



Interfaces Cerebro-Cloud para la predicción de actividades de imaginación motriz

Autor:

Ing. Freddy Julian Riascos Salas

Director:

Mg. Ing. Jaime Andrés Riascos Salas (Potsdam Embodied Cognition
Group PECoG)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 4 de marzo de 2025 y el 22 de abril de 2025.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar.	5
2. Identificación y análisis de los interesados	7
3. Propósito del proyecto	7
4. Alcance del proyecto	8
5. Supuestos del proyecto.	8
6. Product backlog.	9
7. Criterios de aceptación de historias de usuario	11
8. Fases de CRISP-DM	12
9. Desglose del trabajo en tareas	14
10. Diagrama de Gantt	19
11. Planificación de Sprints	26
12. Normativa y cumplimiento de datos (gobernanza)	29
13. Gestión de riesgos	30
14. Sprint Review	32
15. Sprint Retrospective	33

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	4 de marzo de 2025
1	Se completa hasta el 5 punto inclusive	20 de marzo de 2025
2	Se completa hasta el 8 punto inclusive	28 de marzo de 2025

Acta de constitución del proyecto

Buenos Aires, 4 de marzo de 2025

Por medio de la presente se acuerda con el Ing. Freddy Julian Riascos Salas que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Interfaces Cerebro-Cloud para la predicción de actividades de imaginación motriz” y consistirá en *evaluar una interfaz cerebro-computadora (Brain-Computer Interface, BCI) con soporte en la nube para la detección de patrones de imaginación motriz*. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de \$25000, con fecha de inicio el 4 de marzo de 2025 y fecha de presentación pública el octubre de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Xprende
Xprendetech S.A

Mg. Ing. Jaime Andrés Riascos Salas
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

Las interfaces cerebro-computadora (*Brain-Computer Interfaces*, BCI) han emergido como una tecnología innovadora que permite la comunicación directa entre el cerebro humano y dispositivos externos. En particular, la predicción de actividades de imaginación motriz a través de BCI han cobrado relevancia en campos que varían desde la rehabilitación, robótica, control de prótesis hasta sistemas domóticos, videojuegos y realidad virtual.

La integración de las BCI con tecnologías en la nube permite el almacenamiento, procesamiento y análisis eficiente de grandes volúmenes de datos cerebrales. Esto favorece la aplicación de algoritmos avanzados de aprendizaje automático y mejora la precisión de la interpretación de señales cerebrales.

1.1 Conceptos fundamentales

Interfaces cerebro-computadora

Los BCI son sistemas que registran la actividad cerebral mediante técnicas como la electroencefalografía (EEG) y traducen estas señales en comandos computacionales. Existen distintos tipos de BCI:

- Invasivas: electrodos implantados directamente en el cerebro.
- No invasivas: uso de sensores externos como EEG, MEG o fNIRS.

Imaginación motriz

La imaginación motriz se refiere a la capacidad de representar mentalmente movimientos sin ejecutarlos físicamente. Durante este proceso, se activan patrones específicos en la corteza motora, los cuales pueden ser detectados mediante EEG y utilizados para el control de dispositivos externos.

Computación en la nube y BCI

El uso de servicios en la nube permite procesar grandes volúmenes de datos EEG en tiempo real obteniendo beneficios como:

- Almacenamiento y procesamiento escalable de datos cerebrales.
- Acceso remoto para análisis colaborativo.
- Implementación de modelos de aprendizaje automático en infraestructura distribuida.

1.2 Problema actual

Al presente, las personas con discapacidades motoras severas enfrentan grandes dificultades en la interacción con su entorno. Los sistemas actuales de BCI presentan limitaciones en términos de precisión, latencia, accesibilidad, recopilación, análisis y clasificación de las señales EEG. Normalmente, estos datos se encuentran contaminados por distintos artefactos biológicos, tales como señales cardíacas, respiratorias o músculos, como también por ruidos externos.

Así mismo, la dimensión de estos datos, dada por la cantidad de canales y señal de tiempo, crea una problema de procesamiento y dimensionalidad. Todas estas dificultades evitan que

el clasificador reciba características latentes de la señal y así realizar la predicción de forma adecuada y rápida.

1.3 Solución propuesta

La interfaz Cerebro-Cloud sugerida integra un modelo de predicción basado en aprendizaje automático con una arquitectura en la nube que permita la adquisición, procesamiento y transmisión de señales cerebrales en tiempo real. Esto proporcionará una solución más precisa, escalable y accesible para el control de dispositivos mediante imaginación motriz.

En comparación con el estado del arte actual, la solución se destaca en:

- Precisión mejorada: uso de modelos de inteligencia artificial optimizados para la interpretación de señales eléctricas (electroencefalografía, EEG).
- Reducción de latencia: procesamiento distribuido en la nube.
- Accesibilidad: plataforma escalable con acceso a la información para usuarios y especialistas.

Este interfaz BCI-Cloud tiene como objetivo mejorar la calidad de vida de personas con discapacidades motoras al proporcionar una herramienta en la nube para la comunicación y control de dispositivos protésicos.

El proyecto se enmarca dentro de un programa de innovación tecnológica de la empresa Xprende, que cuenta con financiamiento para su ejecución.

1.4 Descripción funcional y diagrama en bloques

La solución propuesta consta de los siguientes módulos principales:

- Adquisición de señales EEG: sensores no invasivos capturan la actividad cerebral del usuario.
- Preprocesamiento de datos: filtrado y eliminación de ruido en las señales EEG.
- Modelo de predicción: algoritmos de aprendizaje automático analizan los datos y determinan la intención motriz.
- Transmisión en la nube: los datos procesados se envían a servidores remotos para análisis y almacenamiento.
- Interfaz usuario-dispositivo: una interfaz que traduce la predicción en comandos para dispositivos externos, como prótesis o interfaces de control.

En la figura 1 se presenta el diagrama de bloques del sistema BCI-Cloud. Se observa que el usuario inicial genera datos con el sensor EEG. Luego envía los datos a un sistema de preprocesamiento. Una vez que los datos se encuentran óptimos se envían a la nube. Seguidamente el modelo seleccionado se entrena. Finalmente el modelo predice el movimiento imaginado en la interfaz de usuario.

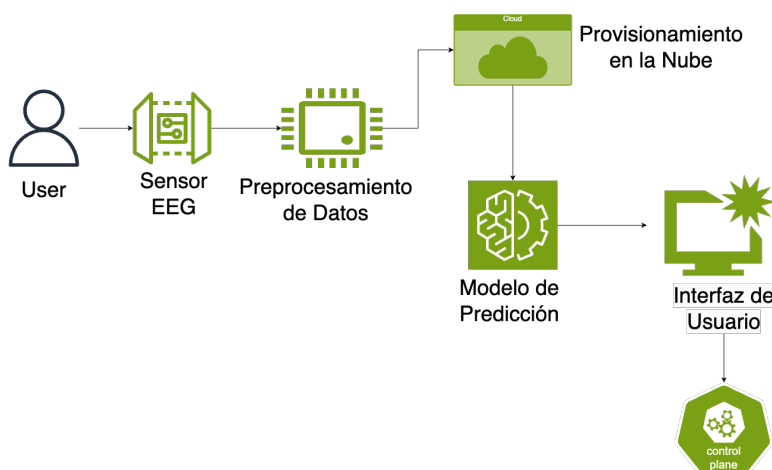


Figura 1. Diagrama del sistema BCI-Cloud.

2. Identificación y análisis de los interesados

A continuación, se presentan los principales actores involucrados en el desarrollo del proyecto y su respectiva función:

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Ing. Miguel Amaya	Xprendetech S.A	-
Responsable	Ing. Freddy Julian Riascos Salas	FIUBA	Alumno
Orientador	Mg. Ing. Jaime Andrés Riascos Salas	Potsdam Embodied Cognition Group PECoG	Director del Trabajo Final
Usuario final	Paciente	-	-

A continuación las principales características de cada interesado.

