



UNIVERSIDAD  
DE GRANADA

TRABAJO FIN DE GRADO  
INGENIERÍA INFORMÁTICA

# Dockerización de Aplicación Paralela y Distribuida para Clasificación de EEGs: Análisis de Viabilidad y Rendimiento

---

DockEEG

**Autor**

Fernando Cuesta Bueno

**Director**

Juan José Escobar Pérez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

---

Granada, septiembre de 2025







UNIVERSIDAD  
DE GRANADA

# DockEEG

---

Dockerización de Aplicación Paralela y Distribuida para  
Clasificación de EEGs: Análisis de Viabilidad y Rendimiento

**Autor**

Fernando Cuesta Bueno

**Director**

Juan José Escobar Pérez



# **DockEEG: Dockerización de Aplicación Paralela y Distribuida para Clasificación de EEGs: Análisis de Viabilidad y Rendimiento**

Fernando Cuesta Bueno

**Palabras clave:** palabra\_clave1, palabra\_clave2, palabra\_clave3, .....

## **Resumen**

Poner aquí el resumen.





# **DockEEG: Dockerization of a Parallel and Distributed Application for EEG Classification: Feasibility and Performance Analysis**

Fernando Cuesta Bueno

**Keywords:** Keyword1, Keyword2, Keyword3, ....

## **Abstract**

Write here the abstract in English.



---

Yo, **Fernando Cuesta Bueno**, alumno de la titulación Graduado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77150866B, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Fernando Cuesta Bueno

Granada a 5 de septiembre de 2025.



---

D. **Juan José Escobar Pérez**, Profesor del Área de XXXX del Departamento de Lengua de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado ***Título del proyecto, Subtítulo del proyecto***, ha sido realizado bajo su supervisión por **Nombre Apellido1 Apellido2 (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 202 .

**Los directores:**

**Nombre Apellido1 Apellido2 (tutor1)**      **Nombre Apellido1 Apellido2 (tutor2)**



# Agradecimientos

Poner aquí agradecimientos...





# Índice general

<b>Acrónimos</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Objetivos . . . . .	3
1.2.1. Objetivos específicos . . . . .	3
<b>2. Gestión del Proyecto</b>	<b>5</b>
2.1. Tareas . . . . .	5
2.2. Planificación temporal . . . . .	9
2.3. Estimación de costes . . . . .	10
<b>3. Estado del arte</b>	<b>13</b>
3.1. Sección . . . . .	13
3.1.1. Sub-seccion . . . . .	13
<b>4. Propuesta principal</b>	<b>15</b>
4.1. Background . . . . .	15
4.2. Herramientas utilizadas . . . . .	15
4.3. Implementación . . . . .	15
<b>5. Experimentación</b>	<b>17</b>
5.1. Setup experimental . . . . .	17
5.1.1. Escenario 1: bla bla bla ... . . . .	17
5.2. Resultados y discusión . . . . .	17
<b>6. Conclusiones y trabajo futuro</b>	<b>19</b>
6.1. Contribuciones . . . . .	19
6.2. Retos y trabajo futuro . . . . .	19
<b>7. Bibliografía</b>	<b>21</b>
<b>A. Anexo A</b>	<b>23</b>



# Índice de figuras

2.1. Diagrama de Gantt del proyecto . . . . .	9
---	---



# Índice de tablas

2.1. Planificación temporal de tareas y horas estimadas . . . . .	9
2.2. Costes estimados de hardware para el proyecto . . . . .	11
2.3. Costes estimados de recursos humanos para el proyecto . . .	11
2.4. Coste total estimado del proyecto . . . . .	11



# Acrónimos

[Ejemplo de introducción y referencia de acrónimos Programmable Logic Controller (PLC). Para introducir nuevos acrónimos, ir al fichero `acro_list.tex`]

**PLC** Programmable Logic Controller





# Capítulo 1

## Introducción

### 1.1. Motivación

### 1.2. Objetivos

Analizar la viabilidad y las limitaciones del uso de contenedores, concretamente Docker, para encapsular y ejecutar aplicaciones de alto rendimiento (HPC) en arquitecturas heterogéneas modernas —como big.LITTLE— y entornos multiplataforma, con el fin de facilitar su portabilidad, uso y adopción por parte de la comunidad científica.

