

GUÍA PARA LA INSTALACIÓN DE ELMER

Proyecto final de Semestre

Estudiantes:

- Juan Esteban Neira Díaz
- David Santiago Rodriguez Ortiz
- Nicolás Alexei Durán Amariles

Profesor:

Carlos Andrés Gómez Vasco

Junio-2025

Resumen En el presente documento se hace un recuento de las capacidades del software ELMER para la simulación de sistemas físicos, mostrando los distintos módulos de simulación implementados junto con las herramientas y algoritmos de los que dispone el programa para la solución de problemas, detallando el método de elementos finitos el cual es la base del software, también esta una revisión detallada de como instalar el programa en Windows y distribuciones Linux para finalmente hacer una demostración de como se ve un proyecto general usando ELMER.

Palabras clave

Elmer, instalación, diferencias finitas



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Índice

1. Introducción	3
2. MODELOS MATEMÁTICOS Y COMPUTACIONALES	4
2.1. PRINCIPALES MODELOS MATEMÁTICOS	6
2.2. HERRAMIENTAS DE SOLUCION	8
3. DESCARGA PARA <i>WINDOWS</i>	11
3.1. Descarga de Elmer	11
3.2. Descargar la documentación	12
3.3. Paraview	14
4. INSTALACIÓN PARA <i>WINDOWS</i>	15
4.1. Instalando Elmer	16
4.2. Intalando Paraview	20
5. Windows- Camino de Elmer	22
5.1. El camino a Elmer	22
5.2. Camino a Paraview	23
5.3. Correr Paraview desde ElmerGUI	23
6. Creando un primer proyecto en Elmer	25
7. Introducción rápida a Paraview	26
7.1. Descripción de un problema sencillo	26
8. INSTALACION EN LINUX	32
8.1. Actualizar y Limpiar	32
8.2. Instalar Dependencias	32
8.3. Librerías de Java	33
8.4. Clonar código fuente y crear build	33
8.5. Configurar Cmake	33
8.6. Configurar Variables de entorno	34
9. FLUJO DE TRABAJO	35

1. Introducción

Elmer es un sistema de simulación multifísico de código abierto desarrollado principalmente por el centro de tecnologías de información para la ciencia (CSC-IT). Principalmente Elmer fue desarrollado por universidades, institutos de investigación y la industria finlandesa. Tras su publicación en 2005, el uso y desarrollo de *Elmer* se volvió de carácter internacional.

Este software incluye modelos físicos de dinámica de fluidos, mecánicas estructural, electromagnetismo, transferencia de calor, acústica, entre otros. Los modelos implementados se describen a partir de ecuaciones diferenciales los cuales se resuelven utilizando modelos de *elementos finitos* (MEF).

Dado que es un programa complejo de desarrollo interdisciplinar se han utilizado diferentes lenguajes de programación tal como se puede observar en la figura 1. Por la importancia de **Elmer** para la comunidad científica y la dificultad que puede llegar a representar para algunas personas para su instalación, en este documento se encuentra el paso a paso de su instalación.

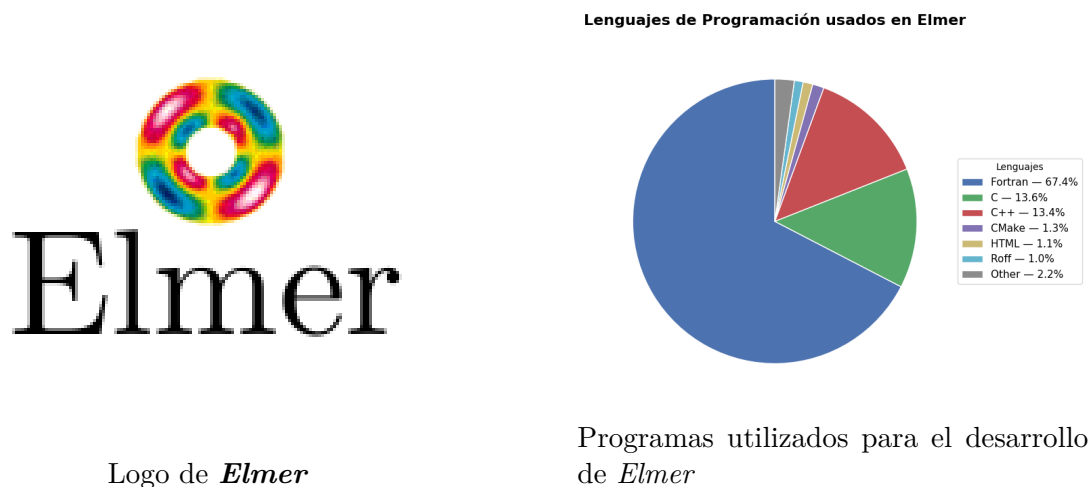


Figura 1: Resumen visual sobre el software *Elmer*.

2. MODELOS MATEMÁTICOS Y COMPUTACIONALES

ELMER busca ser un software que permita la simulación la mayor cantidad de sistemas físicos en busca de obtener la mayor generalidad de casos posible y abarcar una base general que permita construir simulaciones físicas completas con motivo de investigación y desarrollo industrial.

ELMER cuenta con una extensa lista de sistemas físicos integrados y tiene una guía de documentación donde explica los modelos matemáticos tomados en cuenta para las implementaciones en software y como se traducen del lenguaje físico-matemático a los archivos necesarios para la simulación.

Método de los Elementos Finitos (FEM): Discretización, Polinomios de Interpolación y Método de Galerkin

El Método de los Elementos Finitos (FEM) es una poderosa técnica numérica para resolver ecuaciones diferenciales parciales (EDP) que describen fenómenos físicos. A diferencia del método de diferencias finitas, FEM se basa en la discretización del dominio en elementos finitos y la aproximación de la solución dentro de cada elemento mediante funciones de forma.

1. Discretización del Dominio

El primer paso en FEM es dividir el dominio continuo Ω de la ecuación diferencial en un número finito de subdominios más pequeños y simples, llamados *elementos finitos*. Estos elementos pueden ser unidimensionales (líneas), bidimensionales (triángulos, cuadriláteros) o tridimensionales (tetraedros, hexaedros). Los puntos donde los elementos se conectan se denominan *nodos*.

Consideremos un dominio Ω discretizado en N_e elementos. La solución aproximada $u_h(\mathbf{x})$ se define como una suma de las contribuciones de cada elemento.

2. Polinomios de Interpolación (Funciones de Forma)

Dentro de cada elemento finito, la solución desconocida $u(\mathbf{x})$ se aproxima localmente mediante una combinación lineal de los valores de la función en los nodos del elemento y de funciones predefinidas, conocidas como *funciones de forma* o *funciones de base de interpolación*.

Para un elemento e , la solución aproximada $u_h^e(\mathbf{x})$ se expresa como:

$$u_h^e(\mathbf{x}) = \sum_{j=1}^{n_e} N_j^e(\mathbf{x}) u_j^e$$

donde:

- n_e es el número de nodos en el elemento e .
- $N_j^e(\mathbf{x})$ son las funciones de forma (polinomios de interpolación) asociadas al nodo j del elemento e .
- u_j^e es el valor de la solución aproximada en el nodo j del elemento e .

Las funciones de forma $N_j^e(\mathbf{x})$ tienen propiedades clave:

- Tienen un valor de 1 en su propio nodo y 0 en los otros nodos del mismo elemento: $N_j^e(\mathbf{x}_k) = \delta_{jk}$ (donde δ_{jk} es la delta de Kronecker).
- Son usualmente polinomios de bajo orden (lineales, cuadráticos) que garantizan continuidad entre elementos (C0 continuidad).
- Son funciones con soporte local, es decir, son distintas de cero solo en el elemento al que pertenecen.

La solución global $u_h(\mathbf{x})$ se construye ensamblando estas aproximaciones locales, utilizando funciones de forma globales $\Phi_j(\mathbf{x})$ que se forman a partir de la suma de las funciones de forma locales que se solapan en los nodos compartidos:

$$u_h(\mathbf{x}) = \sum_{j=1}^{N_d} \Phi_j(\mathbf{x}) U_j$$

donde N_d es el número total de grados de libertad (valores nodales) en el dominio, y U_j son los valores de la solución aproximada en los nodos globales.

3. Método de Galerkin

El método de Galerkin es una técnica fundamental para derivar el sistema de ecuaciones algebraicas a partir de la formulación variacional de la EDP.

Consideremos una EDP genérica:

$$L(u) = f \quad \text{en } \Omega$$

donde L es un operador diferencial y f es una función de fuente. En el método de Galerkin, buscamos una solución aproximada $u_h(\mathbf{x})$ que satisfaga la *forma débil* de la EDP. Esto se logra multiplicando la EDP por una función de peso (o función de prueba) $w(\mathbf{x})$ y integrando sobre el dominio Ω :

$$\int_{\Omega} w(\mathbf{x}) L(u_h) d\Omega = \int_{\Omega} w(\mathbf{x}) f d\Omega$$

Para suavizar la diferenciabilidad requerida de u_h y transferir derivadas a la función de peso, se aplica integración por partes (o Teorema de Green). Esto resulta en la forma variacional:

$$a(w, u_h) = F(w)$$

donde $a(\cdot, \cdot)$ es una forma bilineal y $F(\cdot)$ es una forma lineal.

La clave del método de Galerkin es elegir las funciones de peso $w(\mathbf{x})$ del mismo espacio funcional que las funciones de base utilizadas para aproximar la solución $u_h(\mathbf{x})$. Es decir, las funciones de peso son las propias funciones de forma globales $\Phi_i(\mathbf{x})$:

$$w(\mathbf{x}) = \Phi_i(\mathbf{x}) \quad \text{para } i = 1, \dots, N_d$$

Sustituyendo $u_h(\mathbf{x}) = \sum_{j=1}^{N_d} \Phi_j(\mathbf{x}) U_j$ y $w(\mathbf{x}) = \Phi_i(\mathbf{x})$ en la forma variacional, obtenemos:

$$a(\Phi_i, \sum_{j=1}^{N_d} \Phi_j U_j) = F(\Phi_i)$$

Por la linealidad de la forma bilineal, esto se convierte en un sistema de ecuaciones lineales:

$$\sum_{j=1}^{N_d} U_j a(\Phi_i, \Phi_j) = F(\Phi_i) \quad \text{para } i = 1, \dots, N_d$$

Este sistema se puede escribir en forma matricial como:

$$\mathbf{K}\mathbf{U} = \mathbf{F}$$

donde:

- \mathbf{K} es la *matriz de rigidez global*, con elementos $K_{ij} = a(\Phi_i, \Phi_j)$.
- \mathbf{U} es el *vector de incógnitas nodales* (los U_j que buscamos).
- \mathbf{F} es el de carga global, con elementos $F_i = F(\Phi_i)$.