- Orientador: el Mg. Ing. Andrés Salas es experto en desarrollar interfaces cerebro-maquina y dará orientación con la definición de los requerimientos y el desarrollo del sistema BCI-Cloud.
- Usuario final: usuario con discapacidad motora severa, quien se beneficiaría directamente del control del dispositivo y la interfaz.

3. Propósito del proyecto

La intención del proyecto es diseñar y desarrollar una plataforma basada en BCI-Cloud que facilite la predicción de actividades de imaginación motriz en personas con discapacidades motoras severas. A través de la integración de inteligencia artificial y computación en la nube, se busca ofrecer una solución innovadora que permita mejorar la interacción con el entorno mediante el control preciso de dispositivos electrónicos y protésicos.

4. Alcance del proyecto

El proyecto comprenderá los siguientes componentes:

- Modelo de predicción basado en algoritmos de aprendizaje automático para la interpretación de señales EEG. Se evaluarán distintos modelos como CNNs, RNNs y *Transformers* para determinar el que mejor performe en términos de precisión y latencia.
- Datos adquiridos de sensores EEG, que se utilizarán para entrenar y validar el modelo. Se garantizará que la adquisición de datos cumpla con estándares de calidad y se preprocesen para eliminar ruido.
- Código de la infraestructura en la nube.
- Documentación técnica y científica, que detalla el diseño, implementación y validación del sistema.
- Pruebas y validaciones realizadas con usuarios objetivo para evaluar el desempeño y precisión del modelo. Se incluirán métricas clave que garanticen el óptimo desempeño del sistema.
- Desarrollo y entrenamiento del modelo de predicción de actividades de imaginación motriz, con comparaciones entre distintos enfoques de inteligencia artificial.
- Integración con una infraestructura en la nube escalable y segura en *Amazon Web Services*.
- Adquisición y procesamiento de señales cerebrales mediante sensores EEG.
- Evaluación del sistema con usuarios finales para validar su precisión y usabilidad.
- Generación de documentación técnica para futuras mejoras e implementación.
- Optimización del procesamiento en la nube para reducir latencia y mejorar la eficiencia del sistema.

El presente proyecto no incluye:

- Desarrollo de hardware EEG propio. Para ello se utilizarán dispositivos comerciales disponibles en el mercado.
- Implementación de interfaces cerebro-máquina más allá de la imaginación motriz.
- Integración con sistemas de salud o bases de datos clínicas.

5. Supuestos del proyecto

- Disponibilidad de datos EEG de calidad: se asume que los datos recopilados mediante sensores EEG serán suficientes y de calidad adecuada para el entrenamiento del modelo sin necesidad de un preprocesamiento excesivo.
- Recursos computacionales: se cuenta con acceso a instancias de cómputo en la nube, tales como las brindadas por *Amazon Web Services*, que permitan el entrenamiento y despliegue del modelo de inteligencia artificial sin limitaciones de procesamiento o almacenamiento.

- Factibilidad técnica de integración: se considera viable la integración entre los sensores EEG, la infraestructura en la nube y la interfaz de usuario.
- Condiciones regulatorias favorables: no existen restricciones legales o normativas que impidan la recopilación y procesamiento de datos EEG.

6. Product backlog

1. Épica 1 Adquisición y procesamiento de datasets EEG

1.1. HU1

Como ingeniero, quiero obtener señales EEG desde datasets públicos para alimentar el modelo de predicción.

Dificultad: 5

Complejidad: 4

Incertidumbre: 3

Suma: 12 → *Story Points*: 13

Prioridad: 1

1.2. HU2

Como ingeniero, quiero que las señales EEG sean filtradas y normalizadas automáticamente para mejorar la calidad del entrenamiento del modelo.

Dificultad: 4

Complejidad: 2

Incertidumbre: 2

Suma: 8 → *Story Points*: 8

Prioridad: 2

2. Épica 2 Inteligencia artificial y modelado

2.1. HU3

Como ingeniero, quiero entrenar un modelo de inteligencia artificial con señales EEG preprocesadas para predecir actividades de imaginación motriz con alta precisión.

Dificultad: 5

Complejidad: 5

Incertidumbre: 4

Suma: 14 → *Story Points*: 21

Prioridad: 3

2.2. HU4

Como ingeniero, quiero optimizar el modelo de inteligencia artificial para que su inferencia sea menor a 5000 ms, alcance al menos un 80 % de precisión y exponga sus predicciones mediante una API REST.

Dificultad: 4
Complejidad: 3
Incertidumbre: 3
Suma: 10 → *Story Points*: 13
Prioridad: 4

3. Épica 3 Infraestructura en la nube

3.1. HU5

Como ingeniero, quiero que el procesamiento de datos EEG ocurra en *Amazon Web Services* AWS Lambda para mejorar la escalabilidad del sistema.

Dificultad: 5
Complejidad: 4
Incertidumbre: 3
Suma: 12 → *Story Points*: 13
Prioridad: 5

3.2. HU6

Como ingeniero, quiero desplegar automáticamente el modelo entrenado en AWS para facilitar su mantenimiento, escalabilidad y monitoreo.

Dificultad: 4
Complejidad: 3
Incertidumbre: 3
Suma: 10 → *Story Points*: 13
Prioridad: 6

4. Épica 4 Interfaz de usuario y seguridad

4.1. HU7

Como usuario final, quiero visualizar mis señales EEG en una interfaz gráfica para entender cómo se interpretan mis actividades cerebrales.

Dificultad: 5
Complejidad: 3
Incertidumbre: 2
Suma: 10 → *Story Points*: 13
Prioridad: 7

4.2. HU8

Como ingeniero, quiero que los datos EEG sean almacenados y transmitidos de manera segura para cumplir con las normativas de privacidad.

Dificultad: 5
Complejidad: 5
Incertidumbre: 2

Suma: 12 \rightarrow *Story Points*: 13

Prioridad: 8

5. Épica 5 Gestión del proyecto, documentación y defensa final

5.1. HU9

Como ingeniero del proyecto, quiero planificar, documentar y estructurar el trabajo desarrollado, para asegurar la trazabilidad y resultados del trabajo final.

Dificultad: 5

Complejidad: 3

Incertidumbre: 2

Suma: 10 \rightarrow *Story Points*: 13

Prioridad: 7

7. Criterios de aceptación de historias de usuario

1. Épica 1 Adquisición y procesamiento de datasets EEG

1.1. Criterios de aceptación HU1

- Se deben verificar al menos 2 repositorios públicos de datasets que estén accesibles y disponibles para su descarga.
- Los datos deben almacenarse automáticamente en una ruta definida del proyecto, sin intervención manual.
- Se debe validar que el formato de los datos descargados sea compatible con la tubería de procesamiento.
- Los datos almacenados se deben organizar en una estructura jerárquica que difiera entre los repositorios consultados.
- Se debe crear un archivo de documentación que describa la información de los datos, tales como fuentes usadas, formatos y cantidad de muestras.

1.2. Criterios de aceptación HU2

- Se debe aplicar al menos un filtro a las señales EEG para eliminar frecuencias no deseadas.
- Los datos deben estar escalados y asegurar que todos los canales estén en la misma escala.
- Se debe crear un archivo de documentación que describa los filtros aplicados, método de normalización y duración del proces.
- Se debe crear un archivo de documentación que describa los filtros aplicados, método de normalización y duración del proces.
- Se debe generarse un gráfico comparando la señal original y la procesada conocer la efectividad del filtrado y normalización.

2. Épica 2 Inteligencia artificial y modelado

2.1. Criterios de aceptación HU3

- El modelo de predicción debe alcanzar una precisión mínima del 85 % en validación cruzada.
- Se deben entrenar al menos tres arquitecturas y seleccionar la mejor.
- Se deben generar logs detallados del proceso de entrenamiento con métricas de desempeño.

2.2. Criterios de aceptación HU4

- El modelo optimizado debe tener un tiempo de inferencia menor a 5000 ms.
- Se debe medir la latencia en diferentes condiciones de carga y garantizar estabilidad.
- Los resultados de predicción deben ser accesibles en tiempo real a través de una API REST.

