



Aplicaciones de modelos de lenguaje de Inteligencia Artificial (LLM) en la ambientación narrativa

Autor:

Ing. Mario Gómez Alonso

Director:

Por ser definido (Por ser definida)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 17 de octubre de 2023 y el 5 de diciembre de 2023.

Índice

1. Descripción técnica-conceptual del proyecto a realizar	5
2. Identificación y análisis de los interesados	6
3. Propósito del proyecto	7
4. Alcance del proyecto	7
5. Supuestos del proyecto.	7
6. Requerimientos	8
7. Historias de usuarios (<i>Product backlog</i>).	9
8. Entregables principales del proyecto	10
9. Desglose del trabajo en tareas	10
10. Diagrama de Activity On Node.	11
11. Diagrama de Gantt	11
12. Presupuesto detallado del proyecto	13
13. Gestión de riesgos	13
14. Gestión de la calidad	15
15. Procesos de cierre	17

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	17 de octubre de 2023
1	Se completa hasta el punto 5 inclusive	31 de octubre de 2023
2	Aplicadas correcciones sobre la versión anterior y se completa hasta el punto 9 inclusive	7 de noviembre de 2023
3	Aplicadas correcciones sobre la versión anterior y se completa hasta el punto 12 inclusive	15 de noviembre de 2023
4	Aplicadas correcciones sobre la versión anterior y se completa el plan en su totalidad	23 de noviembre de 2023
5	Aplicadas correcciones sobre la versión anterior	3 de diciembre de 2023

Acta de constitución del proyecto

Buenos Aires, 17 de octubre de 2023

Por medio de la presente se acuerda con el Ing. Mario Gómez Alonso que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Aplicaciones de modelos de lenguaje de Inteligencia Artificial (LLM) en la ambientación narrativa”, consistirá esencialmente en la implementación de un servidor web que aloje una Inteligencia Artificial de tipo LLM que genere contenido narrativo a través de servicios REST, y tendrá un presupuesto preliminar estimado de 695 h (de las cuales 80h son opcionales) de trabajo y \$5.396.529,2 (del cual \$551.020,8 es el coste de las horas opcionales), con fecha de inicio 17 de octubre de 2023 y fecha de presentación pública agosto de 2024.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Ing. Hans Manuel Grenner Noguerón
Critical Match

Por ser definido
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El proyecto a realizar se enfoca en ampliar las herramientas que ofrece el producto Critical Match, creado por la empresa con el mismo nombre y el cual consta actualmente de una aplicación para Android y iOS. Concretamente, se trata de un proyecto personal donde la empresa Critical Match será el cliente.

La principal función de la App es permitir a sus usuarios la creación o búsqueda de salas de juego, también conocidas como mesas, donde pueden unirse a otros jugadores para organizar partidas de rol, ya sea en línea o de forma presencial. Las mesas ofrecen un marco narrativo para que los usuarios seleccionen la que más se ajuste a sus preferencias.

Es en este punto donde se desea ampliar la funcionalidad mediante la incorporación de servicios de Inteligencia artificial para generar, a partir de un contexto narrativo, una amplia gama de elementos adicionales. Por ejemplo, personajes no jugables que se integren coherentemente en el mundo propuesto, eventos que enriquezcan la trama y objetos, entre otros.

El estado del arte no está suficientemente avanzado en este aspecto. Las soluciones de Inteligencia Artificial como ChatGPT, aunque competentes, no suelen alcanzar los criterios de profundidad y precisión más exigentes, lo que no siempre satisface las expectativas de los jugadores de rol.

Por otro lado, existen sitios web como WorldAnvil que ofrece a sus usuarios la posibilidad de crear y gestionar sus mundos de ficción, donde la compleción de todas y cada una de las categorías y elementos del mundo recae exclusivamente en el propio usuario. Es habitual que los moderadores de las partidas utilicen estas herramientas para compilar todo el contexto narrativo en un único lugar. La empresa Critical Match quiere incorporar esta funcionalidad en su aplicación en un futuro cercano.

El objetivo principal del proyecto es crear una Inteligencia Artificial capaz de dar mejores resultados que otros motores como ChatGPT en un contexto de generación de texto de ambientación narrativa. Esto implica desarrollar un sistema de IA altamente especializado y entrenado específicamente para la generación de contenido narrativo relacionado con partidas de rol, ofreciendo una profundidad y precisión que supere las limitaciones de las soluciones generales.

