



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona



# Post-procesamiento y visualización de conjuntos de datos geofísicos electromagnéticos

Trabajo Final de Grado

Grado en Ingeniería Informática  
Especialidad en Ingeniería del Software

Autor: Jesús Andrés Velázquez Leccese

Director: Octavio Castillo Reyes

24/06/2021

# Resumen

El subsuelo de la Tierra contiene recursos naturales que son fundamentales para el desarrollo regional. Obtener imágenes precisas de fuentes de agua, minerales y energía en las profundidades de la superficie es un paso clave para su gestión y explotación. Las imágenes geofísicas nos permiten obtener mapas detallados del interior de la Tierra. Esto se logra analizando las deformaciones y los campos electromagnéticos (EM) medidos en la superficie. PETGEM es un modelador de datos electromagnéticos capaz de resolver problemas realistas en el campo de la geofísica de exploración. Sin embargo, la versión actual de PETGEM no cuenta con módulos de postprocesado para la visualización de resultados de simulación. Este último punto marca la génesis y las pautas para el desarrollo de este proyecto, cuyo objetivo final es implementar un módulo que permita y facilite la visualización de resultados de ejecuciones PETGEM. Como principales características del nuevo módulo de visualización destacan la flexibilidad, sencillez e interoperabilidad.

# Resum

El subsòl de la Terra conté recursos naturals que són fonamentals per al desenvolupament regional. Obtener imatges precises de fonts d'aigua, minerals i energia en les profunditats de la superfície és un pas clau per a la seva gestió i explotació. Les imatges geofísiques ens permeten obtenir mapes detallats de l'interior de la Terra. Això s'aconsegueix analitzant les deformacions i els camps electromagnètics (EM) mesurats a la superfície. PETGEM és un modelador de dades electromagnètics capaç de resoldre problemes realistes en el camp de la geofísica d'exploració. No obstant això, la versió actual de PETGEM no compta amb mòduls de postprocessat per a la visualització de resultats de simulació. Aquest últim punt marca la gènesi i les pautes per al desenvolupament d'aquest projecte, l'objectiu final és implementar un mòdul que permeti i faciliti la visualització de resultats d'execucions PETGEM. Com a principals característiques de el nou mòdul de visualització destaquen la flexibilitat, senzillesa i interoperabilitat.

# Abstract

The Earth's subsoil contains natural resources that are essential for regional development. Obtaining accurate images of water, mineral and energy sources deep below the surface is a key step in their management and exploitation. Geophysical images allow us to obtain detailed maps of the interior of the Earth. This is achieved by analyzing the deformations and electromagnetic (EM) fields measured at the surface. PETGEM is an electromagnetic data modeler capable of solving realistic problems in the field of exploration geophysics. However, the current version of PETGEM does not have post-processing modules for displaying simulation results. This last point marks the genesis and guidelines for the development of this project, whose final objective is to implement a module that allows and facilitates the visualization of the results of PETGEM executions. The main characteristics of the new display module include flexibility, simplicity and interoperability.

# Índice general

<b>1. Contextualización .....</b>	<b>7</b>
1.1    Introducción.....	7
1.2    Términos y conceptos.....	8
1.3    Descripción del problema .....	9
1.4    Stakeholders .....	10
<b>2. Justificación.....</b>	<b>11</b>
2.1    Visualización de los resultados.....	11
2.2    Usabilidad .....	13
<b>3. Alcance .....</b>	<b>13</b>
3.1    Objetivos .....	14
3.2    Requisitos no funcionales.....	15
3.3    Riesgos.....	15
<b>4. Metodología de trabajo .....</b>	<b>16</b>
4.1    Sistema de control de versiones .....	17
4.2    Sistema de gestión de proyecto Agile.....	17
<b>5. Planificación inicial .....</b>	<b>18</b>
5.1    Descripción de las tareas .....	18
5.1.1    Gestión del proyecto .....	18
5.1.2    Desarrollo.....	19
5.1.3    Documentación y comunicación .....	20
5.2    Recursos.....	20
5.3    Estimaciones .....	21
5.4    Diagrama de Gantt.....	23
<b>6. Gestión de riesgos.....</b>	<b>24</b>
6.1    Inexperiencia en las tecnologías necesarias.....	24
6.2    Bugs.....	24
6.3    Limitaciones no conocidas en las tecnologías utilizadas .....	25
6.4    Plazos de entrega fijados.....	25
<b>7. Presupuesto .....</b>	<b>26</b>
7.1    Identificación y estimación de los costes .....	26
7.1.1    Recursos humanos .....	26
7.1.2    Generales.....	27
7.1.3    Contingencia.....	28

7.1.4	Imprevistos .....	29
7.1.5	Presupuesto final.....	29
7.2	Control de gestión.....	30
<b>8.</b>	<b>Especificación del sistema PETGEM .....</b>	<b>31</b>
8.1	Flujo de trabajo sin visualización .....	31
8.2	Flujo de trabajo con visualización .....	32
8.3	Postprocesado rápido de nuevos resultados .....	35
<b>9.</b>	<b>Desarrollo.....</b>	<b>36</b>
9.1	Jupyter Notebook .....	36
9.1.1	Jupyter Notebook en PETGEM .....	36
9.1.2	Lectura y modificación de ficheros .....	38
9.1.3	Ejecución de PETGEM en Jupyter Notebook .....	39
9.2	Imagen Docker.....	40
9.2.1	Instalación y ejecución de la imagen Docker .....	40
9.2.2	Ejecución de PETGEM en la imagen Docker .....	42
9.3	Módulo de visualización .....	43
9.3.1	Postprocesado de resultados.....	43
9.3.2	Visualización de resultados.....	44
9.3.3	ParaView .....	46
9.3.4	Postprocesado rápido de nuevos resultados: validación.....	47
<b>10.</b>	<b>Leyes y regulaciones.....</b>	<b>48</b>
10.1	Licencias de software utilizadas .....	48
<b>11.</b>	<b>Planificación final y desviaciones.....</b>	<b>49</b>
11.1	Diagrama de Gantt.....	53
11.1.1	Planificación inicial .....	53
11.1.2	Planificación final .....	54
11.2	Desviaciones en los objetivos y presupuesto del proyecto .....	55
11.3	Metodología de trabajo final .....	55
11.4	Etapas actuales del proyecto .....	55
<b>12.</b>	<b>Sostenibilidad.....</b>	<b>56</b>
12.1	Dimensión económica .....	56
12.2	Dimensión ambiental .....	57
12.3	Dimensión social .....	57
<b>13.</b>	<b>Conclusiones y trabajo futuro .....</b>	<b>58</b>
13.1	Conclusiones.....	58

13.2 Trabajo futuro.....	59
<b>14. Referencias .....</b>	<b>60</b>

## Índice de tablas

Tabla 1: Recursos utilizados en el proyecto. Fuente: Elaboración propia.....	21
Tabla 2: Resumen de las tareas del TFG. Fuente: Elaboración propia.....	22
Tabla 3: impacto, probabilidad y horas de desviación por cada riesgo. Fuente: Elaboración propia. .....	24
Tabla 4: Sueldos estimados por rol y hora. Fuente: Elaboración propia, con los datos de (glassdoor, 2021).....	26
Tabla 5: Costes de personal por actividad. Fuente: Elaboración propia.....	27
Tabla 6: Costes estimados para los gastos generales. Fuente: Elaboración propia.....	28
Tabla 7: Costes de contingencia. Fuente: Elaboración propia.....	28
Tabla 8: Costes presupuestados para los imprevistos. Fuente: Elaboración propia.....	29
Tabla 9: Costes estimados del proyecto. Fuente: Elaboración propia.....	29
Tabla 10: Recursos utilizados en el proyecto. Fuente: Elaboración propia.....	50
Tabla 11: Resumen de las tareas del TFG. Fuente: Elaboración propia.....	52

# Índice de figuras

Figura 1: Comparativo de campo eléctrico total para el componente x entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021) .....	9
Figura 2: Visualización del resultado de PETGEM con un modelo 3D. Fuente: (PETGEM website, 2021).....	10
Figura 3: Tablero Kanban del proyecto PETGEM. Fuente: Elaboración propia. ....	17
Figura 4: Diagrama de Gantt de la planificación inicial. Fuente: Elaboración propia. ....	23
Figura 5: Flujo de trabajo sin visualización de PETGEM. Fuente: Elaboración propia. ....	31
Figura 6: Comparativo de campo eléctrico total para el componente x entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021).....	32
Figura 7: Flujo de trabajo con visualización de PETGEM. Fuente: Elaboración propia. ....	33
Figura 8: Visualización del resultado de PETGEM con ParaView. Fuente: Elaboración propia. ....	34
Figura 9: Visualización de la amplitud de campo eléctrico para una lista de puntos. Fuente: Elaboración propia.....	34
Figura 10: Flujo de trabajo del postprocesado rápido de nuevos resultados. Fuente: Elaboración propia.....	35
Figura 11: Interfaz gráfica de Jupyter Notebook con Script de PETGEM. Fuente: Elaboración propia. ....	37
Figura 12: Celda con parámetros de PETGEM. Fuente: Elaboración propia.....	37
Figura 13: Lectura de archivos en Jupyter Notebook. Fuente: Elaboración propia.....	38
Figura 14: Modificación de fichero en Jupyter Notebook. Fuente: Elaboración propia.....	38
Figura 15: Guardar un fichero en Jupyter Notebook. Fuente: Elaboración propia. ....	39
Figura 16: Ejecución de PETGEM en Jupyter Notebook. Fuente: Elaboración propia.....	39
Figura 17: Directorio de trabajo en el ordenador de la imagen Docker. Fuente: Elaboración propia. ....	40
Figura 18: Dockerfile de la imagen de PETGEM. Fuente: Elaboración propia. ....	41
Figura 19: Ejecución de PETGEM en la imagen Docker. Fuente: Elaboración propia. ....	42
Figura 20: Ejecución de PETGEM a través de Jupyter Notebook dentro de la imagen Docker. Fuente: Elaboración propia.....	43
Figura 21: Pseudocódigo del módulo de postprocesado de PETGEM. Fuente: Elaboración propia. ....	44
Figura 22: Comparativo de campo eléctrico total para el componente x entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021).....	45
Figura 23: Visualización de la amplitud de campo eléctrico para una lista de puntos. Fuente: Elaboración propia.....	45
Figura 24: Visualización del resultado de PETGEM con ParaView (Campo magnético). Fuente: Elaboración propia.....	46
Figura 25: Visualización del resultado de PETGEM con ParaView (Campo electromagnético). Fuente: Elaboración propia.....	47
Figura 26: Diagrama de Gantt de la planificación inicial. Fuente: Elaboración propia. ....	53
Figura 27: Diagrama de Gantt de la planificación final. Fuente: Elaboración propia. ....	54

# 1. Contextualización

El proyecto "Post-procesamiento y visualización de conjuntos de datos geofísicos electromagnéticos" se trata de mi trabajo final de grado (TFG), para obtener el grado de Ingeniería Informática con mención en la Especialidad de Ingeniería de Software, de la Facultad de Informática de Barcelona (FIB), de la Universidad Politécnica de Cataluña. Este proyecto tendrá como director al doctor Octavio Castillo Reyes, profesor del departamento de Arquitectura de Computadores de la FIB, y tendrá un periodo de duración de aproximadamente 5 meses, en el periodo comprendido de Enero a Junio de 2021.

## 1.1 Introducción

El proyecto "Post-procesamiento y visualización de conjuntos de datos geofísicos electromagnéticos", además de desarrollarse en el marco de la Facultad de Informática de Barcelona (FIB), también se desarrollará conjuntamente bajo el contexto del proyecto PIXIL, que está coordinado por el grupo de aplicaciones de geociencias del Barcelona Supercomputing Center (BSC) (Interreg POCTEFA, 2021).

PIXIL (Interreg POCTEFA) (PIXIL, 2021) es un proyecto de colaboración científico-tecnológica transfronterizo y multidisciplinario, que tiene como objetivo desarrollar las herramientas más punteras para analizar el subsuelo terrestre, con especial interés en favorecer el crecimiento de la energía geotérmica en la región. Este proyecto tiene un perfil predominante de investigación, cuyos resultados son avances en las tres disciplinas que conforman la imagen geofísico-moderna: Geofísica, Matemática Aplicada y Computación.

El objetivo principal de este TFG es el desarrollo de un módulo para la visualización de resultados dentro del programa de PETGEM (Parallel Edge-based Tool for Geophysical Electromagnetic Modelling, (Castillo Reyes, PETGEM, 2018)). PETGEM es un código HPC (high performance computing) para la simulación de campos electromagnéticos de fuente controlada en tres dimensiones (3D CSEM) que surgen en el contexto de la geofísica de exploración. El código está escrito principalmente en Python 3 y se basa en la pila de software científico Python.

PETGEM se desarrolla como código abierto bajo una licencia open-source (BSD-v3) en el Departamento de Aplicaciones en Ciencia e Ingeniería (CASE) del Barcelona Supercomputing Center (BSC).

La versión actual de PETGEM no realiza tareas de visualización por sí mismo, sino que una vez que se ha obtenido la solución de un modelo 3D CSEM, ésta debe procesarse de forma no automática y posterior mediante el uso de un programa de visualización externo.

Por todo lo anterior, este documento de TFG describe la metodología y resultados para la implementación de un módulo que permita y facilite la visualización de resultados de ejecuciones PETGEM. Como principales características del nuevo módulo de visualización destacan la flexibilidad, sencillez e interoperabilidad. A continuación, se describen y detallan todas las tareas y resultados derivados de este TFG.

## 1.2 Términos y conceptos

- **High performance computing**

La computación de alto rendimiento (Wikipedia, 2021) (*high-performance computing* o HPC en inglés) implica usar la potencia de cálculo para resolver problemas complejos en ciencia, ingeniería y gestión. Para lograr este objetivo, la computación de alto rendimiento se apoya en tecnologías computacionales como los *clusters*, los supercomputadores o la computación paralela.

- **3D CSEM**

3D CSEM o método electromagnético de fuente marina o terrestre controlada (Castillo Reyes, UPCommons, 2021) se ha convertido en una técnica fundamental para reducir las ambigüedades en la interpretación de conjuntos de datos geofísicos mediante el mapeo de las variaciones de conductividad en el subsuelo.

- **Método Electromagnético (EM)**

Método que se basa en el estudio del subsuelo a través de los cambios en las propiedades eléctricas y magnéticas que poseen los materiales que lo componen (Geología web, 2021). A través de este método se puede estudiar con mucho detalle los primeros metros del subsuelo, o se pueden alcanzar varios kilómetros de profundidad, pero con menor resolución en los datos obtenidos.

- **Gmsh**

Es un generador de malla de elementos finitos 3D de código abierto con un motor CAD y un postprocesador integrados (Gmsh, 2021). Su objetivo de diseño es proporcionar una herramienta de mallado rápida, liviana y fácil de usar con entrada paramétrica y capacidades de visualización avanzadas.

- **VTK**

Kit de herramientas de visualización o VTK (MEDTEC, 2021) es una librería multiplataforma que se distribuye mediante licencia *Open Source* para la manipulación y la visualización de datos científicos. VTK es utilizado en distintos campos científicos para la investigación y el desarrollo, análisis de imágenes médicas, procesamiento de imágenes, renderizado, etc.

- **HDF5**

El formato de datos jerárquico (Wikipedia, 2021) (HDF) es un conjunto de formatos de archivo, diseñado para almacenar y organizar grandes cantidades de datos. Las bibliotecas HDF y las herramientas asociadas están disponibles bajo una licencia liberal similar a BSD para uso general.



### 1.3 Descripción del problema

Este TFG surge con la idea de proveer de mejores capacidades de postprocesado y visualización de resultados de simulaciones PETGEM.

PETGEM no realiza la visualización por sí mismo, sino que genera un archivo de salida (en el formato hdf5) con las respuestas electromagnéticas en las posiciones de los receptores, y también proporciona valores de tiempo para evaluar el rendimiento. Con lo que, una vez obtenida la solución, ésta debe procesarse posteriormente mediante el uso de un programa externo de visualización. La Figura 1 muestra un postprocesado de datos típico que consiste en la generación de gráficas de amplitud de campo eléctrico en puntos discretos del dominio.

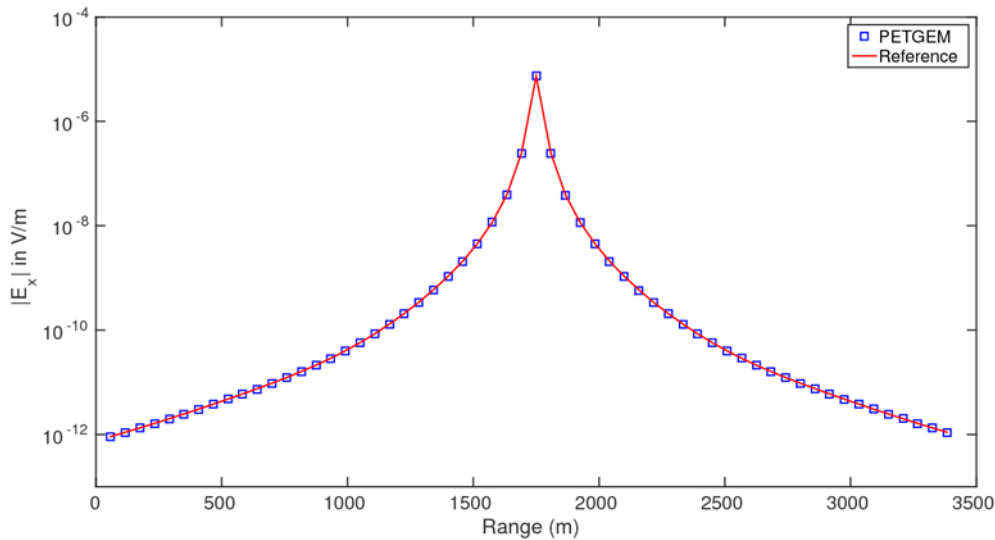


Figura 1: Comparativo de campo eléctrico total para el componente  $x$  entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021)

Respecto a la visualización del archivo de salida en el formato hdf5, éste sólo se puede visualizar en un formato 2D. Por esta razón, el módulo de visualización a desarrollarse en este trabajo pretende solucionar esta limitante y ofrecer la opción de visualizar el resultado en un formato 3D, y además ofrecer más información relevante en la visualización de la simulación, como se puede observar en la Figura 2.