Los elementos de la matriz de rigidez y del vector de carga se calculan integrando las funciones de forma (y sus derivadas) sobre cada elemento finito individualmente, y luego ensamblando estos resultados locales en la matriz y el vector globales. La resolución de este sistema lineal proporciona los valores nodales de la solución aproximada.

2.1. PRINCIPALES MODELOS MATEMÁTICOS

De la extensa lista de modelos matemáticos implementados se pueden distinguir las siguientes categorías principales:

1. Mecánica de Fluidos y Fenómenos de Transporte:
 - a) Ecuación de calor: Descripción del flujo de calor en el tiempo con base en la conservación de la energía, se puede considerar como un caso especial de difusión-convección.
 - b) Ecuaciones de Navier-Stokes: Descripción generalizada del comportamiento de los fluidos Newtonianos y no-Newtonianos teniendo en cuenta la conservación de la masa.
 - c) Ecuación de difusión-convección: Sistemas de difusión en fluidos donde se ve afectada la velocidad de flujo.

- d)* Reacción-Convección: Descripción de refractantes o trazadores en un fluido transportados por un flujo.
- e)* Ecuación de Reynolds para flujo de películas finas: Descripción de fluidos para flujo en canales estrechos donde se pueden despreciar fuerzas inerciales.

2. Mecánica de Sólidos:

- a)* Elasticidad Lineal: Uso de desplazamiento de campo en las ecuaciones de Navier, se pueden calcular materiales anisotropicos y cargas térmicas.
- b)* Elasticidad Finita: Descripción de deformaciones finitas sobre un material usando como base la configuración original del material para no asumir la región que ocupa el material deformado.
- c)* Ecuaciones de Cascaron para Elasticidad Clásica: Ecuaciones que describen la deformación de cascarones delgados, es decir superficies curvas descritas por su área media y grosor.
- d)* Ecuaciones de Elasticidad de Placa: Placas elásticas lineales deformables basadas en el modelo de Reissner y Mindlin.
- e)* Elasticidad unidimensional de Vigas: Ecuaciones para vigas estructurales y sus estados de equilibrio en presencia de esfuerzos, se utiliza el equilibrio de fuerzas y momentos experimentados en la sección transversal de la viga.
- f)* Adición de Resortes y Masas Puntuales: Objetos puntuales con comportamiento de masa o resorte que pueden ser añadidos en los nodos de un dominio.

3. Acústica:

- a)* Modelo de Helmholtz: Solucion de la ecuacion de Helmholtz o transformada de Fourier de la ecuacion de onda, se puede tener en cuenta condiciones de frontera especiales y efectos de conveccion.
- b)* Ecuaciones de Navier-Stokes linealizadas en el dominio de la frecuencia: Ecuación acústica de Navier-Stokes tomando en cuenta efectos de viscosidad, compresibilidad y conducción del calor.
- c)* Ecuación de Onda: Solucion de la ecuación de onda generalizada en el dominio del tiempo.
- d)* Movimiento de Ondas de Alta Amplitud en el Aire: Propagación no lineal de ondas en fluidos compresibles usando la ecuacion de Navier-Stokes.

4. Electromagnetismo:

- a)* Electrostática: Descripcion del campo basado en la distribucion de carga en un dielectrico.
- b)* Conducción de corriente estática: Modelo usando corrientes volumétricas y perdidas de potencia por calentamiento de Joule.

- c)* Circuitos y Solucionador Dinámico: Cuando se debe modelar un circuito teniendo en cuenta efectos magnéticos no despreciables.
- d)* Helmholtz Vectorial para Ondas Electromagnéticas:
- e)* Ecuación de Inducción Magnética: Descripción de fluidos y gases conductores y la inducción generada por su movimiento acoplado con la ecuación de Navier-Stokes y la ecuación de calor.
- f)* Electrostática Dimensionalmente Reducida: En algunas distribuciones de carga tridimensionales que puedan representarse como planos de carga, el problema se puede reducir a una dimension.
- g)* Ecuación de Poisson-Boltzmann: Ecuacion modificada para hallar el campo electrico en entornos con cargas libres.
- h)* Estimación de Pérdidas Usando Series de Fourier: Uso de series de Fourier para calcular perdidas de potencia en sistemas electromagnéticos teniendo en cuenta la ecuación de Steinemtzt.
- i)* Solucionador de Corriente de Bobina: Modelo para generalizar bobinas que se desvían de casos ideales donde los cables se modelan como bucles cerrados.

5. Otros Modelos Físicos:

- a)* Electro-cinética: Casos donde propiedades típicas de fluidos cargados llevan a cambios en su volumen o su interacción con recipientes.
- b)* Dinámica de Partículas: Simulación de partículas sujetas a interacción, ya sea gravitatoria de contacto o distintos potenciales.
- c)* Superficies Libres: Busqueda de la superficie para un fluido en condiciones de flujo especificas.
- d)* Fenómenos de Cambio de Fase: Sustancias en transición de fase.

también existe maneras de implementar arreglos o mallas variables en el tiempo, ademas de poder obtener cantidades derivadas como flujos, divergencia, vórtice etc.

Al estar en continuo desarrollo debido a su naturaleza de código abierto, existen modelos que estan implementados pero es desaconsejado su uso debido a estar obsoletos o en fase experimental y por lo tanto tener limitaciones por ser modelos incompletos

2.2. HERRAMIENTAS DE SOLUCION

ELMER cuenta con una amplia gama de sofisticados algoritmos computacionales basados en distintos métodos numéricos y módulos matemáticos para la simulación y solución de los sistemas fisicos anteriormente mencionados.

El Método de Elementos Finitos MEF

La discretización de los sistemas físicos presentados para su adecuada solución usando como base el MEF permite presentar una solución general al conjunto de ecuaciones diferenciales parciales que describen estos fenómenos. Esto implica trabajar en los siguientes pasos:

- **Discretización del Dominio:** El dominio en el que se halla el sistema se debe dividir de manera conveniente para cada caso dependiendo de sus dimensiones y morfología para poder crear una región de solución, donde todas las divisiones se conectan en nodos, a este arreglo que se obtiene al discretizar se le conoce como malla, mallado o mesh.
- **Funciones de Forma:** Dentro de cada división o subdominio se crean funciones de forma que aproximen la solución, el método mas general es el uso de polinomios de interpolación cuyo grado depende de los nodos de la región.
- **Formulación Débil:** Las ecuaciones diferenciales se transforman en una forma integral (débil), típicamente utilizando el método de Galerkin o el métodos variacionales, que es luego resuelta numéricamente, este paso funciona como un modo temprano de evaluación del error del método.
- **Sistemas de Ecuaciones:** Finalmente la formulación de las funciones de forma en cada región permiten llegar a un sistema de ecuaciones algebraicas cuya solución permite hallar el valor de las variables en los puntos nodales.

Solucionadores de Sistemas Lineales

Una vez que las ecuaciones se han discretizado y ensamblado en un sistema lineal (o linealizado), Elmer ofrece diversas estrategias para resolverlo. La elección del solucionador adecuado es crucial para la eficiencia y convergencia:

- **Solucionadores Directos:**
 - **SPOOLES/MUMPS/SuperLU:** Son librerías robustas y computacionalmente muy detalladas que resuelven sistemas lineales y son óptimos para obtener alta precisión en sistemas con moderados grados de libertad.
- **Solucionadores Iterativos:**
 - **Métodos de Krylov:** ELMER cuenta con varios métodos iterativos de Krylov, como **GMRES (Generalized Minimal Residual)**, **BiCGStab (Biconjugate Gradient Stabilized)**, **CG (Conjugate Gradient)**, etc, los cuales permiten la solución de sistemas grandes y de mallas muy finas.
 - **Precondicionadores:** La efectividad de los solucionadores iterativos depende en gran medida del preconditionamiento. Elmer utiliza preconditionadores como **ILU (Incomplete LU factorization)**, **Jacobi**, **multi-grid**, entre otros, para mejorar la condición de los sistemas de ecuaciones y acelerar la convergencia.

Algoritmos para Problemas Transitorios y No Lineales

Varios de los problemas pueden implicar problemas donde el mallado depende del tiempo o donde se presenta no linealidad, para ello se tienen los siguientes metodos:

- **Integración Temporal:** Para problemas transitorios, se utilizan esquemas de integración temporal (por ejemplo, métodos de Euler implícitos/explicitos, o métodos de Runge-Kutta) para avanzar la solución paso a paso en el tiempo, pueden llegar a ser mas costosos pero para varios casos son la única opción.
- **Métodos para No Linealidades:** Los problemas no lineales se resuelven típicamente mediante esquemas iterativos. El método de **Newton-Raphson** es comúnmente utilizado para linealizar el sistema de ecuaciones en cada iteración y resolverlo hasta alcanzar una convergencia dentro de una tolerancia deseada. Esto se aplica a no linealidades geométricas (grandes deformaciones), materiales (plasticidad, hiperelasticidad) o de contacto.
- **Acoplamiento de Físicas:** Una capacidad clave de ELMER es su flexibilidad para acoplar diferentes solucionadores (físicas) dentro de un mismo problema. Esto se puede hacer de forma *monolítica* (resolviendo todas las ecuaciones acopladas simultáneamente) o *secuencial* (resolviendo cada física por separado y actualizando las interfaces de forma iterativa), la capacidad de acoplamiento es clave para los modelos mas detallados y complejos que requieren tener en cuenta la mayor cantidad de variables y condiciones posibles.