Adicionalmente, se busca dividir esta funcionalidad en varios servicios REST (*Representational State Transfer Protocol* o Protocolo de Transferencia de Estado Representacional) para que la aplicación de Critical Match pueda acceder a las funciones de generación de contenido de forma remota. Esto permitirá a los organizadores de las mesas de rol acceder a las capacidades de la IA, integrando de manera transparente la ambientación narrativa generada en sus partidas y facilitando la creación de un mundo de ficción coherente y atractivo para los jugadores.

Se prevé que la Inteligencia Artificial se aloje en un servidor que sea accesible a través de peticiones REST (figura 1). Esto implica que el sistema de IA estará desplegado en un servidor remoto que estará constantemente en línea y disponible por el protocolo previamente mencionado.

En resumen, la propuesta de valor de este proyecto es la implementación de una serie de servicios REST que serán accedidos por aplicaciones externas, principalmente Critical Match.

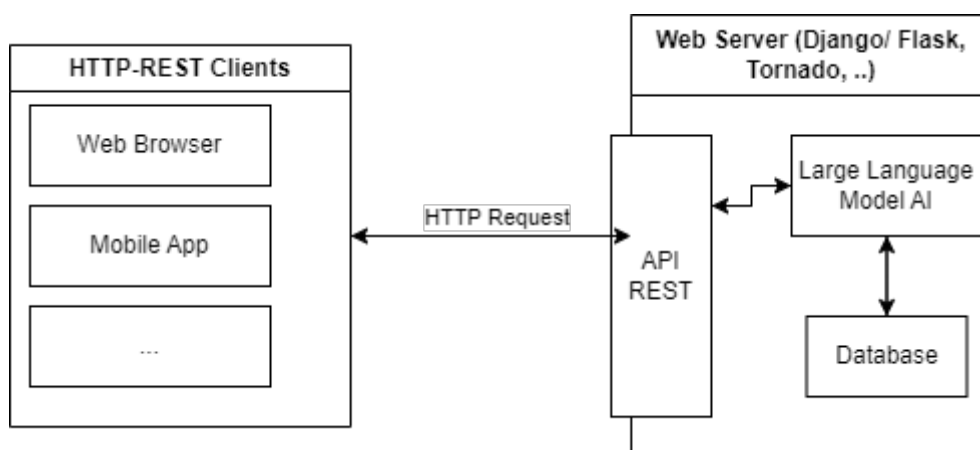


Figura 1. Diagrama en bloques del sistema.

Estos servicios permitirán la generación de elementos de ambientación narrativa, basados en el contexto del mundo proporcionado por los usuarios.

Esta iniciativa supone una evolución natural de la App Critical Match. Actualmente el producto dispone de la funcionalidad de crear mesas y reunir jugadores y esta propuesta añade valor al facilitar a los organizadores de las partidas la creación de contenido atractivo para todos los participantes.

Además, este proyecto podría encajar en otros ámbitos que actualmente están enfocados en el “World Building” al proporcionar a creadores de mundos y escritores herramientas para expandir sus ideas.

Este proyecto a su vez es escalable y permite añadir nuevos servicios con el paso del tiempo. Esto podría llevar en un futuro a la posibilidad de ofrecer un director de juego automatizado por Inteligencia Artificial que sea de interpretar las reglas y acciones de sus jugadores, llevando a cabo partidas de rol de manera autónoma.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Ing. Hans Manuel Grenner Noguerón	Critical Match	co-fundador de Critical Match
Responsable	Ing. Mario Gómez Alonso	FIUBA	Alumno
Director	Por ser definido	Por ser definida	Director Trabajo final

De esta lista podemos destacar:

- Cliente y usuario final: Ing. Hans Manuel Grenner Noguerón. Aporta contexto necesario sobre la App Critical Match y es el interesado de las herramientas que se desarrollarán en el proyecto. Punto de contacto entre el responsable del proyecto y el equipo de Critical Match.