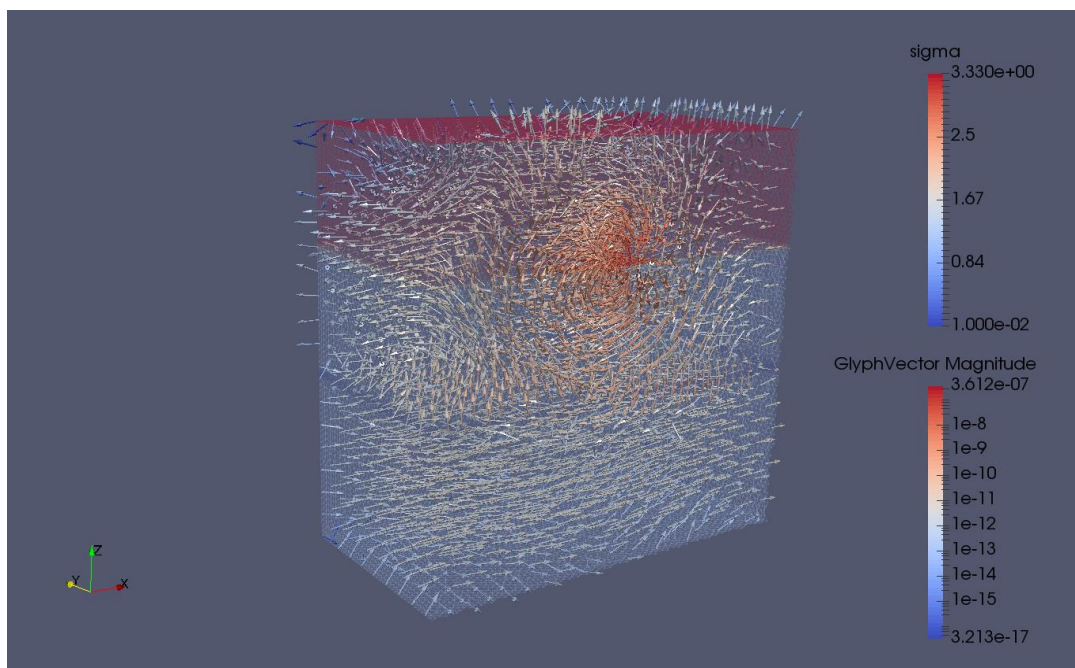


Figura 2: Visualización del resultado de PETGEM con un modelo 3D. Fuente: (PETGEM website, 2021)

Otra limitante que se tiene es a la hora de calcular las respuestas electromagnéticas en nuevos puntos discretos del mismo modelo, con lo que el usuario deberá de establecer estos nuevos puntos de interés y realizar la ejecución de la simulación por completo otra vez. Por lo cual con la implementación del nuevo módulo en PETGEM, se minimizará el tiempo de espera en este proceso, ya que, con el nuevo módulo, no será necesario lanzar una nueva simulación puesto que toda la información requerida para calcular las respuestas electromagnéticas en nuevos puntos estará almacenada en un fichero de salida. El acceso a dicho fichero y la gestión de su contenido se realizará a través del módulo de visualización desarrollado en este TFG.

## 1.4 Stakeholders

Los Stakeholders son las partes o actores que están interesados en el proyecto, por lo tanto, son las partes que se beneficiarán, serán afectadas o serán partícipes del desarrollo del proyecto. Estas partes son de gran importancia en el momento de la planificación del proyecto, ya que son las que definirán tanto los requisitos funcionales como los no funcionales que tienen que cumplir el sistema. A continuación, se listan los Stakeholders de este proyecto.

- **Tutor del TFG:** El tutor de este proyecto, Octavio Castillo Reyes, es uno de los actores con mayor interés en este proyecto. Está implicado en la dirección y asesoramiento del proyecto, con el objetivo de la finalización correcta de este.
- **Autor del software (PETGEM):** El autor del software, Octavio Castillo Reyes, es uno de los actores que se verá beneficiado por la mejora en el software PETGEM.
- **Usuarios de PETGEM:** son una de las partes más importantes de este proyecto, y son los principales que se beneficiarán de las mejoras que se van a introducir en el software,

concretamente en la visualización de los resultados. Los principales usuarios que tiene PETGEM son los científicos que estudian el ámbito de la geofísica.

- **Desarrollador:** El proyecto consta de un único desarrollador, en este caso yo, Jesús Velázquez, seré el encargado de desarrollar un módulo para la visualización de los resultados dentro del kernel de PETGEM.
- **Proyecto PIXIL (Interreg POCTEFA):** Este proyecto es una de las partes interesadas en el desarrollo de un módulo de visualización dentro del kernel de PETGEM, porque este proyecto es el encargado de financiar el desarrollo del software PETGEM.

## 2. Justificación

Como se ha mencionado anteriormente PETGEM, es una herramienta para la simulación de campos electromagnéticos en modelos 3D CSEM reales, que surgen en el contexto de la geofísica. Una de las características diferenciadoras entre otros modeladores electromagnéticos y PETGEM consiste en que este último se desarrolla bajo un entorno *open source* o de código abierto, lo que proporciona una gran ventaja frente al resto de software de pago. Sin embargo, PETGEM no cuenta con una herramienta interna de visualización para sus resultados. Es por esto por lo que este proyecto plantea mejorar esta faceta del software.

### 2.1 Visualización de los resultados

Lo primero que nos planteamos a la hora de mejorar esta fase, es si PETGEM debería diseñar su propia herramienta de visualización interna o si por el contrario debería interactuar con una herramienta de visualización ya existente. Nuestra decisión fue la de interactuar con una herramienta ya existente, ya que diseñar una herramienta de visualización integrada en PETGEM sería un proyecto demasiado ambicioso para un TFG, y además en el mercado ya existen buenas herramientas de visualización, que solucionan nuestro problema.

Después de haber decidido, que usaremos una herramienta externa para poder visualizar nuestro resultado, tenemos que decidir qué tipo de herramienta de visualización es la más conveniente para el proyecto. Para ello tenemos que escoger un software que cuente con el soporte de ficheros en formato VTK (que es el formato de salida que tendrán los ficheros de soluciones de PETGEM), y además este software ha de ser *open source* (de código abierto).

Una vez establecido el tipo de herramientas con el que queremos interactuar, ahora tenemos que ver en el mercado, cuáles son las herramientas disponibles que más nos favorezcan. Las herramientas que hay actualmente y las que consideramos que son las adecuadas para este proyecto son: OpenGL, PyVista y ParaView.

- **OpenGL**

(Open Graphics Library o OpenGL) es una especificación estándar que define una API multilinguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La API contiene más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos.

Si PETGEM interactuará con OpenGL, lo tendría que hacer de la siguiente forma, PETGEM generará un script de salida en el lenguaje Python (usando la API de OpenGL) y posteriormente el usuario ejecuta este script, que mostrará el resultado de la simulación del modelo 3D CSEM.

Si el usuario quisiera modificar los datos obtenidos o modificar cualquier cosa de la visualización, tendría que modificar el script de Python y añadir las funcionalidades necesarias para poder personalizar la visualización. Con lo que el usuario deberá tener conocimientos de Python y de la API de OpenGL.

- **PyVista**

Es un módulo auxiliar para *Visualization Toolkit* (VTK) que adopta un enfoque diferente para interactuar con VTK. Este módulo proporciona una documentación que expone el poderoso *backend* de visualización de VTK, para facilitar la creación rápida de prototipos, el análisis y la integración visual de conjuntos de datos con referencias espaciales.

Si PETGEM interactuará con PyVista, lo tendría que hacer de la siguiente forma, PETGEM generará un script de salida en el lenguaje Python (usando la API de PyVista) y posteriormente el usuario ejecuta este script, que mostrará el resultado de la simulación del modelo 3D CSEM.

Si el usuario quisiera modificar los datos obtenidos o modificar cualquier cosa de la visualización, como por ejemplo colores o modo de visualizar (como por ejemplo modelo 2D o 3D, o solo visualizar los puntos del modelo), tendría que modificar el script de Python y añadir las funcionalidades necesarias para poder personalizar la visualización. Con lo que el usuario deberá tener conocimientos de Python y de la API de PyVista.

- **ParaView**

Si PETGEM interactuará con ParaView, lo tendría que hacer de la siguiente forma, PETGEM generará un fichero de salida en el formato VTK y posteriormente el usuario podrá ver el resultado de la simulación del modelo 3D CSEM en ParaView.

Si el usuario quisiera modificar los datos obtenidos o modificar cualquier cosa de la visualización, como por ejemplo colores o modo de visualizar (como por ejemplo modelo 2D o 3D, o solo visualizar los puntos del modelo), lo puede hacer directamente desde la interfaz de la aplicación. Con lo que el usuario no necesitará el conocimiento de programación.

Por lo tanto, hemos decidido optar por ParaView, porque es una aplicación, que permite a los usuarios crear visualizaciones rápidamente para analizar sus datos, utilizando técnicas cualitativas y cuantitativas. Además, la exploración de datos se puede realizar de forma interactiva en 3D mediante su interfaz gráfica, o mediante la programación, utilizando la API de ParaView que es compatible con Python (lenguaje en el que está escrito PETGEM).

Todas las herramientas mencionadas cuentan con una API que es compatible con Python, para la exploración de los datos. Pero algo con lo que cuenta ParaView y que no poseen las herramientas PyVista, OpenGL, es una interfaz gráfica para la modificación de la visualización y de los datos utilizados en el resultado de la simulación. Con lo cual, estas 2 herramientas añaden una complejidad para los usuarios que quieran interactuar con la visualización del modelo 3D.

Es por estos motivos que hemos decidido que el módulo de visualización de PETGEM interactúe con ParaView. Aunque se recomienda el uso de la herramienta ParaView con PETGEM, existe absoluta libertad para utilizar otras herramientas de visualización que soporten ficheros en formato VTK.

## 2.2 Usabilidad

Una vez que se ha propuesto una solución al problema de la visualización de los resultados, se discutirá la usabilidad del código PETGEM. Con la implementación del nuevo módulo de visualización se mejorará el grado de usabilidad, sin embargo, para mejorarlo en mayor grado se ha propuesto el desarrollo e implementación de una imagen Docker junto con Jupyter Notebook. Este enfoque es robusto, confiable y flexible.

El motivo por el cual se decidió implementar una imagen de Docker es que nos permite tener una mayor portabilidad y facilidad de uso del código PETGEM. Esto se debe a que la imagen de Docker contiene todas las dependencias necesarias para poder ejecutar PETGEM, lo que simplifica el proceso de instalación y ejecución del código. Dado que las imágenes de Docker pueden ejecutarse en cualquier sistema operativo (o entorno de trabajo), se reduce la posibilidad de errores o incompatibilidad con la herramienta. Con esta implementación los usuarios contarán con 2 opciones para ejecutar PETGEM en sus ordenadores (la instalación propia de PETGEM o el uso de la imagen Docker).

Además, se añade una capa más de facilidad para los usuarios, esta capa es la de Jupyter Notebook en la que se tiene un script como documento notebook de Jupyter, en el cual el usuario puede modificar todos los campos necesarios para ejecutar la simulación de un modelo en PETGEM, directamente desde la interfaz del notebook de Jupyter (en la que podrá modificar los puntos seleccionados para la simulación del modelo).

## 3. Alcance

En este apartado hablaremos sobre los principales objetivos que se quieren conseguir con este proyecto y de los principales requisitos que se deben de cumplir para conseguirlos. Posteriormente se hablará de los posibles riesgos u obstáculos, que podrían presentarse durante el desarrollo de este proyecto.

El proyecto completo consta de desarrollar un módulo para la visualización de resultados dentro del kernel de PETGEM, ya que PETGEM no realiza la visualización por sí mismo como hemos mencionado anteriormente. Este módulo se encargará de generar los ficheros de salida necesarios

en el formato VTK, para luego posteriormente poder visualizar la solución de la simulación tanto en 3D como en 2D, en la herramienta de visualización ParaView.

Además, una vez obtenida la solución, los usuarios contarán con la funcionalidad de poder cambiar los puntos que habían establecido, en los que la simulación calcula los campos electromagnéticos en cada uno de estos puntos. Esto lo podrán hacer sin la necesidad de volver a ejecutar toda la simulación, sino que solo se calculará los campos electromagnéticos en cada uno de estos nuevos puntos.

PETGEM no dependerá directamente de ParaView para la visualización de sus resultados, sino que recomendará el uso de esta herramienta. Esto se debe a que los ficheros de salida con el resultado de la simulación se podrán visualizar en cualquier herramienta de visualización de datos multiplataforma que implemente el formato VTK.

### 3.1 Objetivos

Como hemos mencionado anteriormente el principal objetivo de este proyecto es el de generar un módulo para la visualización de resultados dentro del kernel de PETGEM. Los objetivos específicos que definimos a continuación pretenden contribuir al alcance del objetivo general. Cabe mencionar que para cada uno de estos objetivos específicos se tiene que cumplir ciertos requisitos funcionales

- El usuario ha de poder ver el resultado de la simulación en una visualización 2D
  - El formato del fichero de salida tiene que ser el formato VTK.
  - El usuario ha de poder ver la visualización de la simulación en ParaView en 2D (ParaView es la herramienta recomendada de visualización).
  - El fichero de salida ha de ser compatible con otras herramientas de visualización (no obliga al usuario a usar la herramienta de ParaView).
  - La visualización tiene que contener la comparativa de los campos eléctricos en cada uno de los puntos seleccionados por el usuario.
- Tener un único módulo en el kernel de PETGEM que sea el encargado de generar todos los ficheros de salida para la posterior visualización
  - El módulo ha de ser capaz de escribir los datos de salida en paralelo
  - El módulo generar un fichero con la visualización en 3D
  - El módulo generar un fichero con la visualización en 2D
- El usuario pueda ver el resultado de la simulación del modelo 3D CSEM, en cualquier herramienta de visualización
  - Los ficheros de salida tienen que estar en un formato genérico para ser usados en más de una herramienta de visualización
- Funcionalidad que le permita al usuario volver a ejecutar la simulación del modelo 3D CSEM, en otros puntos de preferencias, sin la necesidad de volver a ejecutar toda la simulación
  - Implementar un sistema sencillo en el que el usuario pueda cambiar los puntos en los que quiere generar la simulación.
  - Al cambiar los puntos no se tiene que ejecutar toda la simulación otra vez para obtener los resultados en esos puntos.

## 3.2 Requisitos no funcionales

Una vez establecidos los objetivos del proyecto y los requisitos necesarios para que se cumplan estos objetivos, a continuación, estableceremos los requisitos no funcionales o de calidad, que se tiene que cumplir para el correcto funcionamiento del software.

- **Usabilidad:** El software tiene que ser lo más fácil posible de utilizar para todos los tipos de usuarios. Se espera que el usuario no tenga inconvenientes en la visualización de los resultados del software.
- **Velocidad y latencia:** El software debe tener una velocidad de ejecución rápida en función de la complejidad del modelo 3D CSEM. Se espera que el nuevo módulo que se encargará de la visualización no añada demasiada latencia a la obtención de los resultados del modelo 3D CSEM.
- **Capacidad y escalabilidad:** El nuevo módulo de visualización tiene que ser capaz de gestionar todos los tipos de complejidad en los módulos 3D CSEM que el usuario quiera simular.

## 3.3 Riesgos

Durante el periodo de desarrollo de un proyecto, es importante tener en cuenta que existen diferentes tipos de riesgos que pueden afectar a la planificación inicial del proyecto y a la calidad de este, por lo que es importante conocer de antemano los posibles riesgos que se pueden tener, y contar con un plan de actuación para minimizar lo máximo posible el impacto de estos. A continuación, se mencionan los posibles riesgos que tiene este proyecto.

- **Inexperiencia en las tecnologías necesarias:** En el momento de iniciar el proyecto, el desarrollador no cuenta con el suficiente conocimiento en las tecnologías que se usarán para desarrollar el software. En el caso de que el tiempo de aprendizaje sea mayor al esperado, esto afectará directamente a la planificación del proyecto, con lo que se tendrá que reducir el tiempo de otras tareas planificadas.
- **Bugs:** Durante el desarrollo de un software siempre existe la posibilidad de que se cometan errores en el código (bugs). Estos errores en el código o bugs pueden ser de varios tipos, unos que son más fáciles de detectar y solucionar que otros.

Los que suelen ser más fáciles de detectar y solucionar, son los que nos encontramos cuando estamos desarrollando una funcionalidad o cuando nos percatamos del error antes de tener el código en producción (disponible para los usuarios). Por el contrario, los que suelen ser más difíciles de detectar y solucionar, son los que nos encontramos al final del proyecto o cuando un usuario está usando el software, ya que estos requieren de grandes cambios para solucionarlos.

Es por esto, que se debe tener muy en cuenta la detección y solución de estos posibles errores en la planificación inicial del proyecto.

- **Limitaciones no conocidas en las tecnologías utilizadas:** Es posible que, durante el desarrollo del software, se descubran limitaciones o incompatibilidades en las herramientas o tecnologías que se usan o interactúan con el software, y como

consecuencia esto afectará a la planificación del proyecto, porque se tendrá que asignar tiempo de otras tareas, a buscar e implementar una solución a este problema.

- **Plazos de entrega fijados:** Como ya sabemos, este proyecto es un trabajo final de grado, por lo cual la fecha de entrega ya está establecida por la universidad. Estos se tienen en cuenta en la planificación inicial del proyecto, pero es posible que puedan surgir incidencias, bugs o bloqueos, y se puedan ver comprometidas ciertas funcionalidades que formaban parte de la entrega inicial, esto ocasionará una pérdida de calidad del proyecto final, con el fin de garantizar la entrega en el plazo establecido.

## 4. Metodología de trabajo

La elección de la metodología de trabajo es una de las decisiones más condicionantes de los proyectos, ya que caracterizará la forma en la que se trabajará, siendo esta decisiva a la hora de alcanzar los objetivos.

Los dos principales tipos de metodología que se utilizan actualmente son Waterfall y Agile. Para este proyecto se ha escogido la metodología Agile, porque es la más adecuada, esto se debe a que, con esta metodología de trabajo la planificación del proyecto se va a ir adaptando a las diferentes situaciones y circunstancias que vayan surgiendo en el desarrollo de este.

En concreto la metodología Agile que se ha escogido es Kanban. En el momento de tomar esta decisión se han tenido en cuenta otras metodologías Agile, como pueden ser algunas de las que se han estudiado en la carrera, una de las más relevantes es Scrum. Scrum es una metodología en la que es muy importante las funciones que realizan los distintos roles del equipo y en la que se centra el desarrollo en ciclos temporalmente cortos y de una duración fija (normalmente de 2 semanas).

Debido a que este proyecto solo será desarrollado por una persona, se ha optado por usar Kanban en vez de Scrum, porque Kanban es una metodología que se basa en el desarrollo continuo, trabajando en un pequeño número de tareas de forma fluida y simultánea, y además en esta metodología no se les da mucha importancia a los distintos roles del equipo.

Para seguir la metodología Kanban, se generará un tablero Kanban que estará dividido en cuatro columnas (To-do, in progress, test y Done), las cuales representarán el estado actual de cada historia de usuario, y además cada vez que se finalice una historia de usuario se comprobará la planificación actual del proyecto, con el fin de tomar las medidas necesarias para cumplir el plazo de entrega establecidos.

Para facilitar el desarrollo de este proyecto siguiendo esta metodología, se usarán las siguientes herramientas.



## 4.1 Sistema de control de versiones

- **GitHub:** es un sistema de control de versiones gratuita y de código abierto. Es muy utilizado en el desarrollo de software, en proyectos cooperativos para llevar a cabo toda la gestión de versiones del código, mediante repositorios y las ramas que hay en estos.