Manejo de la Malla y Post-Procesamiento

Existen metodos de integracion computacional que permiten trabajar de manera adecuada con la malla:

- **Manejo de Mallas:** ElmerGrid, una utilidad complementaria, se encarga de la generación, conversión y manipulación de mallas. ElmerSolver puede realizar **adaptación de malla** (refinamiento o engrosamiento) de forma adaptativa durante la simulación para mejorar la precisión en regiones de interés, este post procesamiento suele incurrir en mayor costo computacional pero es imprescindible en casos donde se requiere alcanzar una tolerancia establecida.
- **MATC (Matrix and Tensor Calculator):** Una biblioteca numérica integrada que permite la evaluación de expresiones matemáticas complejas en el archivo de entrada de ElmerSolver, lo que ofrece gran flexibilidad para definir propiedades de materiales, condiciones de contorno o funciones fuente de forma personalizada.

Es evidente entonces que ELMER cuenta con un eficaz catalogo que permite la solucion de un sin fin de problemas físicos de todo tipo.

3. DESCARGA PARA *WINDOWS*

En este capítulo se realizará el paso a paso para la instalación, se utiliza *Windows 10* como sistema operativo, pero el proceso aquí presentado funciona para sistemas *Windows* anteriores a *Windows 10* de **64 bits**.

3.1. Descarga de Elmer

Para instalar la última versión de *Elmer*, puede acceder al sitio oficial a través del siguiente enlace: <https://www.nic.funet.fi/pub/sci/physics/elmer/>. En este link se encuentra la interfaz que se encuentra en la figura 2

Index of /pub/sci/physics/elmer

Name	Last modified	Size	Description
 Parent Directory		-	
 anim/	2010-11-10 19:28	-	
 bin/	2021-02-04 13:48	-	
 courses/	2021-03-04 12:04	-	
 doc/	2022-08-02 11:38	-	
 macros/	2014-08-29 17:28	-	
 slides/	2021-02-04 14:47	-	
 src/	2021-02-04 13:45	-	
 tests/	2021-03-07 23:57	-	
 webinar/	2021-12-22 13:27	-	
 Readme.txt	2021-02-04 13:53	813	
 tests.filepart	2025-06-24 05:00	29M	

Figura 2: Primera interfaz

En esta interfaz se pueden observar diferentes carpetas, para descargar *Elmer* se debe ingresar a la carpeta "bin" después "Windows".

Index of /pub/sci/physics/elmer/bin/windows













Name	Last modified	Size	Description
 Parent Directory		-	
 old-gcc9/	2021-04-06 15:07	-	
 rel9.0/	2022-08-02 11:45	-	
 scripts/	2023-12-18 09:21	-	
 ElmerFEM-gui-mpi-Windows-AMD64.exe	2025-06-24 02:09	159M	
 ElmerFEM-gui-mpi-Windows-AMD64.zip	2025-06-24 02:09	219M	
 ElmerFEM-gui-nompi-Windows-AMD64.exe	2025-06-24 02:08	150M	
 ElmerFEM-gui-nompi-Windows-AMD64.zip	2025-06-24 02:08	210M	
 ElmerFEM-nogui-mpi-Windows-AMD64.exe	2025-04-23 06:22	94M	
 ElmerFEM-nogui-mpi-Windows-AMD64.zip	2025-04-23 06:21	130M	
 ElmerFEM-nogui-nompi-Windows-AMD64.exe	2025-04-22 20:12	89M	
 ElmerFEM-nogui-nompi-Windows-AMD64.zip	2025-04-22 20:12	124M	
 ElmerFEM-tests.tar.gz	2021-04-06 14:46	25M	
 Readme1st.txt	2021-03-08 21:26	2.6K	
 RunElmer.bat	2021-12-01 21:32	164	

Figura 3: Segunda interfaz

Hay cuatro categorías principales para seleccionar, tales como 'gui-mpi', 'gui-nompi', 'nogui-mpi' y 'nuginompi'. También hay algunas etiquetadas con 'rel9.0', que son compilaciones de la versión oficial más reciente y deben considerarse como las versiones estables. Los demás archivos no etiquetados con 'rel9.0' son las compilaciones diarias ('nightly builds') y representan las versiones más recientes. Las compilaciones diarias son, de hecho, bastante estables, por lo que es recomendable empezar con una de ellas.

Para la primera instalación de Elmer en *Windows*, se recomienda seleccionar 'gui-mpi' y 'rel9.0'. Los archivos con extensión .exe son instaladores para Windows. Así, de toda la lista, se descarga este archivo:

- **ElmerFEM-gui-mpi-Windows-AMD64-rel9.0.exe**

Si decide empezar con la compilación diaria, entonces descargue este archivo:

- **ElmerFEM-gui-mpi-Windows-AMD64.exe**

3.2. Descargar la documentación

Para descargar la documentación se vuelve nuevamente a la interfaz 1 (imagen 2), después se ingresa a la carpeta "doc", aquí se encuentra la documentación más reciente.

Index of /pub/sci/physics/elmer/doc

Name	Last modified	Size	Description
 Parent Directory		-	
 old/	2009-08-04 13:52	-	
 ElmerDocumentation.tar.gz	2023-02-09 14:47	24M	
 ElmerDocumentation.zip	2023-02-09 14:47	24M	
 ElmerGUIManual.pdf	2023-04-06 19:01	637K	
 ElmerGridManual.pdf	2023-04-06 19:01	1.3M	
 ElmerModelsManual.pdf	2024-06-07 12:54	1.8M	
 ElmerOverview.pdf	2023-04-06 19:01	159K	
 ElmerParamManual.pdf	2022-08-02 11:39	190K	
 ElmerProgrammersTutorial.pdf	2022-10-17 14:19	165K	
 ElmerSolverManual.pdf	2023-04-06 19:01	1.6M	
 ElmerTutorials-nonGUI.pdf	2023-04-06 19:01	1.6M	
 ElmerTutorials.pdf	2023-04-06 19:01	13M	
 GetStartedElmer.pdf	2023-04-06 19:01	8.1M	
 MATCManual.pdf	2023-04-06 19:01	163K	
 README.txt	2015-10-28 09:33	2.3K	
 TexStart.pdf	2023-04-06 19:01	224K	
 TexTemplate.pdf	2023-04-06 19:01	208K	
 tutorials-CL-files.tar.gz	2023-04-06 19:06	2.5M	
 tutorials-CL-files.zip	2023-04-06 19:06	2.5M	
 tutorials-GUI-files.tar.gz	2023-04-06 19:06	19M	
 tutorials-GUI-files.zip	2023-04-06 19:06	18M	

Figura 4: Interfaz de la carpeta "doc"

De esta carpeta se descarga el archivo:

- ElmerDocumentation.zip

En el archivo .zip se encuentran 4 documentos PDF importantes:

- GetStarted.
- ElmerTutorials.
- ElmerGUI Manual
- ElmerSolverManual.

3.3. Paraview

Paraview es una herramienta comunmente utilizada para la visualización de resultados generados por simulaciones que general *Elmer*.

No es completamente necesario instalar esta herramienta para que se pueda utilizar **Elmer**, pero será necesario para visualizar los resultados, para descargar *Paraview* se hace con el siguiente enlace: <https://www.paraview.org/download/>

Descargue un archivo .msi como se observa en la figura 5, en este caso se descargó el siguiente archivo.

- *ParaView-6.0.0-RC1-Windows-Python3.12-msvc2017-AMD64.msi*

Release Candidates

Previews of ParaView's next release. Full suite of ParaView tools, including the ParaView [MS-MPI](#).

Unzip .zip packages in a directory with a very short path if you encounter an error mess

- 📄 [ParaView-6.0.0-RC1-Windows-Python3.12-msvc2017-AMD64.msi](#)
- 📄 [ParaView-6.0.0-RC1-MPI-Windows-Python3.12-msvc2017-AMD64.msi](#)
- 📄 [ParaView-6.0.0-RC1-Windows-Python3.12-msvc2017-AMD64.zip](#)
- 📄 [ParaView-6.0.0-RC1-MPI-Windows-Python3.12-msvc2017-AMD64.zip](#)

Figura 5: Interfaz de la página Paraview

4. INSTALACIÓN PARA *WINDOWS*

Después de todo el proceso de descarga y de guardarlo en una carpeta, se debería tener algo como en la figura 6.

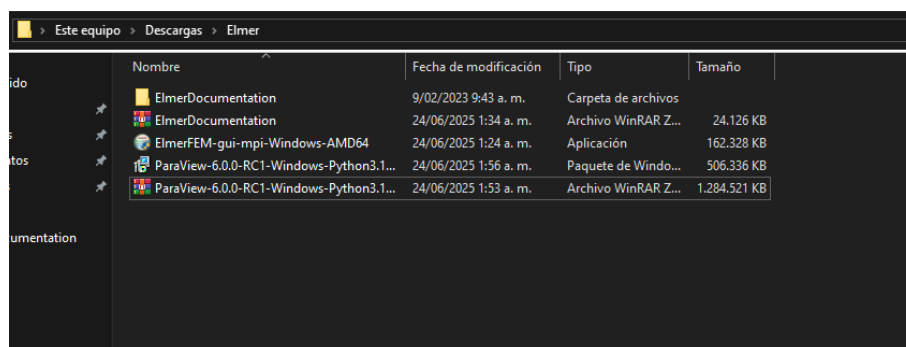


Figura 6: Carpeta base de descarga

En la carpeta se puede observar el ejecutable *ElmerFEM-gui-mpi-Windows-AMD64*, este instalará ***ElmerGUI***, ***ElmerSolver***, ***ElmerSolver_mpi*** y ***Microsoft MPI***.

ElmerGUI es la interfaz gráfica de usuario (GUI por sus siglas en inglés) de Elmer. Permite interactuar con el software de una manera visual, facilitando la configuración de simulaciones, la definición de modelos, entre otros; Sin necesidad de escribir código directamente.

ElmerSolver es el solucionador principal de Elmer. Es el motor de cálculo que realiza las simulaciones físicas. Toma la información de la configuración del modelo (creada, por ejemplo, con ElmerGUI) y ejecuta los cálculos numéricos para resolver las ecuaciones diferenciales parciales que describen el fenómeno físico que estás simulando. Este solucionador funciona de forma serial, es decir, utiliza un solo procesador o núcleo de la CPU para realizar los cálculos.

