x1 + 20, Y0 - espacio, x1 + 20, Y0, grosor,

InsertarTexto(Lienzo, Estratos[i] + " m", x1 + 20, Y0 - espacio /

Y0 = Y0 - espacio;
}

DibujarLineaEntrecortada(Lienzo, xCentro, y1, xCentro, y2, grosor, Co

Y0 = y1 + altura * proporcioAltura;
// PARTE NODOS
for (int i = 0; i < Estratos.Length; i++) {
    espacio = Estratos[i] * proporcioAltura;

    Color color = ColorTranslator.FromHtml(coloresTierra[i]);

    if (i == ElementoSeleccionado) {
        DibujarRectanguloEntrecortado(Lienzo, x1, Y0 - espacio, x2, e
    }
    if (i == ElementoSeleccionado) {
        DibujarLineaEntrecortada(Lienzo, x1, Y0, x2, Y0, 2, Color.White
        DibujarLineaEntrecortada(Lienzo, x1, Y0 - espacio, x2, Y0 - e
    }

    // Parte Nodos

    DibujarPunto(Lienzo, xCentro, Y0, 6, Color.BlanchAlmond);
    InsertarTexto(Lienzo, (2 * i + 1).ToString(), xCentro + 10, Y0 -
    if (i == ElementoSeleccionado) {
        DibujarPunto(Lienzo, xCentro, Y0, 6, Color.Blue);
        DibujarCirculos(Lienzo, xCentro, Y0, 8, 1, Color.White, false
    }
    if (i - 1 == ElementoSeleccionado) {
        DibujarPunto(Lienzo, xCentro, Y0, 6, Color.Blue);
        DibujarCirculos(Lienzo, xCentro, Y0, 8, 1, Color.White, false
    }

    DibujarPunto(Lienzo, xCentro, Y0 - espacio / 2, 4, Color.BlanchAlmond);
    InsertarTexto(Lienzo, (2 * i + 2).ToString(), xCentro - 25, Y0 -
    if (i == ElementoSeleccionado) {
        DibujarPunto(Lienzo, xCentro, Y0 - espacio / 2, 4, Color.Red);
        DibujarCirculos(Lienzo, xCentro, Y0 - espacio / 2, 6, 1, Color
    }

    Y0 = Y0 - espacio;
```



```
    }

    DibujarPunto(Lienzo, xCentro, Y0, 6, Color.BlanchedAlmond);
    InsertarTexto(Lienzo, (Estratos.Length * 2 + 1).ToString(), xCentro +
if (ElementoSeleccionado+1 == Estratos.Length) {
    DibujarPunto(Lienzo, xCentro, Y0, 6, Color.Blue);
    DibujarCirculos(Lienzo, xCentro, Y0, 8, 2, Color.White, false);
}
}
private void DibujarEstratos(Graphics Lienzo, double[] Estratos, double x
// X1, Y1, X2, Y2: son las cordenados donde se presentan todos los es
Font fuenteEstrato = new Font("Arial", 11, FontStyle.Italic);
Font fuenteNodos = new Font("Arial", 10, FontStyle.Italic);
Font fuenteCota = fuente;

string[] coloresTierra = {
    "#D2B48C", // Tan
    "#BC8F8F", // Rosy Brown
    "#8B6969", // Rosy Brown
    "#F4A460", // Sandy Brown
    "#DAA520" // Goldenrod
}; // 9 colores de tierra

double espacio = 0;
double Y0;
double altura = 0;

for (int i = 0; i < Estratos.Length; i++) {
    altura += Estratos[i];
}

double xCentro = x2 / 2;
double proporcioAltura = Math.Abs(y2 - y1) / altura;
Y0 = y1+altura*proporcioAltura;
double qq = 0;
for (int i = 0; i < Estratos.Length; i++) {
    espacio = Estratos[i] * proporcioAltura;

    Color color = ColorTranslator.FromHtml(coloresTierra[i]);

    DibujarRectangulo(Lienzo, x1, Y0-espacio, x2, espacio, grosor, color);

    DibujarLinea(Lienzo, x1, Y0, x2, Y0, 1, Color.Blue);
    DibujarLinea(Lienzo, x1, Y0-espacio, x2, Y0-espacio, 1, Color.Blue);

    InsertarTexto(Lienzo, "Ele. " + (i + 1), x2 - 80, Y0-espacio + 2);

    DibujarAcotar(Lienzo, x1 + 20, Y0-espacio, x1 + 20, Y0, grosor, Color.Black);

    InsertarTexto(Lienzo, Estratos[i] + " m", x1 + 20, Y0-espacio/2-5);

    // Parte Nodos

    DibujarLinea(Lienzo, xCentro, y1, xCentro, y2, grosor, Color.White);

    DibujarPunto(Lienzo, xCentro, Y0, 6, Color.BlanchedAlmond);
```



```
        InsertarTexto(Lienzo, (2*i + 1).ToString(), xCentro + 10, Y0 - 20);

        DibujarPunto(Lienzo, xCentro, Y0-espacio/2, 4, Color.BlanchedAlmond);
        InsertarTexto(Lienzo, (2*i + 2).ToString(), xCentro -25, Y0 - espacio);

        Y0 = Y0 - espacio;

    }

    DibujarPunto(Lienzo, xCentro, Y0, 6, Color.BlanchedAlmond);
    InsertarTexto(Lienzo, (Estratos.Length*2+1).ToString(), xCentro + 10, Y0);

}

private void dibujo1() {
    // Si no existe el Bitmap, créalo con las dimensiones del PictureBox
    if ( drawingBitmap == null ) {
        drawingBitmap = new Bitmap(this.ptbGrafico1.Width,this.ptbGrafico1.Height);
    }

    double wLienzo; double hLienzo;          // Dimensiones del Lienzo
    double scala = 10;                       // multiplicado por 10 la escala

    // Limpiamos el Lienzo de trabajo
    DibujarEliminarLienzo();

    // Creacion del Lienzo de trabajo
    // Crea un objeto Graphics a partir del Bitmap
    Graphics Lienzo = Graphics.FromImage(drawingBitmap);

    //Graphics Lienzo = this.ptbGrafico1.CreateGraphics();

    // Dimensiones del Lienzo de trabajo
    wLienzo = this.ptbGrafico1.Size.Width;
    hLienzo = this.ptbGrafico1.Size.Height;

    // Dimensiones basicas
    double XX = 0.5 * wLienzo;
    double YY = 0.3 * hLienzo;

    // Informacion del lienzo
    this.lblTamanoWH.Text = "Ancho=" + wLienzo + "\nAlto=" + hLienzo;

    // Fuente de trabajo
    Font fuente = new Font("Arial", 14);

    // Colores Hexadecimales
    Color colorMarco = ColorTranslator.FromHtml("#333333");
    Color colorCielo = ColorTranslator.FromHtml("#87CEEB");
    Color colorTierra = ColorTranslator.FromHtml("#D2B48C");
    Color colorSuperficie = ColorTranslator.FromHtml("#333333");
    Color colorBase = ColorTranslator.FromHtml("#333333");
    Color colorCota = ColorTranslator.FromHtml("#0000FF");
    Color colorTextoCota = ColorTranslator.FromHtml("#0000FF");
    Color colorCarga = ColorTranslator.FromHtml("#FF0000");
}
```




```
// Fodo del trabajo color cielo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, hLienzo, 1, colorCielo, true);

// Marco Lienzo de trabajo
DibujarRectangulo(Lienzo, 0, 0, wLienzo, hLienzo, 2, colorMarco, false);

// Terreno
DibujarRectangulo(Lienzo, 0, YY, wLienzo, 0.7 * hLienzo, 0, colorTerreno);