El flujo de trabajo que seguiremos con esta herramienta será la siguiente, se tendrán dos ramas principales, la *master* y la *development*. La rama *master* será en la que se tendrá el código ya testeado y listo para producción, y la rama *development* será la rama en la que se irán añadiendo las funcionalidades terminadas. Para cada historia de usuario se creará una nueva rama desde *development* y tendrán como nombre el *ID* de la historia de usuario, en estas se desarrollarán el código correspondiente a cada historia. Una vez finalizada y testeada la historia, se añadirá a la rama *development*.

Una vez que se tengan unas cuantas historias de usuarios finalizadas, se añadirá la rama de *development* a la rama de *master*, esto se realizará cíclicamente hasta tener todas las historias de usuario finalizadas. Esto se hace para mantener la rama de *master* actualizada.

## 4.2 Sistema de gestión de proyecto Agile

- **Taiga:** es un sistema de gestión de proyectos gratuita y de código abierto. Tiene funcionalidades para crear y puntuar las historias de usuario, para gestionar el *backlog*, y para hacer el seguimiento de las tareas. Se usará esta herramienta para llevar un control de las historias de usuario que se creen, y así poderlas visualizar en un tablero de forma que en todo momento podamos ver en el estado que están, la planificación global y la planificación de cada *sprint*.

El flujo de trabajo que se realizará en esta herramienta será el siguiente, como hemos mencionado anteriormente se creará un tablero con cuatro columnas (*To-do*, *in progress*, *test* y *Done*) y las historias de usuario se irán moviendo en estas a medida que vayan variando su estado. Cada historia de usuario tendrá un *ID*, una descripción y un conjunto de tareas que se realizarán en esta historia para poder finalizarla.

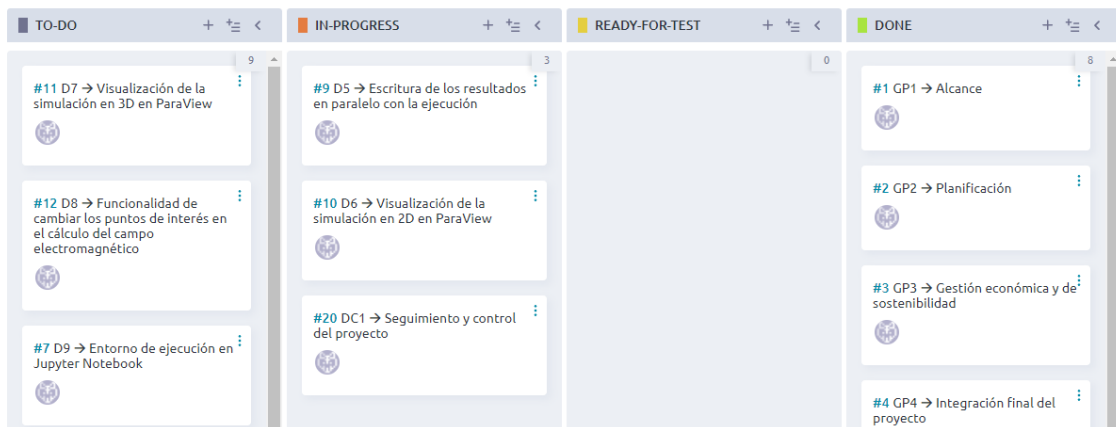


Figura 3: Tablero Kanban del proyecto PETGEM. Fuente: Elaboración propia.

# 5. Planificación inicial

La duración de este proyecto es de 5 meses aproximadamente, concretamente el inicio de este es el día 25 de enero con el comienzo de la gestión del proyecto, y finaliza con la lectura de la entrega final en el mes de junio (aproximadamente el día 28). La carga media de trabajo es de unas 25 horas semanales, que pueden variar dependiendo de la tarea que se esté realizando. Por lo tanto, el proyecto consta de unas 540 horas aproximadas de trabajo.

Este proyecto estará dividido en 3 principales grupos de tareas, las cuales son, gestión del proyecto, desarrollo, y documentación y comunicación. Desde el 23 de febrero al 22 de marzo se realizará el grupo de tareas de gestión del proyecto, en el cual la carga de trabajo consta de unas 65 horas.

El grupo de tareas de desarrollo empezará el día 25 de enero, implementando la metodología de trabajo Kanban, y acabará aproximadamente el 27 de mayo. La carga de trabajo de este grupo de tareas será de unas 340 horas.

Y por último el grupo de tareas de documentación y comunicación, empezará el mismo día que empieza el grupo de desarrollo (25 de enero) y finalizará el 28 de junio aproximadamente. Este periodo de tiempo se debe a que este grupo está conformado por tareas que se realizan constantemente a lo largo del desarrollo del software. La carga de trabajo en este grupo será de 135 horas.

## 5.1 Descripción de las tareas

Como ya hemos comentado, las tareas que se realizarán en este proyecto están divididas en 3 grupos: gestión del proyecto, desarrollo, y documentación y comunicación. Las tareas que están en el grupo de desarrollo son las que están relacionadas con la parte técnica del proyecto.

En cada tarea se detalla una breve descripción de qué es lo que se hará, una estimación del tiempo que tomará realizarlas y las dependencias que tiene respecto a las demás tareas.

### 5.1.1 Gestión del proyecto

- **GP1 → Alcance (25 h):** Realizar un documento donde se especifica la descripción del proyecto, su justificación, el alcance y explicación de la metodología de trabajo que se va a seguir. Dependencias: Ninguna.
- **GP2 → Planificación (10 h):** Redactar un documento en el que se encuentra la propuesta de la planificación inicial, su correspondiente estimación y diagrama de Gantt y la gestión de los posibles riesgos. Dependencias: GP1.
- **GP3 → Gestión económica y de sostenibilidad (10 h):** Elaborar un documento con la estimación del presupuesto de costes y gestión y el informe de sostenibilidad. Dependencias: GP2.
- **GP4 → Integración final del proyecto (20 h):** Redactar un documento final con la integración de toda la información de los tres documentos anteriores. Dependencias: GP3.

## 5.1.2 Desarrollo

- **D1 → Especificación de los requisitos (20 h):** Se identificará y especificarán los requisitos y funcionalidades que tendrá el software. Para eso se estudiará en profundidad las tecnologías que se usarán y las posibles tecnologías que se podrían usar que hay en el mercado. Dependencias: Ninguna.
- **D2 → Preparación del entorno (10 h):** Se descargará todos los programas y librerías necesarias para la gestión, diseño y programación y se creará la estructura del proyecto. Dependencias: D1.
- **D3 → Familiarizarse con VTK y ParaView (40 h):** Antes de empezar a programar, se familiarizará con las librerías de VTK y de ParaView, y se hará una prueba, para comprobar el correcto funcionamiento de la estructura del proyecto. Dependencias: D2.
- **D4 → Definir la estructura del modelo en función de los datos de entrada (30 h):** Se establecerá la estructura que tendrán los ficheros de salida en formato VTK, dependiendo de la información que el usuario introduzca antes de ejecutar la simulación. Por ejemplo, la solución obtenida dependerá de la información que contenga el modelo 3D CSEM o si el usuario establece que solo quiere el resultado de la simulación en 2D o 3D. Dependencias: D1, D2, D3.
- **D5 → Escritura de los resultados en paralelo con la ejecución (40 h):** Se programará el módulo de visualización dentro del *kernel* de PETGEM, para que la escritura de los ficheros de salida, con los resultados de la simulación se ejecute en paralelo junto a la ejecución de la simulación. Dependencias: D4.
- **D6 → Visualización de la simulación en 2D en ParaView (30 h):** Se definirá la información del resultado que se visualizará en el software de ParaView en el formato de 2D. Dependencias: D4.
- **D7 → Visualización de la simulación en 3D en ParaView (50 h):** Se definirá la información del resultado que se visualizará en el software de ParaView en el formato de 3D. Dependencias: D4.
- **D8 → Funcionalidad de cambiar los puntos de interés en el cálculo del campo electromagnético (70 h):** Se programará una nueva funcionalidad en el software, la cual le permite al usuario volver a calcular el campo electromagnético en los nuevos puntos de interés que establezca. Esto se hará sin la necesidad de volver a ejecutar por completo la simulación del modelo 3D CSEM para obtener la solución. Dependencias: D5, D6, D7.
- **D9 → Entorno de ejecución en Jupyter Notebook (20 h):** Se creará un entorno de ejecución del módulo de postprocesado de PETGEM para los usuarios. Este entorno estará implementado en la herramienta de Jupyter Notebook. Dependencias: D8.
- **D10 → Realización de los test para ParaView (35 h):** Se realizarán *test* manuales y unitarios, para comprobar el correcto funcionamiento de todas las funcionalidades del software y de la correcta visualización de los resultados en el software ParaView. Dependencias: D8.
- **D11 → Realización de los test de compatibilidad con otra herramienta (15 h):** Se realizarán *test* manuales y unitarios, para comprobar que se visualiza de manera correcta los resultados obtenidos de la simulación en el resto de software de visualización que se han considerado, estos softwares son PyVista y OpenGL. Dependencias: D5, D7.

### 5.1.3 Documentación y comunicación

- **DC1 → Seguimiento y control del proyecto (- h):** En esta tarea, se irán apuntando todas las tareas realizadas y los inconvenientes que surjan, para ir actualizando la planificación del proyecto. Como esto se hará constantemente no se le asignará un tiempo de duración. Dependencias: Ninguna.
- **DC2 → Documentación de la fase de seguimiento (30 h):** Se dedicará este tiempo a preparar la entrega y hacer una reunión con el tutor del proyecto, Octavio Castillo, para que se tenga conocimiento del progreso del proyecto. Dependencias: GP4.
- **DC3 → Comunicación (15 h):** Realizar reuniones frecuentemente con el tutor del proyecto, Octavio Castillo, para comentar la evolución, las dudas y los problemas que surjan en el desarrollo del proyecto. Dependencias: Ninguna.
- **DC4 → Documentación de la fase final (60 h):** Se dedicará este tiempo a terminar la redacción del documento final. Dependencias: CD2.
- **DC5 → Preparación de la presentación final (30 h):** Preparación de la presentación final del proyecto. Dependencias: CD4.

## 5.2 Recursos

A continuación, se mencionan los recursos necesarios para el desarrollo del proyecto.

Código	Nombre	Descripción	Tareas
	RECURSOS HUMANOS		
OC	Octavio Castillo	Director del proyecto	DC2, DC3
JV	Jesús Velázquez	Desarrollador del proyecto	Todas
	SOFTWARE Y SERVICIOS		
GH	GitHub	Herramienta de gestión de versiones y repositorios.	D
O365	Office 365	Herramienta de Microsoft en la que se usa: Word, Excel, PowerPoint.	GP, DC
DR	Drive	Plataforma para almacenar archivos en la nube y compartirlos	GP, DC1, DC2, DC4, DC5
ME	Meet	Herramienta de Google para reuniones.	DC2, DC3
AT	Atenea	Plataforma para la comunicación con los profesores	GP
PyC	PyCharm	Entorno de desarrollo específicamente para el lenguaje de Python.	D
PV	ParaView	Herramientas de visualización de modelos en 3D y 2D.	D2, D3, D5, D6, D7, D8, D9, D10, DC3

PyV	PyVista	Herramientas de visualización de modelos en 3D y 2D.	D2, D11
OpGL	OpenGL	Herramientas de visualización de modelos en 3D y 2D.	D2, D11
JN	Jupyter Notebook	Es un entorno informático interactivo basado en la web para crear documentos de Jupyter Notebook.	D9
MES	meshio	Hay varios formatos de malla disponibles para representar mallas no estructuradas. meshio puede leer, escribir y realizar conversiones entre los distintos formatos de mallas	D4, D5, D7, D8, D10, D11
TA	Taiga	Plataforma de gestión de proyectos para startups, desarrolladores ágiles y diseñadores	DC

Tabla 1: Recursos utilizados en el proyecto. Fuente: Elaboración propia.

### 5.3 Estimaciones

A continuación, se muestran resumidamente las tareas que se realizarán en este TFG.

Código	Tarea	Duración (horas)	Dependencias	Recursos
GP	GESTIÓN DEL PROYECTO	Total: 65		
GP1	Alcance	25	-	JV, DR, O365, AT
GP2	Planificación	10	GP1	JV, DR, O365, AT
GP3	Gestión económica y de sostenibilidad	10	GP2	JV, DR, O365, AT
GP4	Integración final del proyecto	20	GP3	JV, DR, O365, AT
D	DESARROLLO	Total: 340		
D1	Especificación de los requisitos	20	-	JV, PyC, GH
D2	Preparación del entorno	10	D1	JV, PyC, GH, PV, PyV, OpGL
D3	Familiarizarse con VTK y ParaView	40	D2	JV, PyC, PV
D4	Definir la estructura del modelo en función de los datos de entrada	30	D1, D2, D3	JV, PyC, GH, MES
D5	Escritura de los resultados en paralelo con la ejecución	40	D4	JV, PyC, GH, PV, MES
D6	Visualización de la simulación en 2D en ParaView	30	D4	JV, PyC, GH, PV
D7	Visualización de la simulación en 3D en ParaView	50	D4	JV, PyC, GH, PV, MES

D8	Funcionalidad de cambiar los puntos de interés en el cálculo del campo electromagnético.	50	D5, D6, D7	JV, PyC, GH, PV, MES
D9	Entorno de ejecución en Jupyter Notebook	20	D8	JV, PyC, GH, PV, JN
D10	Realización de los test para ParaView	35	D6, D7, D8	JV, PyC, GH, PV, MES
D11	Realización de los test de compatibilidad con otra herramienta	15	D5, D7	JV, PyC, GH, PyV, OpenGL, MES
DC	DOCUMENTACIÓN Y COMUNICACIÓN	Total: 135		
DC1	Seguimiento y control del proyecto	-	-	JV, TA, DR, O365
DC2	Documentación de la fase de seguimiento	30	GP4	JV, OC, DR, O365, TA, ME, Gmail
DC3	Comunicación	15	-	JV, OC, O365, PyC, PV, ME, TA, Gmail
DC4	Documentación de la fase final	60	CD2	JV, DR, O365, TA
DC5	Preparación de la presentación final	30	CD4	JV, DR, TA, O365
	TOTAL	540		

Tabla 2: Resumen de las tareas del TFG. Fuente: Elaboración propia.

# 5.4 Diagrama de Gantt

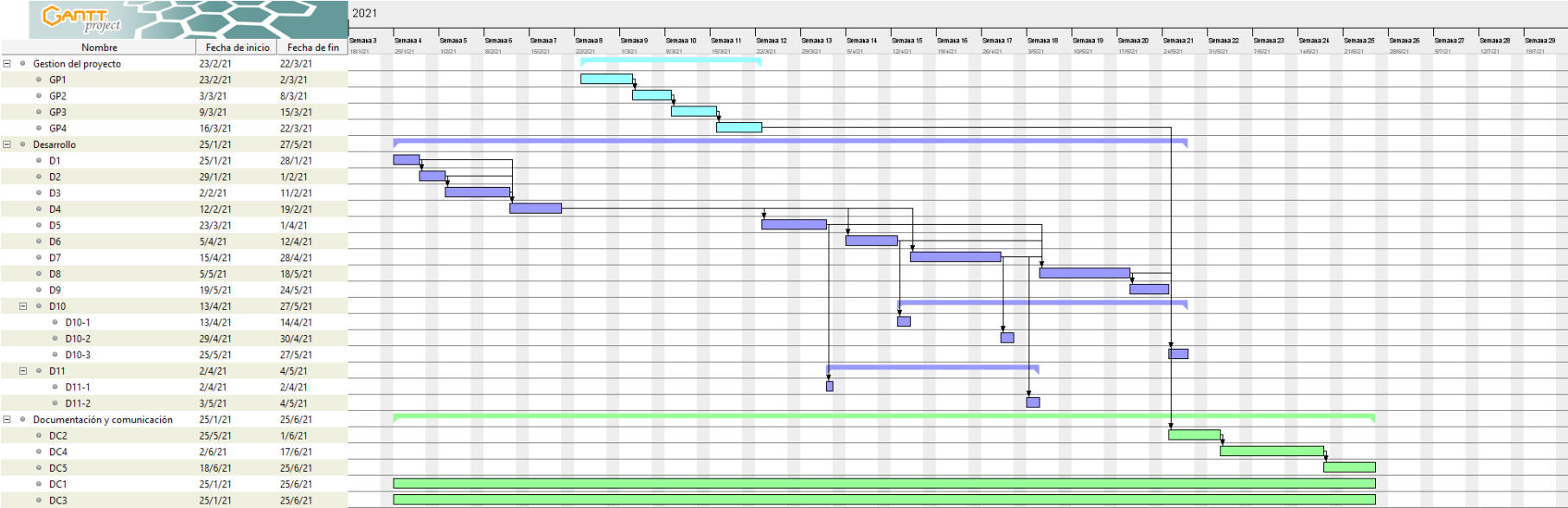


Figura 4: Diagrama de Gantt de la planificación inicial. Fuente: Elaboración propia.

## 6. Gestión de riesgos

Como ya hemos visto anteriormente, existe la posibilidad de que surjan ciertas situaciones durante el desarrollo de este proyecto, que afecten significativamente a la calidad de este. Por lo cual se ha calculado para cada uno de estos riesgos, el impacto que tendrían en la planificación del proyecto, la probabilidad de que ocurrieran y las horas aproximadas que se tendrían que trabajar para solucionar estos problemas.

Riesgos	Impacto	Probabilidad	Horas aprox.
Inexperiencia en las tecnologías necesarias	Alta	Media (60%)	50
Bugs	Medio	Media (65%)	30
Limitaciones no conocidas en las tecnologías utilizadas	Medio	Baja (30%)	40
Plazos de entrega fijados	Medio	Baja (25%)	20

Tabla 3: impacto, probabilidad y horas de desviación por cada riesgo. Fuente: Elaboración propia.

A continuación, se explicarán las acciones a realizar y los planes alternativos que se han considerado en uno o más de los riesgos antes mencionados.

### 6.1 Inexperiencia en las tecnologías necesarias

Este riesgo es uno de los que tienen más impacto en el proyecto, porque puede influir mucho en la planificación inicial, ya que se puede dar la situación de que la curva de aprendizaje sea mayor a la esperada, lo que ocasionará que se necesite más tiempo en el desarrollo del código.

Las medidas que se tomarán para minimizar al máximo este riesgo son, informarse de las dificultades de la implementación de las tecnologías a usar, y sobreestimar el número de horas en las tareas que están relacionadas con el desarrollo de código usando estas tecnologías. De esta forma tendremos un margen de error para garantizar la calidad del proyecto (el tiempo estimado es de unas 50 horas aproximadamente). No se necesitarán recursos adicionales para gestionar este riesgo.

### 6.2 Bugs

Uno de los problemas más comunes en el desarrollo de software son los bugs o errores en el código, estos suelen ser frecuentes cuando se está desarrollando con una nueva tecnología o con una que no conocemos.