ElmerSolver_mpi es una versión paralela del solucionador de Elmer. La extensión "_mpi"(que significa Message Passing Interface) indica que esta versión está optimizada para ejecutarse en múltiples procesadores o núcleos de CPU simultáneamente. Esto es crucial para simulaciones muy grandes y complejas, ya que distribuye la carga de trabajo entre varios recursos computacionales, acelerando significativamente el tiempo de cálculo. La instalación de *gui-mpi* incluye Microsoft MPI para habilitar esta capacidad.

4.1. Instalando Elmer

Para realizar la instalación se debe ejecutar *ElmerFEM-gui-mpi-Windows-AMD64*, al presionar se encontrará este error.

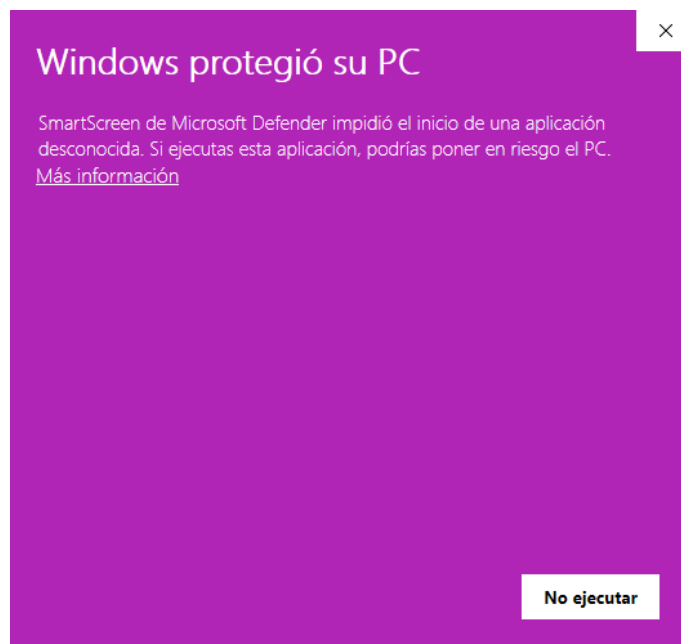


Figura 7: Protección de Windows

La primera ventana tiene un botón que dice "Más información", después de presionar este botón aparecerá la opción de **ejecutar de todas formas**.

A partir de aquí simplemente se instala con los ajustes de default hasta que se llegue a la ventana de **opciones de instalación**, aquí se puede añadir el camino a Elmer, se debe asegurar de presionar el botón **Add Elmer to the system path for ...**, aparecerán 2 opciones después de esos puntos suspensivos, si no está seguro seleccione "for current user", adicionalmente **marque la casilla Create Elmer Desktop Icon**", tal como se puede observar en la figura 8

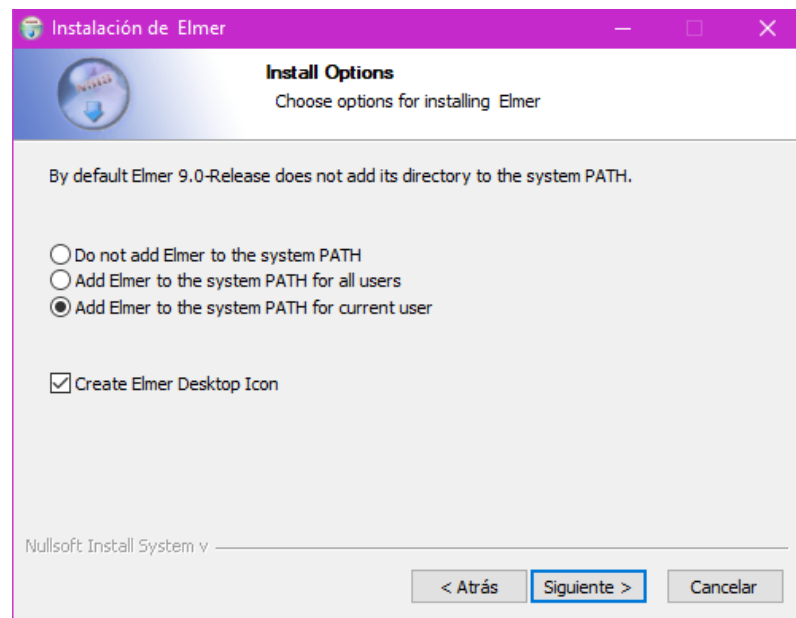


Figura 8: Pestaña Opciones de instalación

A partir de aquí nuevamente acepte los ajustes de default y se debería instalar Elmer correctamente, se llegará a una pestaña en la cuál se muestran los componentes a instalar, deben estar seleccionados todos como en la figura 9

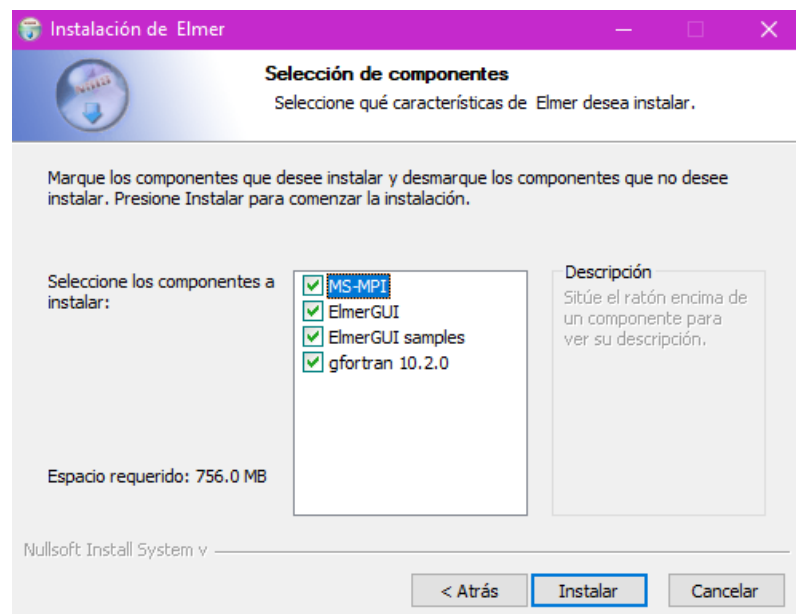


Figura 9: Componentes a instalar

Al seguir se encontrará con la pestaña de instalación 10.

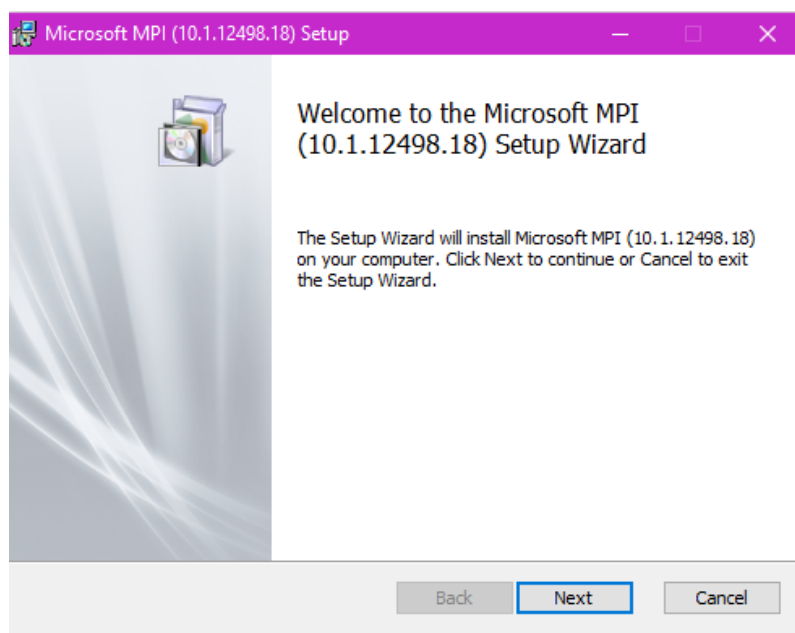


Figura 11: Caption

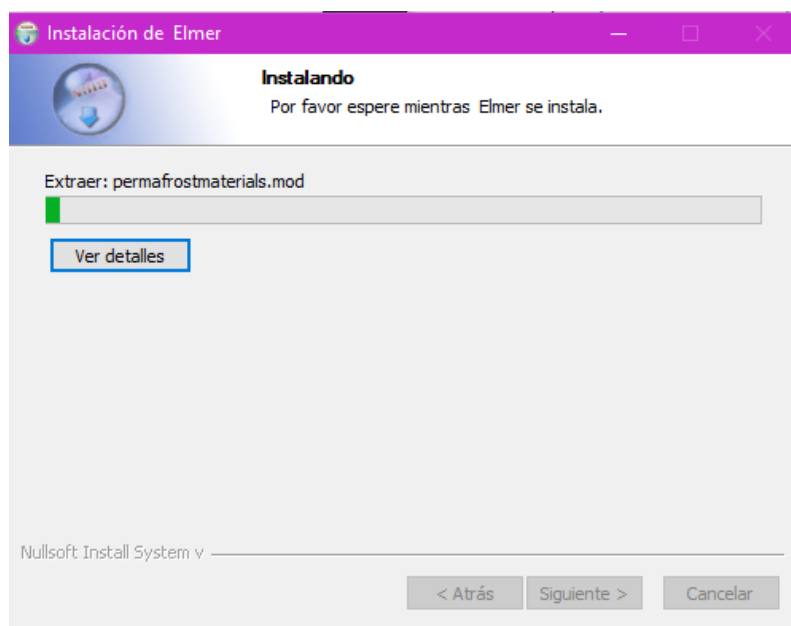


Figura 10: Instalando

Después del proceso de instalación aparecerá la pestaña de la figura 11.