3. Épica 3 Infraestructura en la nube

3.1. Criterios de aceptación HU5

- La infraestructura debe ser escalable, permitiendo hasta 1000 eventos concurrentes.
- Se debe garantizar un 85.9 % de disponibilidad del servicio en producción.
- *Amazon Web Services* Lambda debe procesar eventos EEG en tiempo real con ejecución máxima de 15 segundos.

3.2. Criterios de aceptación HU6

- El modelo debe estar expuesto a través de una API REST en AWS.
- Se debe mantener un historico de versiones del modelo.
- El modelo debe integrarse con logs y métricas para monitoreo.

4. Épica 4 Interfaz de usuario y seguridad

4.1. Criterios de aceptación HU7

- La interfaz debe permitir visualizar las señales EEG en gráficos.
- Se debe permitir la exportación de datos en formatos CSV y JSON.

4.2. Criterios de aceptación HU8

- Los datos EEG deben ser cifrados en tránsito y en reposo.
- El sistema debe incluir una política de retención y eliminación de datos.

5. Épica 5 Gestión del proyecto, documentación y defensa final

5.1. Criterios de aceptación HU9

- La documentación técnica debe incluir diagramas, descripciones de componentes, procesos y arquitectura.

8. Fases de CRISP-DM

Comprensión del negocio

Objetivo: desarrollar un modelo de inteligencia artificial que analice señales EEG para predecir la intención de movimiento en usuarios, lo que facilitarían aplicaciones en neurorehabilitación y control de dispositivos.

Valor agregado: automatización del análisis de señales cerebrales para mejorar la accesibilidad a tecnologías BCI.

Métricas de éxito:

- Precisión mínima del modelo: 85 %.
- Latencia de predicción menor a 5000 ms.
- Disponibilidad del sistema: 85 % en producción.
- Cumplimiento de normativas de seguridad.

Comprensión de los datos

- Tipo de datos: señales de series temporales EEG obtenidas de dispositivos de adquisición neurofisiológica.
- Fuentes: sensores EEG comerciales como *OpenBCI*, *Emotiv* o bases de datos públicas.
- Cantidad: al menos 100,000 registros diarios procesados en *Amazon Web Services*.
- Calidad: se esperan encontrar ruidos, artefactos musculares y variabilidad intersujeto.

Preparación de los datos

Las características claves a tener en cuenta para las señales EEG son:

- Bandas de frecuencia.
- Espectrogramas de señales EEG.
- Características temporales obtenidas con algoritmos de *STFT* y *Wavelet*.

Las transformaciones que podrían ser requeridas para las señales EEG son:

- Filtrado pasa bandas.
- Normalización de señales.
- Eliminación de artefactos usando algoritmos de análisis de componentes principales.

Modelado

El problema se define como una clasificación de señales EEG para la predicción de imaginación motriz. Mientras las arquitecturas candidatas para este problema son redes neuronales como CNNs, RNNs y *Transformers*.

Evaluación del modelo

Se podría utilizar el *accuracy* para conocer la precisión de aciertos en la predicción de clases de imaginación motriz. Además el *F1-score* y la matriz de confusión para obtener el balance entre precisión y exhaustividad y los falsos positivos y falsos negativos.

Despliegue del modelo

El modelo se implementará con un sistema basado en la nube con *Amazon Web Services* Lambda para procesamiento, una API REST para enviar los datos a la nube y una fuente de almacenamiento que proporcione un costo beneficio en los datos durante el tiempo.

9. Desglose del trabajo en tareas

Cantidad total de horas: 580.

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU1	Investigar y comparar datasets públicos EEG	8 h	Alta
HU1	Descargar datasets seleccionados y limpiar estructura de carpetas	8 h	Alta
HU1	Analizar formato de datos y convertir si es necesario	8 h	Media
HU1	Crear scripts para carga, validación y parsing	8 h	Alta
HU1	Automatizar descarga periódica desde repositorios	8 h	Media
HU1	Conversión y unificación de formatos	8 h	Media
HU1	Verificación de integridad de los archivos descargados	8 h	Alta
HU1	Almacenamiento y organización de datos EEG	8 h	Alta
HU1	Documentación del proceso de adquisición	4 h	Baja
HU1	Evaluación legal/licencia de uso de los datasets	6 h	Media
HU1	Generación de logs y reportes de adquisición	4 h	Baja
HU2	Investigar técnicas de preprocesamiento EEG	8 h	Alta
HU2	Implementación de función de filtrado por banda	8 h	Alta
HU2	Implementación de detección y corrección de artefactos simples	8 h	Alta
HU2	Implementación de métodos de normalización	8 h	Alta
HU2	Diseño de pipeline automático de procesamiento por lote	8 h	Alta
HU2	Pruebas con datos reales y comparación visual	6 h	Media
HU2	Generación de logs por archivo procesado	4 h	Baja
HU2	Creación de gráficas de validación por dataset	6 h	Media
HU2	Evaluación de calidad de la señal post-procesamiento	6 h	Media
HU2	Desarrollo de manejo de errores en procesamiento	8 h	Media
HU2	Documentación del pipeline de procesamiento	8 h	Baja
HU2	Generación de logs y reportes de adquisición	4 h	Baja

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU3	Estudio de modelos adecuados para señales EEG	8 h	Alta
HU3	Seleccionar arquitectura del primer modelo (baseline)	8 h	Alta
HU3	Preparar datasets de entrenamiento y test con etiquetas (baseline)	8 h	Alta
HU3	Entrenar modelo con logs y métricas (baseline)	8 h	Alta
HU3	Validación cruzada y testing (baseline)	8 h	Medio
HU3	Implementar early stopping y checkpointing (baseline)	8 h	Medio
HU3	Evaluar métricas clave como accuracy, precision, recall, o F1 (baseline)	8 h	Alta
HU3	Seleccionar arquitectura del segundo modelo (alternativo)	8 h	Alta
HU3	Preparar datasets de entrenamiento y test con etiquetas (alternativo)	6 h	Alta
HU3	Entrenar modelo base con logs y métricas (alternativo)	6 h	Alta
HU3	Validación cruzada y testing (alternativo)	6 h	Medio
HU3	Implementar early stopping y checkpointing (alternativo)	6 h	Medio
HU3	Evaluar métricas clave como accuracy, precision, recall, o F1 (alternativo)	6 h	Alta
HU3	Análisis comparativo entre modelos	6 h	Media
HU4	Afinar hiperparámetros y simplificar arquitectura	8 h	Alta
HU4	Evaluación comparativa del modelo con diferentes configuraciones de batch sizes, hardware y frameworks	8 h	Alta
HU4	Medir tiempo de inferencia en entorno controlado	8 h	Alta
HU4	Desarrollo de una API REST para server las predicciones	8 h	Alta
HU4	Añadir arquitectura de contenedores para la API REST y modelo para despliegue rápido	8 h	Alta
HU4	Realizar test de carga con locust a la API REST	6 h	Media
HU4	Documentar API y latencia de respuesta	6 h	Media

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU4	Despliegue del modelo en un entorno cloud con aceleración	8 h	Alta
HU5	Diseñar la arquitectura en AWS	8 h	Alta
HU5	Codificar las funciones que recibirán y preprocesarán los datasets	8 h	Alta
HU5	Crear scripts para emular eventos desde datasets	6 h	Media
HU5	Configurar límites de timeout y memoria en Lambda	8 h	Alta
HU5	Simular carga de 500 a 1000 eventos para probar escalabilidad	8 h	Media
HU5	Configurar logs, métricas y alertas para el procesamiento de los datasets	8 h	Media
HU5	Documentar la arquitectura y diagramas de arquitectura	6 h	Media
HU5	Validar tiempo de ejecución menor o igual a 15 s	6 h	Media
HU6	Crear script de empaquetado del modelo y sus dependencias	8 h	Alta
HU6	Configurar pipeline CI/CD para despliegue automático	8 h	Alta
HU6	Definir e implementar pruebas automatizadas antes del despliegue	8 h	Alta
HU6	Investigar la plataforma para alojar el modelo productivo	8 h	Alta
HU6	Desplegar el modelo productivo en la plataforma seleccionada	8 h	Alta
HU6	Implementar un mecanismo para almacenar versiones del modelo y realizar rollbacks	8 h	Media
HU6	Habilitar logging y métricas con AWS CloudWatch	8 h	Media
HU6	Configurar alarmas básicas de disponibilidad y errores	8 h	Media
HU6	Documentar el flujo de despliegue automático y monitoreo	8 h	Media
HU7	Implementar componente de visualización	8 h	Alta
HU7	Exportar datos visualizados a CSV y JSON	8 h	Alta
HU7	Conectar frontend con backend de predicción	8 h	Alta
HU7	Realizar pruebas de usabilidad	4 h	Media
HU7	Documentar guía básica de uso	4 h	Baja