// Superficie y base del terreno
DibujarLinea(Lienzo, 0, YY, wLienzo, YY, 5, colorSuperficie);
DibujarLinea(Lienzo, 0, hLienzo, wLienzo, hLienzo, 10, colorBase);

// Medir la profundidad
DibujarAcotar(Lienzo, wLienzo - 50, YY, wLienzo - 50, hLienzo, 3, colorAcotar);
InsertarTexto(Lienzo, "5 m", wLienzo - 45, 0.65 * hLienzo, fuente, colorAcotar);

// Cargas Distribuidas y Puntuales
DibujarCargaPuntual(Lienzo, XX, YY, XX, YY - 150, 15, 4, colorCarga);
InsertarTexto(Lienzo, "150 Tonf", XX + 5, YY - 150, fuente, colorCarga);
DibujarCargaDistribuida(Lienzo, wLienzo / 3, YY, 2 * wLienzo / 3, YY, 10, 4, colorCarga);
InsertarTexto(Lienzo, "10 Tonf/m", 2 * wLienzo / 3 - 40, YY - 70, fuente, colorCarga);

// Generacion de estratos del suelo
double[] Estratos = { 8, 6, 7, 8, 9, 10, 5 };
DibujarEstratos(Lienzo, Estratos, 0, YY, wLienzo, hLienzo, 2, fuente, colorEstratos);

// Nodos del Analisis de suelos
// Nodos del Analisis del agua
//DibujarNodos(Lienzo, Estratos, wLienzo / 2, 0.3 * hLienzo, wLienzo / 2, hLienzo, 2, fuente, colorNodos);

// Asigna el Bitmap al PictureBox para mostrarlo
this.ptbGrafico1.Image = drawingBitmap;
}

private void DibujoEdificio() {
    byte nPisos = 5;
    byte nColumnas = 4;
    byte yCol = 20;
    byte xCol = 30;

    // Creacion del Lienzo de trabajo
    Graphics Lienzo = this.ptbGrafico1.CreateGraphics();

    // Dimensiones del Lienzo de trabajo
    int wLienzo = this.ptbGrafico1.Size.Width;
    int hLienzo = this.ptbGrafico1.Size.Height;

    Color colorEdificio = ColorTranslator.FromHtml("#333333");

    for (int i = 0; i < nColumnas; i++) {
        for (int j = 0; j < nPisos; j++) {
            DibujarLinea(Lienzo, XX + i * yCol, YY - j * yCol, XX + i * yCol, YY - (j + 1) * yCol, 1, colorEdificio);
            DibujarLinea(Lienzo, XX + i * xCol, YY - j * xCol, XX + (i + 1) * xCol, YY - j * xCol, 1, colorEdificio);
            DibujarPunto(Lienzo, XX + i * xCol, YY - j * xCol, 5, colorEdificio);
        }
    }
}
```



```
}
private void DibujarEliminarLienzo() {
    // Dimensiones del Lienzo de trabajo
    double wLienzo = this.ptbGrafico1.Size.Width;
    double hLienzo = this.ptbGrafico1.Size.Height;
    // Creacion del Lienzo de trabajo
    Graphics Lienzo = this.ptbGrafico1.CreateGraphics();
}
void DibujoMapaColores() {
    // MAPA DE COLORES
    // Creacion del Lienzo de trabajo
    Graphics Lienzo = this.ptbGrafico1.CreateGraphics();

    string[] coloresHex = {
        "#000080", "#000096", "#0000ad", "#0000c4", "#0000db", "#0000f2",
        "#0079ff", "#0091ff", "#00a9ff", "#00c1ff", "#00d9ff", "#00f1ff",
        "#00ff92", "#00ff7a", "#00ff62", "#00ff4a", "#00ff32", "#00ff1a",
        "#6dff00", "#85ff00", "#9dff00", "#b5ff00", "#cdf00", "#e5ff00",
        "#ffa100", "#ff8981", "#ff7181", "#ff5981", "#ff4181", "#ff2981",
        "#c80000"
    };

    for (int i = 0; i < coloresHex.Length; i++) {
        Color color = ColorTranslator.FromHtml(coloresHex[i]);
        DibujarPunto(Lienzo, 10 + i * 6, 50, 10, color);
    }
}

#endregion

#region DIBUJO DE EN PICTUREBOX

// SECCION DE FUNCIONES Y PROCEDIMIENTOS

private void DibujarCargaDistribuida(Graphics Lienzo, double x1, double y1,
    double espacio = Math.Abs(x2 - x1) / num_Cargas;
    for (int i = 0; i <= num_Cargas; i++) {
        DibujarCargaPuntualEntrecortada(Lienzo, x1 + espacio * i, y1, x1 + espacio * i, y1);
    }
    DibujarLineaEntrecortada(Lienzo, x1, y1 - 50, x2, y2 - 50, grosor, color);
    DibujarLinea(Lienzo, x1, y1, x2, y2, 1, color);
}

private void InsertarTexto(Graphics Lienzo, string texto, double x, double y,
    Brush brush = new SolidBrush(color);
    Lienzo.DrawString(texto, fuente, brush, (int)x, (int)y);
}

private void DibujarLineaEntrecortada(Graphics Lienzo, double x1, double y1, double x2, double y2,
    using (Pen pen = new Pen(color, grosor)) {
        pen.DashStyle = System.Drawing.Drawing2D.DashStyle.Dash;
        Lienzo.DrawLine(pen, (int)x1, (int)y1, (int)x2, (int)y2);
    }
}

private void DibujarCuadrado(Graphics Lienzo, double x1, double y1, double x2, double y2,
    Pen pen = new Pen(color, grosor);
    Rectangle xy = new Rectangle((int)x1, (int)y1, (int)lado, (int)lado);
}
```



```
Lienzo.DrawRectangle(pen, xy);
if (relleno == true) {
    Brush brush = new SolidBrush(color);
    Lienzo.FillRectangle(brush, xy);
}
}
private void DibujarRectangulo(Graphics Lienzo, double x1, double y1, do
    Pen pen = new Pen(color, (int)grosor);
    Rectangle xy = new Rectangle((int)x1, (int)y1, (int)ancho, (int)alto);
    Lienzo.DrawRectangle(pen, xy);
    if (relleno == true) {
        Brush brush = new SolidBrush(color);
        Lienzo.FillRectangle(brush, xy);
    }
}
private void DibujarRectanguloEntrecortado(Graphics Lienzo, double x1, do
    Pen pen = new Pen(color, (int)grosor);
    Rectangle xy = new Rectangle((int)x1, (int)y1, (int)ancho, (int)alto);
    pen.DashStyle = System.Drawing.Drawing2D.DashStyle.Dash;
    Lienzo.DrawRectangle(pen, xy);
    if (relleno == true) {
        Brush brush = new SolidBrush(color);
        Lienzo.FillRectangle(brush, xy);
    }
}

#endregion

#region COMANDOS EXTRA - MAXIMIZAR Y MINIMIZAR

private void Form1_Resize(object sender, EventArgs e) {
    dibujo1();
}

#endregion COMANDOS EXTRA - MAXIMIZAR Y MINIMIZAR

#region TRABAJO CON EN CHART1, CHART2, CHART3

private void cboTipoGraficaDesplazamiento_SelectedIndexChanged(object send
    // GRAFICAR RESULTADOS
    // 1: Grafica Total          2: Grafica de Nodos Impar          3: Grafic
    //Convert.ToDouble(this.dgvTablaRestriciones.Rows[i].Cells[1].Value);
    if ( analisis == true) {
        GraficaDesplazamiento((int)this.cboTipoGraficaDesplazamiento.Sele
        chart1.MouseMove += Chart1_MouseMove;
    }
}

private void GraficaPresiones(int TipoGrafica) {
    // 1: Grafica Total          2: Grafica de Nodos Impar          3: Grafic
```