Para finalizar encontrará la última ventana de instalación, tal como en la figura 12

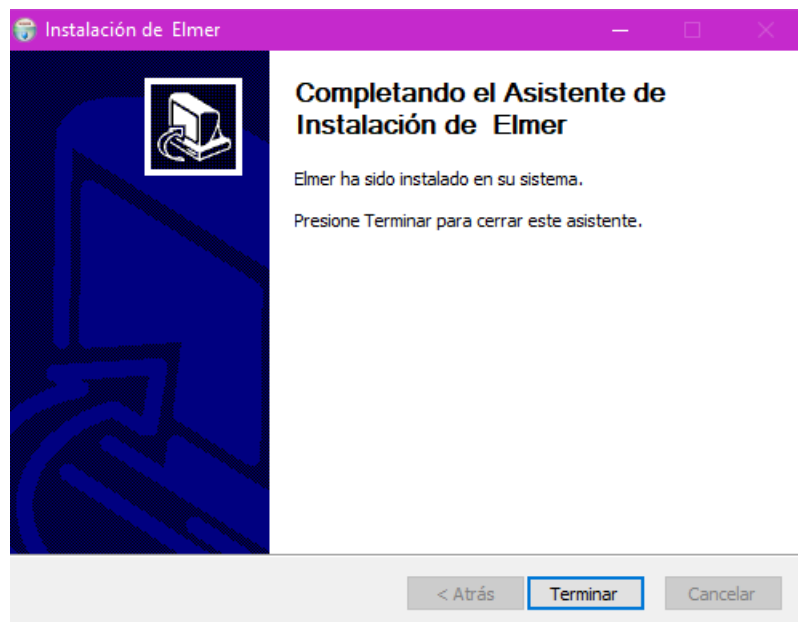


Figura 13: Caption

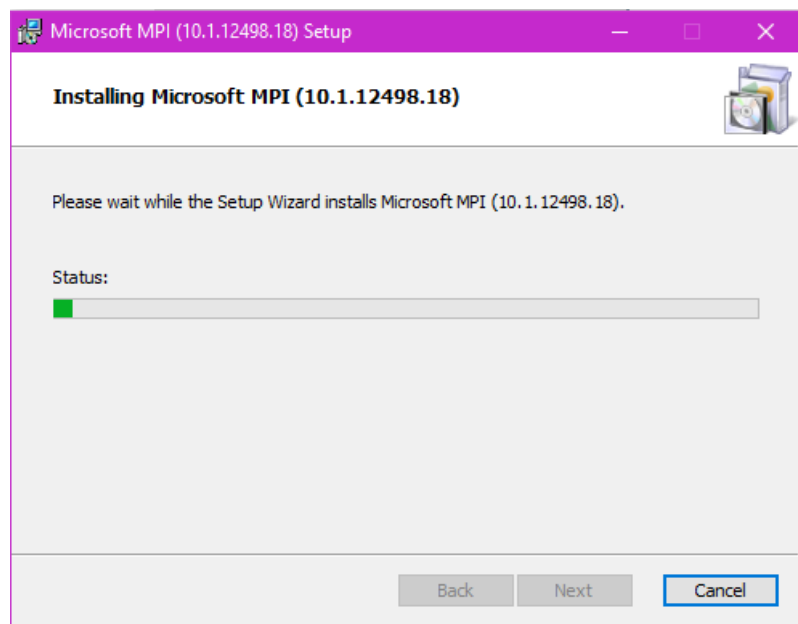


Figura 12: Pantalla final de instalación

En caso de que le aparezca un error como en la figura 14 durante el proceso de instalación, presiones el botón **Ok**. Este mensaje aparece si ha instalado anteriormente *Microsoft MPI*. Esto ocurrirá cada vez que instale una versión más reciente de *Elmer* con *MPI*.

Después de este proceso aparecerá la ventana final del asistente de instalación de *Elmer*, como se puede observar en la figura 13.

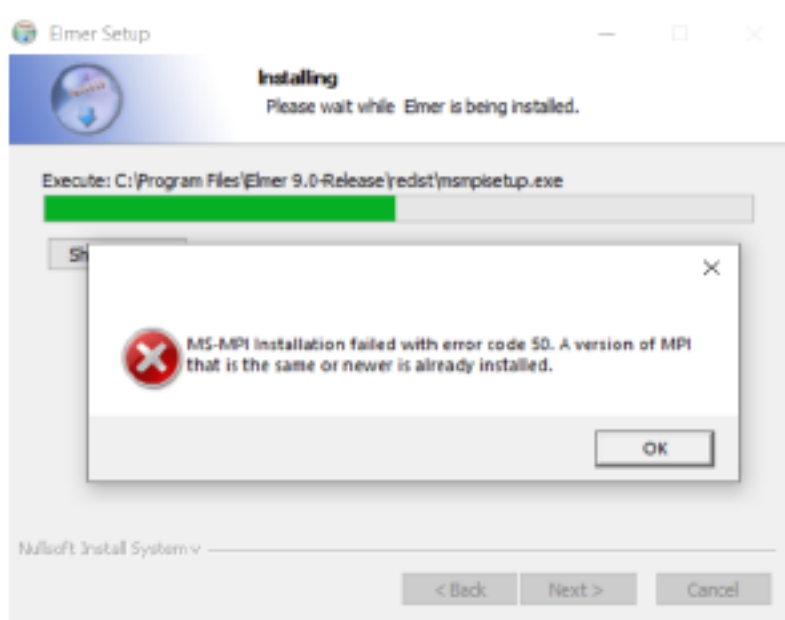


Figura 14

Aquí ya se tiene *Elmer* instalado, en caso de que lo abra y ejecute algún comando, se encontrará un error **Unable to start Paraview**, dado que aún no se ha instalado la solución a correr el código y presionar **Start ElmerVTK**.

4.2. Intalando Paraview

En una sección anterior ya se explicó el proceso de descarga del instalador, se debe realizar doble click en este para empezar el proceso de instalación. Simplemente se deben aceptar todas las opciones pre-seleccionadas de instalación y se instalará correctamente.

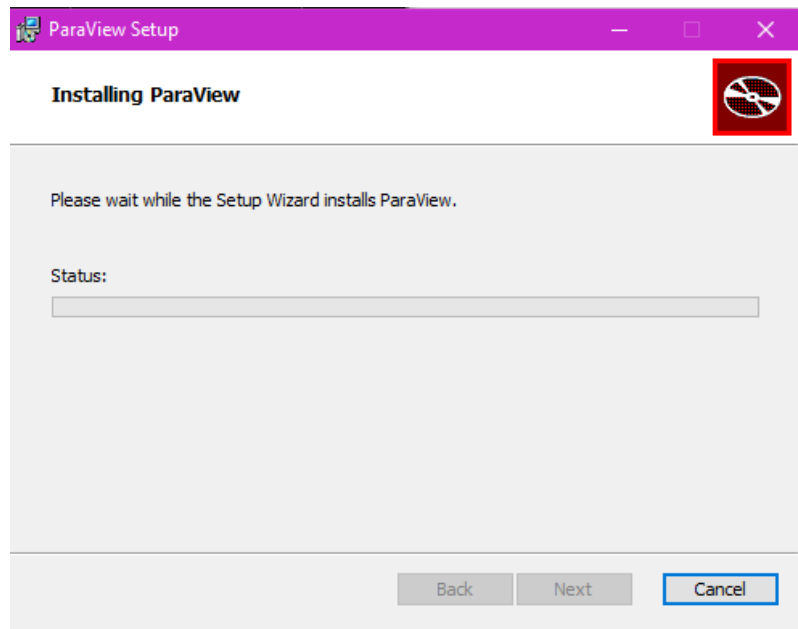


Figura 15: Instalación de Paraview

El instalador no añadirá Paraview al camino en el sistema de Elmer, por lo tanto esto se debe realizar manualmente.

5. Windows- Camino de Elmer

5.1. El camino a Elmer

Si se establece un camino a *Paraview*, entonces *ElmerGUI* podrá correr *Paraview* para visualizar los archivos **.vtu**.

Después de instalar *Paraview*, se debe ir a configuraciones de Windows , después escribir en el buscador **variables** y aparecerá la opción de **editar variables del entorno de esta cuenta** (figura 16).

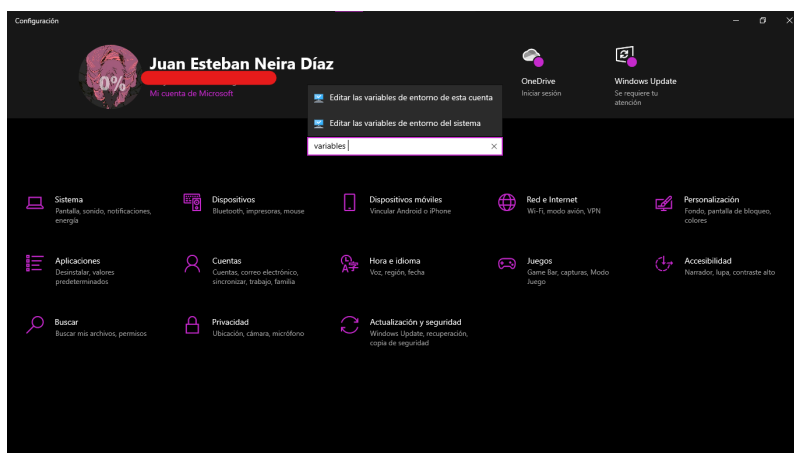


Figura 16: Configuraciones de Windows

Cuando se instaló *Elmer*, el camino a este se creó. Para verificar que el camino existe, en la ventana después de presionar **editar variables del entorno de esta cuenta** seleccionar editar en el último camino creado, que es el de *Elmer* (figura 17).

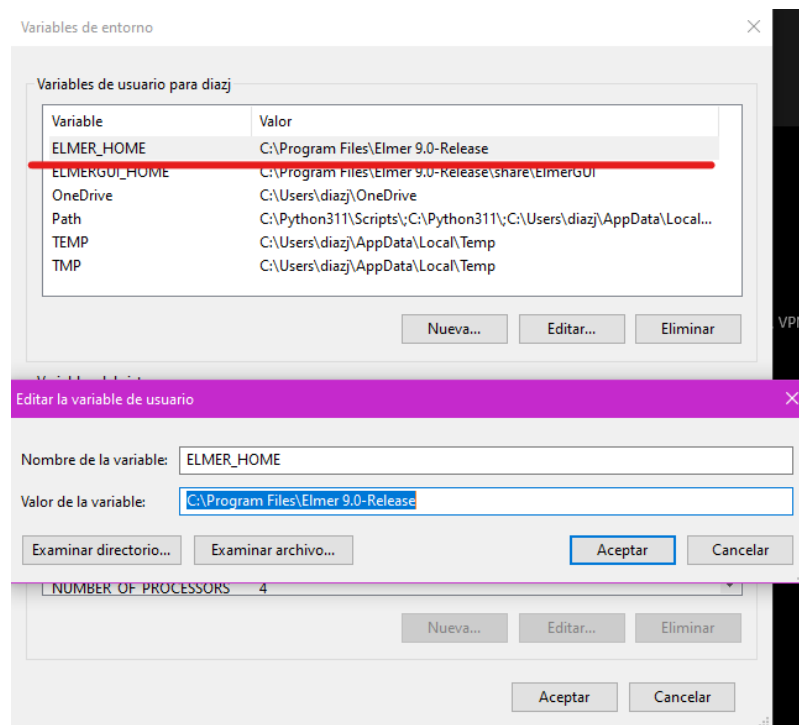


Figura 17: Variables

5.2. Camino a Paraview

Mientras tenga la pestaña de la figura 17 abierta, presione en el botón nuevo y añada el camino a Paraview.