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU8	Cifrar datos en tránsito con HTTPS / TLS	6 h	Alta
HU8	Cifrar datos en reposo con servicios de AWS	6 h	Alta
HU8	Configurar políticas de IAM para acceso restringido	6 h	Alta
HU8	Definir e implementar políticas y reglas de retención de datos	4 h	Media
HU8	Documentar cumplimiento normativo básico	4 h	Media
HU8	Configurar un registro y monitoreo de accesos	4 h	Alta
HU9	Reuniones iniciales de planificación y definición de alcance del proyecto	8 h	Alta
HU9	Elaboración del cronograma y planificación detallada	8 h	Alta
HU9	Diseño de cronograma de trabajo	8 h	Alta
HU9	Redacción del capítulo de introducción y objetivos	8 h	Alta
HU9	Redacción del capítulo de metodología	8 h	Alta
HU9	Redacción del capítulo de resultados y discusión	8 h	Alta
HU9	Redacción de conclusiones y recomendaciones	8 h	Alta
HU9	Revisión general y edición del documento final	8 h	Alta
HU9	Diseño y elaboración de presentación para defensa	8 h	Alta
HU9	Ensayos de presentación y defensa con retroalimentación	8 h	Alta

10. Diagrama de Gantt

En las siguientes figuras se puede observar las tablas del diagrama de Gantt

Name	Begin date	End date
Investigar y comparar datasets públicos EEG	4/11/25	4/11/25
Descargar datasets seleccionados y limpiar estructura de carpetas	4/14/25	4/14/25
Analizar formato de datos y convertir si es necesario	4/15/25	4/15/25
Analizar formato de datos y convertir si es necesario	4/16/25	4/16/25
Crear scripts para carga, validación y parsing	4/17/25	4/17/25
Automatizar descarga periódica desde repositorios	4/18/25	4/18/25
Conversión y unificación de formatos	4/21/25	4/21/25
Verificación de integridad de los archivos descargados	4/22/25	4/22/25
Almacenamiento y organización de datos EEG	4/23/25	4/23/25
Documentación del proceso de adquisición	4/24/25	4/24/25
Evaluación legal/licencia de uso de los datasets	4/28/25	4/28/25
Generación de logs y reportes de adquisición	4/29/25	4/29/25
Investigar técnicas de preprocesamiento EEG	4/30/25	4/30/25
Implementación de función de filtrado por banda	5/1/25	5/1/25
Implementación de detección y corrección de artefactos simples	5/2/25	5/2/25
Implementación de métodos de normalización	5/5/25	5/5/25
Diseño de pipeline automático de procesamiento por lote	5/6/25	5/6/25
Pruebas con datos reales y comparación visual	5/7/25	5/7/25

Figura 2. Tabla de tareas de Gantt.

Name	Begin date	End date
Generación de logs por archivo procesado	5/8/25	5/8/25
Creación de gráficas de validación por dataset	5/9/25	5/9/25
Evaluación de calidad de la señal post-procesamiento	5/12/25	5/12/25
Desarrollo de manejo de errores en procesamiento	5/13/25	5/13/25
Documentación del pipeline de procesamiento	5/14/25	5/14/25
Generación de logs y reportes de adquisición	5/15/25	5/15/25
Estudio de modelos adecuados para señales EEG	5/16/25	5/16/25
Seleccionar arquitectura del primer modelo (baseline)	5/19/25	5/19/25
Preparar datasets de entrenamiento y test con etiquetas (baseline)	5/20/25	5/20/25
Entrenar modelo con logs y métricas (baseline)	5/21/25	5/21/25
Validación cruzada y testing (baseline)	5/22/25	5/22/25
Implementar early stopping y checkpointing (baseline)	5/23/25	5/23/25
Evaluar métricas clave como accuracy, precision, recall, o F1 (baseline)	5/26/25	5/26/25
Seleccionar arquitectura del segundo modelo (alternativo)	5/27/25	5/27/25
Preparar datasets de entrenamiento y test con etiquetas (alternativo)	5/28/25	5/28/25
Entrenar modelo base con logs y métricas (alternativo)	5/29/25	5/29/25
Validación cruzada y testing (alternativo)	5/30/25	5/30/25
Implementar early stopping y checkpointing	6/2/25	6/2/25

Figura 3. Tabla de tareas de Gantt.

Name	Begin date	End date
Evaluar métricas clave como accuracy, precision, recall, o F1 (alternativo)	6/3/25	6/3/25
Análisis comparativo entre modelos	6/4/25	6/4/25
Afinar hiperparámetros y simplificar arquitectura	6/5/25	6/5/25
Evaluación comparativa del modelo con diferentes configuraciones de batch sizes, hardware y frameworks	6/6/25	6/6/25
Medir tiempo de inferencia en entorno controlado	6/9/25	6/9/25
Desarrollo de una API REST para server las predicciones	6/10/25	6/10/25
Añadir arquitectura de contenedores para la API REST y modelo para despliegue rápido	6/11/25	6/11/25
Realizar test de carga con locust a la API REST	6/12/25	6/12/25
Documentar API y latencia de respuesta	6/13/25	6/13/25
Despliegue del modelo en un entorno cloud con aceleración	6/16/25	6/16/25
Diseñar la arquitectura en AWS	6/17/25	6/17/25
Codificar las funciones que recibirán y preprocesarán los datasets	6/18/25	6/18/25
Crear scripts para emular eventos desde datasets	6/19/25	6/19/25
Configurar límites de timeout y memoria en Lambda	6/20/25	6/20/25
Simular carga de 500 a 1000 eventos para probar escalabilidad	6/23/25	6/23/25
Configurar logs, métricas y alertas para el procesamiento de los datasets	6/24/25	6/24/25
Documentar la arquitectura y diagramas de arquitectura	6/25/25	6/25/25
Validar tiempo de ejecución menor o igual a 15 s	6/26/25	6/26/25

Figura 4. Tabla de tareas de Gantt.

Name	Begin date	End date
Crear script de empaquetado del modelo y sus dependencias	6/27/25	6/27/25
Configurar pipeline CI/CD para despliegue automático	6/30/25	6/30/25
Definir e implementar pruebas automatizadas antes del despliegue	7/1/25	7/1/25
Investigar la plataforma para alojar el modelo productivo	7/2/25	7/2/25
Desplegar el modelo productivo en la plataforma seleccionada	7/3/25	7/3/25
Implementar un mecanismo para almacenar versiones del modelo y realizar rollbacks	7/4/25	7/4/25
Habilitar logging y métricas con AWS CloudWatch	7/7/25	7/7/25
Configurar alarmas básicas de disponibilidad y errores	7/8/25	7/8/25
Documentar el flujo de despliegue automático y monitoreo	7/9/25	7/9/25
Implementar componente de visualización	7/10/25	7/10/25
Exportar datos visualizados a CSV y JSON	7/11/25	7/11/25
Conectar frontend con backend de predicción	7/14/25	7/14/25
Realizar pruebas de usabilidad	7/15/25	7/15/25
Documentar guía básica de uso	7/16/25	7/16/25
Cifrar datos en tránsito con HTTPS / TLS	7/17/25	7/17/25
Cifrar datos en reposo con servicios de AWS	7/18/25	7/18/25
Configurar políticas de IAM para acceso restringido	7/21/25	7/21/25
Definir e implementar políticas y reglas de retención de datos	7/22/25	7/22/25

Figura 5. Tabla de tareas de Gantt.