```
this.chart2.Series.Clear();
MessageBox.Show(TipoGrafica.ToString());
// 1: Grafica Total
if (TipoGrafica == 1) {
    for (int ee = 0; ee < numCondi; ee++) {
        Series nodo = new Series {
            Name = "Nodo " + (ee + 1).ToString(),
            ChartType = SeriesChartType.Line,
            BorderWidth = 2
        };
        this.chart2.Series.Add(nodo);
        for (int i = 0; i < 20; i++) {
            nodo.Points.AddXY(mtiempo[i, 0], mPresiones[i, ee]);
        }
    }
}

private void GraficaIsocronas(int TipoGrafica) {

    // 1: Grafica Total          2: Grafica de Nodos Impar          3: Grafic

    this.chart3.Series.Clear();
    MessageBox.Show(TipoGrafica.ToString());
    // 1: Grafica Total
    if (TipoGrafica == 1) {
        for (int ee = 0; ee < numNodos; ee++) {
            Series nodo = new Series {
                Name = "Nodo " + (ee + 1).ToString(),
                ChartType = SeriesChartType.Line,
                BorderWidth = 2
            };
            this.chart3.Series.Add(nodo);
            for (int i = 0; i < 20; i++) {
                nodo.Points.AddXY(mtiempo[i, 0], mDesplazamientos[i, ee]);
            }
        }
    }

    // 2: Grafica de un solo

    // Dar Formato a la grafica
    this.chart3.ChartAreas[0].AxisX.Title = "Tiempo (s)";
    this.chart3.ChartAreas[0].AxisY.Title = "Presion (KPa)";

    this.chart3.Titles.Clear();
    this.chart3.Titles.Add("Gráfica de las Isocronas");
    this.chart3.Titles[0].Font = new Font("Arial", 16, FontStyle.Bold);
}
#endregion

#region CINTA OPCIONES

/* CINTA DE OPCIONES - ARCHIVOS */
private void NuevoToolStripMenuItem_Click(object sender, EventArgs e) {
    // Limpiar las tablas de los datos
}
```



```
limpiarTabla(this.dgvTablaMateriales,1,5);
limpiarTabla(this.dgvTablaElementos,3,3);
limpiarTabla(this.dgvTablaRestriciones,3,3);
limpiarTabla(this.dgvTablaCargas,2,3);

// Cargar Datos iniciales

}
private void AbrirToolStripMenuItem_Click(object sender,EventArgs e) {
    // Utiliza OpenFileDialog para seleccionar el archivo JSON
    using ( OpenFileDialog openFileDialog = new OpenFileDialog() ) {
        openFileDialog.Filter = "Archivo JSON (*.json)|*.json";
        openFileDialog.Title = "Abrir archivo JSON";

        // Mostrar el diálogo y cargar si el usuario selecciona un archivo
        if ( openFileDialog.ShowDialog() == DialogResult.OK ) {
            try {
                // Obtener la ruta completa del archivo seleccionado
                string rutaArchivo = openFileDialog.FileName;

                // Leer el contenido del archivo
                string jsonString = File.ReadAllText(rutaArchivo);

                // Cargar el JSON en un objeto JObject
                var jsonObject = JObject.Parse(jsonString);

                // Mostrar la información en los TextBox
                //txtNombre.Text = jsonObject["Nombre"]?.ToString();

                // Convertir el JArray a una matriz de enteros y mostrarla
                string[,] matrizMateriales = ConvertirJArrayAMatriz(jsonObject["Materiales"]);
                string[,] matrizElementos = ConvertirJArrayAMatriz(jsonObject["Elementos"]);
                string[,] matrizRestriciones = ConvertirJArrayAMatriz(jsonObject["Restriciones"]);
                string[,] matrizCargas = ConvertirJArrayAMatriz(jsonObject["Cargas"]);

                // Mostrar la matriz cargada en las tablas
                imprimirM(matrizMateriales,this.dgvTablaMateriales);
                imprimirM(matrizElementos,this.dgvTablaElementos);
                imprimirM(matrizRestriciones,this.dgvTablaRestriciones);
                imprimirM(matrizCargas,this.dgvTablaCargas);
                ActualizarEncabezadosTablaDatos();
                MessageBox.Show($"Se cargo el proyecto sin problemas");
            } catch ( Exception ex ) {
                MessageBox.Show($"Error al cargar la información: {ex.Message}");
            }
        }
    }
}
private void guardarToolStripMenuItem_Click(object sender,EventArgs e) {
    // Guardar Materiales
    string[,] matrizMateriales = obtenerDatosTabla(this.dgvTablaMateriales);

    // Guardar Elementos
    string[,] matrizElementos = obtenerDatosTabla(this.dgvTablaElementos);

    // Guardar Restriciones
    string[,] matrizRestriciones = obtenerDatosTabla(this.dgvTablaRestriciones);
}
```



```
// Guardar Cargas
string[,] matrizCargas = obtenerDatosTabla(this.dgvTablaCargas);

// Crear un objeto JSON usando JObject
var jsonObject = new JObject {
    ["proyecto"] = "Análisis consolidatorio usando 5 elementos",
    ["fecha"] = "3-11-2024",
    ["modelador"] = "Denis Rudy Lupaca Quispe",
    ["matrizMateriales"] = ConvertirMatrizAJArray(matrizMateriales),
    ["matrizElementos"] = ConvertirMatrizAJArray(matrizElementos),
    ["matrizRestriciones"] = ConvertirMatrizAJArray(matrizRestriciones),
    ["matrizCargas"] = ConvertirMatrizAJArray(matrizCargas)
};

// Utiliza SaveFileDialog para seleccionar la ruta del archivo JSON
using ( SaveFileDialog saveFileDialog = new SaveFileDialog() ) {
    saveFileDialog.Filter = "Archivo JSON (*.json)|*.json";
    saveFileDialog.Title = "Guardar archivo JSON";
    // Mostrar el diálogo y guardar si el usuario selecciona "Guardar"
    if ( saveFileDialog.ShowDialog() == DialogResult.OK ) {
        try {
            // Obtener la ruta completa del archivo seleccionado
            string rutaArchivo = saveFileDialog.FileName;
            // Escribir el archivo JSON en la ruta seleccionada
            File.WriteAllText(rutaArchivo, jsonObject.ToString());
            MessageBox.Show("Se guardo correctamente el modelo.");
        } catch ( Exception ex ) {
            MessageBox.Show($"Error al guardar el modelo: {ex.Message}");
        }
    }
}

private void salirToolStripMenuItem_Click(object sender, EventArgs e) {
    // Muestra un cuadro de diálogo de confirmación
    DialogResult result = MessageBox.Show("¿Está seguro que desea salir?");

    // Si el usuario elige 'Si', se cierra el formulario
    if ( result == DialogResult.Yes ) {
        salir = true;
        this.Close(); // Cierra el formulario actual
    }
    // Si elige 'No', el formulario permanece abierto
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e) {
    // Muestra el cuadro de diálogo de confirmación
    if ( salir == false ) {
        DialogResult result = MessageBox.Show("¿Está seguro que desea salir?");

        // Si el usuario elige 'No', cancela el cierre del formulario
        if ( result == DialogResult.No ) {
            e.Cancel = true; // Cancela el evento de cierre
        }
    }
}

/* CINTA DE OPCIONES - VER */
private void panumElementosDatosToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControl1.Visible = !this.TabControl1.Visible; // Cambia la visibilidad
}
```