Las medidas que se tomarán para minimizar lo máximo posible el impacto de este riesgo, es realizar test en cada una de las tareas, para asegurarnos de detectar los bugs que surjan y



solucionarlo lo más rápidamente posible. En el caso de que surjan bugs en tareas ya finalizadas, se adaptará la planificación creando una nueva tarea asignada para solucionar este error. El tiempo dedicado en estas nuevas tareas será contabilizado como tiempo de solución de bugs (aproximadamente 30 horas). No se necesitarán recursos adicionales para gestionar este riesgo.

### 6.3 Limitaciones no conocidas en las tecnologías utilizadas

Es un problema que puede surgir en cualquier momento durante el desarrollo de las funcionalidades del software, y que nos impide la creación de esta.

Las medidas que se tomarán para evitar este riesgo se detallan a continuación. Antes de iniciar el proceso de desarrollo del software se estudiará en profundidad las tecnologías que usaremos para asegurarnos de que no tiene limitación en las funcionalidades que se crearán. En el caso de que se descubran ciertas limitaciones una vez empezado el proceso de desarrollo, las medidas que se tomarán para solucionar los problemas dependerán del punto en que se encuentre el desarrollo.

En el caso de que se encuentren al principio del proceso de desarrollo, la medida que se tomará será considerar sustituir la tecnología actual por otra que nos permita realizar todas las funcionalidades esperadas, o bien seguir con la misma tecnología, pero se resolverá esta limitación utilizando otra herramienta o tecnología.

Y en el caso de que se encuentren a la mitad o al final del proceso de desarrollo, la medida que se tomará será la mencionada anteriormente, la de parchear esta limitación utilizando otra herramienta o tecnología. No se necesitarán recursos adicionales para gestionar este riesgo.

### 6.4 Plazos de entrega fijados

La fecha de entrega establecida del proyecto es en junio, por lo que la planificación inicial se ha hecho teniendo en cuenta este factor y se ha adaptado el proyecto al tiempo disponible para realizarlo. Este problema surge cuando se presentan ciertas situaciones que nos hacen tener desviación en la planificación y estas afectan a la calidad del trabajo.

Por lo tanto, las medidas que se tomarán para evitar estas situaciones son, tener buenas prácticas en el desarrollo del código para evitar posibles bugs y planificar un tiempo estimado para solucionar esto imprevistos (el tiempo planificado es de 20 horas aproximadamente). No se necesitarán recursos adicionales para gestionar este riesgo.

# 7. Presupuesto

Una vez hecha la planificación inicial, en la que se identifican todas las tareas que se llevarán a cabo, se procederá a especificar el coste final del proyecto incluyendo el desglose del coste de cada categoría.

## 7.1 Identificación y estimación de los costes

Para llevar a cabo correctamente el desarrollo del software, se necesitan una serie de recursos humanos y materiales. En los cuales, los recursos humanos son las personas que trabajarán en el proyecto, y los recursos materiales son el software, el hardware y el lugar en los que se desarrolla el proyecto.

Con el objetivo de calcular correctamente el coste de cada uno de estos recursos que son necesarios, se tendrán en consideración los siguientes factores: coste de recursos humanos, los costes generales (software, hardware, instalaciones, etc.), costes de contingencia y los costes de los posibles imprevistos.

### 7.1.1 Recursos humanos

El coste de recursos humanos hace referencia a los diferentes tipos de roles y número de personas que son necesarias para realizar las tareas descritas en el diagrama de Gantt. Pero para poder asignar el coste de cada tarea, primero se tiene que establecer un salario aproximado para cada uno de los roles y establecer un número aproximado de personas que conformarán el proyecto.

En este proyecto están involucradas dos personas, las cuales son, Octavio Castillo y yo, Jesús Velázquez. Octavio realizará el rol de *Product Owner* o dueño del producto y yo realizaré el rol de desarrollador. Estas aproximaciones de sueldos se han hecho a partir del sueldo mínimo estipulado por la FIB (Facultad de Informática de Barcelona de la UPC) para estudiantes en prácticas y también según la información obtenida en la página de búsqueda de sueldos y remuneraciones glassdoor (glassdoor, 2021).

Rol	Salario / hora (bruto)	Salario / hora (bruto) + Seguridad Social (x1.3)
<i>Product Owner</i> (Dueño del producto)	25€	32,5€
Desarrollador	9€	11,7€

Tabla 4: Sueldos estimados por rol y hora. Fuente: Elaboración propia, con los datos de (glassdoor, 2021)

Luego de definir el costo de cada rol, en la Tabla 4. A continuación, podemos calcular el coste de cada tarea a partir del sueldo por hora de cada persona, una vez calculado junto con el coste de la Seguridad Social. Podemos ver el cálculo en la Tabla 5.

Código	Tarea	Duración (horas)	Horas		Coste (€)
			Jesús	Octavio	
GP	GESTIÓN DEL PROYECTO	Total: 65			761
GP1	Alcance	25	25		293
GP2	Planificación	10	10		117
GP3	Gestión económica y de sostenibilidad	10	10		117
GP4	Integración final del proyecto	20	20		234
D	DESARROLLO	Total: 340			3.979
D1	Especificación de los requisitos	20	20		234
D2	Preparación del entorno	10	10		117
D3	Familiarizarse con VTK y ParaView	40	40		468
D4	Definir la estructura del modelo en función de los datos de entrada	30	30		351
D5	Escritura de los resultados en paralelo con la ejecución	40	40		468
D6	Visualización de la simulación en 2D en ParaView	30	30		351
D7	Visualización de la simulación en 3D en ParaView	50	50		585
D8	Funcionalidad de cambiar los puntos de interés en el cálculo del campo electromagnético.	50	50		585
D9	Entorno de ejecución en Jupyter Notebook	20	20		234
D10	Realización de los test para ParaView	35	35		410
D11	Realización de los test de compatibilidad con otra herramienta	15	15		176
DC	DOCUMENTACIÓN Y COMUNICACIÓN	Total: 135			1.944
DC1	Seguimiento y control del proyecto	-	-	-	-
DC2	Documentación de la fase de seguimiento	30	20	10	559
DC3	Comunicación	15	7,5	7,5	332
DC4	Documentación de la fase final	60	60		702
DC5	Preparación de la presentación final	30	30		351
	<b>TOTAL</b>	<b>540</b>			<b>6.684</b>

Tabla 5: Costes de personal por actividad. Fuente: Elaboración propia.

## 7.1.2 Generales

Los costes generales son los que calculamos de forma global, sin calcular su coste a nivel de tareas. En este proyecto, los gastos generales se dividen en las siguientes categorías: hardware, software y recursos materiales. Los gastos que se representan en la categoría de hardware y de software, son los componentes y los softwares que se usarán en el día a día para el desarrollo del proyecto y para la comunicación.

En la categoría de recursos materiales, se consideran todos los gastos relacionados con el área de trabajo, algunos de estos son: el suministro de la electricidad, el servicio de internet, los materiales de oficina, entre otros. Para facilitar el cálculo de estos servicios, se ha decidido tomar como

referencia el precio de un espacio de coworking en Barcelona para determinar el gasto global del espacio de trabajo.

Para cada uno de los recursos utilizados se calcula su amortización siguiendo esta fórmula:

$$\frac{\text{Costes del equipo (€)}}{\text{Vida útil} * \text{Días laborables al año} * \text{Dedicación diaria (horas)}} * \text{Dedicación del TFG (horas)}$$

Donde la vida útil es de 4 años, los días laborables al año son 220, la dedicación diaria son 5 horas y las horas dedicadas en el TFG son 540.

Recursos	Coste (€)	Amortización (€) (*)
<b>HARDWARE</b>	1.429	176
Portátil MSI Prestige 15 A11SCS-033XES (MSI, 2021)	1300	160
Monitor BenQ GW2480 (BENQ, 2021)	129	16
<b>SOFTWARE</b>	69	-
GitHub	Gratuito	-
Office 365	69 anual	-
Meet	Gratuito	-
Atenea	Gratuito	-
PyCharm	Gratuito	-
ParaView	Gratuito	-
PyVista	Gratuito	-
OpenGL	Gratuito	-
Jupyter Notebook	Gratuito	-
<b>RECURSO MATERIALES</b>	1.500	-
Espacio de trabajo (Meet BCN, 2021)	1.500 (5 meses)	-
<b>TOTAL</b>	2.998	

Tabla 6: Costes estimados para los gastos generales. Fuente: Elaboración propia.

(\*): Amortización a los 5 meses por la duración del proyecto

### 7.1.3 Contingencia

El coste de contingencia es dinero que se le añade al presupuesto del proyecto, con el fin de tener un margen de seguridad, para el caso de que surgieran imprevistos que no se habían anticipado. Este coste se calcula con un porcentaje del valor total del proyecto, en el caso del sector informático se suelen fijar un rango de entre el 10% y el 20%. En este proyecto se ha escogido un margen de seguridad del 15%.

En la Tabla 7 se puede visualizar el cálculo concreto en cada uno de los tipos de costes del proyecto, en los que se le asigna un sobrecoste de contingencia.

Tipos de costes	Coste (€)	Contingencia (%)	Coste Contingencia (€)
Coste de recursos humanos	6.684	15	1.003
Costes generales	2.998	15	450
<b>TOTAL</b>			1.453

Tabla 7: Costes de contingencia. Fuente: Elaboración propia.

#### 7.1.4 Imprevistos

Como ya hemos visto en el apartado de gestión de riesgos, se han especificado las acciones y los planes alternativos que se realizarán en caso de que ocurrieran uno o más de los riesgos antes mencionados. En el caso de que se realicen algunas de estas acciones o planes alternativos, ocasionarían un impacto económico en el proyecto, es por esto por lo que es necesario calcular el dinero que se destinará a la solución de estos riesgos.

A continuación, en la Tabla 8 se pueden visualizar los costes asociados a cada imprevisto que se pueden encontrar en el desarrollo del proyecto.

Riesgo	Probabilidad (%)	Tiempo (Horas)	Coste Estimado (€)	Coste Presupuestado (€)
Inexperiencia en las tecnologías necesarias	60%	50	585	351
Bugs	65%	30	351	229
Limitaciones no conocidas en las tecnologías utilizadas	30%	40	468	141
Plazos de entrega fijados	25%	20	436	109
<b>TOTAL</b>				<b>830</b>

Tabla 8: Costes presupuestados para los imprevistos. Fuente: Elaboración propia.

El coste presupuestado es la provisión en base a la probabilidad de que ocurran posibles riesgos.

En el caso de que no ocurriera un imprevisto, el dinero asignado a este se distribuirá en los demás riesgos que si se tienen que solucionar. Y en el mejor de los casos, en el que no ocurran ninguno de estos riesgos, todo el dinero se destinará a las tareas de desarrollo, con el fin de mejorar las funcionalidades del software y la experiencia de usuario.

#### 7.1.5 Presupuesto final

Finalmente, el presupuesto final está determinado por la suma de todos los gastos anteriores (el precio está redondeado al alza).

Tipos de costes	Coste (€)
Coste de recursos humanos	6.684
Costes generales	2.998
Costes de contingencia	1.453
Costes de imprevistos	830
<b>TOTAL</b>	<b>11.965</b>

Tabla 9: Costes estimados del proyecto. Fuente: Elaboración propia

## 7.2 Control de gestión

En este apartado se explicará la forma en la que se llevará a cabo el control de los gastos del proyecto y la detección de las posibles desviaciones en el presupuesto que se pudieran tener. Este control contiene los siguientes indicadores, con los que podremos estimar si estamos dentro del presupuesto inicial, o si por el contrario está ocurriendo alguna desviación no esperada. Estos indicadores son:

- **Desviación total de horas:**  
horas totales estimadas – horas totales reales
- **Desviación total de costes:**  
Coste total estimado – Coste total real
- **Desviación total del coste del personal:**  
Coste personal estimado – Coste personal real
- **Desviación del coste de recursos humanos por tarea:**  
(Coste estimado – Coste real) \* horas
- **Desviación del coste de la tarea según el número de horas:**  
(horas estimadas – horas reales) \* Coste por hora
- **Desviación total del coste general:**  
Coste general estimado – Coste general real
- **Desviación total del coste de imprevistos:**  
Coste de imprevistos estimado – Coste de imprevistos real

Para realizar esta supervisión y facilitar la visualización de los gastos y de las posibles variaciones que se puedan tener, se creará una hoja de Excel con todos los indicadores mencionados anteriormente, de esta forma podremos realizar un análisis del presupuesto y de los gastos realizados, con el fin de prever si será necesario utilizar el dinero estimado en el coste de contingencia.

# 8. Especificación del sistema

## PETGEM

Antes de empezar a explicar los cambios y las mejoras que se han realizado en este proyecto, se explicará cómo es el flujo de trabajo actual que tiene PETGEM en la ejecución de una simulación de un modelo 3D CSEM, y posteriormente se explicará cómo será el nuevo flujo de trabajo que tendrá la nueva versión de PETGEM, en la cual estarán implementadas las mejoras y los cambios que se tienen establecidos para este proyecto.

### 8.1 Flujo de trabajo sin visualización

El flujo de trabajo actual de las ejecuciones PETGEM se describe en la Figura 5.

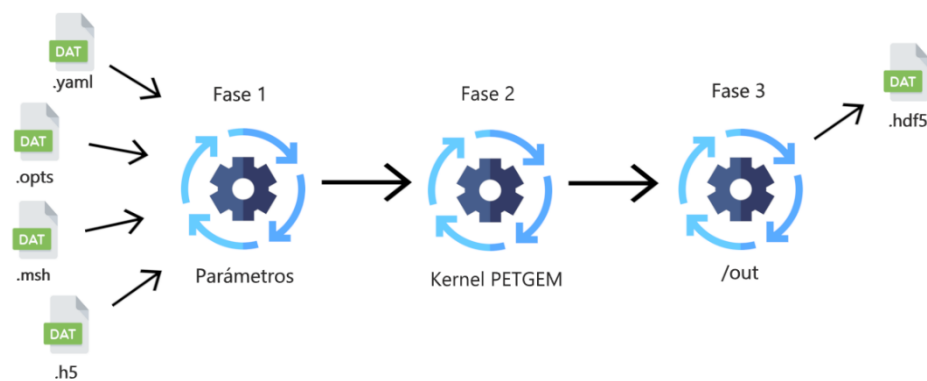


Figura 5: Flujo de trabajo sin visualización de PETGEM. Fuente: Elaboración propia.

Como podemos observar en la Figura 5 el flujo de trabajo comienza por la fase 1, en la que el usuario debe introducir los ficheros necesarios para la ejecución del kernel de PETGEM. Los ficheros que debe introducir el usuario en esta fase son:

- `params.yaml` : archivo donde se definen los parámetros del modelo 3D CSEM / MT
- `petsc.opts` : archivo donde se definen las opciones para PETSc
- `fichero.h5`: una lista de posiciones de los receptores en formato hdf5
- `fichero.msh` : un archivo de malla (la versión actual admite archivos de malla de Gmsh)
- Un modelo de conductividad asociado con los materiales definidos en el archivo de malla

Una vez que todos los datos necesarios estén disponibles para iniciar la simulación de PETGEM, el usuario ejecuta el kernel de PETGEM. Con esta acción comienza la segunda fase del flujo de trabajo, la cual se centra en la ejecución del kernel para la obtención de los datos de la simulación.

En cuanto finaliza la ejecución del kernel, se obtienen los resultados del modelo 3D CSEM y se genera un archivo con el formato hdf5 que contiene los resultados de la simulación. Este fichero se almacena en el directorio de salida establecido por el usuario (directorio: /out). En el momento en el que se genera el fichero de salida finaliza la ejecución del kernel de PETGEM, por lo tanto, finaliza la segunda fase.

A continuación, comienza la fase 3, la cual se centra en la visualización de los resultados de la simulación, en donde el usuario puede analizar el resultado a través de una herramienta de visualización que sea compatible con el formato hdf5. A continuación, en la Figura 6 se puede observar un ejemplo de la visualización de los resultados de un modelo 3D CSEM.

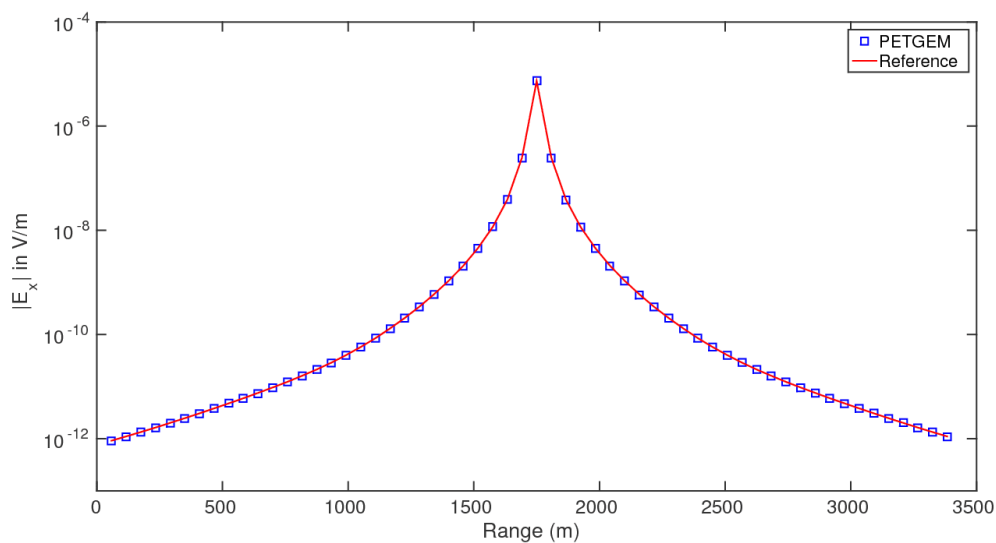


Figura 6: Comparativo de campo eléctrico total para el componente x entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021)

Con esta versión de PETGEM el usuario solo puede visualizar una gráfica generada a partir de los datos obtenidos del fichero de salida (el fichero con formato hdf5)

## 8.2 Flujo de trabajo con visualización

Una vez que se ha explicado el flujo de trabajo actual que tiene PETGEM, ahora pasaremos a describir el nuevo flujo de trabajo que se tendrá en la nueva versión de PETGEM. A continuación, en la Figura 7 podemos observar cómo es el nuevo flujo de trabajo.





Figura 7: Flujo de trabajo con visualización de PETGEM. Fuente: Elaboración propia.

La nueva versión de PETGEM ha realizado cambios y mejoras en el flujo de trabajo. Aunque las principales mejoras se realizan en la fase de ejecución del kernel (fase 2) y en la fase de visualización de los resultados (fase 3), estas modifican el flujo de trabajo, sin embargo, cambios realizados en la fase 1 no afectan al flujo de trabajo, pero si afectan a la forma en que tiene el usuario de interactuar con el software. Estas mejoras consisten en la implementación de una imagen Docker y la utilización de la interfaz de Jupyter Notebook. La imagen Docker le permite al usuario realizar la ejecución de PETGEM en un entorno controlado, y la interfaz de Jupyter Notebook facilita la interacción que tiene el usuario con la aplicación.