Documentar cumplimiento normativo básico	7/23/25	7/23/25
Configurar un registro y monitoreo de accesos	7/24/25	7/24/25
Elaboración del cronograma y planificación detallada	7/25/25	7/25/25
Reuniones iniciales de planificación y definición de alcance del proyecto	7/28/25	7/28/25
Diseño de cronograma de trabajo	7/29/25	7/29/25
Redacción del capítulo de introducción y objetivos	7/30/25	7/30/25
Redacción del capítulo de metodología	7/31/25	7/31/25
Redacción del capítulo de resultados y discusión	8/1/25	8/1/25
Redacción de conclusiones y recomendaciones	8/4/25	8/4/25
Revisión general y edición del documento final	8/5/25	8/5/25
Diseño y elaboración de presentación para defensa	8/6/25	8/6/25
Ensayos de presentación y defensa con retroalimentación	8/7/25	8/7/25

Figura 6. Tabla de tareas de Gantt.

En las siguientes figuras se puede observar el diagrama de Gantt del proyecto. El color amarillo representa tareas no técnicas.

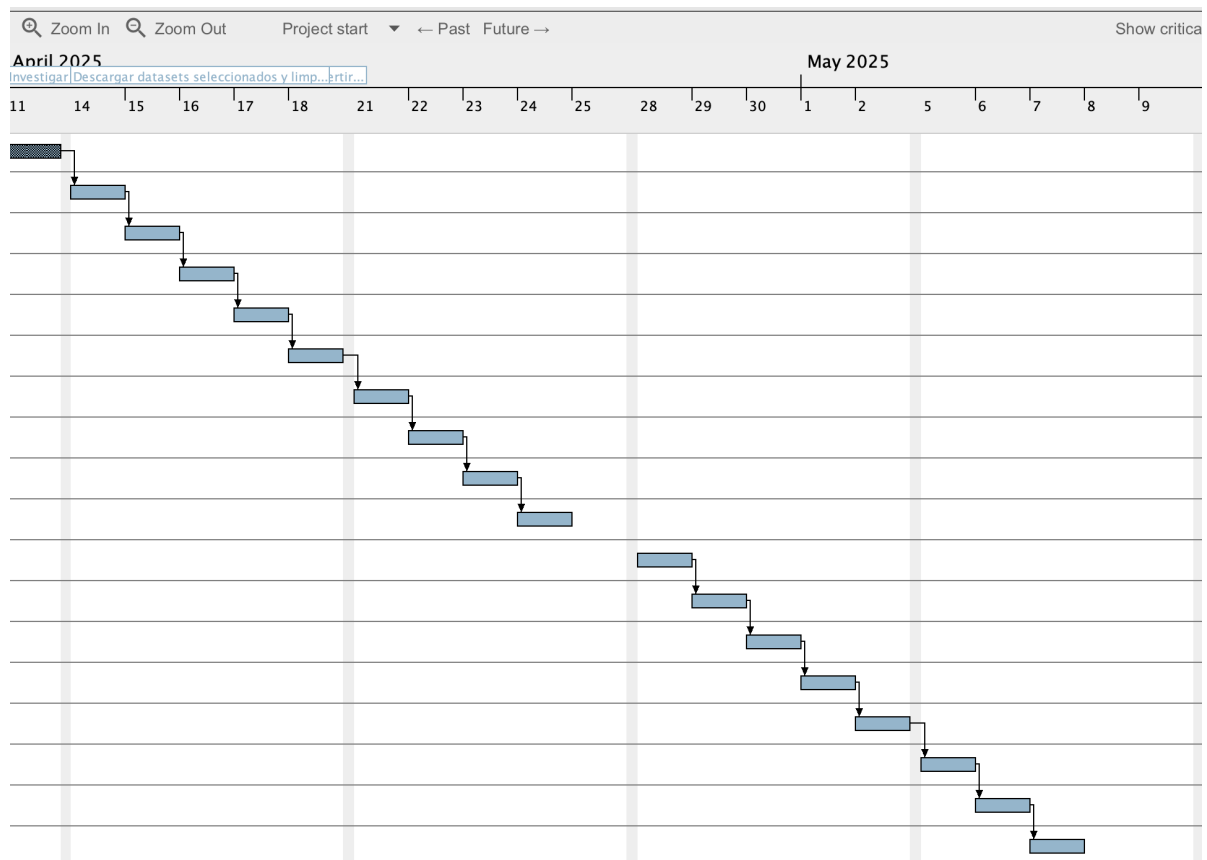


Figura 7. Tabla de tareas de Gantt.

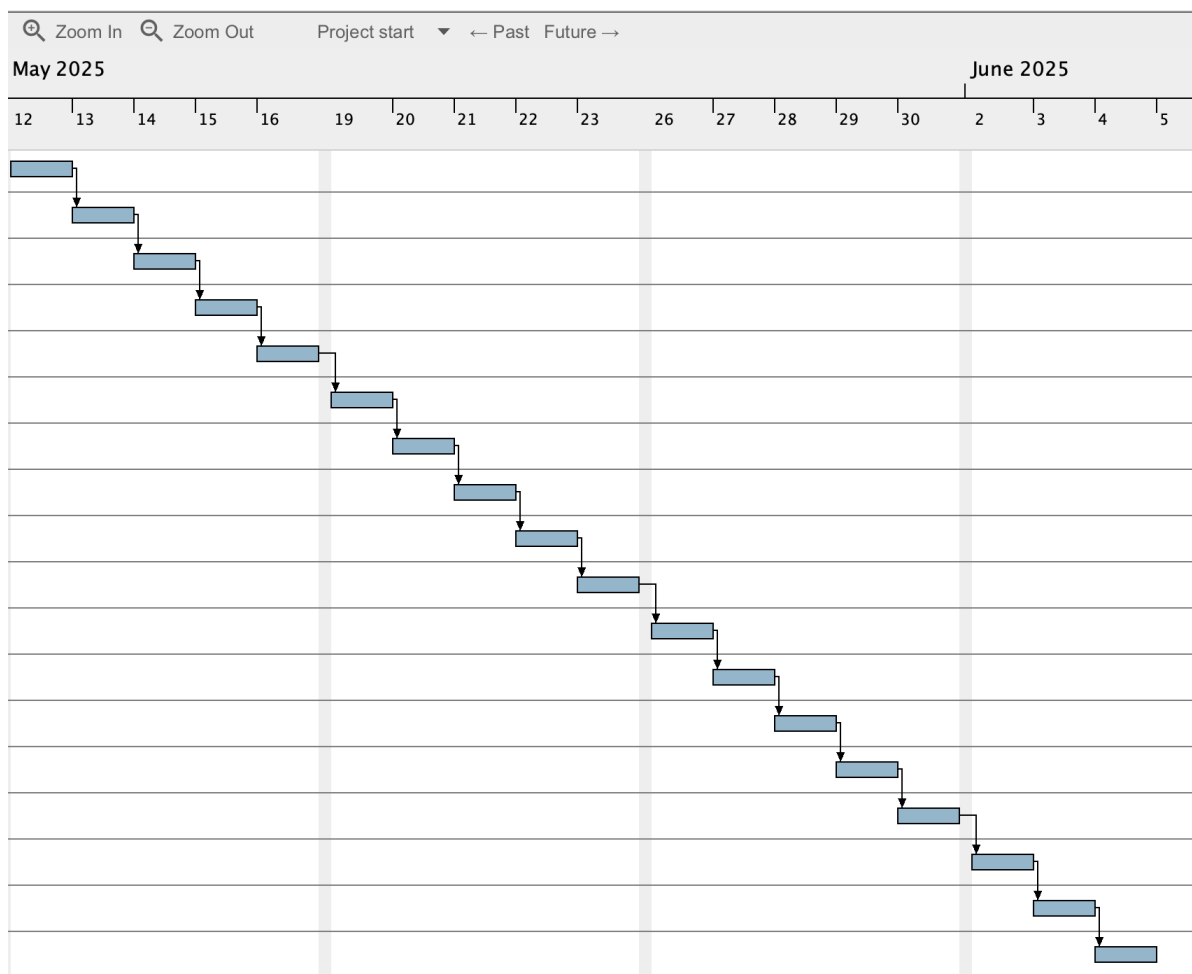


Figura 8. Tabla de tareas de Gantt.