```
        if ( this.TabControl1.Visible == true ) {
            this.panumElementosDatosToolStripMenuItem.Text = "Ocultar Datos";
        } else {
            this.panumElementosDatosToolStripMenuItem.Text = "Mostrar Datos";
        }
        dibujo1();
    }
    private void panumElementosDeResultadosToolStripMenuItem_Click(object sender, EventArgs e) {
        this.TabControlResultados.Visible = !this.TabControlResultados.Visible;
        if ( this.TabControlResultados.Visible == true ) {
            this.panumElementosDeResultadosToolStripMenuItem.Text = "Ocultar Datos";
        } else {
            this.panumElementosDeResultadosToolStripMenuItem.Text = "Mostrar Datos";
        }
        dibujo1();
    }
    /* CINTA DE OPCIONES - DEFINIR */
    private void propiedadesToolStripMenuItem_Click(object sender, EventArgs e) {
        this.TabControl1.SelectedIndex = 0;
        if ( this.TabControl1.Visible == true ) {
            this.panumElementosDatosToolStripMenuItem.Text = "Mostrar Datos";
        } else {
            this.TabControl1.Visible = true;
            this.panumElementosDatosToolStripMenuItem.Text = "Ocultar Datos";
        }
        dibujo1();
    }
    private void elementosToolStripMenuItem_Click(object sender, EventArgs e) {
        this.TabControl1.SelectedIndex = 1;
        if ( this.TabControl1.Visible == true ) {
            this.panumElementosDatosToolStripMenuItem.Text = "Mostrar Datos";
        } else {
            this.TabControl1.Visible = true;
            this.panumElementosDatosToolStripMenuItem.Text = "Ocultar Datos";
        }
        dibujo1();
    }
    private void restriccionesToolStripMenuItem_Click(object sender, EventArgs e) {
        this.TabControl1.SelectedIndex = 2;
        if ( this.TabControl1.Visible == true ) {
            this.panumElementosDatosToolStripMenuItem.Text = "Mostrar Datos";
        } else {
            this.TabControl1.Visible = true;
            this.panumElementosDatosToolStripMenuItem.Text = "Ocultar Datos";
        }
        dibujo1();
    }
    private void cargasToolStripMenuItem_Click(object sender, EventArgs e) {
        this.TabControl1.SelectedIndex = 3;
        if ( this.TabControl1.Visible == true ) {
            this.panumElementosDatosToolStripMenuItem.Text = "Mostrar Datos";
        } else {
            this.TabControl1.Visible = true;
            this.panumElementosDatosToolStripMenuItem.Text = "Ocultar Datos";
        }
        dibujo1();
    }
}
```



```
/* CINTA DE OPCIONES - ANALISIS */
private void ejecutarToolStripMenuItem_Click(object sender,EventArgs e) {
    // llamanda al boton ejecutar
    EnviarInformacionForm1("tab-Ejecutar","");
}
private void informeDeEjecuciónToolStripMenuItem_Click(object sender,EventArgs e) {
    // se obtubo un analisis de 3 nodos 5 elementos
    // es como un reporte de los resultlados que se tuvieron
    EnviarInformacionForm1("tab-InfEjecutar","");
}

/* CINTA DE OPCIONES - REPORTE */
private void generarLatexToolStripMenuItem_Click(object sender,EventArgs e) {
    // mandamos las tablas de datos y resultado, tambien informaciones
    // a un nuevo formulario
    //EnviarInformacionForm1("tab-EjecutarLatex","");
    this.TabControlResultados.SelectedIndex = 7;
}
private void cómoEjecutarEnLaTeXToolStripMenuItem_Click(object sender,EventArgs e) {
    // hacemos paso a paso de como generar y copiar el codigo a latex
    // asi como llevamos la carpeta img, a proyecto latex
    EnviarInformacionForm1("tab-ComoEjcutarLatex","");
}
private void generarExcelToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 8;
}

/* CINTA DE OPCIONES - INFORMACION */
private void sobreSueloMEFTToolStripMenuItem_Click(object sender,EventArgs e) {
    EnviarInformacionForm1("tab-SobrePrograma","");
}
private void asesorToolStripMenuItem_Click(object sender,EventArgs e) {
    // Informacion del asesor del trabajo
    // Dr. Ricardo Romero
    EnviarInformacionForm1("tab-AsesorTesis","");
}
private void autorToolStripMenuItem_Click(object sender,EventArgs e) {
    // Informacion de tesita
    string tipoFormularioAA = "tab-AutorTesis";
    string textoMensaje = "Hola autor";
    EnviarInformacionForm1(tipoFormularioAA,textoMensaje);
}

/* CINTA DE OPCIONES - AYUDA */
private void manualDeUsorToolStripMenuItem_Click(object sender,EventArgs e) {
    EnviarInformacionForm1("tab-ManualUso","");
}
private void tutorialToolStripMenuItem_Click(object sender,EventArgs e) {
    EnviarInformacionForm1("tab-TutorialUso","");
}

public static int[,] ConvertStringToMatrix(string matrizString) {
    // Divide el string en filas usando la coma como delimitador
    string[] filas = matrizString.Split(',');

    // Determina el número de filas y columnas
    int rowCount = filas.Length;
}
```




```
int colCount = filas[0].Split(' ').Length;

// Inicializa la matriz numérica
int[,] matrizNumerica = new int[rowCount, colCount];

// Rellena la matriz numérica con los valores del string
for ( int i = 0; i < rowCount; i++ ) {
    // Divide cada fila en sus elementos (números) usando el espacio
    int[] elementos = filas[i].Split(' ').Select(int.Parse).ToArray()

    for ( int j = 0; j < colCount; j++ ) {
        matrizNumerica[i,j] = elementos[j];
    }
}
return matrizNumerica;
}

public static string[,] ConvertStringToMatrix2(string matrizString) {
    // Divide el string en filas usando la coma como delimitador
    string[] filas = matrizString.Split(',');

    // Determina el número de filas y columnas
    int rowCount = filas.Length;
    int colCount = filas[0].Split(' ').Length;

    // Inicializa la matriz de strings
    string[,] matriz = new string[rowCount, colCount];

    // Rellena la matriz con los valores del string
    for ( int i = 0; i < rowCount; i++ ) {
        // Divide cada fila en sus elementos usando el espacio como delimitador
        string[] elementos = filas[i].Split(' ');
        for ( int j = 0; j < colCount; j++ ) {
            matriz[i,j] = elementos[j];
        }
    }
    return matriz;
}

// CINTA - RESULTADOS
private void rigidezToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 0;
}

private void hidraulicaToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 1;
}

private void kHToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 2;
}

private void asentamientoToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 3;
}

private void presiónToolStripMenuItem_Click(object sender, EventArgs e) {
    this.TabControlResultados.SelectedIndex = 4;
}
```



```
    }

    // CINTA - GRAFICOS
    private void asentamientosToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.tabControl2.SelectedIndex = 2;
        this.tabControl2.SelectedIndex = 4;
    }

    private void presionesToolStripMenuItem_Click(object sender, EventArgs e)
    {

    }

    #endregion CINTA OPCIONES

    #region BOTONES CALCULAR Y EJECUTAR - integracion latex

    private void GenerarReporteTextoLatex() {
        string salto = "$@"
        \vspace{{0.5cm}}
        ";
        string nuevaPagina = "$@"
        \newpage
        ";
        // Parametros Iniciales para Latex
        string nombreProyecto = "Calculo de suelo con cinco elementos";