Una manera de determinar con precisión la ruta de instalación de Paraview es la siguiente: abra el Explorador de archivos y seleccione la unidad del sistema, como por ejemplo C:. Luego, acceda a la carpeta Archivos de programa (Program Files) y, dentro de ella, localice y abra la carpeta correspondiente a Paraview. A continuación, ingrese a la carpeta bin, donde encontrará un archivo denominado Paraview.exe; esta es la ubicación que usted debe ingresar en la variable de entorno PATH.

5.3. Correr Paraview desde ElmerGUI

Abra la aplicación **ElmerGUI** (figura 18), seleccione **Run** y debería ver una ventana que diga **Start Paraview** (figura 19), posteriormente se verá corriendo Paraview.

Si encuentra una ventana de error como critical: In unknown, line 0', significa que no encontró un archivo **.vtu** válido y mostrará una pantalla gris. (Figura 20)



Figura 18: Aplicación en el sistema

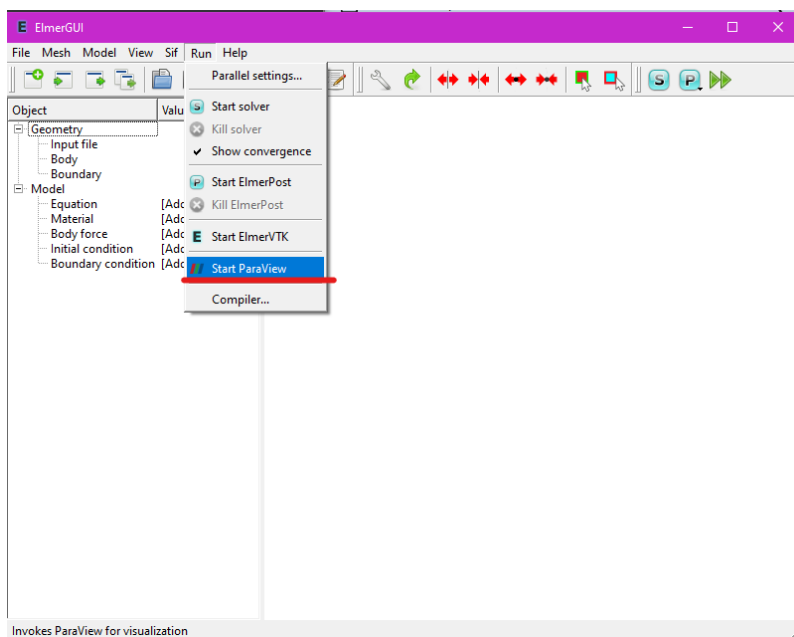


Figura 19: Elmer corriendo

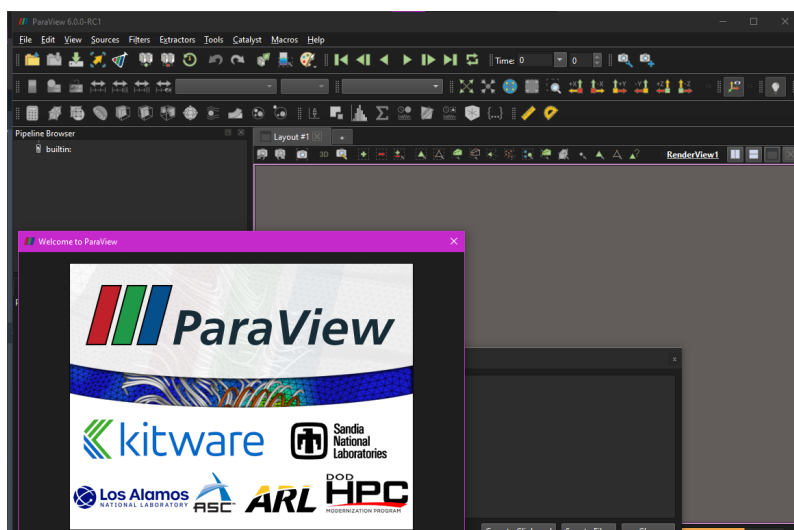


Figura 20: Paraview abierto después de correr Elmer

6. Creando un primer proyecto en Elmer

En esta caso se cargará uno de los ejemplos que da *Elmer*, se toma el ejemplo *ElasticHookNonLinear*, en *Elmer*, se selecciona la opción **FILE**, posteriormente se selecciona **Loas Mesh**, se abrirá una ventana en el cuál se navega hasta la carpeta del ejemplo y se carga, posteriormente *Elmer* cargará la malla (figura 22).

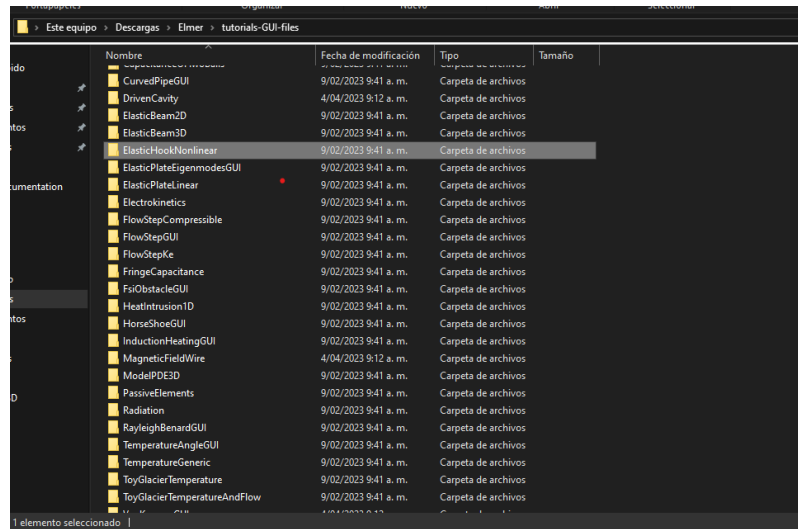


Figura 21: Carpeta que contiene los archivos necesarios para correr una simulación

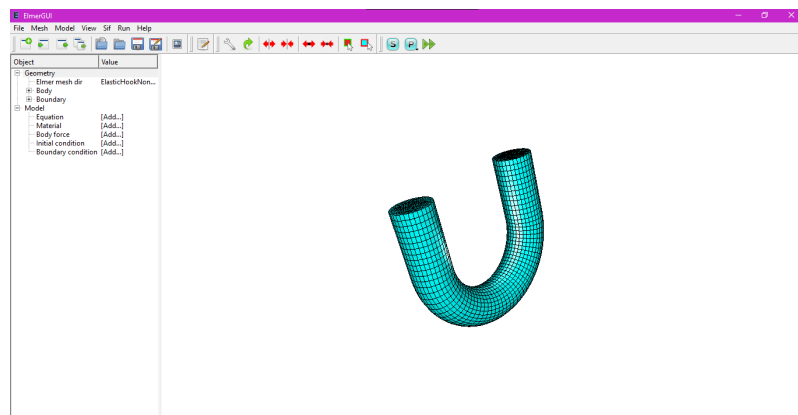


Figura 22: Malla cargada en *Elmer*

7. Introducción rápida a Paraview

Después de la ventana de la figura 20, y de cargar un archivo .vtu válido, aparecerá un botón verde ".apply", presionando este botón la geometría aparecerá.

Se realiza un ejemplo simple para verificar que funcione correctamente.

7.1. Descripción de un problema sencillo

Una estructura en forma de L (ver figura 23) se calienta mediante una fuente de calor interna, cuya magnitud es de 1 W/m^3 . La densidad de la estructura es de 1 kg/m^3 y la conductividad térmica es de 1 W/mK . Todas las fronteras Γ_i se mantienen a una temperatura constante de 0 K . El problema es resolver la distribución de temperatura en la estructura. Matemáticamente, el problema a resolver es

$$\begin{cases} -\kappa \Delta T &= \rho f & \text{en } \Omega \\ T &= 0 & \text{en } \Gamma \end{cases} \quad (2,1)$$

donde κ es la conductividad térmica, T es la temperatura y f es la fuente de calor. Se asume que la densidad y la conductividad térmica son constantes.

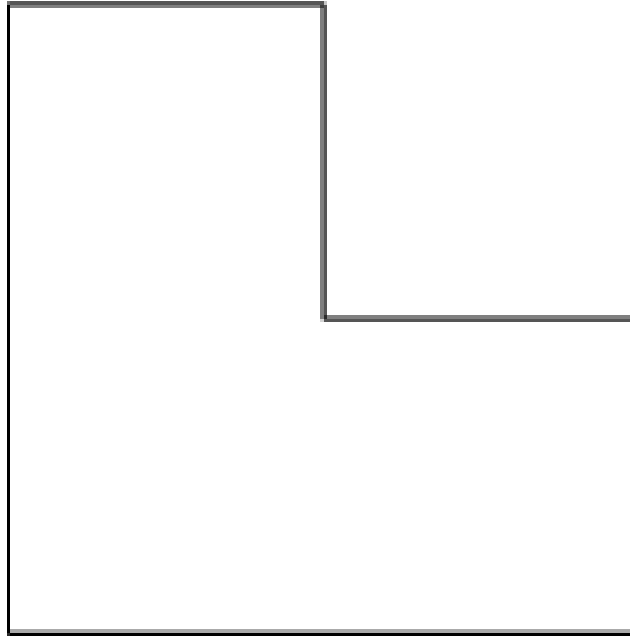


Figura 23: Imagen del sistema físico

Posteriormente se debe tener una carpeta con archivos que definan la estructura del problema físico (figura 24), se importa la carpeta a Elmer y se genera la malla (figura 25).