#### 1.2.1. Objetivos específicos

- **OB1.** Investigar el estado del arte en el ámbito de la tecnología de contenedores y su aplicación en entornos de computación de alto rendimiento.
- **OB2.** Diseñar e implementar un conjunto de experimentos para evaluar el rendimiento de aplicaciones HPC contenerizadas en diferentes arquitecturas y plataformas.
- **OB3.** Comparar el rendimiento de las aplicaciones contenerizadas con su ejecución nativa en diferentes entornos y arquitecturas, identificando las ventajas y desventajas de cada enfoque.
- **OB4.** Analizar los resultados obtenidos en los experimentos para identificar las limitaciones y desafíos asociados al uso de contenedores en entornos HPC, así como establecer futuras líneas de investigación.



## Capítulo 2

# Gestión del Proyecto

### 2.1. Tareas

#### Tareas de planificación del proyecto

- **Definición de objetivos y alcance del proyecto**  
Establecer claramente los objetivos principales y específicos del proyecto, así como el alcance y las limitaciones.
- **Planificación temporal**  
Desarrollar un cronograma detallado que incluya todas las fases del proyecto, desde la investigación inicial hasta la redacción final del informe.
- **Asignación de recursos**  
Identificar y asignar los recursos necesarios, tanto humanos como materiales, para llevar a cabo el proyecto de manera eficiente.
- **Gestión de riesgos**  
Identificar posibles riesgos que puedan afectar al desarrollo del proyecto y establecer planes de contingencia para mitigarlos.
- **Comunicación y seguimiento**  
Establecer canales de comunicación efectivos entre todos los miembros del equipo y realizar un seguimiento regular del progreso del proyecto para asegurar que se cumplen los plazos establecidos.

#### Tareas del OB1 (Estudio del estado del arte)

- **Revisión del estado del arte en HPC**  
Estudiar la evolución histórica y tendencias actuales en la computación de alto rendimiento.

- **Análisis del uso de contenedores en HPC**

Revisar tecnologías de contenedores aplicadas a entornos científicos y de alto rendimiento. Comparar contenedores frente a máquinas virtuales en cuanto a eficiencia, overhead y portabilidad en HPC.

- **Estudio del uso de GPU en aplicaciones HPC**

Revisar el papel de las GPUs en la aceleración de aplicaciones científicas y de ingeniería. Identificar librerías y frameworks para programación en GPU. Analizar casos de éxito en la integración de GPU en entornos HPC.

- **Investigación sobre el soporte de GPU en contenedores**

Revisar soluciones actuales para ejecutar GPUs dentro de contenedores. Analizar el grado de compatibilidad con diferentes sistemas operativos y arquitecturas. Estudiar el impacto en rendimiento del uso de GPUs en entornos contenerizados en comparación con la ejecución nativa.

### **Tareas del OB2 (Diseño e implementación de la propuesta)**

- **Selección de la aplicación o problema HPC a estudiar**

Se seleccionará una aplicación representativa del ámbito HPC, justificando su elección en función de su relevancia científica, disponibilidad de código abierto y viabilidad técnica para su ejecución en diferentes plataformas y entornos contenerizados.

- **Preparar entornos y dependencias**

Se identificarán y documentarán las librerías y herramientas necesarias, incluyendo MPI, OpenMP y CUDA. Se garantizará la homogeneidad de las configuraciones en todos los sistemas de prueba y se detallarán los requisitos específicos para cada plataforma (Linux, Windows, macOS).

- **Diseñar y construir imágenes de contenedor**

Se desarrollarán Dockerfiles reproducibles que incluyan todas las dependencias necesarias, asegurando soporte para GPU mediante NVIDIA Container Toolkit. Las imágenes serán versionadas y almacenadas en un registro para facilitar su reutilización y trazabilidad.

- **Definir casos de prueba y parámetros de ejecución**

Se establecerán experimentos mononodo variando el número de hebras, experimentos multinodo con diferentes cantidades de nodos y casos combinados que exploren el espacio de parámetros hebras  $\times$  nodos.