Como podemos observar en la Figura 7 la fase 1 no cambia con respecto a la versión anterior de PETGEM, por lo cual el usuario tendrá que seguir introduciendo los mismos ficheros para poder ejecutar la simulación. Sin embargo, ahora en la fase dos, se añade un módulo de Post-procesamiento en el kernel de PETGEM, este nuevo módulo se encarga de procesar los resultados de la simulación y convertirlos en un fichero con el formato VTK y en una imagen JPG, la cual contiene una gráfica que representa estos resultados. En esta fase se sigue manteniendo la construcción del fichero con el formato hdf5 en el cual se almacenan los resultados de la simulación.

Una vez que finaliza la fase 2 se pasa a la fase 3, donde además de seguir contando con la posibilidad de usar una herramienta de visualización para ver los datos almacenados en el fichero de salida con el formato hdf5 (Figura 6), se han incorporado 2 opciones diferentes para la visualización e interpretación de los resultados.

La primera opción consiste en el uso de un fichero de salida en formato VTK y usar la herramienta de visualización ParaView. Esta herramienta permite generar vistas 3D de los resultados (malla y campos electromagnéticos) de la simulación. En la Figura 8 se puede observar un ejemplo de cómo se visualizarán los resultados almacenados en el fichero VTK.

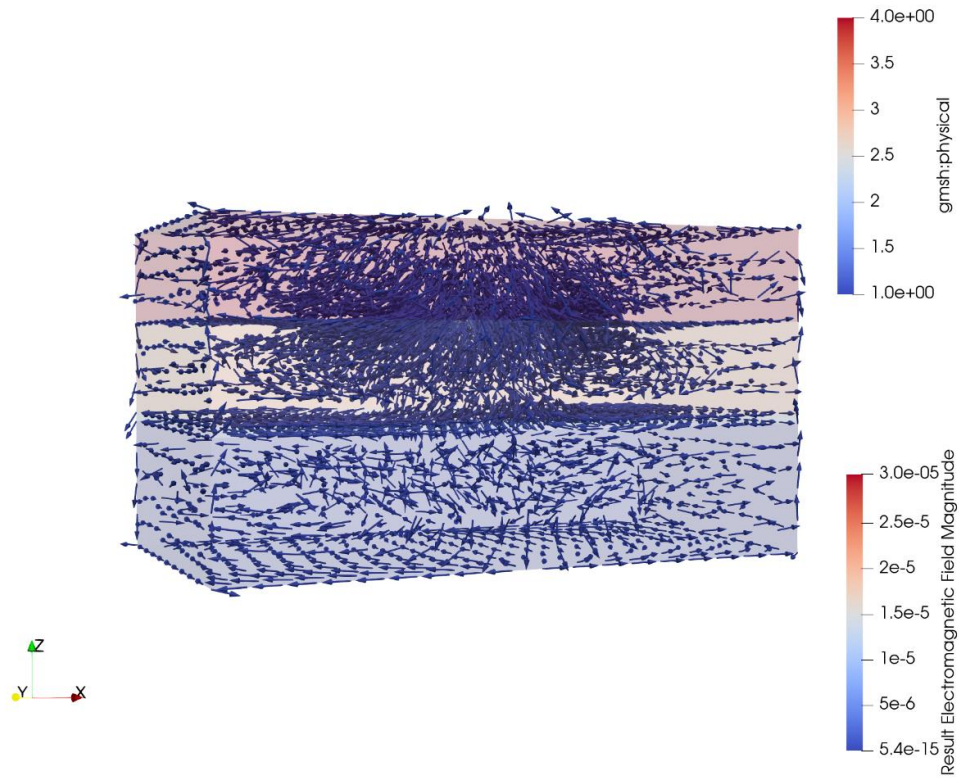


Figura 8: Visualización del resultado de PETGEM con ParaView. Fuente: Elaboración propia.

La segunda opción añadida para interpretar los resultados es la imagen JPG, esta imagen contiene una gráfica representando los datos obtenidos en la simulación. Con esta gráfica los usuarios podrán interpretar los resultados a partir del análisis de la amplitud de campo eléctrico para una lista de puntos dados. En la Figura 9 se puede observar un ejemplo de dicha visualización.

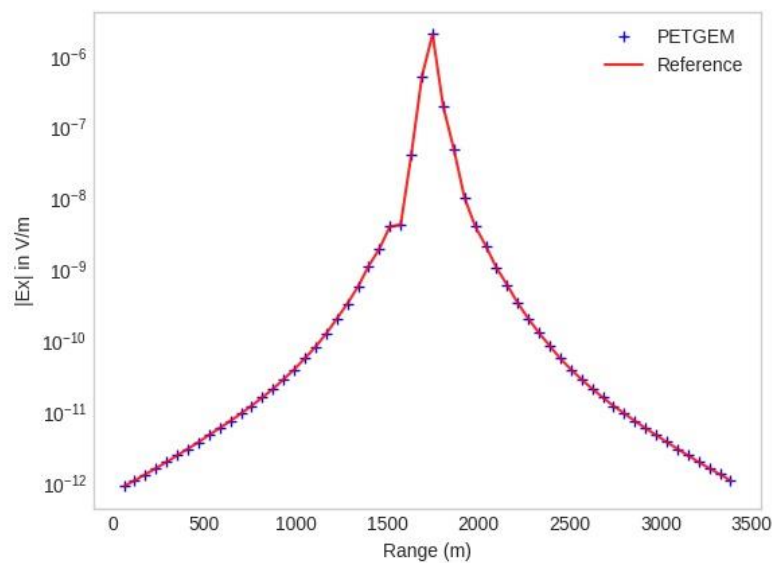


Figura 9: Visualización de la amplitud de campo eléctrico para una lista de puntos. Fuente: Elaboración propia.

### 8.3 Postprocesado rápido de nuevos resultados

La versión de PETGEM desarrollada en este TFG cuenta con la característica de que los usuarios pueden ejecutar el mismo modelo 3D CSEM nuevamente cambiando los puntos de interés seleccionados, sin tener que ejecutar toda la simulación nuevamente. Gracias a esta nueva funcionalidad, los usuarios pueden realizar un mayor número de pruebas sobre el mismo modelo 3D CSEM en un intervalo de tiempo mucho más reducido.

Esta nueva funcionalidad no modifica el nuevo flujo de trabajo, esto se consigue almacenando una malla, la cual contiene los resultados de los campos electromagnéticos en todos los puntos del modelo 3D CSEM, lo que nos permite realizar menos procesos para obtener el resultado deseado.

El flujo de trabajo para obtener estos resultados solicitados es el siguiente:

1. Interpolación en los nuevos puntos de interés de respuestas electromagnéticas (eléctricas y magnéticas)
2. Ejecutar el postprocesado, para generar la imagen JPG que contiene la interpretación de los resultados de la interpolación

Este flujo de trabajo se puede observar en la siguiente imagen, en la Figura 10.

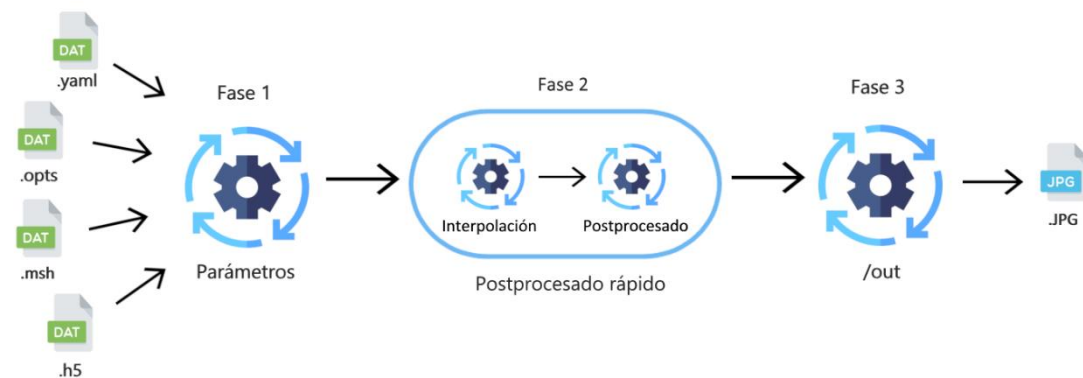


Figura 10: Flujo de trabajo del postprocesado rápido de nuevos resultados. Fuente: Elaboración propia.

# 9. Desarrollo

En este apartado se explicarán las implementaciones de las diversas funcionalidades agregadas en la nueva versión de PETGEM. Los apartados que se explicaran a continuación se pueden dividir en tres módulos principales, los cuales son la integración de interfaz de Jupyter Notebook, la creación de una imagen Docker en la cual se ejecuta la herramienta de PETGEM y por último las mejoras realizadas en la visualización de los resultados obtenidos en las simulaciones de PETGEM. Estas mejoras están enfocadas en aumentar la facilidad de uso del software PETGEM.

## 9.1 Jupyter Notebook

Jupyter Notebook es una aplicación cliente-servidor, que permite crear y compartir documentos web en formato JSON que siguen un esquema de versiones y una lista ordenada de celdas de entrada y de salida. Estas celdas contienen código, texto (formato Markdown), fórmulas y ecuaciones matemáticas, o también contenido multimedia (Rich Media), etc. El programa se ejecuta desde una aplicación web del lado del cliente que se puede ejecutar en cualquier navegador estándar.

En una sola interfaz, los usuarios pueden escribir, documentar y ejecutar código, visualizar datos, realizar cálculos y ver resultados. Esta interfaz incluye la ventaja de la organización del código en unidades independientes, es decir, los bloques de código específicos se pueden ejecutar individualmente.

### 9.1.1 Jupyter Notebook en PETGEM

Se ha decidido incluir la herramienta Jupyter Notebook dentro del software de PETGEM, con el fin de facilitar el uso de la herramienta a los usuarios. Desde la interfaz gráfica de Jupyter Notebook el usuario podrá visualizar y modificar los ficheros requeridos para la ejecución de la simulación en PETGEM, además de gestionar los ficheros, el usuario podrá ejecutar la simulación desde esta interfaz y posteriormente visualizar los resultados con las respectivas herramientas de visualización.

En la Figura 11 se puede observar cómo es la interfaz desde la cual el usuario gestiona la ejecución de PETGEM.

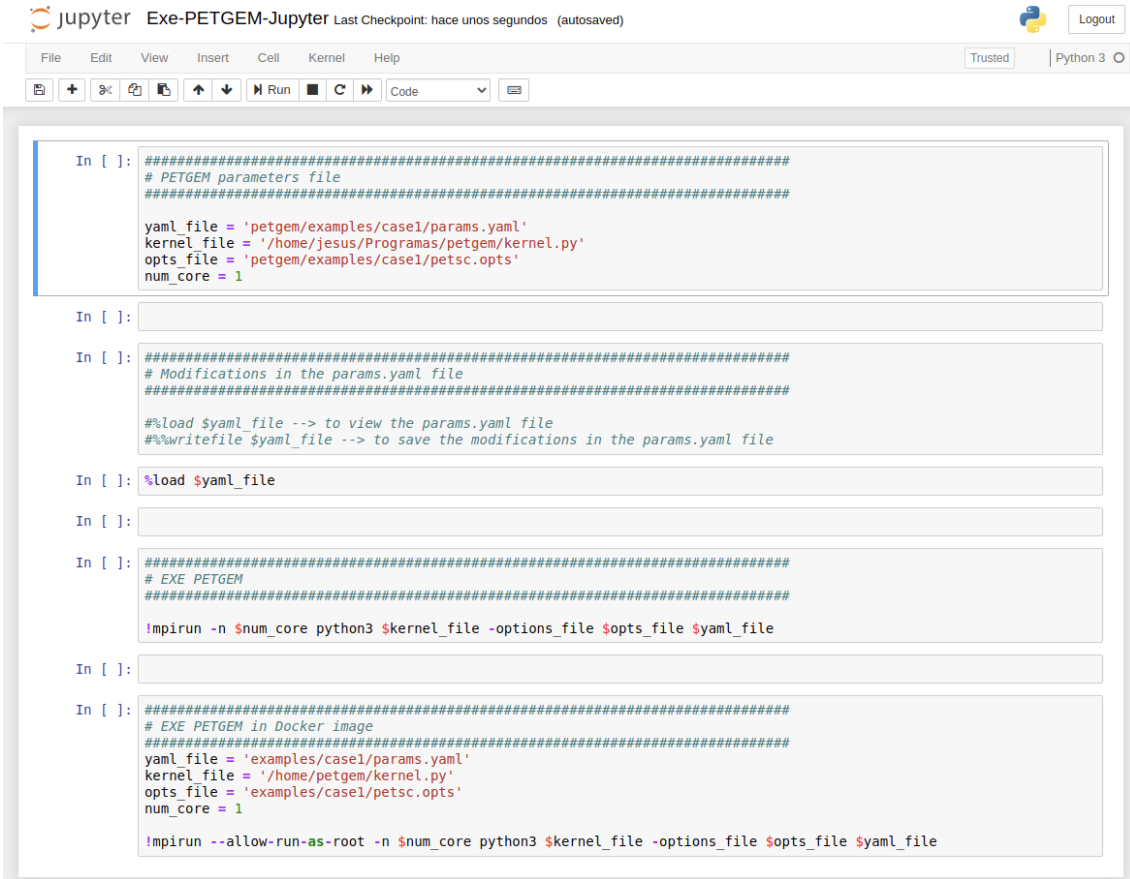


Figura 11: Interfaz gráfica de Jupyter Notebook con Script de PETGEM. Fuente: Elaboración propia.

Como podemos observar en la Figura 11, una vez el usuario tenga el script de ejecución que PETGEM proporciona, deberá rellenar los *path* con las posiciones de los ficheros que se le solicita (esto se puede ver en la primera celda del script), posteriormente se le proporciona una celda donde el usuario podrá visualizar y modificar los ficheros que él desee, sin la necesidad de salir de la interfaz gráfica (estas funcionalidades se encuentran en la segunda celda del script de ejecución).

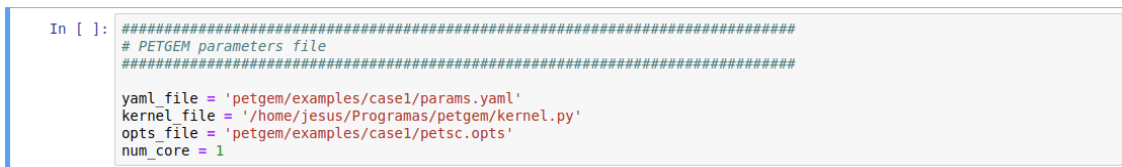


Figura 12: Celda con parámetros de PETGEM. Fuente: Elaboración propia.

Una vez que el usuario haya introducido todos los datos necesarios (habiendo ejecutado la celda donde se solicitan estos, Figura 12) podrá ejecutar la celda donde se encuentra la sentencia de ejecución del kernel de PETGEM, una vez hecho esto podrá seguir la ejecución de la simulación a través de la salida de la terminal de la interfaz (esta se encuentra debajo de la celda que se ejecuta).

## 9.1.2 Lectura y modificación de ficheros

A continuación, se mostrará cómo el usuario puede ver y modificar los ficheros solicitados para la ejecución de la simulación de un modelo 3D CSEM en PETGEM.

```
In [ ]: #####
# Modifications in the params.yaml file
#####

#%load $yaml_file --> to view the params.yaml file
#%writefile $yaml_file --> to save the modifications in the params.yaml file

In [ ]: %load $yaml_file
```

Figura 13: Lectura de archivos en Jupyter Notebook. Fuente: Elaboración propia.

Como podemos observar en la Figura 13 cuando el usuario quiere visualizar el contenido de un fichero, lo hace ejecutando la sentencia “%load” seguido de la dirección donde se encuentra el fichero. Una vez que se ejecuta la sentencia se muestra el contenido del fichero en la celda ejecutada, podemos observar un ejemplo de esta en la siguiente Figura 14.

```
In [ ]: # %load petgem/examples/case1/params.yaml
#####
# PETGEM parameters file
#####
# Model parameters
model:
  mesh_file: examples/case1/DIPOLE1D.msh # Mesh file (gmsh format v2)
  basis_order: 1 # Vector basis order (1,2,3,4,5,6)
  frequency: 2.0 # Frequency
  src_position: [1750., 1750., -975.] # Source position (xyz)
  src_azimuth: 0 # Source rotation in xy plane
  src_dip: 0 # Source rotation in xz plane
  src_current: 1. # Source current
  src_length: 1. # Source length
  sigma_horizontal: [1., 0.01, 1., 3.3333] # Horizontal conductivity for each material
  sigma_vertical: [1., 0.01, 1., 3.3333] # Vertical conductivity for each material
  receivers_file: examples/case1/receiver_pos.h5 # Receiver positions file (xyz)

# Execution parameters
run:
  cuda: False # Flag to activate/deactivate cuda support

# Output parameters
output:
  directory: examples/case1/out/ # Directory for output (results)
  directory_scratch: examples/case1/tmp/ # Directory for temporal files
```

Figura 14: Modificación de fichero en Jupyter Notebook. Fuente: Elaboración propia.

Seguidamente el usuario podrá modificar el fichero visualizado o crear un nuevo fichero. Una vez que desee guardar este fichero tendrá que escribir la sentencia “%%write” seguido del *path* del fichero (en donde se incluye el nombre del fichero), en caso de que exista el fichero se sobrescribirá, un ejemplo de este proceso se puede observar en la siguiente imagen Figura 15.

```

In [ ]: #####
# Modifications in the params.yaml file
#####

#%load $yaml_file --> to view the params.yaml file
#%writefile $yaml_file --> to save the modifications in the params.yaml file

In [6]: %%writefile $yaml_file
#####
# PETGEM parameters file
#####
# Model parameters
model:
  mesh_file: examples/case1/DIPOLE1D.msh # Mesh file (gmsh format v2)
  basis_order: 1 # Vector basis order (1,2,3,4,5,6)
  frequency: 2.0 # Frequency
  src_position: [1750., 1750., -975.] # Source position (xyz)
  src_azimuth: 0 # Source rotation in xy plane
  src_dip: 0 # Source rotation in xz plane
  src_current: 1. # Source current
  src_length: 1. # Source length
  sigma_horizontal: [1., 0.01, 1., 3.3333] # Horizontal conductivity for each material
  sigma_vertical: [1., 0.01, 1., 3.3333] # Vertical conductivity for each material
  receivers_file: examples/case1/receiver_pos.h5 # Receiver positions file (xyz)

# Execution parameters
run:
  cuda: False # Flag to activate/deactivate cuda support

# Output parameters
output:
  directory: examples/case1/out/ # Directory for output (results)
  directory_scratch: examples/case1/tmp/ # Directory for temporal files

Overwriting petgem/examples/case1/params.yaml

```

Figura 15: Guardar un fichero en Jupyter Notebook. Fuente: Elaboración propia.