3. Propósito del proyecto

El propósito del proyecto es el desarrollo de un servidor web que aloje una Inteligencia Artificial de tipo LLM. Esta IA será entrenada para la generación de contexto y contenido narrativo. El servidor podrá recibir peticiones a través del protocolo API REST, donde se alimentará a la IA con un contexto narrativo. El servidor responderá las peticiones con la salida aportada por el modelo dependiendo de la entrada de texto y el tipo de servicio solicitado.

4. Alcance del proyecto

El alcance del proyecto incluye lo siguiente:

- Desarrollo de un servidor web.
- Entrenamiento de una Inteligencia Artificial de tipo LLM.
- Análisis y tratamiento de los datos utilizados para el entrenamiento de la IA.
- La integración de un conjunto esencial de servicios REST.
- Servicios adicionales, si entran dentro de las horas estimadas del proyecto.

El alcance del proyecto no incluye:

- Página web.
- Securización del servidor web ni de ninguno de sus servicios REST.
- Rendimiento de los servicios que garanticen tiempos de respuesta cercanos a tiempo real.

En resumen, el proyecto consistirá de un prototipo que aporte funcionalidad en un tiempo de respuesta razonable, pero es posible que no se ajuste a los requerimientos de un sistema puesto en producción. Con esto aclarado, la intención siempre será desarrollar el proyecto aportando el mayor rendimiento posible.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- No se desarrollará la Inteligencia Artificial desde cero, si no que se utilizará una pre-entrenada y de código abierto. Partiendo de esa base, se hará un proceso de *fine-tuning* para especializar el modelo en los servicios de generación de contenido narrativo.
- No se dispone de un set de datos por parte del cliente. Se analizarán sets de datos con licencia abierta en la red y se emplearán aquellos que se ajusten al proyecto.
- Comunicación fluida con el Equipo de Critical Match para seguimiento del proyecto y aportación de ideas.

- El proyecto se realizará con una máquina cuyas especificaciones permitan tanto el desarrollo del código como su ejecución, además del proceso de *fine-tuning* de la Inteligencia artificial.
- La disponibilidad para realizar las tareas del proyecto, las cuales se describirán más adelante, será de entre 20 y 30 horas semanales.

6. Requerimientos

1. Requerimientos del servidor:

- 1.1. El servidor debe alojar y administrar la información relativa al *dataset* y la configuración del modelo LLM.
- 1.2. El servidor debe contar con los procesos necesarios para entrenar la Inteligencia Artificial.
- 1.3. Al ser desplegado, el servidor deberá iniciar el módulo LLM y utilizarlo en el procesamiento de peticiones entrantes.
- 1.4. El servidor debe dar acceso a clientes externos a través del protocolo API REST.
- 1.5. Los servicios REST deben aceptar entrada de texto en varios formatos: pdf, txt, word o texto plano en el cuerpo de la petición.
- 1.6. Los servicios REST devolverán la respuesta en formato simple HTML para su cómoda visualización en un navegador.
- 1.7. El prototipo del servidor debe de tener una disponibilidad del 100 % durante las pruebas y aceptar la carga de trabajo de peticiones individuales. Opcionalmente, sería deseable que maneje el mayor número de peticiones simultáneas posible.

2. Requerimientos del módulo LLM:

- 2.1. El módulo LLM debe aceptar entrada de texto y generar texto como salida.
- 2.2. El módulo LLM, si el texto de entrada es legible, debe aportar una respuesta con un detalle y profundidad razonables, además de ser coherente con las instrucciones recibidas.
- 2.3. El módulo debe disponer de múltiples métodos para los cuales, dada la misma entrada de texto, devolverá información enfocada en un aspecto concreto de la narrativa. Por ejemplo, un método centrado en la generación de personajes, otro en la descripción de localizaciones, un tercero de eventos y así sucesivamente.
- 2.4. El tiempo de respuesta del módulo LLM debe estar, teniendo en cuenta el *hardware* con el que se dispone, la extensión del texto de entrada y que no se garantiza un rendimiento cercano a tiempo real, en un rango de tiempo razonable para un servicio REST (no más de 5 minutos).

3. Requerimiento de testing:

- 3.1. El proyecto contará con test automáticos que probarán el correcto funcionamiento de los módulos no relacionados con el modelo LLM.
- 3.2. El cliente podrá validar los test automáticos a través de un análisis del código.
- 3.3. Se validará con el cliente tanto el módulo LLM como el servidor.