        // Iniciando con Latex
        string inicioLatex = "$@"\documentclass[a4paper,12pt]{{report}}
        \usepackage{{adjustbox}}
        \title{{nombreProyecto}}
        \author{{Suelos MEF}}
        \date{{\today}}

        \begin{{document}}

        \maketitle";
        string finLatex = "$@"\end{{document}}";

        // DATOS INICIALES
        string datosSuelo = "$@"
        \begin{{center}}
        \textbf{{\LARGE DATOS}}
        \end{{center}}
        ";

        string datosPropiedadSuelo = "$@"
        \noindent \textbf{{\large 1. Datos de propiedad del Suelo}}\
        \noindent \textit{{A conumCantCargasnuación, los tipos de suelos creados:}}\
        ";
        datosPropiedadSuelo = datosPropiedadSuelo + GeneraTablaDatosLatex(this.dgvTablaMa

        string datosElementosSuelo = "$@"
        \noindent \textbf{{\large 2. Datos de Elementos del Suelo}}\
        \noindent \textit{{A conumCantCargasnuación, todos los elementos creados:}}\
        ";
        datosElementosSuelo = datosElementosSuelo + GeneraTablaDatosLatex(this.dgvTablaE
```



```
string datosCondicionFronteraSuelo = "$@"
\noindent \textbf{\large 3. Datos de las condiciones de frontera del Suelo}}\
\noindent \textit{A conumCantCargasnuación, todos los elementos creados:}}\
";
datosCondicionFronteraSuelo = datosCondicionFronteraSuelo + GeneraTablaDatosLatex

string datosCagasSuelo = "$@"
\noindent \textbf{\large 4. Datos de las cargas aplicadas en el suelos del Suelo}}\
\noindent \textit{A conumCantCargasnuación, todos los elementos creados:}}\
";
datosCagasSuelo = datosCagasSuelo + GeneraTablaDatosLatex(this.dgvTablaCargas);

// Convertir la matriz a código LaTeX
//string matrizLatex = ConvertirMatrizALatex(matriz);
string latexDatos = datosSuelo + salto + datosPropiedadSuelo + salto + datosElementos;
//string latexCalculo = nuevaPagina+calculoSuelo + salto+calculoKeSuelo+salto+calculoHuelo;
//string latexResultados = ""; //nuevaPagina+ resultadosSuelo+salto+ resultadosAsesores;

//this.txtLatex.Text = inicioLatex + latexDatos +latexCalculo+latexResultados+finLatex;
this.txtLatex.Text = inicioLatex + latexDatos + finLatex;

string maiLatex = PasarTablaLatex(this.dgvQ);
string textoPruebaTabla = GeneraTablaMatrizLatex(this.dgvK);

}

private string GeneraKeLatex() {
// Agregar el título
string salto = "$@"
\vspace{{0.5cm}}
";
string matrizKe = "$@";
// Agregar datos (a partir de la fila 3)
for ( int e = 0; e < numElementos; e++ ) {
matrizKe = matrizKe + salto + "$@"\textbf{\textit{Elemento {{e+1}}}}}}\
";
int inicioFila = e * 3 + 1;
int finalFila = inicioFila + 2;
double[,] M = submatrix(KKensablando, inicioFila, finalFila, 0, 2);
matrizKe = matrizKe+ ConvertirMatrizALatex(M, "K_" + (e + 1).ToString() + "=", "KN/m3");
}
return matrizKe;
}

private string GeneraKKLatex() {
// Agregar el título
string KKglobal = "$@";
KKglobal = "$@"\textbf{\textit{Matriz Global K}}}}\
";
KKglobal = KKglobal + ConvertirMatrizALatex(K, "K=", "KN/m3");
return KKglobal;
}

private string GeneraHHLatex() {
// Agregar el título
string HHglobal = "$@";
HHglobal = "$@"\textbf{\textit{Matriz Global H}}}}\
";
HHglobal = HHglobal + ConvertirMatrizALatex(H, "H=", "KN/m3");
}
```



```
return HHglobal;
}

private string GeneraQQLatex() {
// Agregar el título
string QQglobal = "$@";
QQglobal = "$@" + "\\textbf{{\\textit{{Matriz Global Q}}}}\\\\";
QQglobal = QQglobal + ConvertirMatrizALatex(Q, "Q=", "KN/m3");
return QQglobal;
}

private string GeneraHeLatex() {
// Agregar el título
string salto = "$@"
\\vspace{{0.5cm}}
";
string matrizKe = "$@";
// Agregar datos (a partir de la fila 3)
for ( int e = 0; e < numElementos; e++ ) {
matrizKe = matrizKe + salto + "$@" + "\\textbf{{\\textit{{Elemento {{e + 1}}}}}}\\\\";
int inicioFila = e * 2 + 1;
int finalFila = inicioFila + 1;
double[,] M = submatrix(HHensablando, inicioFila, finalFila, 0, 1);
matrizKe = matrizKe + ConvertirMatrizALatex(M, "H_" + (e + 1).ToString() + "=", "KN/m3");
}
return matrizKe;
}

private string GeneraQeLatex() {
// Agregar el título
string salto = "$@"
\\vspace{{0.5cm}}
";
string matrizKe = "$@";
// Agregar datos (a partir de la fila 3)
for ( int e = 0; e < numElementos; e++ ) {
matrizKe = matrizKe + salto + "$@" + "\\textbf{{\\textit{{Elemento {{e + 1}}}}}}\\\\";
int inicioFila = e * 3 + 1;
int finalFila = inicioFila + 2;
double[,] M = submatrix(QQensablando, inicioFila, finalFila, 0, 1);
matrizKe = matrizKe + ConvertirMatrizALatex(M, "Q_" + (e + 1).ToString() + "=", "KN/m3");
}
return matrizKe;
}

private string GeneraTablaDatosLatex(DataGridView Tabla) {
StringBuilder latexTable = new StringBuilder();

int numFilas = Tabla.RowCount;
int numColum = Tabla.ColumnCount;

// Iniciar la tabla LaTeX
latexTable.AppendLine("\\\\begin{adjustbox}{max width=\\textwidth}");
latexTable.Append("\\\\begin{tabular}");
}
```



```
// Crear los encabezados y columnas
if(Tabla== this.dgvTablaMateriales ) {
latexTable.AppendLine("{1 1 1 1 1 1}");
} else if ( Tabla == this.dgvTablaElementos ) {
latexTable.AppendLine("{1 1 1 1}");
} else if (Tabla==this.dgvTablaRestriciones) {
latexTable.AppendLine("{1 1 1 1}");
} else if (Tabla==this.dgvTablaCargas) {
latexTable.AppendLine("{1 1 1 1}");
} else {
latexTable.AppendLine("{|C{0.5cm}|C{2.5cm}|C{2.5cm}|}");
MessageBox.Show("No se encontro encabezado");
}

// Agregar una línea de separación para los encabezados
latexTable.AppendLine("\\hline");
// Agregar las celdas de encabezado (si hay encabezado)
for ( int i = 0; i < numColum; i++ ) {
latexTable.Append(Tabla.Columns[i].HeaderText);
if ( i < numColum - 1 )
latexTable.Append(" & "); // Separar con "&"
}
latexTable.AppendLine(" \\\\ \\hline");

// Agregar los datos de las filas
foreach ( DataGridViewRow row in Tabla.Rows ) {
if ( row.IsNewRow ) conumCantCargasnue; // Ignorar la fila en blanco
for ( int i = 0; i < numColum; i++ ) {
latexTable.Append(row.Cells[i].Value?.ToString() ?? ""); // Obtener el valor de
if ( i < numColum - 1 )
latexTable.Append(" & "); // Separar con "&"
}
latexTable.AppendLine(" \\\\");
}
latexTable.AppendLine("\\hline");