- **Automatización y orquestación**

Se implementarán scripts en para automatizar la ejecución de lotes de pruebas, así como la recogida y almacenamiento de logs y resultados.

- **Instrumentación y métricas**

Se instrumentará la aplicación para medir tiempos totales de ejecución, uso de CPU y otros recursos. Se calcularán métricas como aceleración, eficiencia, throughput y overhead comparando la ejecución en contenedor frente a la nativa. Se generarán gráficos comparativos para el análisis de resultados.

- **Reproducibilidad y trazabilidad**

Se mantendrá un repositorio con los Dockerfiles, scripts y documentación del proyecto. Se etiquetarán las versiones de las imágenes y dependencias para asegurar la reproducibilidad de los experimentos.

### Tareas del OB3 (Evaluación de rendimiento)

- **Definición de criterios de comparación**

Se establecerán las métricas principales para la comparación de rendimiento y se garantizará la paridad de configuraciones (versiones de compiladores, librerías, drivers).

- **Ejecución de pruebas comparativas**

Se ejecutarán las mismas baterías de experimentos tanto en modo nativo como en contenedor. Se registrarán logs completos de cada ejecución.

- **Recopilación y organización de resultados**

Se guardarán los tiempos de ejecución y métricas de uso de recursos, clasificando los datos según plataforma (Linux, Windows, macOS) y tipo de acelerador (CPU, GPU). Se establecerá un formato homogéneo para los ficheros de resultados (CSV o base de datos).

- **Visualización de resultados comparativos**

Se generarán gráficos y tablas que destaquen los casos extremos (mejores y peores comportamientos), facilitando la interpretación de los resultados.

### Tareas del OB4 (Análisis de resultados)

- **Revisión sistemática de los resultados experimentales**

Se analizarán de manera estructurada los datos obtenidos en las pruebas, comparando el rendimiento entre ejecución nativa y contenerizada en las distintas plataformas (Linux, Windows, macOS) y ante el uso o

no de aceleradores (CPU, GPU). Se identificarán tendencias generales, anomalías y comportamientos consistentes.

- **Análisis cuantitativo del rendimiento**

Se calcularán diferencias absolutas y relativas entre ejecución nativa y contenerizada, estimando overheads medios y por caso. Se evaluará la escalabilidad en cada escenario y se aplicará análisis estadístico para validar la significancia de las diferencias observadas.

- **Análisis cualitativo**

Se identificarán ventajas no estrictamente de rendimiento, como portabilidad, reproducibilidad y facilidad de despliegue, así como limitaciones observadas relacionadas con drivers de GPU, gestión de red en contenedores y problemas de compatibilidad.

- **Detección de desafíos en la adopción de contenedores en HPC**

Se evaluará la complejidad asociada a la configuración, despliegue y mantenimiento de entornos contenerizados en HPC, incluyendo la integración de aceleradores como GPU y la gestión de dependencias específicas.

- **Propuesta de líneas de investigación futura**

A partir de los resultados y desafíos identificados, se propondrán posibles líneas de trabajo futuro.

### Tareas de redacción del informe final

- **Estructuración del informe**

Se definirá un esquema claro y lógico para el informe, asegurando que cada sección fluya de manera coherente hacia la siguiente.

- **Redacción de secciones técnicas**

Se describirán detalladamente los métodos, experimentos y resultados obtenidos, utilizando un lenguaje técnico preciso y adecuado para la audiencia objetivo.

- **Incorporación de gráficos y tablas**

Se incluirán visualizaciones que apoyen y clarifiquen los puntos clave del informe, asegurando que estén correctamente etiquetadas y referenciadas en el texto.

- **Revisión y edición**

Se realizará una revisión exhaustiva del informe para corregir errores gramaticales, mejorar la claridad y coherencia del texto, y asegurar que todas las referencias bibliográficas estén correctamente citadas.

- **Formato y presentación**  
Se aplicarán las normas de formato establecidas por la institución o publicación a la que se destina el informe, asegurando una presentación profesional y uniforme.