1. Se verifica que en la carpeta estén las condiciones de frontera.

Nombre	Fecha de modificación	Tipo	Tamaño
angle.grd	26/06/2025 9:04 p. m.	Archivo GRD	1 K
case_t0001.vtu	26/06/2025 9:04 p. m.	Archivo VTU	22 K
egproject.xml	26/06/2025 9:04 p. m.	Microsoft Edge H...	69 K
ELMERSOLVER_STARTINFO	26/06/2025 9:04 p. m.	Archivo	1 K
mesh.boundary	26/06/2025 9:04 p. m.	Archivo BOUNDA...	2 K
mesh.elements	26/06/2025 9:04 p. m.	Archivo ELEMENTS	8 K
mesh.header	26/06/2025 9:04 p. m.	Archivo HEADER	1 K
mesh.nodes	26/06/2025 9:04 p. m.	Archivo NODES	6 K

Figura 24: Imagen con archivos de frontera

2. Importar archivo

Inicie ElmerGUI desde la línea de comandos o haciendo clic en el ícono en su escritorio. Aquí describimos los pasos esenciales en ElmerGUI escribiendo el procedimiento de clic.

La malla se da en formato ElmerGrid en el archivo `angle.grd` en el directorio de ejemplos de ElmerGUI, cargue este archivo.

- File
- Open -> angle.grd

Debería obtener su malla y verificar en la ventana de resumen del Modelo que consta de 341 nodos y 300 elementos bilineales. Si la malla se importó con éxito, su ventana debería verse como en la figura 25.

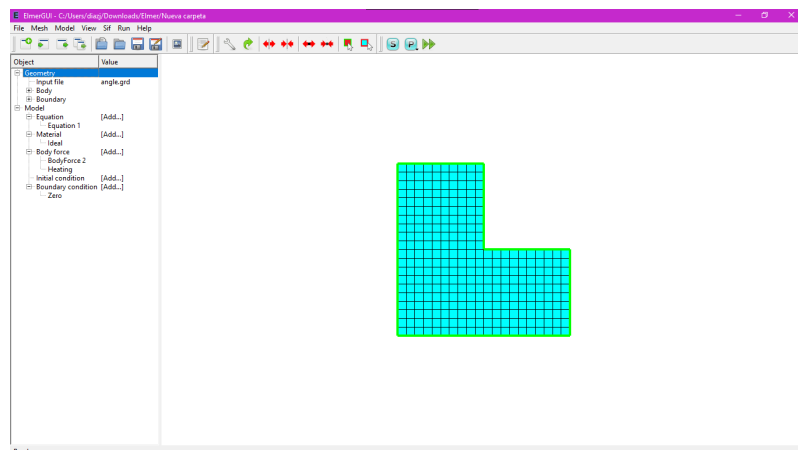


Figura 25: Malla generada por archivos que definen las condiciones iniciales del problema

3. Estableciendo parámetros físicos del sistema

Después de tener la malla, comenzamos a revisar el menú Modelo de arriba hacia abajo. En la sección de Configuración (Setup) elegimos cosas relacionadas con toda la simulación, como nombres de archivo, pasos de tiempo, constantes, etc. La simulación se realiza en coordenadas cartesianas bidimensionales y en

estado estacionario. Solo se necesita una iteración en estado estacionario ya que el caso es lineal.

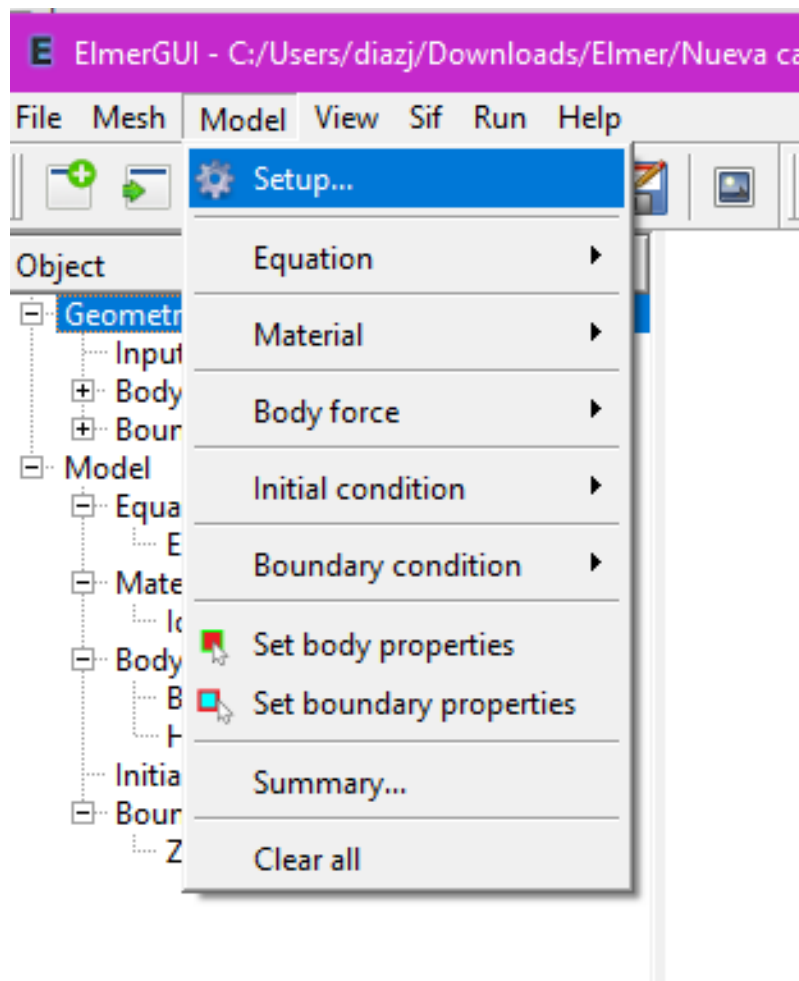


Figura 26: Ventana de ajuste de parámetros

- Model
 - Setup
 - Simulation Type = Steady state
 - Steady state max. iter = 1

Elija Aceptar para cerrar la ventana.

En la ventana de Ecuaciones, elegimos las ecuaciones y parámetros relevantes relacionados con su solución. En este caso, solo tenemos una ecuación – la ecuación de calor. Al definir Ecuaciones y Materiales, es posible asignarlos a los cuerpos inmediatamente. En este caso, solo tenemos un cuerpo y una frontera y, por lo tanto, es más sencillo.

- Model
 - Equation
 - Add
 - ◊ Name = Heat Equation

- ◊ Apply to bodies = 1
- ◊ Heat Equation
- ◊ Active = on
- Apply
- OK

La sección de Materiales incluye todos los parámetros del material.

- Model
 - Material
 - Add
 - ◊ Name = Ideal
 - ◊ Apply to bodies = 1
 - ◊ General
 - ◊ Density = 1.0
 - ◊ Heat Equation
 - ◊ Heat Conductivity = 1.0
 - Apply
 - OK

Una Fuerza de Cuerpo (Body Force) representa el lado derecho de una ecuación que en este caso representa la fuente de calor.

- Model
 - Body Force
 - Add
 - ◊ Name = Heating
 - ◊ Heat Source = 1.0
 - ◊ Apply to bodies = 1
 - Apply
 - OK

No se requieren condiciones iniciales en el caso de estado estacionario. En este caso, solo tenemos una frontera y la configuramos a cero.

- Model
 - BoundaryCondition
 - Add
 - ◊ Heat Equation
 - ◊ Temperature = 0.0
 - ◊ Name = Zero
 - ◊ Apply to boundaries = 1
 - Apply
 - OK

4. Generando el archivo .sift.

Para la ejecución, ElmerSolver necesita los archivos de malla y el archivo de comandos. Ahora hemos definido básicamente toda la información para que

ElmerGUI escriba el archivo de comandos. Después de escribirlo, también podemos inspeccionar visualmente el archivo de comandos.

- Sif
 - Generate
 - Edit ->look how your command file came out

5. Guardando el archivo

Antes de que podamos ejecutar el solucionador, debemos guardar los archivos en un directorio. Al guardar el proyecto, todos los archivos necesarios para reiniciar el caso se guardarán en el directorio de destino.

- File
 - Save Project

6. Corriendo la simulación

Después de haber guardado los archivos con éxito, podemos iniciar el solucionador.

- Run
 - Start solver

Una vista de convergencia aparece automáticamente mostrando los cambios relativos de cada iteración. Como el caso es lineal, solo se requirió una iteración para la solución y la segunda solo es necesaria para verificar la convergencia. El registro de salida resultante se muestra en la figura 27 y 28. Nota: si encuentra problemas en la fase de solución y necesita editar la configuración, siempre recuerde guardar el proyecto antes de la ejecución.

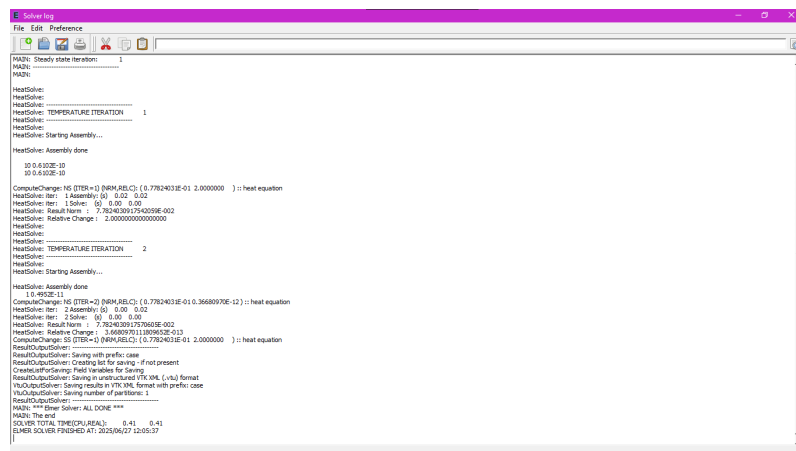


Figura 27: Solver Log

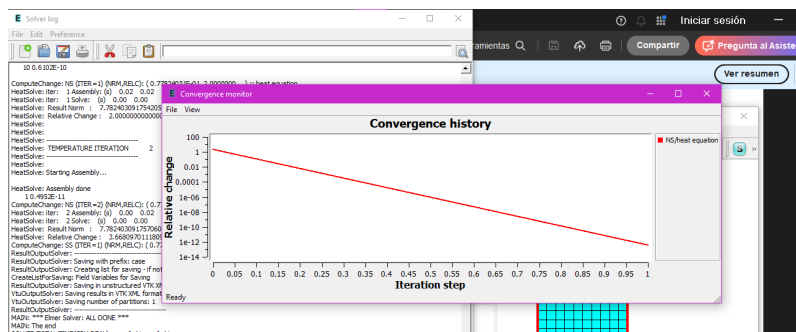


Figura 28: Historia de convergencia

7. Ver los resultados en Paraview

Para ver los resultados, usamos Paraview.