### 9.1.3 Ejecución de PETGEM en Jupyter Notebook

Una vez que se han proporcionado todos los ficheros necesarios para la ejecución de la simulación del modelo 3D CSEM seleccionado, el usuario iniciará la ejecución de la simulación desde la interfaz, desde donde podrá observar si ha ocurrido algún error en la ejecución de PETGEM. Este proceso se puede observar en la siguiente imagen Figura 16.

```

In [*]: #####
# EXE PETGEM
#####

!mpirun -n $num_core python3 $kernel_file -options_file $opts_file $yaml_file

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%                PETGEM                %%%
%%%    Parallel Edge-based Tool for Electromagnetic Modelling    %%%
%%%
%%%
%%%
%%% (c) Octavio Castillo-Reyes                %%%
%%%    Barcelona Supercomputing Center, 2020                %%%
%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Data preprocessing
Nodal coordinates
Mesh connectivity
Edges connectivity
Faces connectivity
Faces-edges connectivity
DOFs connectivity

```

Figura 16: Ejecución de PETGEM en Jupyter Notebook. Fuente: Elaboración propia.

Como podemos observar en la Figura 11, en el script existen dos formas de ejecutar PETGEM, la primera la cual es la que se ha explicado en esta sección, y una segunda la cual se ejecuta desde dentro de la imagen Docker de PETGEM, y esta se detalla en el siguiente apartado.

## 9.2 Imagen Docker

Docker es un software que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos, en otras palabras, Docker nos permite aislar aplicaciones (software) dentro de contenedores y hacer que funcionen en cualquier lugar.

Es por esto que la decisión de crear una imagen Docker que contenga el software de PETGEM y sus requisitos y además recursos adicionales, viene dada por las ventajas que aporta esta tecnología para facilitar a los usuarios la instalación y ejecución de software. Esto se debe a que con Docker se añade una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos, lo cual nos permite que los usuarios de PETGEM puedan ejecutar PETGEM a partir de una imagen Docker instalada en sus ordenadores, sin importar que sistema operativo tenga instalado.

Al realizar la instalación de PETGEM por este método, el usuario debe de tener en cuenta que este método de instalación no requiere de ningún requisito, pero la imagen Docker de PETGEM no cuenta con la herramienta de ParaView instalada, por lo cual el usuario debe de instalar este software en su ordenador para poder visualizar los resultados obtenidos a través de la imagen Docker de PETGEM.

### 9.2.1 Instalación y ejecución de la imagen Docker

Para iniciar la instalación de la imagen Docker el usuario debe de instalar Docker en su ordenador y posteriormente descargarse la imagen de PETGEM. Una vez que se completen estos pasos se tiene que crear un directorio de trabajo donde se cuente con el Dockerfile y un directorio, el cual será un directorio compartido (no volátil) entre su ordenador y el contenedor Docker.

A continuación, en la Figura 17 podemos ver un ejemplo del cual sería un directorio de trabajo. Y posteriormente en la Figura 18 podemos ver el contenido del Dockerfile.

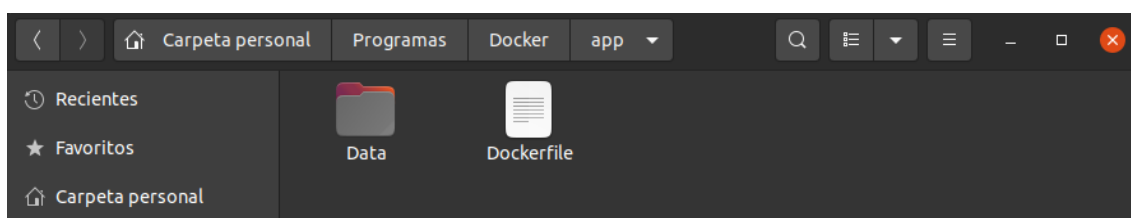


Figura 17: Directorio de trabajo en el ordenador de la imagen Docker. Fuente: Elaboración propia.



```

1 # Docker file for a slim Ubuntu-20.04 Python3 image
2
3 #Install Ubuntu-20.04
4 FROM ubuntu:20.04
5
6 #Install python 3.8.5
7 RUN apt-get update \
8     && apt-get install -y python3.8
9 RUN apt-get install -y firefox
10 RUN apt update
11 RUN apt install -y python3-pip
12 RUN ln -s /usr/bin/python3 /usr/bin/python
13
14 #Install Jupyter (Jupyter Notebook)
15 RUN pip install jupyter
16
17 #Install PETCs:
18 RUN pip install numpy
19 RUN pip install scipy
20 RUN pip install singleton-decorator
21 RUN apt-get install -y python3-sphinx
22 RUN apt install -y texlive-latex-extra
23 RUN pip install mpi4py
24 RUN pip install h5py
25 RUN pip install pyyaml
26
27 RUN apt-get install -y gfortran
28 RUN apt-get install -y python3-petsc4py-complex
29 RUN apt install -y git
30 RUN git clone -b maint https://bitbucket.org/petsc/petsc /PETSC
31 RUN apt-get install -y flex
32 RUN apt-get install -y bison
33 RUN cd PETSC/
34 RUN export PETSC_DIR=$PWD
35 RUN export PETSC_ARCH=arch-linux2-c-debug
36 RUN cd PETSC/ && ./configure --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-mpich --download-fblaslapack
--with-scalar-type=complex --download-mumps --download-scalapack --download-parmetis --download-metis
--download-ptscotch --download-cmake && make PETSC_DIR=/PETSC PETSC_ARCH=arch-linux-c-debug all && make
PETSC_DIR=/PETSC PETSC_ARCH=arch-linux-c-debug check && export PATH="${PETSC_DIR}/${PETSC_ARCH}/bin:${PATH}"
37
38
39 #Install PETGEM
40 RUN pip install pyyaml
41 RUN git clone https://github.com/ocastilloreyes/petgem.git /home/petgem
42
43 #Working directory
44 WORKDIR /home

```

Figura 18: Dockerfile de la imagen de PETGEM. Fuente: Elaboración propia.

Como podemos observar en la Figura 18 la imagen Docker se construye a partir del Dockerfile, en el cual se instala PETGEM, todas las dependencias y los recursos, como puede ser Jupyter Notebook, entre otros, a partir del comando RUN. Además de esto también se establece un directorio de trabajo con el comando WORKDIR.

Una vez construida la imagen Docker de PETGEM, el usuario podrá pasar todos los ficheros que quiera entre su ordenador y el contenedor Docker, con el objetivo de pasar los ficheros necesarios del modelo 3D CSEM que desea simular en PETGEM. Esta compartición de datos entre el contenedor Docker y el ordenador del usuario es posible, porque Docker tiene dos formas de almacenar los datos para que no se elimine una vez se detenga el contenedor Docker.

La principal diferencia de estas dos formas de almacenar los datos se encuentra en donde se almacenan los datos, la opción de crear una carpeta compartida (*Mounted Files*) almacena los datos en el ordenador y la opción de crear un volumen (*Docker Volumes*), almacena los datos en la nube.

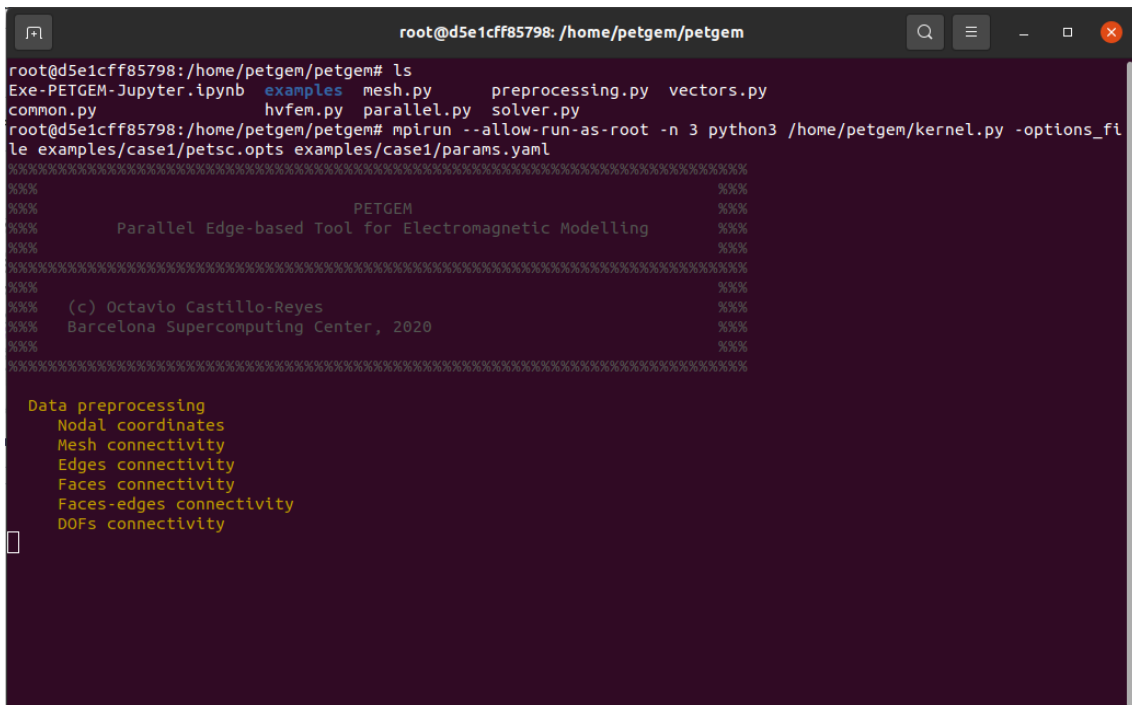
De estas dos formas de almacenamiento, se ha escogido la opción de compartir carpeta (*Mounted Files*) porque aporta más facilidad a la hora de compartir archivos entre el ordenador y el contenedor, por los siguientes motivos:

- Si quieres hacer un *backup* debes hacerlo sin ayuda de Docker
- Es muy útil para entornos de desarrollo, ya que si modificamos el contenido en nuestra computadora automáticamente se modifica en el contenedor y viceversa

Por lo consiguiente el usuario solo deberá de copiar los ficheros que necesite para la ejecución de PETGEM en su directorio compartido, y posteriormente podrá visualizar los resultados obtenidos en este directorio desde su ordenador. Además, esta transferencia de datos se realiza mientras el contenedor se encuentra ejecutándose

## 9.2.2 Ejecución de PETGEM en la imagen Docker

En esta sección se explicará las opciones que tienen para ejecutar PETGEM dentro del contenedor Docker. La imagen PETGEM cuenta con dos opciones para ejecutar la simulación del modelo 3D CSEM, la primera sería ejecutar PETGEM desde el propio terminal del contenedor Docker, con esta opción solo se tiene que ejecutar la sentencia del kernel como se puede observar en la siguiente imagen Figura 19.



```
root@d5e1cff85798: /home/petgem/petgem
root@d5e1cff85798: /home/petgem/petgem# ls
Exe-PETGEM-Jupyter.ipynb  examples  mesh.py  preprocessing.py  vectors.py
common.py                 hvfem.py  parallel.py  solver.py
root@d5e1cff85798: /home/petgem/petgem# mpirun --allow-run-as-root -n 3 python3 /home/petgem/kernel.py -options_file examples/case1/petsc.opts examples/case1/params.yaml
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%                PETGEM                %%%
%%%  Parallel Edge-based Tool for Electromagnetic Modelling  %%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%  (c) Octavio Castillo-Reyes           %%%
%%%  Barcelona Supercomputing Center, 2020 %%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Data preprocessing
  Nodal coordinates
  Mesh connectivity
  Edges connectivity
  Faces connectivity
  Faces-edges connectivity
  DOFs connectivity
```

Figura 19: Ejecución de PETGEM en la imagen Docker. Fuente: Elaboración propia.

La segunda opción de ejecutar PETGEM, es a través de la interfaz de Jupyter Notebook, desde donde se ejecuta la celda especificada para la ejecución en la imagen Docker. Un ejemplo de esta ejecución se puede ver en la siguiente imagen Figura 20.

```

In [*]: #####
# EXE PETGEM in Docker
#####
yaml_file = 'examples/casel/params.yaml'
kernel_file = '/home/petgem/kernel.py'
opts_file = 'examples/casel/petsc.opts'
num_core = 3

!mpirun --allow-run-as-root -n $num_core python3 $kernel_file -options_file $opts_file $yaml_file

#####
%%
%%          PETGEM          %%
%%  Parallel Edge-based Tool for Electromagnetic Modelling  %%
%%
%%#####
%%
%%  (c) Octavio Castillo-Reyes          %%
%%  Barcelona Supercomputing Center, 2020  %%
%%
%%#####

Data preprocessing
Nodal coordinates
Mesh connectivity
Edges connectivity
Faces connectivity
Faces-edges connectivity
DOFs connectivity

```

Figura 20: Ejecución de PETGEM a través de Jupyter Notebook dentro de la imagen Docker. Fuente: Elaboración propia.

### 9.3 Módulo de visualización

En esta sección veremos el módulo de postprocesado del kernel de PETGEM, el cual se encarga de generar los ficheros para la visualización de los resultados y también veremos el modo de visualizar esta información obtenida, esta visualización se realizará a través de ciertas herramientas como pueden ser ParaView y DIPOLE1D.

#### 9.3.1 Postprocesado de resultados

El módulo de postprocesado es el encargado de convertir los resultados obtenidos por los módulos de procesamiento y convertirlos en ficheros con ciertos formatos, los cuales facilitan la visualización de los resultados. A continuación, en la Figura 21 podemos observar el pseudocódigo de este módulo.

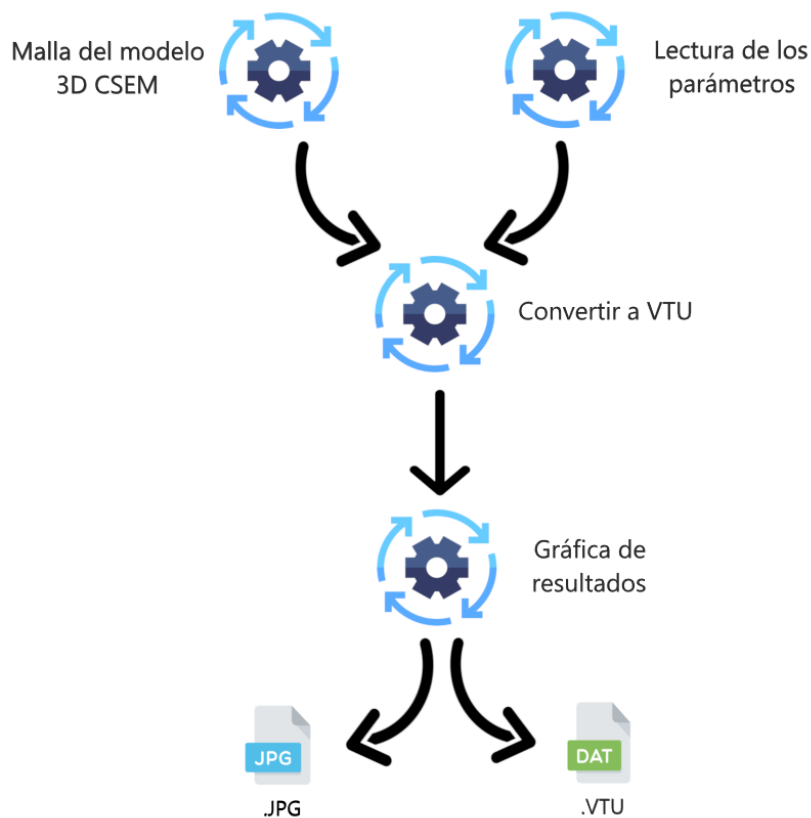


Figura 21: Pseudocódigo del módulo de postprocesado de PETGEM. Fuente: Elaboración propia.

Como podemos observar en la Figura 21 el módulo de postprocesado hace uso de la herramienta meshio para leer la información de la malla del modelo 3D CSEM que se encuentra en el fichero DIPOLE1D.msh.

Posteriormente, junto con la información de la malla y los resultados del campo electromagnético en cada uno de los puntos del modelo 3D CSEM, se procesa esta información y se convierte en un fichero VTK (específicamente en un VTU, vtkUnstructuredGrid) y posteriormente se genera una gráfica con las propiedades escogidas por el usuario, la cual se exporta en un formato JPG.

### 9.3.2 Visualización de resultados

En este apartado veremos los ficheros de salida que nos aporta PETGEM como solución a la simulación del modelo 3D CSEM introducido. Una vez que finaliza la simulación, PETGEM crea 3 fichero en el directorio de salida, estos ficheros son:

- File.vtu
- Plot.jpg
- File.h5

El fichero .h5 lo podemos visualizar con la herramienta DIPOLE1D (se puede usar cual otra herramienta compatible con el formato HDF5) y la gráfica JPG se puede visualizar con cualquier

visualizador de imagen, un ejemplo de estas visualizaciones las tenemos a continuación en la Figura 22 y en la Figura 23.

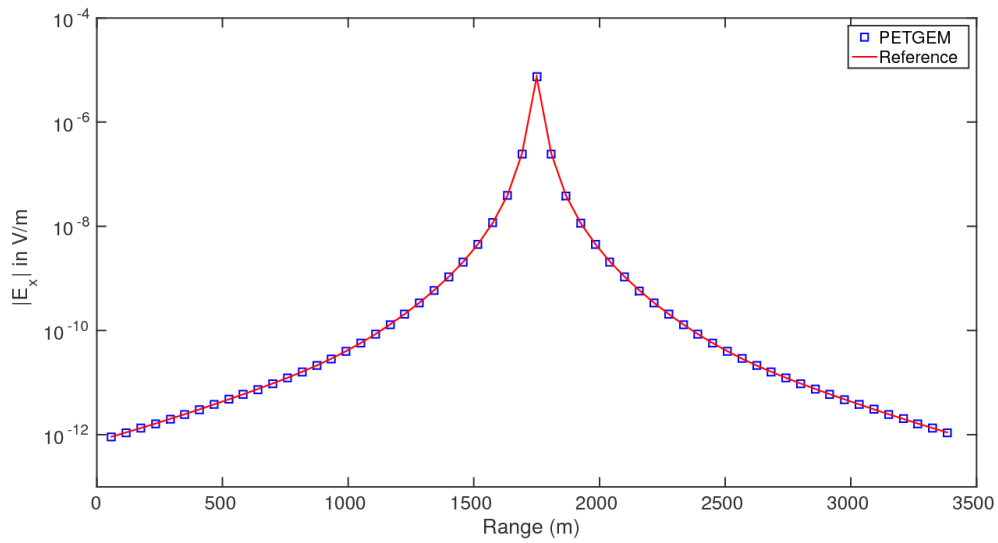


Figura 22: Comparativo de campo eléctrico total para el componente x entre PETGEM y DIPOLE1D. Fuente: (PETGEM website, 2021)

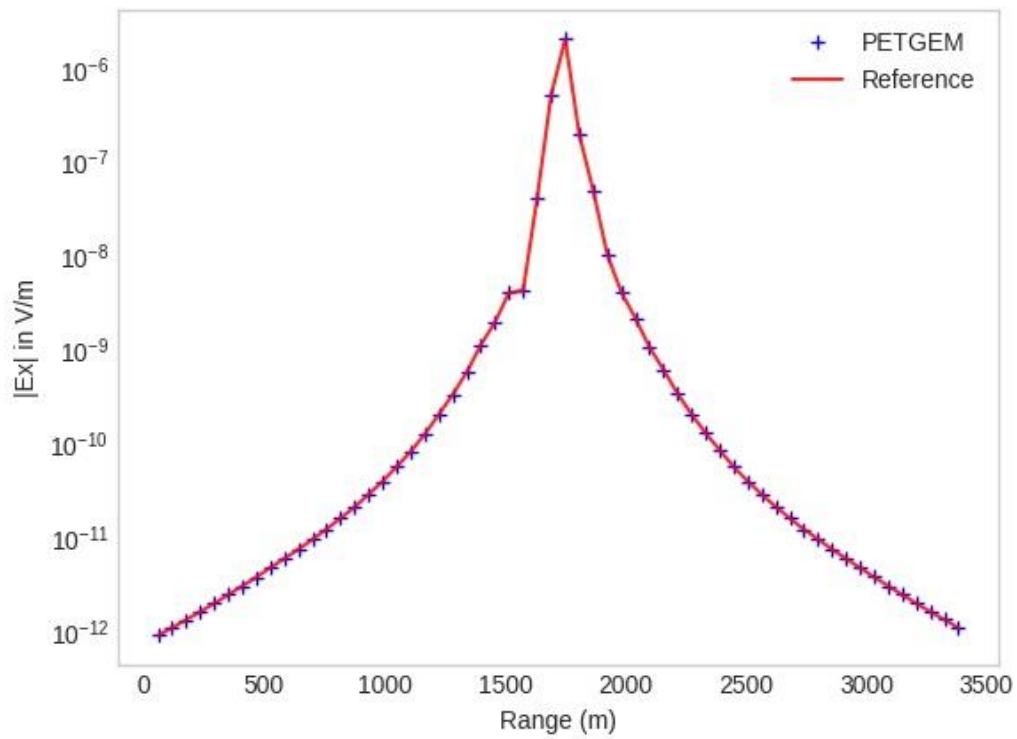


Figura 23: Visualización de la amplitud de campo eléctrico para una lista de puntos. Fuente: Elaboración propia.