4. Requerimientos de documentación:

- 4.1. Se entregará un documento de administrador que explicará brevemente el sistema, qué comandos de ejecución tiene y los requisitos *hardware* necesarios para su correcto funcionamiento. Además, contará con una descripción de todos los servicios REST disponibles en el servidor y un ejemplo de cómo acceder a ellos. Este documento también recopilará las instrucciones de despliegue con la creación del entorno virtual, instalación de librerías y el entrenamiento del modelo, entre otros.
 - 4.2. Se entregará un documento con un breve análisis de los resultados del módulo LLM que se hayan expuesto al cliente durante las reuniones de seguimiento.
5. Requerimientos legales:
- 5.1. Cualquier información del cliente que se utilice en el desarrollo del proyecto estará amparada por derechos de privacidad y propiedad intelectual.
 - 5.2. El set de datos utilizado para el entrenamiento del módulo LLM debe tener una licencia libre o que se ajuste a uso académico.
 - 5.3. Todas las librerías de código empleadas en el desarrollo del proyecto deben tener una licencia de código abierto.

7. Historias de usuarios (*Product backlog*)

A las historias de usuario se les asignará una puntuación (*Story Points*) que seguirá la sucesión de Fibonnaci en un rango de 1 a 13 en función de la dificultad. A mayor dificultad más tiempo de desarrollo se le dedicará a la historia de usuario y mayores serán los riesgos que acarrea.

A continuación, se describen brevemente los roles:

- Administrador: persona o equipo encargado de, sin necesidad de tener conocimiento del código, controlar y monitorizar el estado del sistema en producción.
- Desarrollador: persona o equipo con conocimientos de análisis de datos y/o programación encargado de mantener el software.
- Usuario: conjunto de personas interesadas en consumir el producto que ofrece el sistema.

La lista de historias de usuario es la siguiente:

- Como administrador quiero contar con un servidor web y con las herramientas necesarias para su despliegue y detención. (2 puntos)
- Como desarrollador deseo tener herramientas que comprueben el buen funcionamiento del código. (5 puntos)
- Como desarrollador necesito ser capaz de actualizar el *dataset* y reentrenar la inteligencia artificial. (8 puntos)
- Como usuario quiero enviar peticiones API REST al servidor y, aportando un contexto inicial, obtener ambientación narrativa relativa al servicio que he solicitado. (13 puntos)
- Como usuario necesito adjuntar archivos de texto en sus formatos habituales (pdf, txt, doc) y que el servicio sea capaz de interpretarlos. (2 puntos)

8. Entregables principales del proyecto

- Código fuente.
- Manual de administrador.
- Análisis del *dataset*.
- Recopilación de informes de seguimiento del módulo LLM.
- Memoria técnica.

9. Desglose del trabajo en tareas

1. Tareas del servidor (190 h)

- 1.1. Desarrollo inicial del servidor web y página principal. (20 h)
- 1.2. Implementación del modelo de datos y ORM (*Object-Relational Mapping*). (40 h)
- 1.3. Implementación de pruebas automáticas. (40 h)
- 1.4. Validación de pruebas automáticas. (10 h)
- 1.5. Implementación del comando para reentrenar el módulo LLM. (20 h)
- 1.6. Implementación de los servicios API REST del paquete MVP (*Minimum Viable Product* o Producto Viable Mínimo). (20 h)
- 1.7. Opcional. Implementación de servicios REST adicionales. (40 h)

2. Tareas del módulo LLM (360 h)

- 2.1. Obtención y análisis inicial del *dataset*. (40 h)
- 2.2. Desarrollo de software necesario para el tratamiento de los datos (40 h)
- 2.3. Implementar *pipeline* de procesamiento del *dataset*. (40 h)
- 2.4. *Fine-tuning* del módulo LLM inicial. (40 h)
- 2.5. Implementación de los métodos del módulo LLM para el paquete MVP. (40 h)
- 2.6. Presentación al cliente y análisis inicial de los resultados. (20 h)
- 2.7. Mejora del *pipeline* de procesamiento del *dataset*. (40 h)
- 2.8. *Fine-tuning* final del módulo LLM. (40 h)
- 2.9. Presentación al cliente y análisis final de los resultados. (20 h)
- 2.10. Opcional. Implementación de métodos del módulo LLM adicionales. (40 h)

3. Tareas de documentación (145 h)

- 3.1. Plan de proyecto. (20h)
- 3.2. Presentación del plan de proyecto. (10h)
- 3.3. Manual de administrador. (10h)
- 3.4. Documento de análisis del *dataset*. (25 h)
- 3.5. Recopilación de informes de seguimiento del módulo LLM. (20 h)
- 3.6. Memoria técnica. (40 h)
- 3.7. Presentación y defensa del trabajo final. (20 h)

Cantidad total de horas: 695 h, de las cuales 80 h son opcionales.