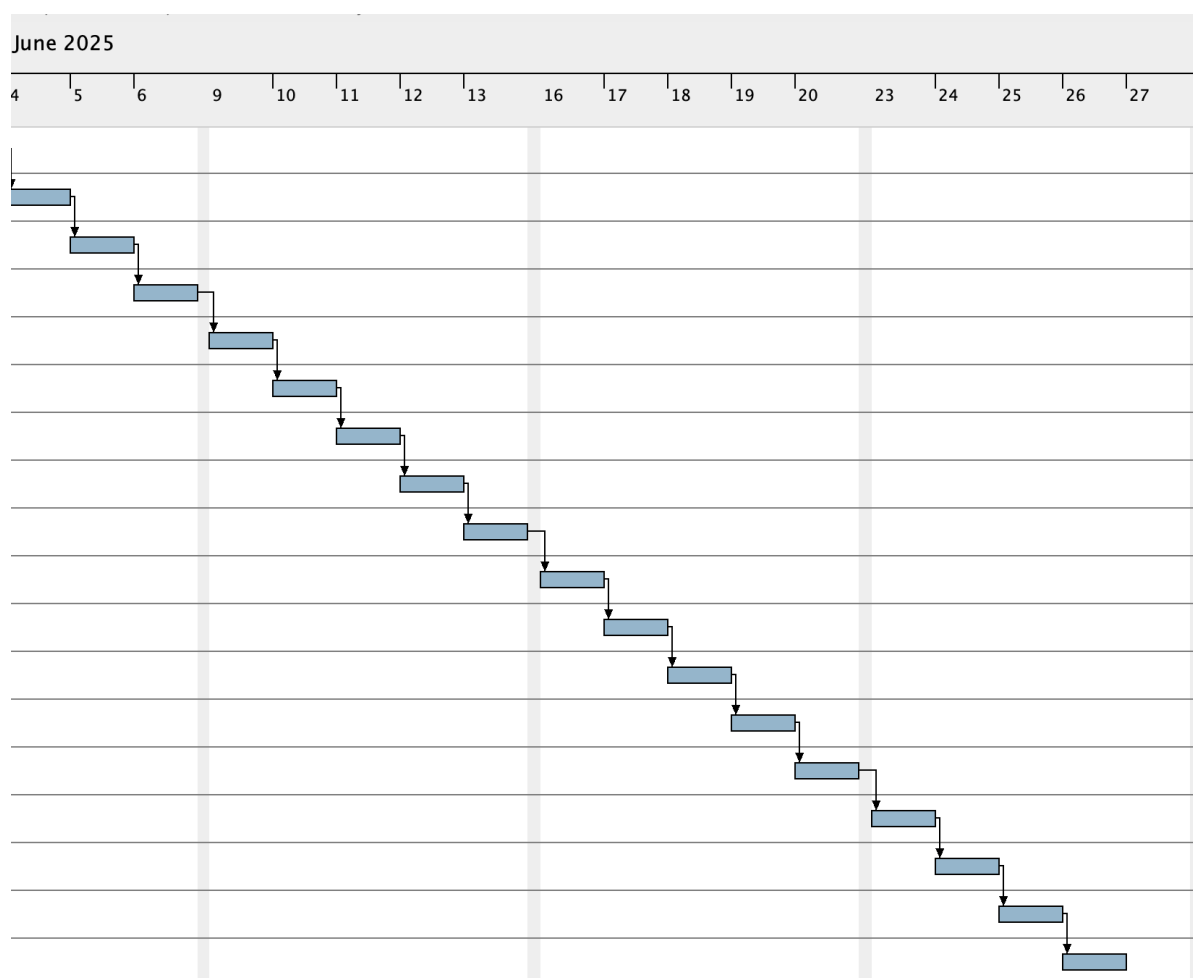


Figura 9. Tabla de tareas de Gantt.

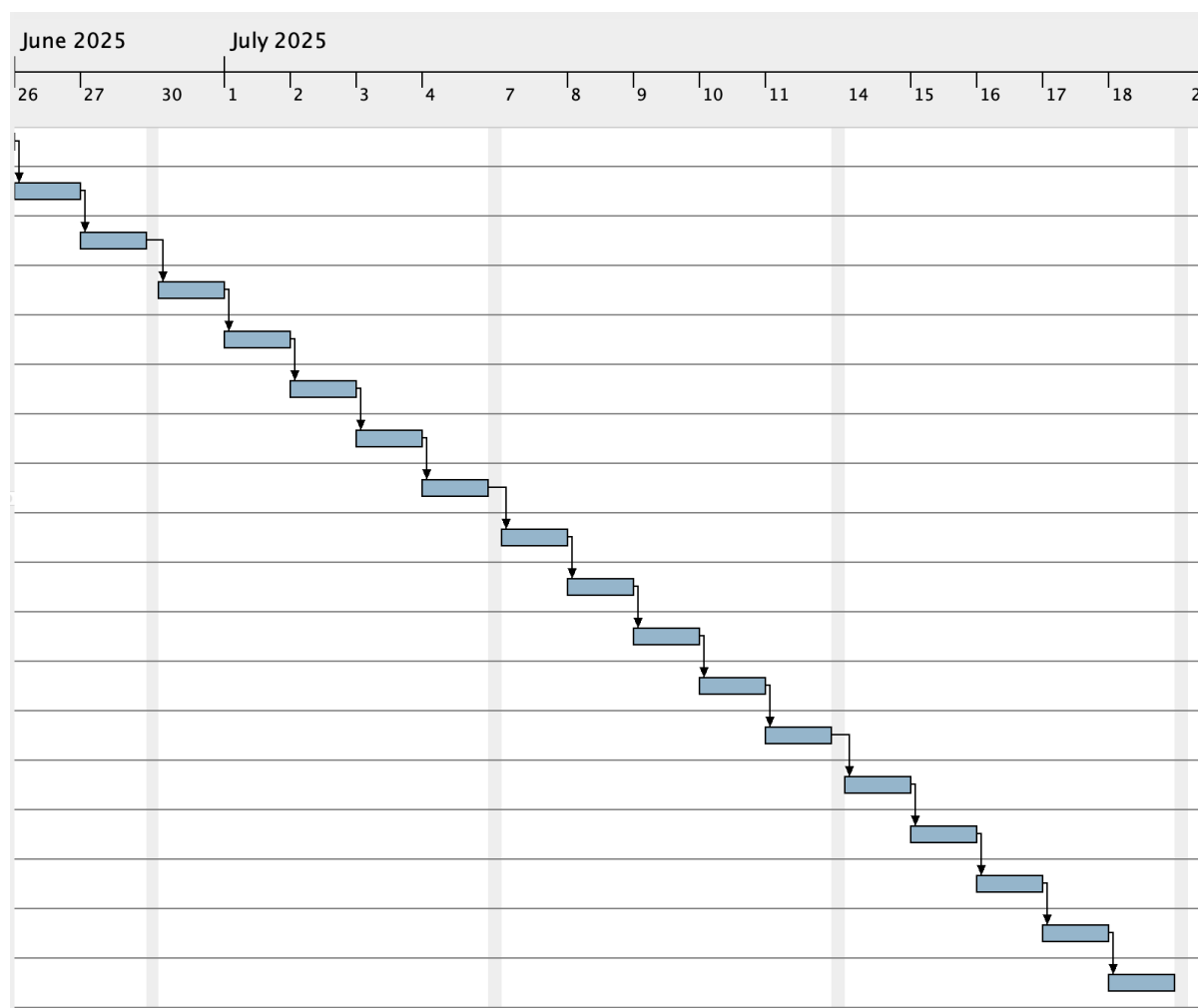


Figura 10. Tabla de tareas de Gantt.