2.2. Planificación temporal

En la tabla 2.1 se presenta una estimación del tiempo necesario para completar cada una de las tareas principales del proyecto, desglosado en horas dedicadas por el desarrollador y el tutor.

Tarea	Desarrollador (h)	Tutor (h)
Planificación	20	5
Estado del arte	40	10
Implementación	85	8
Evaluación	55	7
Análisis	30	5
Informe	30	15
Total	260	50

Tabla 2.1: Planificación temporal de tareas y horas estimadas

En la imagen 2.1 se muestra un diagrama de Gantt que ilustra la distribución temporal de las tareas a lo largo del proyecto.

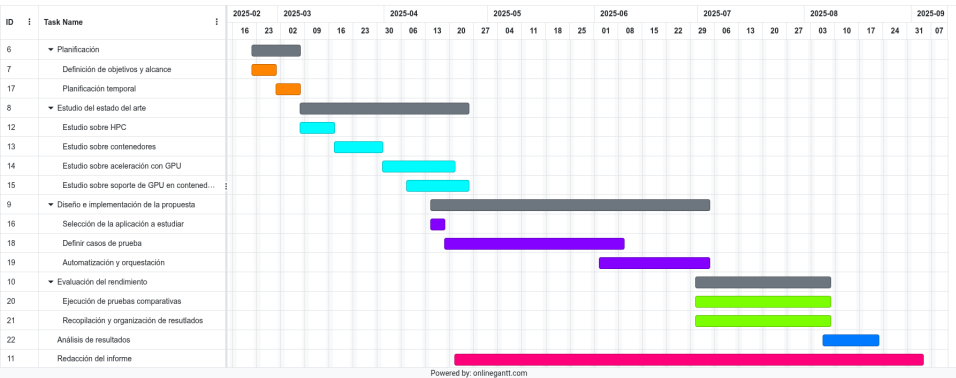


Figura 2.1: Diagrama de Gantt del proyecto

## 2.3. Estimación de costes

A continuación, se detallan los recursos necesarios para llevar a cabo el proyecto, incluyendo hardware, software y recursos humanos, junto con una estimación de los costes asociados.

### Hardware

- Ordenador portátil LG Gram 14Z90Q-G.AA75B, este equipo se utilizará para el desarrollo general del trabajo: creación del código para las pruebas, gestión de las pruebas en el clustery creación de la memoria. Cuenta con un procesador Intel Core i7-1260P, 16 GB de RAM y 512 GB de almacenamiento SSD.
- Ordenador portátil Lenovo Legion 5, utilizado como plataforma principal para la ejecución de pruebas de rendimiento, especialmente aquellas que requieren aceleración por GPU. Este equipo está equipado con un procesador AMD Ryzen 7 4800H (8 núcleos y 16 hilos), 16 GB de memoria RAM DDR4, 512 GB de almacenamiento SSD NVMe y una tarjeta gráfica dedicada NVIDIA GeForce RTX 2060 con 6 GB de VRAM GDDR6. La presencia de la GPU NVIDIA permite la ejecución y evaluación de aplicaciones HPC que hacen uso de CUDA, así como la integración y validación del soporte de GPU en entornos contenerizados mediante NVIDIA Container Toolkit. Además, este equipo facilita la comparación de resultados entre ejecución nativa y contenerizada en escenarios reales de computación de alto rendimiento.
- Ordenador Apple Mac Mini M4, utilizado como plataforma de pruebas para la ejecución y validación de aplicaciones HPC en entornos macOS y arquitectura ARM. Este equipo incorpora un procesador Apple M4 con arquitectura ARM64, núcleos de alto rendimiento y eficiencia, GPU integrada de última generación y 16 GB de memoria unificada. El Mac Mini M4 permitirá evaluar la compatibilidad y el rendimiento de contenedores en sistemas Apple Silicon.

### Software

- Sistema operativo Ubuntu 24.04 LTS. Será la distribución Linux principal con la que vamos a trabajar, tanto en forma nativa, como en los contenedores.
- Sistema operativo Microsoft Windows 11 Home. Será la distribución con la que se ejecutarán las pruebas de compatibilidad y rendimiento en entornos Windows.