10. Diagrama de Activity On Node

A continuación se presenta en la figura 2 el diagrama de *Activity on Node* de las tareas del proyecto.

En la leyenda se indica el color de cada grupo de tareas. Además, se puede diferenciar el camino crítico de 445 horas en rojo, los caminos semicríticos en amarillo y los caminos opcionales en línea discontinua. El valor del tiempo de cada tarea está en horas.

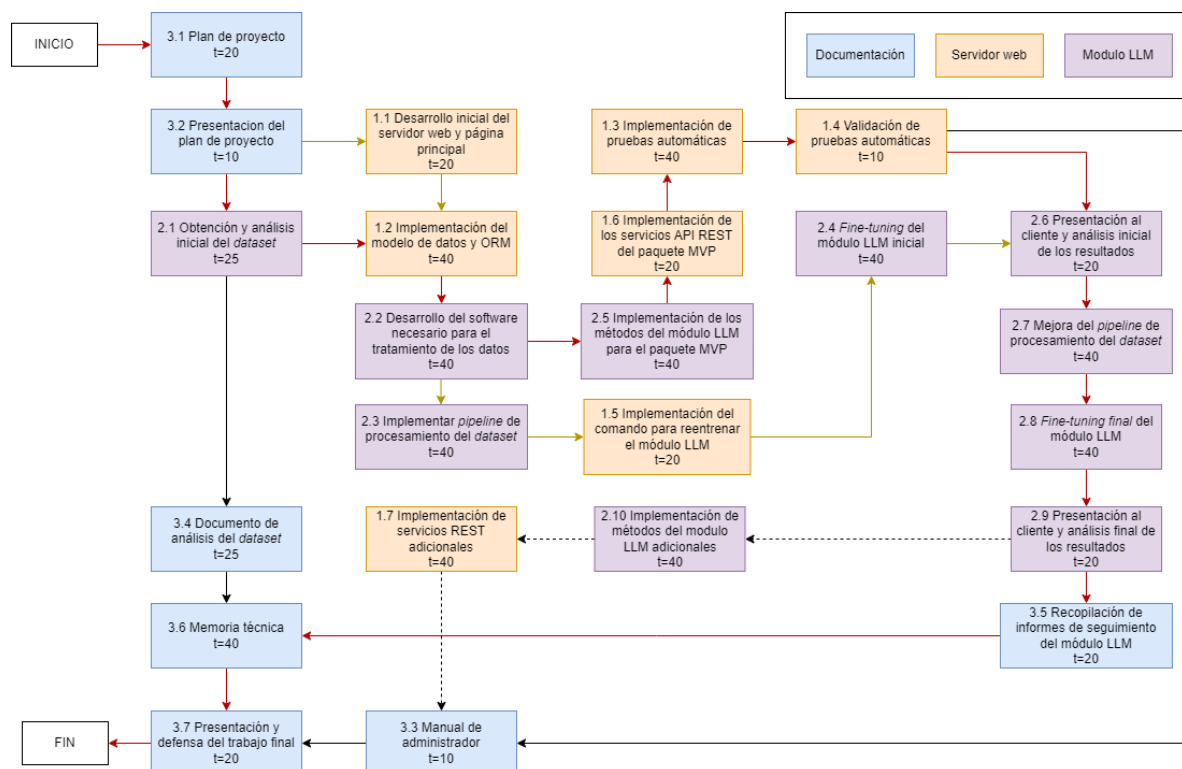


Figura 2. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

En la figura 3 se presenta el diagrama de Gantt del proyecto. Al solo disponer de una persona trabajando en el proyecto, las tareas no pueden paralelizarse y están dispuestas en cascada. Cada día equivale a 4 horas de trabajo en el proyecto de acuerdo a lo descrito en la sección 5.