// Finalizar la tabla LaTeX
latexTable.AppendLine("\\end{tabular}");
latexTable.AppendLine("\\end{adjustbox}");

return latexTable.ToString();
}

private string GeneraTablaMatrizLatex(DataGridView Tabla) {
StringBuilder sb = new StringBuilder();
int numFilas = Tabla.RowCount;
int numColum = Tabla.ColumnCount;
//string textoMatrix = "mathbf{K}_{viga}=";
string cantColum = "";
for(int i=0;i<numColum+2+2;i++ ) {
cantColum = cantColum + "c ";
}
string inicioMatrix = "1-1";
string finMatrix = (numColum+2).ToString() + "-" + (numColum+2).ToString();
sbLatex.AppendLine($"@");
sbLatex.AppendLine($"@\"[");
//sbLatex.AppendLine($"@\"{{{textoMatrix}}}");
}
```



```
sbLatex.AppendLine($"@\\begin{{{array}}}{{{cantColum}}}" );

for ( int j = 0; j < numColum; j++ ) {
sbLatex.Append($"@      & u_{{{j+1}}}" );
}
sbLatex.AppendLine($"@      & \\");
sbLatex.AppendLine($"@      \\cline{{{inicioMatrix}}}" );
sbLatex.AppendLine($"@      \\cline{{{finMatrix}}}" );

for (int i=0;i<numFilas;i++ ) {
sbLatex.Append($"@      \\multicolumn{{{1}}}{{{c}}}{{{}}}" & " ");
for (int j = 0;j < numColum; j++) {
sbLatex.Append($"@      {{{Tabla.Rows[i].Cells[j].Value}}}" & " ");
}
sbLatex.AppendLine($"@      \\multicolumn{{{1}}}{{{c |}}}{{{}}}" & v_{{{i+1}}}" & "\\");
}
sbLatex.AppendLine($"@      \\cline{{{inicioMatrix}}}" );
sbLatex.AppendLine($"@      \\cline{{{finMatrix}}}" );
sbLatex.AppendLine($"@\\end{{{array}}}" );
sbLatex.AppendLine($"@\\");

return sbLatex.ToString();
}

private string PasarTablaLatex(DataGridView Tabla) {
StringBuilder sb = new StringBuilder();

// Iniciar la representación LaTeX de la matriz
sbLatex.AppendLine("");
sbLatex.AppendLine("\\[");
sbLatex.AppendLine("\\begin{bmatrix}");

for ( int i = 0; i < Tabla.RowCount; i++ ) {
for ( int j = 0; j < Tabla.ColumnCount; j++ ) {
sbLatex.Append(Tabla.Rows[i].Cells[j].Value);

// Añadir "&" entre elementos de la fila, excepto el último
if ( j < Tabla.ColumnCount - 1 ) {
sbLatex.Append(" & ");
}
}

// Añadir "\\\" al final de cada fila, excepto la última
if ( i < Tabla.RowCount - 1 ) {
sbLatex.AppendLine(" \\\\");
}
}

// Cerrar el entorno de la matriz
sbLatex.AppendLine();
sbLatex.Append("\\end{bmatrix}");
sbLatex.AppendLine();
sbLatex.Append("\\]");
sbLatex.AppendLine();

return sbLatex.ToString();
}
```



```
// Método para convertir una matriz en formato LaTeX
private string ConvertirMatrizALatex(double[,] matriz, string titulo, string unidad)
{
    StringBuilder sb = new StringBuilder();

    string matLatex = "$@";

    // Iniciar la representación LaTeX de la matriz
    matLatex = matLatex + "$@" + @"\begin{{center}}
    {{{titulo}}}
    \begin{{bmatrix}}
    ";

    for ( int i = 0; i < matriz.GetLength(0); i++ ) {
        for ( int j = 0; j < matriz.GetLength(1); j++ ) {
            matLatex = matLatex + matriz[i,j].ToString();

            // Añadir "&" entre elementos de la fila, excepto el último
            if ( j < matriz.GetLength(1) - 1 ) {
                matLatex = matLatex + " & ";
            }
        }

        // Añadir "\\" al final de cada fila, excepto la última
        if ( i < matriz.GetLength(0) - 1 ) {
            matLatex = matLatex + " \\\\";
        }
    }

    // Cerrar el entorno de la matriz
    matLatex = matLatex + "$@" + @"\end{{bmatrix}}
    {{{unidad}}}$
    \end{{center}}";

    return matLatex;
}

string directoryPath = Application.StartupPath;
string parentDirectory = Directory.GetParent(directoryPath).Parent.Parent.Parent;

// Crea el nombre de la carpeta con el formato "Proyecto + fecha"
string projectFolderName = "Proyecto_" + DateTime.Now.ToString("yyyyMMdd");
string newFolderPath = Path.Combine(parentDirectory, projectFolderName);

// Verifica si la carpeta ya existe, si no, la crea
if ( !Directory.Exists(newFolderPath) ) {
    Directory.CreateDirectory(newFolderPath);
}

// Define el nombre de archivo para la imagen
string fileName = "grafico_guardado.png"; // Puedes cambiar el nombre de la imagen

// Combina la ruta del directorio con el nombre del archivo
string filePath = Path.Combine(newFolderPath, fileName);
```



```
// Guarda la imagen en la carpeta del ejecutable
this.ptbGrafico1.Image.Save(filePath);
}
private void button9_Click(object sender, EventArgs e) {
// Dibujar
dibujo1();
}

#endregion BOTONES CALCULAR Y EJECUTAR

#region LANZAR FORMULARIO HIJO Y PADRE
private void EnviarInformacionForm1(string tipoFormulario, string mensaje) {
// Instancia Form2
Form2 form2 = new Form2();
// Enviar información a Form2 a través de la propiedad
form2.Form1AForm2AA = tipoFormulario;
form2.Form1AForm2BB = mensaje;
form2.ShowDialog();
}
#endregion LANZAR FORMULARIO HIJO Y PADRE

#region EXPORTAR A EXCEL
private void btnExportarExcel_Click(object sender, EventArgs e) {
if (true) { //se ejecuto el programa?
ExportarExcel();
MessageBox.Show("Reporte Generado :)");
}

}

private void ExportarExcel() {

// Crear una nueva aplicación de Excel
var excelApp = new Excel.Application {
Visible = true
};

// Crear un nuevo libro de Excel
var workbook = excelApp.Workbooks.Add();

// INFORMACION
if ( true ) {
// Hoja de Inicio
var hoja0 = (Excel.Worksheet) workbook.Worksheets[1];
hoja0.Name = "Informacion";
ExportarInformacionExcel(hoja0);
}

// DATOS
if ( this.ckbExcelPropiedadSuelo.Checked ) {
// Exportar Dato - Propiedades del suelo
var hoja1 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.W
hoja1.Name = "Propiedades";
```




```
ExportarTablaExcel(this.dgvTablaMateriales, hoja1, "Propiedades del Suelo");
}

if ( this.ckbExcelElementoSuelo.Checked ) {
// Exportar Dato - Propiedades del suelo
var hoja2 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja2.Name = "Elementos";
ExportarTablaExcel(this.dgvTablaElementos, hoja2, "Elementos");
}

if ( this.ckbExcelCondicionFrontera.Checked ) {
// Exportar Dato - Propiedades del suelo
var hoja3 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja3.Name = "Condiciones";
ExportarTablaExcel(this.dgvTablaRestriciones, hoja3, "Condiciones de frontera");
}

if ( this.ckbExcelCargas.Checked ) {
// Exportar Dato - Propiedades del suelo
var hoja4 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja4.Name = "Cargas";
ExportarTablaExcel(this.dgvTablaCargas, hoja4, "Carga aplicada");
}