### 9.3.3 ParaView

A continuación, veremos cómo el usuario visualizará los resultados del VTK a través de la herramienta ParaView. A través de la interfaz de ParaView el usuario podrá seleccionar los datos que quiere visualizar, una de las posibles visualizaciones sería visualizar solo el campo magnético de los puntos o visualizar solo el campo eléctrico en los puntos, o visualizar ambos campos en los puntos del modelo.

Un ejemplo de estas posibles visualizaciones se encuentra en la Figura 24 en la que solo se visualiza el campo eléctrico en todos los puntos del modelo, y en la Figura 25 en la que se visualiza el campo electromagnético del modelo.

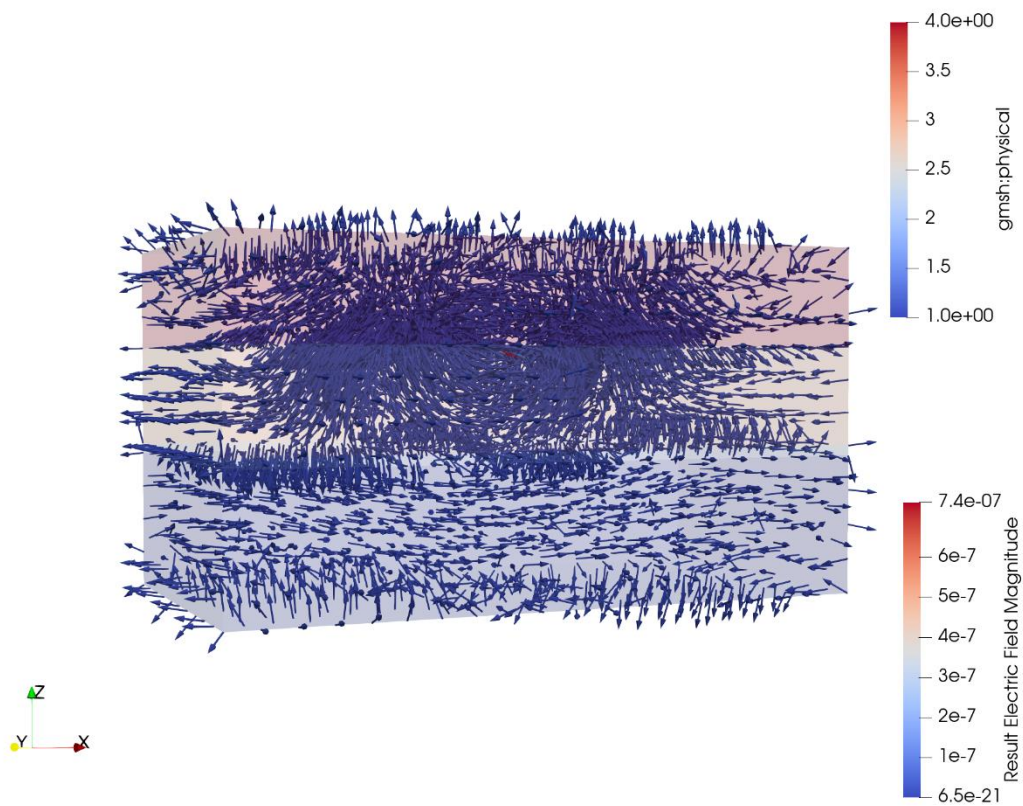


Figura 24: Visualización del resultado de PETGEM con ParaView (Campo magnético). Fuente: Elaboración propia.

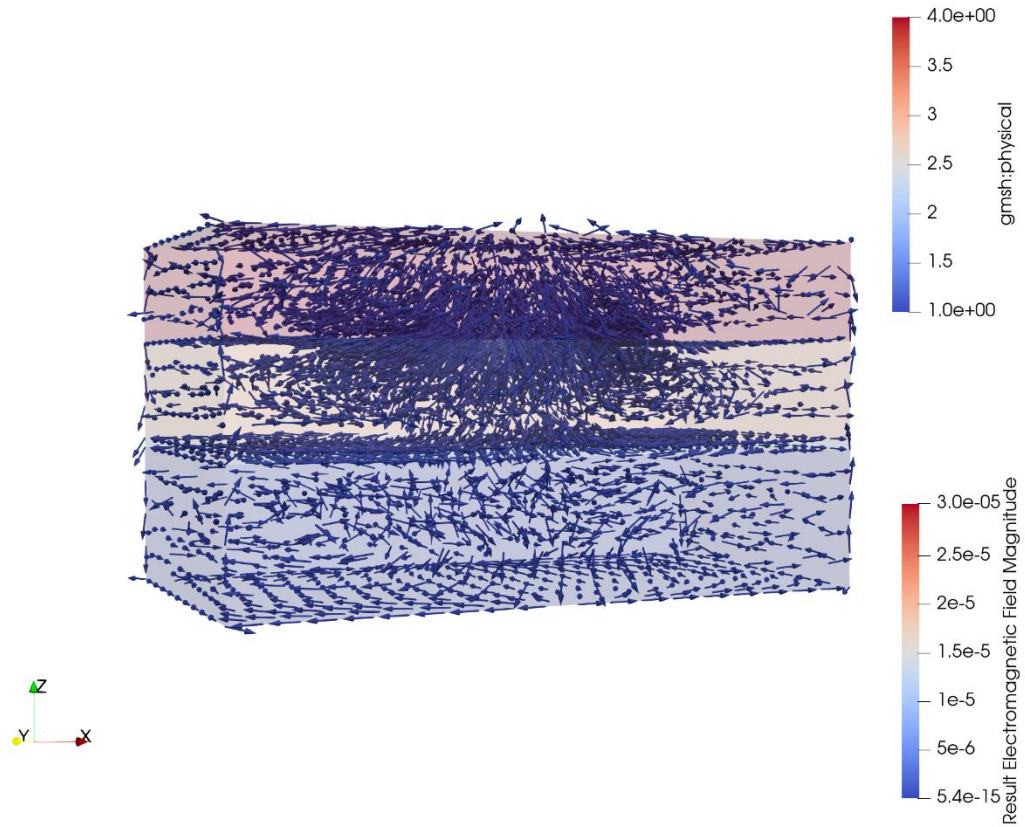


Figura 25: Visualización del resultado de PETGEM con ParaView (Campo electromagnético). Fuente: Elaboración propia.

### 9.3.4 Postprocesado rápido de nuevos resultados: validación

En la situación en la que el usuario quiere ejecutar la simulación del modelo 3D CSEM otra vez, pero esta vez variando los puntos de interés. El usuario tendrá que cambiar estos puntos y ejecutar la simulación otra vez, el proceso que conlleva esto ya lo hemos visto anteriormente en la Figura 10. Por lo tanto, la simulación se ejecuta otra vez y el módulo de postprocesado será el encargado de generar una nueva imagen (plot.JPG) con los resultados en los nuevos puntos establecidos.

# 10. Leyes y regulaciones

Este proyecto no está sujeto a ninguna ley o normativa europea. El proyecto se desarrolla bajo una licencia *open-source* (BSD-v3), la cual es la licencia actual del software PETGEM, además de cumplir con esta licencia, el proyecto debe cumplir y cumple con las regulaciones estipuladas por el proyecto PIXIL (Interreg POCTEFA) (PIXIL, 2021).

## 10.1 Licencias de software utilizadas

Con el fin de mantener la licencia actual del software PETGEM, todos los softwares y servicios de terceros que se usen en PETGEM deben de tener el mismo tipo de licencia o una licencia la cual sea compatible con la licencia *open-source* (BSD-v3).

Para mantener la licencia actual de PETGEM, en este proyecto se ha hecho uso de recursos con licencias de software de código abierto o licencias de software libre permisiva. A continuación, podemos observar las licencias que tiene cada uno de los softwares integrados en PETGEM.

- Docker: Apache License, Version 2.0 (Docker website, 2021)
- Jupyter Notebook: BSD 3-Clause License (Jupyter website, 2021)
- meshio: MIT "Expat" License (meshio website, 2021)
- ParaView: BSD 3-Clause License (ParaView website, 2021)

La licencia BSD-v3 (Open Source Initiative, 2021) es una licencia de software libre permisiva, la cual le permite al usuario del software el uso del código fuente en software no libre, con la condición de que se cumplan los tres siguientes requisitos:

- Las redistribuciones del código fuente deben conservar el aviso de copyright anterior, esta lista de condiciones y el siguiente descargo de responsabilidad.
- Las redistribuciones en formato binario deben reproducir el aviso de copyright anterior, esta lista de condiciones y el siguiente descargo de responsabilidad en la documentación y/u otros materiales suministrados con la distribución.
- Ni el nombre de los titulares de derechos de autor ni los nombres de sus colaboradores pueden usarse para apoyar o promocionar productos derivados de este software sin permiso específico previo y por escrito.

La licencia MIT "Expat" (MIT License, 2021) es una licencia de software libre permisiva, lo que significa que impone muy pocas limitaciones en la reutilización del software dentro de Software propietario, y por lo tanto posee una excelente Compatibilidad de licencia. Esta licencia permite hacer copias, modificaciones, integraciones con otro Software, publicaciones, sublicenciar o vender copias del Software, y permite a las personas a las que se les entregue el Software hacer lo mismo, con la condición de que se incluyan en todas las copias o partes sustanciales del Software la nota del copyright y los derechos que tiene el usuario del software (mencionados anteriormente).



La licencia "Apache License, Version 2.0" (Apache License, 2021) es una licencia de software libre permisiva creada por la Apache Software Foundation. Esta licencia permite al usuario del software usar el software para cualquier propósito, para distribuirlo, modificarlo y distribuir versiones modificadas del software, con la condición de que se incluya el aviso de derecho de autor y el descargo de responsabilidad.

Por consiguiente, se mantiene la licencia *open-source* (BSD-v3) en PETGEM, ya que esta es compatible con las otras dos licencias que tienen los recursos utilizados en el proyecto.

## 11. Planificación final y desviaciones

La planificación inicial del proyecto no ha tenido gran cantidad de desviaciones, por lo que la planificación final es muy similar a la inicial.

La planificación inicial planteaba tener un proyecto, en el que la duración de este sería de unos 5 meses aproximadamente, concretamente empezando el día 25 de enero y finalizando aproximadamente el día 28 de junio. La carga que tendría este proyecto sería de unas 25 horas semanales, por lo cual el proyecto constará de unas 540 horas en total.

La desviación que se ha tenido durante el desarrollo de este proyecto es la incorporación de una nueva tarea de desarrollo, la cual se centra en la creación de una imagen Docker, esta tarea tendrá una carga de trabajo de unas 20h aproximadamente. Teniendo en cuenta esta desviación se ha modificado la planificación inicial, con el fin de terminar el proyecto en la fecha establecida, cumpliendo con todos los objetivos que se habían establecido en la planificación inicial.

Con la incorporación de esta nueva tarea, la planificación final tendrá la misma fecha de finalización (aproximadamente el día 28 de junio), y contará con el mismo número de horas de trabajo semanales (25h), exceptuando una semana en la cual se duplicarán estas horas, en dicha semana, se dedicarán estas horas extras a desarrollar la implementación de la imagen Docker de PETGEM. El proyecto finalmente contará con una carga de trabajo de unas 560 h.

Además, para poder realizar esta nueva tarea de desarrollo, el proyecto debe contar con un nuevo recurso de software, el cual es el software Docker, el cual nos permite crear la imagen Docker previamente mencionada.

A continuación, en la Tabla 10 podemos observar todos los recursos necesarios para desarrollar este proyecto (en esta se destaca el nuevo recurso incorporado).

Código	Nombre	Descripción	Tareas
	RECURSOS HUMANOS		
OC	Octavio Castillo	Director del proyecto	DC2, DC3
JV	Jesús Velázquez	Desarrollador del proyecto	Todas
	SOFTWARE Y SERVICIOS		
GH	GitHub	Herramienta de gestión de versiones y repositorios.	D
O365	Office 365	Herramienta de Microsoft en la que se usa: Word, Excel, PowerPoint.	GP, DC
DR	Drive	Plataforma para almacenar archivos en la nube y compartirlos	GP, DC1, DC2, DC4, DC5
ME	Meet	Herramienta de Google para reuniones.	DC2, DC3
AT	Atenea	Plataforma para la comunicación con los profesores	GP
PyC	PyCharm	Entorno de desarrollo específicamente para el lenguaje de Python.	D
PV	ParaView	Herramientas de visualización de modelos en 3D y 2D.	D2, D3, D5, D6, D7, D8, D9, D10, DC3
PyV	PyVista	Herramientas de visualización de modelos en 3D y 2D.	D2, D11
OpGL	OpenGL	Herramientas de visualización de modelos en 3D y 2D.	D2, D11
JN	Jupyter Notebook	Es un entorno informático interactivo basado en la web para crear documentos de Jupyter Notebook.	D9
MES	meshio	Hay varios formatos de malla disponibles para representar mallas no estructuradas. meshio puede leer, escribir y realizar conversiones entre los distintos formatos de mallas	D4, D5, D7, D8, D10, D11
TA	Taiga	Plataforma de gestión de proyectos para startups, desarrolladores ágiles y diseñadores	DC
DO	Docker	Es un software que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos	D12

Tabla 10: Recursos utilizados en el proyecto. Fuente: Elaboración propia.

A continuación, se muestran resumidamente las tareas que se realizarán en este TFG (en esta se destaca la nueva tarea de desarrollo que se ha incorporado).

Código	Tarea	Duración (horas)	Dependencias	Recursos
GP	GESTIÓN DEL PROYECTO	Total: 65		
GP1	Alcance	25	-	JE, Drive, Word, Atenea
GP2	Planificación	10	GP1	JE, Drive, Word, Atenea
GP3	Gestión económica y de sostenibilidad	10	GP2	JE, Drive, Word, Excel, Atenea
GP4	Integración final del proyecto	20	GP3	JE, Drive, Word, Atenea
D	DESARROLLO	Total: 360		
D1	Especificación de los requisitos	20	-	JE, PyC, GH
D2	Preparación del entorno	10	D1	JE, PyC, GH, PV, PyV, OpenGL
D3	Familiarizarse con VTK y ParaView	40	D2	JE, PyC, PV
D4	Definir la estructura del modelo en función de los datos de entrada	30	D1, D2, D3	JE, PyC, GH
D5	Escritura de los resultados en paralelo con la ejecución	40	D4	JE, PyC, GH, PV
D6	Visualización de la simulación en 2D en ParaView	30	D4	JE, PyC, GH, PV
D7	Visualización de la simulación en 3D en ParaView	50	D4	JE, PyC, GH, PV
D8	Funcionalidad de cambiar los puntos de interés en el cálculo del campo electromagnético.	50	D5, D6, D7	JE, PyC, GH, PV
D9	Entorno de ejecución en Jupyter Notebook	20	D8	JE, PyC, GH, PV, JN
D10	Realización de los test para ParaView	35	D6, D7, D8	JE, PyC, GH, PV
D11	Realización de los test de compatibilidad con otra herramienta	15	D5, D7	JE, PyC, GH, PyV, OpenGL
D12	Creación de una imagen Docker	20	D9	JE, PyC, JN, DO
DC	DOCUMENTACIÓN Y COMUNICACIÓN	Total: 135		

DC1	Seguimiento y control del proyecto	-	-	JE, Taiga, Trello, Drive, Word
DC2	Documentación de la fase de seguimiento	30	GP4	JE, OC, Drive, Word, Taiga, Trello, Meet, Gmail
DC3	Comunicación	15	-	JE, OC, Word, PyC, PV, Meet, Trello, Gmail
DC4	Documentación de la fase final	60	CD2	JE, Drive, Word, Taiga, Trello, Excel
DC5	Preparación de la presentación final	30	CD4	JE, Drive, Taiga, PowerPoint
	TOTAL	560		

Tabla 11: Resumen de las tareas del TFG. Fuente: Elaboración propia.

Descripción de la nueva tarea de desarrollo:

- **D12 → Creación de una imagen Docker (20 h):** Se creará una imagen Docker la cual se publicará para que cualquier usuario que desee usar la herramienta de PETGEM lo pueda hacer desde cualquier sistema operativo, y además no tendrá la necesidad de instalar las dependencias necesarias para ejecutar PETGEM, en esta imagen ya se contará con las dependencias requeridas en PETGEM. Dependencias: D9.