Además, las tareas opcionales no aparecen para no desvirtuar el diagrama estimado. En caso de aplicarse estos objetivos secundarios, se enviará una actualización de dicho diagrama a los interesados.

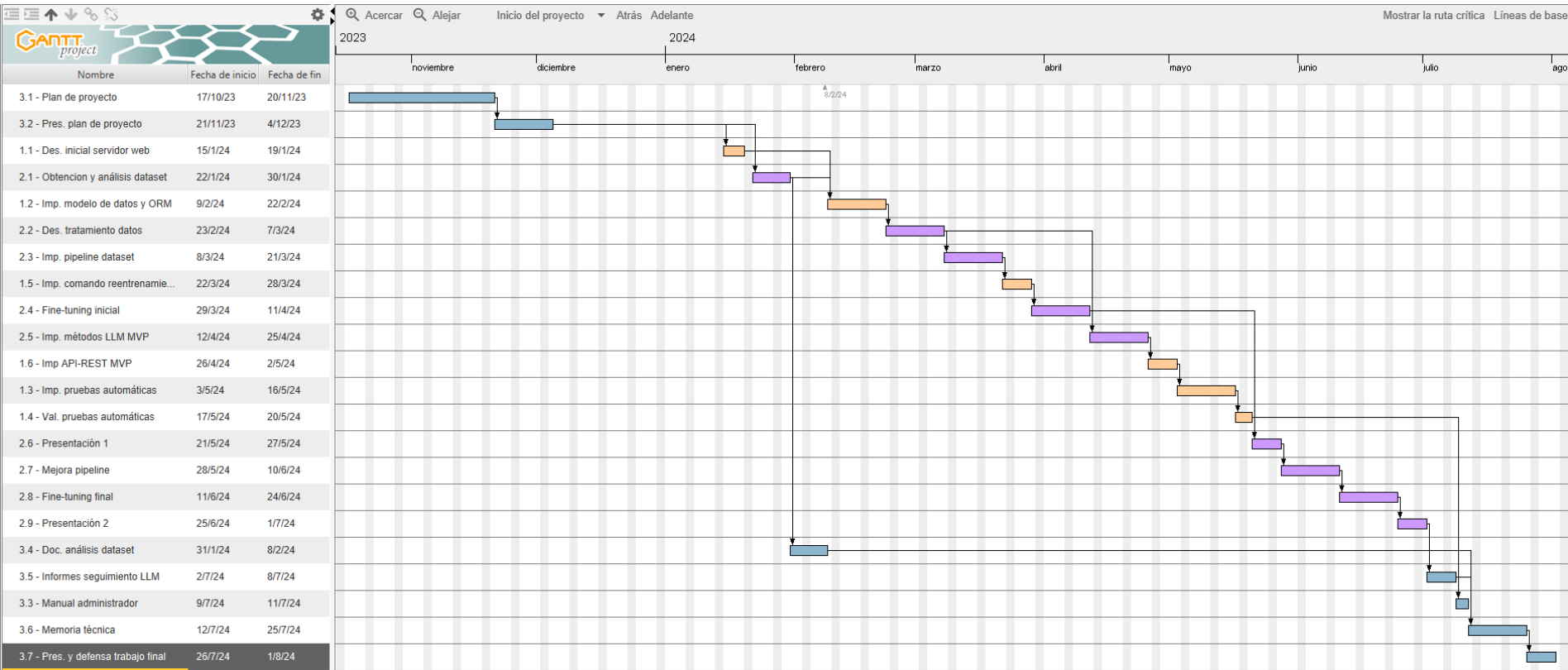


Figura 3. Diagrama de Gantt.

12. Presupuesto detallado del proyecto

Para el cómputo del valor de las horas de ingeniería se ha tomado como referencia el salario medio de un ingeniero software en España en el año 2023, cuyo valor es de €18,08 por hora. Dicho valor se ha convertido a pesos argentinos bajo la tasa representativa del mercado del día 14/11/2023, una equivalencia de 380,96 pesos por euro. Es decir, el valor del trabajo de ingeniería es de \$6.887,76 por hora.