////if ( this.ckbExcelDatosOtro.Checked ) {
////    // Exportar Dato - Propiedades del suelo
////    var hoja5 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
////    hoja5.Name = "OtrosDatos";
////    ExportarTablaExcel(this.dgvTablaRestriciones, hoja5, "Otros - Datos");
////}

// RESULTADOS
if ( this.ckbExcelMatrizK.Checked ) {
// Exportar Resultados - Matriz Rigidez
var hoja6 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja6.Name = "Matriz-Ke";
ExportarMatrizRigidezExcel(hoja6, "Matriz Rigidez por Elemento");

// Exportar Resultados - Matriz Rigidez Global
var hoja7 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja7.Name = "Matriz-K";
ExportarMatrizRigidezGlobalExcel(hoja7, "Matriz Rigidez Global");
}

if ( this.ckbExcelMatrizH.Checked ) {
// Exportar Resultados - Matriz Hidraulica
var hoja8 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja8.Name = "Matriz-He";
ExportarMatrizHidraulicaExcel(hoja8, "Matriz Rigidez Hidraulica");

// Exportar Resultados - Matriz Hidraulica Global
var hoja9 = workbook.Worksheets.Add(Type.Missing, workbook.Worksheets[workbook.Worksheets.Count - 1]);
hoja9.Name = "Matriz-H";
ExportarMatrizHidraulicaGlobalExcel(hoja9, "Matriz Hidraulica Global");
}
```



```
if ( this.ckbExcelMatrizQ.Checked ) {
// Exportar Resultados - Matriz K-H
var hoja10 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja10.Name = "Matriz-Qe";
ExportarMatrizQelicaExcel(hoja10,"Matriz Qe");

// Exportar Resultados - K-H Global
var hoja11 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja11.Name = "Matriz-Q";
ExportarMatrizKHGlobalExcel(hoja11,"Matriz Q Global");
}

if ( this.ckbExcelDesplazamiento.Checked ) {
// Exportar Resultados - Desplazamiento
var hoja12 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja12.Name = "Asentamiento";
ExportarResultadosExcel(this.dgvMatrizEnsablada,hoja12,"Asentamiento");
}

if ( this.ckbExcelPresion.Checked ) {
// Exportar Resultados - Desplazamiento
var hoja13 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja13.Name = "Presion";
ExportarResultadosExcel(this.dgvMatrizEnsablada2,hoja13,"Presion");
}

// GRAFICOS
if ( this.ckbExcelGraficoDesplazamiento.Checked ) {
// Exportar Resultados - Desplazamiento
var hoja14 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja14.Name = "Grafico-Asentamiento";
ExportarGraficoResultadosExcel(this.dgvMatrizEnsablada,hoja14,"Grafico Asentamiet
}

if ( this.ckbExcelGraficoPresion.Checked ) {
// Exportar Resultados - Desplazamiento
var hoja14 = workbook.Worksheets.Add(Type.Missing,workbook.Worksheets[workbook.W
hoja14.Name = "Grafico-Presion";
ExportarGraficoResultadosExcel(this.dgvMatrizEnsablada2,hoja14,"Grafico Presion")
}
}

private void ExportarGraficoResultadosExcel(DataGridView dgv,Excel.Worksheet work
// Agregar el titulo
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range[worksheet.Cells[1,1],worksheet.Cells[1,dgv.Colu
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

// Agregar encabezados (fila 2)
for ( int col = 0; col < dgv.Columns.Count; col++ ) {
worksheet.Columns[col + 1].ColumnWidth = 12;
worksheet.Cells[2,col + 1] = dgv.Columns[col].HeaderText;
worksheet.Cells[2,col + 1].font.bold = true;
worksheet.Cells[2,col + 1].Font.Italic = true;
worksheet.Cells[2,col + 1].Interior.Color = Excel.XlRgbColor.rgbLightSeaGreen; //
```



```
//worksheet.Cells[2,col + 1].Font.Color = Excel.XlRgbColor.rgbWhite; // Color de
worksheet.Cells[2,col + 1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter; /
worksheet.Cells[2,col + 1].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter; /
worksheet.Cells[2,col + 1].WrapText = true; ; // Ajustar texto
}
// Ajustar columnas
var rangoDatos = worksheet.Range[worksheet.Cells[1,1],worksheet.Cells[dgv.Rows.C
rangoDatos.Borders.LineStyle = Excel.XlLineStyle.xlConumCantCargasnuous;
rangoDatos.Borders.Weight = Excel.XlBorderWeight.xlThin;

// Crear un gráfico
var charts = (Excel.ChartObjects) worksheet.ChartObjects();
var chartObject = charts.Add(100,50,1000,700);
var chart = chartObject.Chart;

// Configuración del gráfico
chart.ChartType = Excel.XlChartType.xlXYScatterSmooth;

// Agregar series (T con A, T con B, T con C)
var seriesCollection = chart.SeriesCollection();

// T con A
for ( int g = 1; g <= numNodos; g++ ) {
var seriesA = seriesCollection.NewSeries();
seriesA.Name = "Nodo " + g.ToString();
seriesA.XValues = worksheet.Range[worksheet.Cells[3,1],worksheet.Cells[dgv.Rows.C
seriesA.Values = worksheet.Range[worksheet.Cells[3,g+1],worksheet.Cells[dgv.Rows.C

}

// Agregar título al gráfico
chart.HasTitle = true;
chart.ChartTitle.Text = titulo;

// Deshabilitar la línea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;
System.Threading.Thread.Sleep(100);
}

private void ExportarMatrizQelicaExcel(Excel.Worksheet worksheet, string titulo) {
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

// Agregar datos (a partir de la fila 3)
for ( int e = 0; e < numElementos; e++ ) {
int inicioFila = e * 3 + 1;
int finalFila = inicioFila + 2;
double[,] M = submatrix(QQensablado, inicioFila, finalFila, 0, 1);
var range = worksheet.Range[worksheet.Cells[4 * e + 4, 2], worksheet.Cells[4 * e +
range.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle = Excel.XlLineStyle.xl
worksheet.Cells[4 * e + 5, 1] = "Qe_" + (e + 1).ToString() + "=";
worksheet.Cells[4 * e + 5, 4] = "KN/m2";
```



```
for ( int i = 0; i < rows(M); i++ ) {
for ( int j = 0; j < cols(M); j++ ) {
worksheet.Cells[4 * e + 3 + i + 1,j + 2] = M[i,j];//dgv.Rows[row].Cells[col].Value
}
}
}

// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;
System.Threading.Thread.Sleep(100);
}

private void ExportarMatrizKHGlobalExcel(Excel.Worksheet worksheet,string titulo)
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

worksheet.Cells[rows(Q) / 2 + 4,1] = "H=";
worksheet.Cells[rows(Q) / 2 + 4,cols(Q) + 2] = "KN/m2";
// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;
System.Threading.Thread.Sleep(100);
}

private void ExportarMatrizHidraulicaExcel(Excel.Worksheet worksheet,string titulo)
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