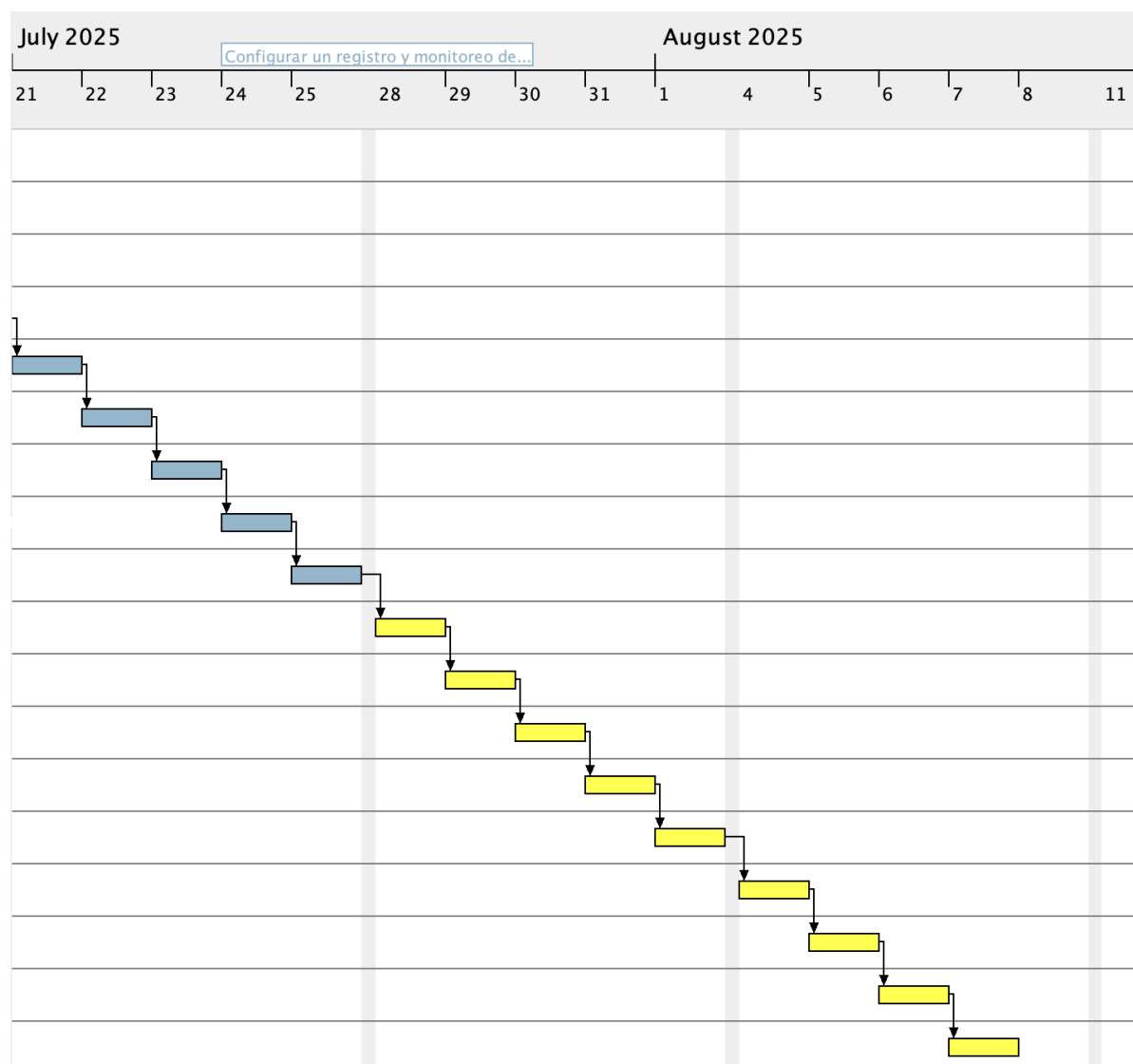


Figura 11. Tabla de tareas de Gantt.

11. Planificación de Sprints

A continuación, se detalla la tabla de planificación de Sprints.

Sprint	HU o fase	Tarea	Horas / SP	Responsable	Completado %
1	HU1	Exploración del dataset MOABB	8	Alumno	0
1	HU1	Análisis de estructuras de datos EEG	8	Alumno	0
1	HU1	Carga y limpieza de datos MOABB	8	Alumno	0
1	HU1	Documentación de la adquisición de datos	8	Alumno	0
2	HU2	Diseño del pipeline de preprocesamiento	8	Alumno	0
2	HU2	Implementación de filtrado y normalización	8	Alumno	0
2	HU2	Evaluación de calidad de señales preprocesadas	8	Alumno	0
2	HU2	Documentación de procesos y scripts	8	Alumno	0
2	HU3	Configuración de entorno virtual y dependencias	4	Alumno	0
2	HU2	Revisión de criterios de aceptación	4	Alumno	0
3	HU3	Definición de arquitectura del modelo	8	Alumno	0
3	HU3	Implementación y pruebas iniciales del modelo	8	Alumno	0
3	HU3	Validación cruzada y análisis de resultados	8	Alumno	0
3	HU3	Documentación del modelo	8	Alumno	70
4	HU4	Benchmarking con distintas configuraciones	8	Alumno	0
4	HU4	Optimización de tiempo de inferencia	8	Alumno	0
4	HU4	Pruebas de carga y latencia	8	Alumno	0
4	HU4	Exposición vía API REST	8	Alumno	0
5	HU5	Diseño arquitectura AWS (Lambda, API Gateway)	8	Alumno	0
5	HU5	Desarrollo funciones Lambda	8	Alumno	0
5	HU5	Pruebas de escalabilidad	8	Alumno	0
5	HU5	Seguridad básica y monitoreo	8	Alumno	0

Sprint	HU o fase	Tarea	Horas / SP	Responsable	Completado %
6	HU6	Configuración CI/CD para despliegue automático	8	Alumno	0
6	HU6	Integración con servicios de monitoreo	8	Alumno	0
6	HU6	Revisión de escalabilidad y rollback	8	Alumno	0
6	HU6	Pruebas y validación en entorno productivo	8	Alumno	0
7	HU7	Diseño de interfaz gráfica inicial	8	Alumno	0
7	HU7	Implementación de visualización de señales	8	Alumno	0
7	HU7	Exportación de datos a CSV y JSON	8	Alumno	0
7	HU7	Pruebas de usabilidad	8	Alumno	0
8	HU8	Implementación de cifrado en tránsito	8	Alumno	0
8	HU8	Implementación de cifrado en reposo	8	Alumno	0
8	HU8	Diseño de política de retención y eliminación	8	Alumno	0
8	HU8	Validación del cumplimiento de normativas	8	Alumno	0
9	HU9 Planificación y memoria	Planificación inicial del proyecto	8	Alumno	0
9	HU9 Planificación y memoria	Redacción del marco teórico	8	Alumno	0
9	HU9 Planificación y memoria	Revisión bibliográfica	8	Alumno	0
9	HU9 Planificación y memoria	Elaboración del índice de memoria	8	Alumno	0
10	HU9 Planificación y memoria	Redacción de capítulos técnicos	8	Alumno	0
10	HU9 Planificación y memoria	Integración de contenidos y estilo	8	Alumno	0
10	HU9 Planificación y memoria	Revisión ortográfica y citación	8	Alumno	0
10	HU9 Planificación y memoria	Creación de material para defensa	8	Alumno	0

Sprint	HU o fase	Tarea	Horas / SP	Responsable	Completado %
11	HU9 Finalización	Revisión global de entregables	8	Alumno	0
11	HU9 Finalización	Simulacros de defensa	8	Alumno	0
11	HU9 Finalización	Correcciones finales y ajustes	8	Alumno	0
11	HU9 Finalización	Presentación ante jurado	8	Alumno	0

12. Normativa y cumplimiento de datos (gobernanza)

En el marco del presente proyecto, se trabaja con datos de señales EEG provenientes de dataset públicos como MOABB (*Mother of All BCI Benchmarks*), un repositorio público ampliamente utilizado en la comunidad científica para el estudio de interfaces cerebro-computadora. A continuación, se analizan los aspectos legales, éticos y técnicos relacionados con la gobernanza de estos datos.