También se utilizará el valor de conversión de euros a pesos para la estimación del coste del *hardware* de trabajo.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario (ARS)	Valor total (ARS)
Horas del Responsable (MVP)	615	6.887,76	4.235.972,4
Horas del Responsable (Opcional)	80	6.887,76	551.020,8
SUBTOTAL			4.786.993,2
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario (ARS)	Valor total (ARS)
Hardware de trabajo	1	609.536	609.536
SUBTOTAL			609.536
TOTAL			5.396.529,2

Se aclara de nuevo que \$551.020,8 del coste total se atribuyen a costes opcionales en caso de que se decida añadir funcionalidad adicional.

13. Gestión de riesgos

a) Se identificaron los siguientes riesgos para el proyecto:

Riesgo 1: no disponer de un modelo de tipo LLM pre entrenado que sea adecuado para el proyecto.

- Severidad (S): 10. Se trata de un riesgo crítico ya que la ausencia de un modelo supondría la imposibilidad de realizar el proyecto.
- Ocurrencia (O): 3. La probabilidad de ocurrencia es baja porque hay múltiples opciones disponibles.

Riesgo 2: *dataset* inadecuado para el *fine-tuning* del modelo.

- Severidad (S): 8. Se trata de un riesgo severo ya que un entrenamiento inadecuado del modelo induciría a resultados insuficientes para el cliente.
- Probabilidad de ocurrencia (O): 5. La probabilidad de ocurrencia es media debido a varios factores. Es necesario encontrar un *dataset* lo suficientemente completo para empezar a procesarlo. Además, no se dispone de conocimiento pleno a la hora de tratar adecuadamente con datos de esta naturaleza. Se supone que se adquirirá experiencia en el desarrollo del curso de especialización.

Riesgo 3: capacidad del *hardware* insuficiente.

- Severidad (S): 5. La severidad del riesgo es media porque provocaría retrasos en los tiempos de *fine-tuning* y de predicción del modelo. Esto podría provocar retrasos en el proyecto y un rendimiento insuficiente en los servicios.
- Ocurrencia (O): 3. La ocurrencia del riesgo es baja porque se cuenta con un equipo con una placa de video de buena calidad. Además, es improbable que dicho equipo se dañe o se pierda.

Riesgo 4: retrasos en el proyecto.

- Severidad (S): 3. La severidad del riesgo es baja ya que el cliente es flexible en la extensión de los plazos.
- Ocurrencia (O): 6. La ocurrencia del riesgo es alta al tratarse de un proyecto que se inicia en mitad del curso de especialización. Es posible que tanto la falta de experiencia como la carga de trabajo externa cause que no se disponga de todo el tiempo efectivo estimado para la realización del proyecto.

Riesgo 5: resultado final insuficiente.

- Severidad (S): 8. Se trata de un riesgo severo debido a que invalida gran parte de los requerimientos del proyecto.
- Ocurrencia (O): 4. Existe la posibilidad de que, a pesar de trabajar correctamente con el Modelo LLM y el *dataset*, el resultado final no sea esperado por el cliente. No existe una forma objetiva de ponderar el éxito del modelo.

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN=S \times O$)

Riesgo	S	O	RPN	S*	O*	RPN*
1. Modelo LLM inadecuado.	10	3	30	10	2	20
2. <i>Dataset</i> inadecuado.	8	5	40	8	3	24
3. Capacidad del <i>hardware</i> insuficiente.	5	3	15	-	-	-
4. Retrasos en el proyecto.	3	6	18	-	-	-
5. Resultado final insuficiente.	8	4	32	6	3	18

Se van a adoptar medidas de mitigación para aquellos riesgos con valores RPN mayores a 30.

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: no disponer de un modelo de tipo LLM pre entrenado que sea adecuado para el proyecto. El plan de mitigación consiste en dedicar tiempo y recursos adicionales en la búsqueda de modelos LLM. Esto incluye considerar modelos en otros lenguajes de programación, siempre que se cuente con la experiencia necesaria. También buscar documentación online o videos de divulgación de modelos de tipo LLM.

- Severidad (S^*): 10. La severidad no cambia ya que la integridad del proyecto depende de un modelo LLM compatible.
- Probabilidad de ocurrencia (O^*): 2. La probabilidad de ocurrencia es menor debido a los recursos adicionales empleados.