# 11.1 Diagrama de Gantt

## 11.1.1 Planificación inicial

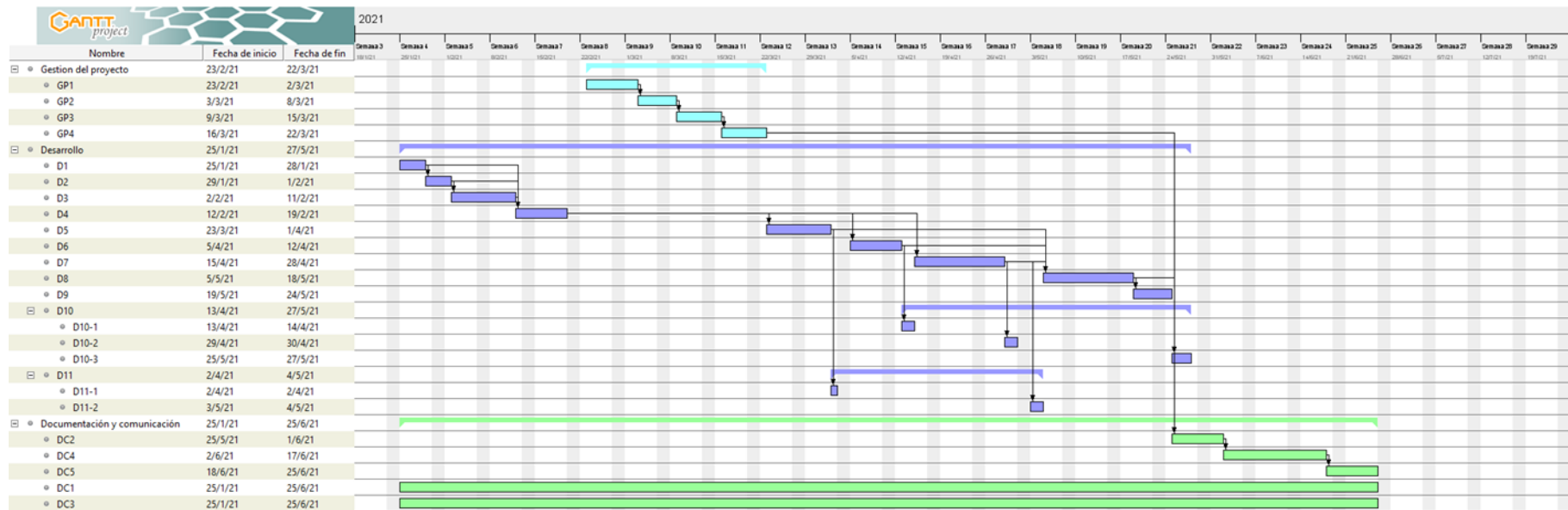


Figura 26: Diagrama de Gantt de la planificación inicial. Fuente: Elaboración propia.

## 11.1.2 Planificación final

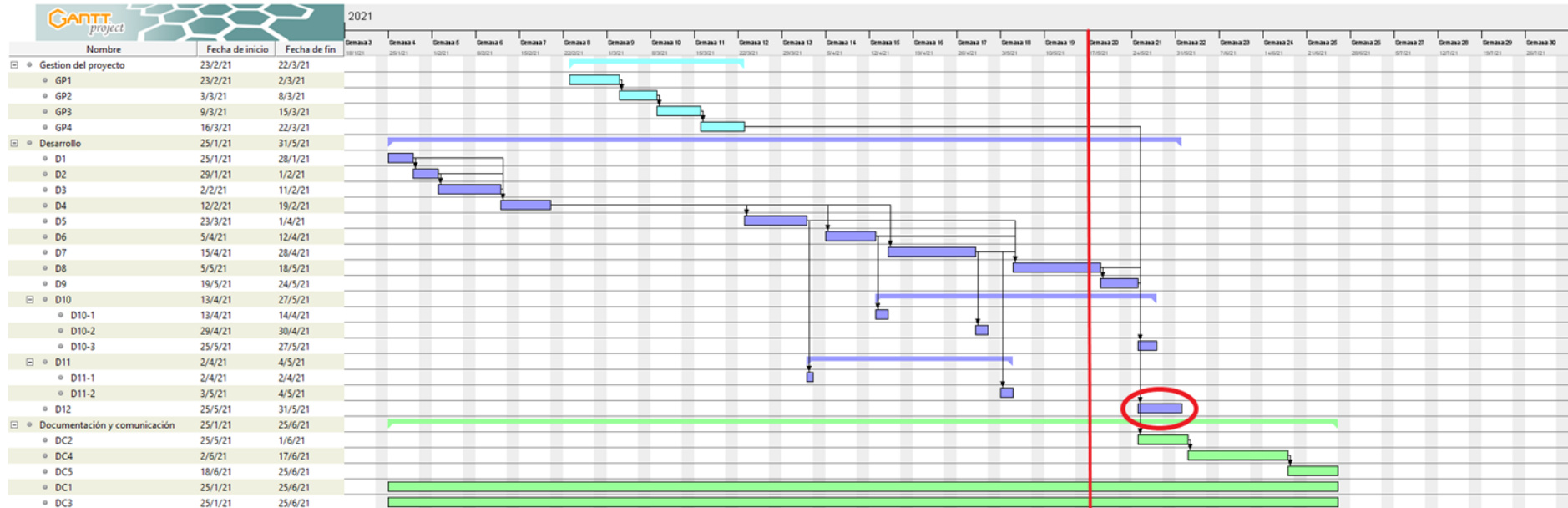


Figura 27: Diagrama de Gantt de la planificación final. Fuente: Elaboración propia.

### Leyenda:

- Círculo rojo: indica la nueva tarea de desarrollo (D12) que se ha incorporado en la planificación final.
- Línea roja: indica la etapa actual del proyecto.

## 11.2 Desviaciones en los objetivos y presupuesto del proyecto

La creación de esta nueva tarea no afectará a los objetivos del proyecto ni al presupuesto del proyecto. Esta desviación no afecta al presupuesto del proyecto, ya que el coste de completar esta tarea es de 234€ y el proyecto cuenta con un presupuesto de contingencia de 830€, por lo cual no se incrementa el coste del proyecto. En lo que afecta esta desviación es en la carga de trabajo del proyecto, por lo que se tendrán que realizar 20h más de las que se habían planificado, esto se puede ver en la Figura 27.

En cuanto a los objetivos del proyecto, esta desviación no los modificará, ya que la implementación de la nueva tarea de desarrollo, no implicará no cumplir un objetivo o el tener que reducir el nivel de un objetivo, sino que todo lo contrario, con la implementación de esta nueva tarea, lo que se consigue es afianzar el requisito no funcional de la usabilidad, ya que con la implementación de la imagen Docker, añadirá una facilidad a los usuarios a la hora de poder ejecutar la herramienta de PETGEM desde cualquier sistema operativo, y sin la necesidad de preocuparse de tener que contar con las dependencias necesarias para poder ejecutar las simulaciones de PETGEM.

## 11.3 Metodología de trabajo final

La metodología del proyecto no ha sido modificada durante el desarrollo de este. La metodología que se está siguiendo para realizar este proyecto es la metodología agile, en concreto la metodología Kanban. Esta metodología de trabajo se mantiene, debido a que es la más adecuada para el proyecto, porque se enfoca en el desarrollo continuo, trabajando en un número reducido de tareas de manera fluida y simultánea, y además nos permite adaptarnos de una forma más eficaz a las posibles desviaciones que surjan en el desarrollo del proyecto, como hemos podido ver anteriormente con la incorporación de una nueva tarea de desarrollo.

El sistema de gestión del proyecto sigue siendo el mismo, se mantiene el tablero Kanban que está dividido en cuatro columnas (To-do, in progress, test y Done), las cuales representarán el estado actual de cada historia de usuario, y además cada vez que se finaliza una historia de usuario se comprueba la planificación del proyecto, con el fin de tomar las medidas necesarias para cumplir el plazo de entrega establecidos.

## 11.4 Etapa actual del proyecto

Como podemos observar en la Figura 27, actualmente el proyecto se encuentra en la semana 20 de la planificación final, en la cual queda por finalizar cuatro tareas de desarrollo (D8, D9, D10 y D12), pero esto no supone una desviación en la planificación final, porque aún hay planificadas dos semanas más de trabajo para las tareas de desarrollo.

El proyecto está muy cerca de su etapa final, especialmente porque para la entrega final del proyecto quedan 6 semanas, en las que hay alrededor de 150 horas de carga de trabajo de las 560 horas totales del proyecto.

# 12. Sostenibilidad

Al realizar la encuesta de EDINSOST sobre el nivel de conocimiento en el ámbito de la sostenibilidad, he podido reflexionar sobre los conocimientos que tengo sobre el ámbito de la sostenibilidad. Me he dado cuenta de que aún me queda por aprender sobre este tema, sobre todo en la dimensión social.

En cuanto a la dimensión económica, he podido analizar que tengo algunos conocimientos generales sobre el proceso de gestión de un proyecto, como puede ser, las diferentes partes económicas: amortización, costes fijos, costes variables, etc., pero son conocimientos muy generales. Y por otra parte me falta aprender las técnicas de planificación de proyectos, la economía social y la economía del bien común.

En el ámbito medioambiental, comprendo los costes y efectos ambientales que tienen los productos relacionados con las TIC a lo largo de su ciclo de vida, pero en el momento de medir el impacto medioambiental, conozco algunos de los indicadores de medición, pero no conozco en profundidad todos los indicadores que se usan en la actualidad.

Por último, en el ámbito social, me sucede lo mismo que en el ámbito medioambiental, y es que conozco las ventajas de las herramientas de colaboración y que el software ha de ser ético, transparente y accesible, pero a la hora de realizar las mediciones, no tengo los conocimientos necesarios sobre los indicadores que se tiene que usar para medir y comparar los resultados en este ámbito.

## 12.1 Dimensión económica

Al analizar la viabilidad de un proyecto, es muy importante considerar su presupuesto, es por eso por lo que se ha realizado un presupuesto detallado con los recursos humanos, generales (materiales), de contingencia y de imprevistos, con el fin de poder estimar de una forma más precisa el costo de la realización del proyecto.

Como se ha mencionado anteriormente, el principal problema actual que tiene PETGEM reside a la hora de visualizar los resultados de la simulación del modelo 3D CSEM, es por eso, que con la implementación de las nuevas funcionalidades que se quieren desarrollar en este proyecto, se solucionara este problema ofreciendo una nueva forma de visualizar los resultados.

En cuanto a la mejora económica, los usuarios podrán aumentar su eficiencia, ya que con las nuevas funcionalidades del software se le facilitará la visualización de los resultados y las ejecuciones de las simulaciones, además al ser un software de código abierto (*open-source*) no les proporcionará un coste a los usuarios.



## 12.2 Dimensión ambiental

Al ser una solución software, el proyecto no tiene un gran impacto medioambiental. Sin embargo, se ha estimado el impacto que tendrán los recursos utilizados en el desarrollo del proyecto, como pueden ser el portátil y el suministro de la electricidad contratada.

Sin embargo, el impacto de estos recursos se puede minimizar. Por ejemplo, el ordenador que se utilizará es de uso personal y además se lleva usando varios años, por lo que se evita el coste medioambiental que se genera al sustituir un ordenador. Otra forma de minimizar el impacto ambiental del proyecto es contratar servicios energéticos de fuentes renovables.

Este software mejorará la capacidad de reproducibilidad y repetibilidad en el campo de la ciencia de las energías renovables, especialmente en el campo de la energía geotérmica. Por lo tanto, mientras se tengan más mejoras en las funcionalidades del software, se mejorará la comprensión en el campo de la geotermia, lo que conducirá a mejoras en este campo, que a su vez producirá mejoras en el medio ambiente, ya que se trata de un tipo de energía renovable.

Además, con las nuevas funcionalidades del software, las ejecuciones de las simulaciones se harán de una forma más eficiente, por lo que se ahorra en el consumo de energía.

## 12.3 Dimensión social

A nivel personal el desarrollo de este proyecto me proporciona conocimientos y experiencia en diversos campos, que incluyen no solo conocimientos en el desarrollo de software y gestión de proyectos (de una forma general), sino también en el ámbito de herramientas de exploración de hidrocarburos, que son softwares que se usan para el cálculo de los campos electromagnéticos en una estructura geológica determinada.

Como ya hemos mencionado anteriormente, el desarrollo de este proyecto mejorará la capacidad de reproducibilidad y repetibilidad en el campo de la ciencia de las energías renovables, especialmente en el campo de la energía geotérmica. Con la mejora de las funciones del software, los científicos podrán reproducir datos de una forma más sencilla, de modo que los datos obtenidos puedan ser verificados directamente de forma más rápida y sencilla.

El software mejorará indirectamente la calidad de vida social, pues gracias a las mejoras que aportará en el ámbito de la geotermia, se producirán mejoras, que beneficiarán la calidad de vida en nuestra sociedad.

Por otra parte, el proyecto tiene una necesidad real de mejorar la experiencia de usuario que se tiene en la visualización de los resultados de los modelos 3D CSEM.

# 13. Conclusiones y trabajo futuro

## 13.1 Conclusiones

PETGEM es un modelador de datos electromagnéticos capaz de resolver problemas realistas en el campo de la geofísica de exploración. Sin embargo, la versión actual de PETGEM no cuenta con módulos de postprocesado para la visualización de resultados de simulación, es por esto por lo que en este proyecto de TFG se han realizado mejoras en el software, por lo que PETGEM aumentará su valor ofreciendo a los usuarios una mayor facilidad para interactuar con el software.

Las conclusiones que podemos sacar del desarrollo de este proyecto pueden establecerse como satisfactorias. Todos los objetivos propuestos al inicio y durante el desarrollo del proyecto, han sido cumplidos a cabalidad.

Un ejemplo de esto es la implementación de un módulo de postprocesado, con el que se ha conseguido generar un fichero de salida en formato VTK, con los resultados de la simulación de PETGEM, los cuales se pueden interpretar usando la herramienta ParaView o cualquier otra herramienta de visualización compatible con este modelo, ofreciéndole la posibilidad al usuario de utilizar varias herramientas para interpretar los datos. Estas visualizaciones se pueden hacer tanto en 2D como en 3D.

Además, para hacer la ejecución de PETGEM más eficiente aún, se ha añadido una nueva funcionalidad que le permite al usuario realizar un postprocesado rápido de nuevos resultados de un modelo 3D CSEM previamente ejecutado.

Con estas mejoras conseguimos que nuestros usuarios, tengan mayor facilidad para la interpretación de los resultados y poder realizar más simulaciones en un menor tiempo.

En este proyecto no solo se ha trabajado en la eficiencia y en la construcción de nuevas funcionalidades para el software, sino que también se ha tenido en cuenta la interacción que tiene el usuario con el software. Y es por esto por lo que también se han realizado mejoras en estas interacciones.

Un ejemplo de esto es la integración que se tiene con la interfaz de Jupyter Notebook, desde donde el usuario puede realizar todas las modificaciones previo a la ejecución de PETGEM y la propia ejecución de PETGEM usando una interfaz gráfica.

Además de esto también se consideró la opción de facilitar la instalación de PETGEM a sus usuarios, sin importar qué sistema operativo estén usando. Por ello, se desarrolló e implementó una imagen Docker, la cual contiene todo lo necesario para ejecutar PETGEM y posteriormente poder visualizar los resultados desde el propio ordenador del usuario.

Cabe mencionar que ParaView es la única dependencia rígida para explotar las capacidades del Docker, esto se debe a que no se ha podido instalar ParaView dentro de la imagen Docker. A pesar de esta limitación, la interacción que tiene el usuario con esta imagen no se ve perjudicada,

ya que el usuario tiene la facilidad de compartir datos desde su ordenador a la imagen Docker, por lo que esta interacción con PETGEM sigue siendo positiva y funcional.

## 13.2 Trabajo futuro

Como trabajo futuro destaca la redacción de un artículo científico que describa el desarrollo, los resultados y conclusiones más relevantes de este TFG. Más concretamente, el hilo conductor del artículo versará sobre el módulo de visualización, postprocesado rápido de nuevos resultados, la interfaz de Jupyter Notebook y la instalación y ejecución con Docker. Este artículo se enviará a un *Journal* arbitrado de alto impacto.

# 14. Referencias

- Apache License. (2021). *Apache License*. Obtenido de <https://github.com/moby/moby/blob/master/LICENSE>
- BENQ. (2021). *Monitor BENQ*. Obtenido de <https://www.benq.eu/es-es/monitor/home-office/gw2480.html>
- Castillo Reyes, O. (2018). *PETGEM*. Obtenido de <https://upcommons.upc.edu/handle/2117/189344>
- Castillo Reyes, O. (2021). *UPCommons*. Obtenido de Portal de acceso abierto al conocimiento de la UPC: <https://upcommons.upc.edu/handle/2117/113681?show=full>
- Docker website. (2021). *Docker website*. Obtenido de <https://docs.docker.com/engine/>
- Geología web. (2021). *Método electromagnético (EM)*. Obtenido de [https://geologiaweb.com/geofisica/metodo-electromagnetico/#:~:text=El%20m%C3%A9todo%20electromagn%C3%A9tico%20\(EM\)%20hace,y%20una%20fuerza%20de%20magnetizaci%C3%B3n.&text=Las%20corrientes%20de%20Foucault%20generan,secundario%20que%20viaja%20al%20receptor](https://geologiaweb.com/geofisica/metodo-electromagnetico/#:~:text=El%20m%C3%A9todo%20electromagn%C3%A9tico%20(EM)%20hace,y%20una%20fuerza%20de%20magnetizaci%C3%B3n.&text=Las%20corrientes%20de%20Foucault%20generan,secundario%20que%20viaja%20al%20receptor)
- glassdoor. (2021). *Buscar sueldos y remuneración*. Obtenido de <https://www.glassdoor.es/Sueldos/index.htm>
- Gmsh. (2021). *A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. Obtenido de <https://gmsh.info/>
- Interreg POCTEFA. (2021). *¿Qué es POCTEFA?* Obtenido de <https://www.poctefa.eu/programa/que-es-poctefa/>
- Jupyter website. (2021). *Jupyter website*. Obtenido de <https://jupyter.org/about>
- MEDTEC. (2021). *VTK*. Obtenido de <https://mt4sd.ulpgc.es/slicer-int/index.php/VTK>
- Meet BCN. (2021). *Meet BCN: Tu espacio de Coworking*. Obtenido de <https://meetbcn.com/>
- meshio website. (2021). *meshio website*. Obtenido de <https://pypi.org/project/meshio/>
- MIT License. (2021). *MIT License*. Obtenido de <https://opensource.org/licenses/MIT>
- MSI. (2021). *Prestige 15*. Obtenido de <https://es.msi.com/Business-Productivity/Prestige-15-A11X/Specification>
- Open Source Initiative. (2021). *The 3-Clause BSD License*. Obtenido de <https://opensource.org/licenses/BSD-3-Clause>
- ParaView website. (2021). *ParaView website*. Obtenido de <https://www.paraview.org/paraview-license/>
- PETGEM website. (2021). *PETGEM website*. Obtenido de <https://petgem.bsc.es/>
- PIXIL. (2021). *Pyrenees Imaging eXperience: an International network*. Obtenido de <https://pixil-project.eu/>

Wikipedia. (2021). *Computación de alto rendimiento*. Obtenido de [https://es.wikipedia.org/wiki/Computaci%C3%B3n\\_de\\_alto\\_rendimiento](https://es.wikipedia.org/wiki/Computaci%C3%B3n_de_alto_rendimiento)

Wikipedia. (8 de febrero de 2021). *Hierarchical Data Format*. Obtenido de [https://en.wikipedia.org/wiki/Hierarchical\\_Data\\_Format](https://en.wikipedia.org/wiki/Hierarchical_Data_Format)