- Sistema operativo macOS Sequoia 15.5. Será la distribución con la que se ejecutarán las pruebas de compatibilidad y rendimiento en entornos Apple.

#### Recursos humanos.

Dispositivo	Descripción	Coste (€)
LG Gram 14Z90Q-G.AA75B	Portátil principal de desarrollo	1 200
Lenovo Legion 5	Portátil de pruebas	1 000
Apple Mac Mini M4	Dispositivo de pruebas Apple	599
<b>Total</b>		<b>2 799</b>

Tabla 2.2: Costes estimados de hardware para el proyecto

En cuanto al software, los sistemas operativos Microsoft Windows 11 Home y macOS Sequoia 15.5 vienen incluidos en los dispositivos correspondientes, por lo que no se ha considerado un coste adicional. El sistema operativo Ubuntu 24.04 LTS es de código abierto y gratuito, por lo que tampoco se ha considerado un coste adicional.

#### Recursos humanos

En la tabla 2.3 se detalla el coste por hora, las horas estimadas y el coste total de los recursos humanos necesarios para llevar a cabo el proyecto.

Recurso	Puesto	€/h	Horas	Total (€)
Fernando Cuesta Bueno	Desarrollador	20	260	5 200
Juan José Escobar Pérez	Tutor/Supervisor	50	50	2 500
<b>Total</b>				<b>7 700</b>

Tabla 2.3: Costes estimados de recursos humanos para el proyecto

#### Coste total del proyecto

El coste total del proyecto se calcula sumando los costes de hardware, software y recursos humanos. En la tabla 2.4 se detalla el coste total estimado.

Concepto	Coste (€)
Hardware	2 799
Software	0
Recursos humanos	7 700
<b>Total</b>	<b>10 499</b>

Tabla 2.4: Coste total estimado del proyecto



## Capítulo 3

# Estado del arte

[En el estado del arte se necesita hacer un estudio tanto sobre la tecnología que soporta el proyecto como sobre el problema que se aborda en él. Se puede estructurar por secciones y se aconseja utilizar referencias a los documentos e información que se describe aquí.]

Como norma general y más en proyectos con carácter investigador, se recomienda añadir un párrafo por cada documento/referencia que estudie del estado del arte, finalizando esta sección con un párrafo explicativo de la novedad/característica que propone, modifica o añade el proyecto sobre dicho estado del arte.]

### 3.1. Sección

#### 3.1.1. Sub-seccion



## Capítulo 4

# Propuesta principal

[En esta sección se ha de introducir y explicar la propuesta principal del trabajo. Se puede y es recomendable dividir en secciones, incluso, este capítulo puede contemplar varios capítulos a su vez.]

### 4.1. Background

### 4.2. Herramientas utilizadas

### 4.3. Implementación



## Capítulo 5

# Experimentación

[Exponer aquí los requisitos, diseño y la implementación de la propuesta realizada. Este capítulo podría integrarse el de la propuesta si es necesario.]

Ejemplo de código en  $\text{\LaTeX}$

### 5.1. Setup experimental

[Se definen aquí los *setups* necesarios así como su configuración para poder evaluar y validar la propuesta de proyecto y objetivos del mismo. Es posible que se divida en secciones correspondientes a escenarios diferentes para evaluar diferentes casos de uso o funcionalidades.]

#### 5.1.1. Escenario 1: bla bla bla ...

### 5.2. Resultados y discusión





## Capítulo 6

# Conclusiones y trabajo futuro

[En este capítulo se presentan las conclusiones obtenidas al llevar a cabo el presente trabajo]

### 6.1. Contribuciones

[En esta sección se presentan las principales contribuciones del trabajo realizado.]

- Contribución 1 ...
- Contribución 2 ...

### 6.2. Retos y trabajo futuro

[Exponer aquí los retos y trabajos futuros.]



## Capítulo 7

# Bibliografía



## Apéndice A

## Anexo A

[En los anexos se expone aquella información que es complementaria a la propia memoria pero que, por su contenido o longitud, no encajan como un capítulo al uso. Piezas de código fuente, explicación en detalle de algoritmos, tablas adicionales, etc., son algunos ejemplos de información que podría ir en un anexo.]