12.1. Evaluación normativa aplicable

Los datos EEG utilizados en este proyecto no contienen información de identificación personal (*PII*), ya que han sido anonimizados previamente por las entidades que los generaron. No obstante, al tratarse de datos biométricos, pueden estar sujetos a regulaciones de privacidad según la jurisdicción. Se considera la siguiente normativa:

- Reglamento General de Protección de Datos (*GDPR - Unión Europea*): establece que los datos biométricos se consideran datos sensibles, y su tratamiento requiere garantías específicas, incluso si están anonimizados.

12.2. Consentimiento de los usuarios

El dataset MOABB incluye únicamente registros cuya recolección fue realizada bajo protocolos éticos con consentimiento informado, documentado por las instituciones que lo generaron. Como usuario del dataset, el proyecto se adhiere a las licencias y restricciones de uso establecidas por los propietarios de los datos.

12.3. Restricciones de uso, compartición o publicación

Los datos EEG empleados provienen de fuentes de acceso abierto, bajo licencias que permiten su uso para fines de investigación, con la condición de no intentar identificar a los sujetos participantes ni redistribuir los datos de forma desautorizada.

12.4. Fuente y licenciamiento de los datos

Los datos utilizados pertenecen al proyecto MOABB (<https://github.com/NeuroTechX/moabb>), que ofrece múltiples datasets estandarizados para evaluación de modelos BCI. Se emplean únicamente los que están públicamente disponibles y cuentan con licencias de uso explícitas para investigación académica.

12.5. Consideraciones éticas y viabilidad legal

Desde el punto de vista legal, el uso de MOABB como fuente de datos es viable para los fines del proyecto, siempre y cuando se respeten los términos de la licencia de uso y no se intente aplicar el sistema a datos personales sin consentimiento.

12.6. Gobernanza de los datos

Se establece como principio del proyecto el cumplimiento del ciclo de vida de datos responsable, incluyendo:

- Almacenamiento seguro y cifrado de los resultados.
- Eliminación periódica de datos temporales.
- Documentación clara sobre el uso y transformación de los datos.
- Evaluación de riesgos asociados a la exposición o mal uso de los datos.

13. Gestión de riesgos

a) Identificación de los riesgos y estimación de sus consecuencias

Riesgo 1: Retrasos en la adquisición o validación de datos

- **Severidad (S): 8**
La falta de datos validados compromete el entrenamiento del modelo y la evaluación general del sistema.
- **Ocurrencia (O): 5**
Aunque los datos MOABB son de acceso abierto, la validación y el preprocesamiento pueden requerir más tiempo del esperado.

Riesgo 2: Dificultad en el despliegue en AWS Lambda por límites técnicos

- **Severidad (S): 7**
Superar límites de ejecución o memoria afectaría la escalabilidad del sistema.
- **Ocurrencia (O): 6**
La ejecución de modelos en tiempo real puede exceder los 15 segundos, especialmente en inferencia no optimizada.

Riesgo 3: Falta de experiencia previa con MLOps y DevOps

- **Severidad (S): 6**
El desconocimiento puede retrasar la integración y despliegue continuo.
- **Ocurrencia (O): 7**
Requiere curva de aprendizaje en herramientas como Docker, GitHub Actions y AWS CI/CD.

Riesgo 4: Vulnerabilidades en la seguridad de datos EEG

- **Severidad (S): 9**
La filtración o uso indebido de datos EEG podría tener consecuencias legales y éticas graves.
- **Ocurrencia (O): 4**
Aunque se usan prácticas de cifrado, errores de configuración o descuidos pueden generar brechas.

Riesgo 5: Sobrecarga de trabajo hacia el final del proyecto

- **Severidad (S): 6**
Puede comprometer la calidad de entregables finales (documentación, defensa).
- **Ocurrencia (O): 6**
Hay alta probabilidad si no se sigue el cronograma de sprints rigurosamente.

b) Tabla resumen de riesgos

ID	Riesgo	Severidad (S)	Ocurrencia (O)	Índice de Riesgo ($IR = S \times O$)
R1	Retrasos en validación de datos EEG	8	5	40
R2	Dificultad técnica en AWS Lambda	7	6	42
R3	Inexperiencia en MLOps	6	7	42
R4	Falla en la seguridad de los datos	9	4	36
R5	Sobrecarga en etapas finales	6	6	36

Cuadro 1. Resumen de riesgos y su índice de prioridad

c) Acciones de mitigación propuestas

- **R1:** establecer un sprint exclusivo para validación temprana de los datos y documentación clara de cada dataset.
- **R2:** realizar pruebas con versiones optimizadas del modelo desde etapas tempranas y monitorear tiempos de ejecución.
- **R3:** capacitación autodidacta continua en herramientas MLOps durante los primeros sprints y asistencia con plantillas preconfiguradas.
- **R4:** usar cifrado TLS/HTTPS en tránsito y AWS KMS en reposo. Definir políticas de IAM estrictas.
- **R5:** mantener reuniones semanales de control de avance. Incorporar buffer de tiempo en sprints finales.

14. Sprint Review

HU	Criterios de Aceptación	Método de Verificación	Observaciones Previstas
HU2 – Preprocesamiento EEG	Se deben aplicar filtros, normalizar señales y evaluar calidad.	Se ejecutarán scripts sobre datasets de prueba y se validará visual y estadísticamente la calidad de la señal.	Se prevé dificultad en señales con alto ruido o mala adquisición.
HU3 – Entrenamiento de Modelo	El modelo debe ser entrenado correctamente y lograr métricas mínimas en validación.	Se comprobará mediante métricas (accuracy, F1-score) en conjunto de validación del pipeline.	Posible ajuste de hiperparámetros y reconsideración del modelo base.
HU5 – Visualización EEG UI	La interfaz debe mostrar las señales y permitir exportar CSV y JSON.	Prueba funcional de la interfaz con simulaciones y test de exportación.	Se prevén ajustes menores en diseño o formatos de salida.
HU6 – Despliegue en AWS	Modelo desplegado con monitoreo y escalabilidad básica configurada.	Verificación mediante logs, alertas y disponibilidad de endpoints de inferencia.	Dependencia de configuración en AWS; posible demora en permisos y acceso.

15. Sprint Retrospective

Sprint (tipo y N)	¿Qué hacer más?	¿Qué hacer menos?	¿Qué mantener?	¿Qué empezar a hacer?	¿Qué dejar de hacer?
Sprint 0 (Planificación – No técnico)	Dedicar más tiempo a la planificación detallada y realista de tareas.	Menos sub-estimación de tiempos de documentación y revisión.	La revisión cruzada de entregables antes de enviar.	Hacer revisiones iterativas del cronograma a medida que avance el proyecto.	Dejar de asumir que todas las tareas técnicas serán lineales en duración.
Sprint 2 (Técnico – Pre-procesamiento EEG)	Invertir más en validar calidad de señal y documentación de scripts.	Menos enfoque en optimizar código antes de validarlo.	Pipeline modular para reusabilidad y claridad.	Usar validaciones automáticas en el flujo de preprocesamiento.	Parar de postergar tareas de documentación para el final.
Sprint 5 (Técnico – Extracción de características)	Hacer más pruebas unitarias en etapas tempranas del desarrollo.	Menos dependencia de notebooks desordenados.	La automatización del pipeline con scripts parametrizables.	Crear un repositorio limpio y separado para funciones reutilizables.	Dejar de trabajar sin control de versiones fino por etapa.
Sprint 9 (Técnico – Entrenamiento y evaluación del modelo)	Probar más métricas y evaluar resultados con mayor profundidad.	Menos iteraciones sin documentación intermedia de resultados.	La evaluación cruzada para robustez de resultados.	Explorar herramientas para visualización de métricas.	Dejar de almacenar resultados sólo localmente, sin respaldo en la nube.