- Run
 - Start Paraview

Para visualizar correctamente la solución, se deben tener activas las opciones señaladas con una línea roja en la figura 29

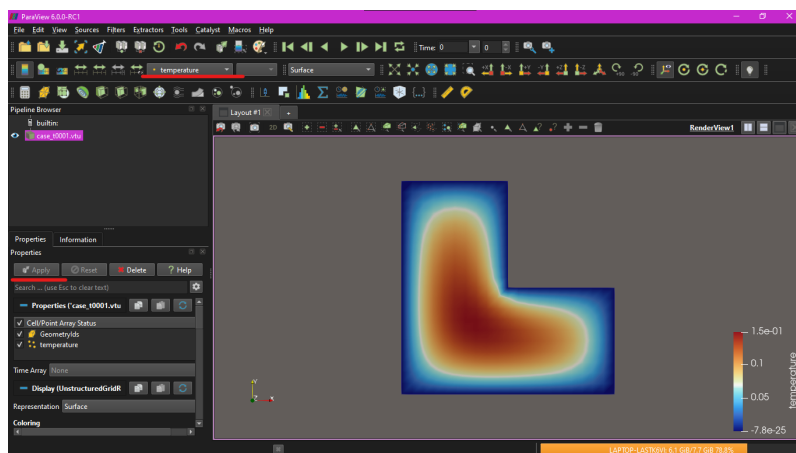


Figura 29: Imagen del sistema físico solucionado en paraview

8. INSTALACION EN LINUX

8.1. Actualizar y Limpiar

Al configurar ElmerFEM en un dispositivo Linux es preferible asegurarse de que el sistema este limpio y actualizado para poder proceder con la instalacion de librerias y herrmientas requeridas por Elmer. Este proceso puede demorarse en la actualizacion, pues se aplica a cada programa del sistema

```
1 sudo apt update
2 sudo apt remove --purge -y python3-paraview || true
3 sudo apt --fix-broken install -y
4 sudo apt autoremove -y
```

- Actualizamos el sistema
- Nos deshacemos de posibles previas instalaciones de paraview que pueden llegar a generar problemas
- Reparamos cualquier archivo a medio instalar
- Limpiamos dependencias sin usar

8.2. Instalar Dependencias

```
1 sudo apt install -y \
2   build-essential git cmake wget \
3   qtbase5-dev qttools5-dev-tools qtscript5-dev libqt5svg5-dev \
4   libqwt-qt5-dev \
5   tetgen netgen \
6   liboce-foundation-dev liboce-modeling-dev liboce-ocaf-dev \
7   liboce-visualization-dev \
8   default-jdk \
9   libvtk9-dev libvtk9-qt-dev \
10  paraview \
11  python3-pyqt5 python3-pyqt5.qtsvg python3-pyqt5.qtwebkit \
   python3-pip
```

Requeriremos de programas como build-essential, git, cmake, wget (Compiladores, control, etc). Los paquetes de desarrollo Qt5 se utilizan en la interfaz visual de ElmerGUI. Tetgen y netgen son generadores de maya. OpenCASCADE permitira importar archivos STEP que se encargan de geometrias en los sistemas. default-jdk se utiliza en la consola de ElmerGUI. Librerias VTK tambien se encargan de la visualizacion y correcto funcionamiento de esta. Finalmente esta ParaView en un version que no cause problemas.

8.3. Librerías de Java

ElmerGUI utiliza internamente algunos componentes basados en Java (por ejemplo, la consola de scripts). Sin embargo, las bibliotecas compartidas de Java (libjvm.so) no son visibles automáticamente para el enlazador del sistema por defecto. Este paso indica al sistema dónde encontrar dichas bibliotecas.

```

1 JAVA_BIN=$(which java)
2 JAVA_HOME=$(dirname "$(dirname "$(readlink -f "$JAVA_BIN")")")
3 echo "$JAVA_HOME/lib" | sudo tee /etc/ld.so.conf.d/java.conf
4 echo "$JAVA_HOME/lib/server" | sudo tee -a /etc/ld.so.conf.d/
   java.conf
5 sudo ldconfig

```

8.4. Clonar código fuente y crear build

Obtenemos el repositorio oficial de ElmerFEM de GitHub y creamos una carpeta de compilación dedicada,

```

1 git clone https://github.com/ElmerCSC/elmerfem.git
2 cd elmerfem
3 mkdir -p build && cd build

```

8.5. Configurar Cmake

CMake es un generador de sistemas de compilación. Lee los archivos de configuración del código fuente de Elmer (llamados CMakeLists.txt) y determina cómo compilarlo en el ordenador.

Comprueba las bibliotecas instaladas (como Qt, VTK u OpenMPI), detecta el compilador y crea todos los archivos Makefile necesarios para que se pueda compilar Elmer correctamente con make.

```

1 cmake .. \
2   -DCMAKE_INSTALL_PREFIX=/usr/local/share/elmersolver \
3   -DDEFAULT_ELMER_HOME=/usr/local/share/elmersolver \
4   -DCMAKE_INSTALL_LIBDIR=lib \
5   -DWITH_ELMER_SOLVER=TRUE \
6   -DWITH_ELMERGUI=TRUE \
7   -DWITH_QWT=TRUE \
8   -DWITH_VTK=TRUE \
9   -DWITH_OCC=FALSE \
10  -DWITH_TETGEN=FALSE \
11  -DWITH_QT5SCRIPT=TRUE \
12  -DWITH_PYTHONQT=FALSE \
13  -DWITH_MATC=TRUE \
14  -DWITH_PARAVIEW=FALSE \
15  -DWITH_QT5=TRUE

```

Las dos primeras líneas indican donde PONER todo (binaries, librerías, módulos) y donde BUSCAR sus propios archivos al correr. EL resto permite activa o desactivar distintos componentes opcionales. A continuación debemos correr:

```
1 make -j "$(nproc)"
2 sudo make install
```

Compilamos con `make -j $ (nproc)`, que ejecuta un trabajo por núcleo de CPU para lograr mayor velocidad, y `sudo make install` que copia todo en `/usr/local/share/elmersolver`.

8.6. Configurar Variables de entorno

Este paso lo realizamos simple despues del `make install` con tal de limpiar previas configuraciones. Este bloque se encarga de establecer rutas de acceso que se utilizaran en el código, para esto creamos y editamos directamente un archivo `profile`, que dentro de la terminal es hacer:

```
1 cat << 'EOF' > ~/.elmer_profile
2
3 export ELMER_HOME=/usr/local/share/elmersolver
4 export PATH="$ELMER_HOME/bin:$PATH"
5 export LD_LIBRARY_PATH="$JAVA_HOME/lib:$JAVA_HOME/lib/server:
6     $ELMER_HOME/lib"
7 export CMAKE_MODULE_PATH="$ELMER_HOME/cmake/Modules:
8     $CMAKE_MODULE_PATH"
9 EOF
```

Acontinuacion para asegurar que las variables de entorno que definimos se carguen automáticamente cada vez que abramos una terminal usamos:

```
1 grep -qxF 'source ~/.elmer_profile' ~/.profile \
2 || echo 'source ~/.elmer_profile' >> ~/.profile
3
4 source ~/.elmer_profile
```

9. FLUJO DE TRABAJO

Inicialmente vamos a tener la herramienta ElmerGrid que se encarga de generar mayas las cuales utilizamos junto a un archivo .sif ,que contiene las fisica del problema (proveniente de distintos solvers incorporados en el codigo). Estos dos son utilizados por la herramienta ElmerSolver que se encargan de ajustar, calcular y generar archivos de salida con la informacion fisica lista para ser vista en ParaView

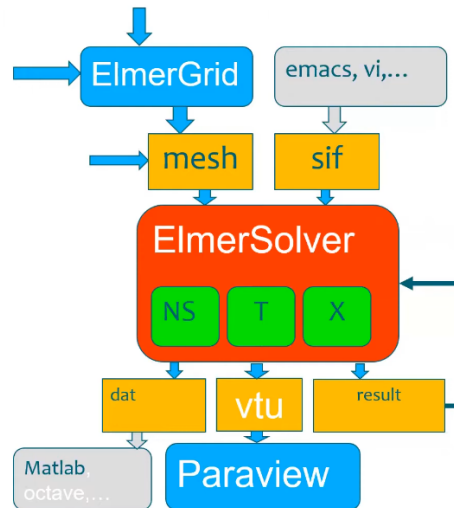


Figura 30: Flujo de trabajo

Los pasos a seguir son:

1. crear una maya en el formato apropiado (necesitamos un .msh) para generar un directorio con archivos tipo: mesh.boundary , mesh.header, mesh.element , mesh.nodes
2. Establecer archivo .sif y utilizar ElmerSolver (genera .vtu)
3. Visualizar con paraview abriendo el archivo .vtu generado anteriormente

Referencias

- [1] Ruokolainen, J., Råback, P., Malinen, M., & Zwinger, T. ElmerFEM [Computer software]. <https://github.com/ElmerCSC/elmerfem>
- [2] CSC – IT Center for Science. *Elmer Documentation Repository* [Online archive]. <https://www.nic.funet.fi/pub/sci/physics/elmer/>