// Agregar datos (a partir de la fila 3)
for ( int e = 0; e < numElementos; e++ ) {
int inicioFila = e * 2 + 1;
int finalFila = inicioFila + 1;
double[,] M = submatrix(HHensablando,inicioFila,finalFila,0,1);
var range = worksheet.Range[worksheet.Cells[3 * e + 4,2],worksheet.Cells[3 * e + 4,4]];
range.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlLineStyleNone;
range.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle = Excel.XlLineStyle.xlLineStyleNone;
worksheet.Cells[3 * e + 4,1] = "He_" + (e + 1).ToString() + "=";
worksheet.Cells[3 * e + 4,4] = "KN/m2";
for ( int i = 0; i < rows(M); i++ ) {
for ( int j = 0; j < cols(M); j++ ) {
worksheet.Cells[3 * e + 3 + i + 1,j + 2] = M[i,j];//dgv.Rows[row].Cells[col].Value
}
}
}

// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;

// Cambio de color de la pestana de la hoja
```



```
System.Threading.Thread.Sleep(100);
}
private void ExportarMatrizHidraulicaGlobalExcel(Excel.Worksheet worksheet, string
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

worksheet.Cells[rows(H) / 2 + 3,1] = "H=";
worksheet.Cells[rows(H) / 2 + 3,cols(H) + 2] = "KN/m2";
// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;

// Cambio de color de la pestana de la hoja
worksheet.Tab.Color = System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.B

System.Threading.Thread.Sleep(100);
}

private void ExportarMatrizRigidezGlobalExcel(Excel.Worksheet worksheet, string ti
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

worksheet.Cells[rows(K)/2 + 4,1] = "K=";
worksheet.Cells[rows(K)/2 + 4,cols(K)+2] = "KN/m2";
// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;

// Cambio de color de la pestana de la hoja
System.Threading.Thread.Sleep(100);
}

private void ExportarMatrizRigidezExcel(Excel.Worksheet worksheet, string titulo)
// Agregar el título
worksheet.Cells[1,1] = titulo;
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

// Agregar datos (a partir de la fila 3)
for ( int e=0;e<numElementos;e++ ) {
int inicioFila = e * 3 + 1;
int finalFila = inicioFila + 2;
double[,] M = submatrix(KKensablando, inicioFila, finalFila, 0,2);
var range = worksheet.Range[worksheet.Cells[4*e+4,2], worksheet.Cells[4*e+6,4]];
worksheet.Cells[4 * e + 5,1] = "Ke_" + (e + 1).ToString() + "=";
worksheet.Cells[4 * e + 5,5] = "KN/m2";
for ( int i = 0; i < rows(M); i++ ) {
```



```
for ( int j = 0; j < cols(M); j++ ) {
worksheet.Cells[4*e+3+i + 1,j + 2] = M[i,j]; //dgv.Rows[row].Cells[col].Value?.ToString()
}
}
}

// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;

// Cambio de color de la pestaña de la hoja
worksheet.Tab.Color = System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.FromName("Blue"));

System.Threading.Thread.Sleep(100);
}

private void ExportarInformacionExcel(Excel.Worksheet worksheet) {
// Agregar el título
worksheet.Cells[1,1] = "INFORMACIÓN DEL REPORTE";
var rangoTitulo = worksheet.Range["A1:G1"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

worksheet.Cells[2,2] = "Suelos MEF";
rangoTitulo = worksheet.Range["B2:G2"];
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Italic = true;
rangoTitulo.Font.Size = 12;
var rangoDatos = worksheet.Range["A6:B6"];
rangoDatos.Borders.Weight = Excel.XlBorderWeight.xlThin;

worksheet.Cells[6,5] = "Hora";
worksheet.Cells[6,6] = DateTime.Now.ToString("HH:mm:ss");
rangoDatos = worksheet.Range["E6:F6"];
rangoDatos.Borders.LineStyle = Excel.XlLineStyle.xlConumCantCargasnuous;
worksheet.Cells[10,2] = "Numero de Elementos";
worksheet.Cells[10,4] = numElementos;

worksheet.Cells[12,2] = "Numero de Cargas Asignadas";
worksheet.Cells[12,4] = numCantCargas;

// Desabilitar la linea de cuadrícula
worksheet.Application.ActiveWindow.DisplayGridlines = false;
}

private void ExportarTablaExcel(DataGridView dgv, Excel.Worksheet worksheet, string titulo) {
// Agregar el título
worksheet.Cells[1,1] = titulo;
rangoTitulo.Merge();
rangoTitulo.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
rangoTitulo.Font.Bold = true;
rangoTitulo.Font.Size = 14;

// Agregar encabezados (fila 2)
for ( int col = 0; col < dgv.Columns.Count; col++ ) {
```



```
worksheet.Columns[col + 1].ColumnWidth = 12;
worksheet.Cells[2,col + 1] = dgv.Columns[col].HeaderText;
worksheet.Cells[2,col + 1].font.bold = true;
worksheet.Cells[2,col + 1].Font.Italic = true;
worksheet.Cells[2,col + 1].Interior.Color = Excel.XlRgbColor.rgbLightSeaGreen; //
//worksheet.Cells[2,col + 1].Font.Color = Excel.XlRgbColor.rgbWhite; // Color de
worksheet.Cells[2,col + 1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter; //
worksheet.Cells[2,col + 1].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter; //
worksheet.Cells[2,col + 1].WrapText = true; ; // Ajustar texto
}
// Cambio de color de la pestana de la hoja
worksheet.Tab.Color = System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.F
//255, 200, 100 naranja
// 200, 200, 200 gris

System.Threading.Thread.Sleep(100);
}

#endregion EXPORTAR A EXCEL

private void InicarcboEstilos() {
this.cboApariencia.Items.Clear();
this.cboApariencia.Items.Add("MidnightBlue");
this.cboApariencia.Items.Add("DarkSlateGray");
this.cboApariencia.Items.Add("DimGray");
this.cboApariencia.Items.Add("SaddleBrown");
this.cboApariencia.SelectedIndex = 0;
}

private void CambiarColorLabel(string estilo) {

Color miColor = Color.MidnightBlue;
if (estilo == "MidnightBlue") {
miColor = Color.MidnightBlue;
} else if (estilo == "DarkSlateGray") {
miColor = Color.DarkSlateGray;
} else if (estilo == "DimGray") {
miColor = Color.DimGray;
} else if (estilo == "SaddleBrown") {
miColor = Color.SaddleBrown;
} else {
miColor = Color.MidnightBlue;
}

this.lbla1.BackColor = miColor;
this.lbla2.BackColor = miColor;
this.lbla3.BackColor = miColor;
this.lbla4.BackColor = miColor;
this.lbla5.BackColor = miColor;
this.lbla6.BackColor = miColor;
this.lbla7.BackColor = miColor;
this.lbla8.BackColor = miColor;
this.lbla9.BackColor = miColor;
this.lbla10.BackColor = miColor;
this.lbla11.BackColor = miColor;
this.lbla12.BackColor = miColor;
this.lbla13.BackColor = miColor;
```



```
this.lbla14.BackColor = miColor;  
}  
  
#endregion OPCIONES DE APARIENCIA  
  
}  
}
```




ANEXO 12: Declaración jurada de autenticidad de tesis



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Denis Rudy Ipara Quispe
identificado con DNI 70650517 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado
Ciencias Física Matemáticas

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

"Desarrollo de una herramienta computacional para la simulación numérica de los asentamientos lineales en suelos usando el método de los elementos finitos"

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 20 de diciembre del 2014

[Firma]
FIRMA (obligatoria)



Huella



ANEXO 13: Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Denis Rudy Lupaca Quispe
identificado con DNI 70650512 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

Ciencias físico Matemáticas
informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“ Desarrollo de una herramienta computacional para la
simulación numérica del asentamiento lineal en suelos
usando el método de los elementos finitos ”

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los “Contenidos”) que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

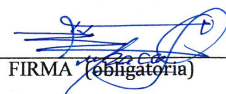
En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 20 de diciembre del 2004


FIRMA (obligatoria)



Huella