Riesgo 2: *dataset* inadecuado para el *fine-tuning* del modelo. El plan de mitigación consiste en dedicar tiempo y recursos adicionales en la búsqueda y análisis de *Dataset*. La redacción del documento de análisis del dataset también ayuda a mitigar el riesgo.

- Severidad (S^*): 8. La severidad del riesgo no cambia.
- Probabilidad de ocurrencia (O^*): 3. La probabilidad de ocurrencia es menor debido a los recursos adicionales empleados.

Riesgo 3: resultado final insuficiente. El plan de mitigación consiste en comunicarse con el cliente para acordar criterios de evaluación del *output* de los servicios que sean razonables. Siempre que sea posible se harán demostraciones del *software* en las reuniones de seguimiento para detectar puntos de mejora antes de la finalización del proyecto. Los test automáticos aportarán cobertura de código en la medida de lo posible.

- Severidad (S^*): 6. La severidad baja al conocer las expectativas del cliente.
- Probabilidad de ocurrencia (O^*): 3. La probabilidad de ocurrencia es menor debido a la demostración preliminar de resultados.

14. Gestión de la calidad

- Req #1.2: el servidor debe contar con los procesos necesarios para entrenar la Inteligencia Artificial.
 - Verificación: consultar la base de datos para observar si los parámetros del módulo LLM se actualizan tras ejecutar el comando de entrenamiento.
 - Validación: en el manual de administrador se explicará el comando que entrena el módulo LLM. Además, de ser requerido, se hará una demostración del rendimiento del modelo antes y después del *fine-tuning*.
- Req #1.3: al ser desplegado, el servidor deberá iniciar el módulo LLM y utilizarlo en el procesamiento de peticiones entrantes.
 - Verificación: confirmar a través de inspección en el código y/o archivos de registro que el proceso de despliegue del servidor web incluye la inicialización del módulo LLM.
 - Validación: se confirmará con el cliente que los servicios REST estén funcionando y su *output* se ajusta a los criterios acordados.
- Req #1.5: los servicios REST deben aceptar entrada de texto en varios formatos: pdf, txt, word o texto plano en el cuerpo de la petición.
 - Verificación: se harán test automáticos que validen que la extracción de texto funciona correctamente.

- Validación: en las peticiones REST se utilizarán archivos adjuntos como parámetros de entrada y se confirmará una respuesta acorde.
- Req #1.6: los servicios REST devolverán la respuesta en formato simple HTML para su cómoda visualización en un navegador.
 - Verificación: inspección de código.
 - Validación: la respuesta de las peticiones REST se visualizará en el navegador.
- Req #2.2: el módulo LLM, si el texto de entrada es legible, debe aportar una respuesta con un detalle y profundidad razonables, además de ser coherente con las instrucciones recibidas.
 - Verificación: se probará el comportamiento del módulo en varias pruebas a lo largo del desarrollo, las cuales podrán incluirse en el documento de análisis del *dataset* o en los informes de seguimiento del módulo LLM.
 - Validación: se confirmará con el cliente que los resultados de las peticiones REST son satisfactorias.
- Req #3.1: el proyecto contará con tests automáticos que probarán el correcto funcionamiento de los módulos no relacionados con el modelo LLM.
 - Verificación: inspección de código.
 - Validación: se describirá brevemente al cliente la lista de tests automáticos, mostrando su código si así se solicita.
- Req #4.1: se entregará un documento de administrador que explicará brevemente el sistema, qué comandos de ejecución tiene y los requisitos *hardware* necesarios para su correcto funcionamiento.
 - Verificación: se seguirá el manual de administrador para comprobar que se despliega el proyecto sin problemas.
 - Validación: el cliente seguirá el manual de administrador para comprobar que se despliega el proyecto sin problemas.
- Req #4.2: se entregará un documento con un breve análisis de los resultados del módulo LLM que se hayan expuesto al cliente durante las reuniones de seguimiento.
 - Verificación: se asegurará que el contenido del documento sea adecuado.
 - Validación: el cliente confirmará que el documento está completo.
- Req #5.1: cualquier información del cliente que se utilice en el desarrollo del proyecto estará amparada por derechos de privacidad y propiedad intelectual.
 - Verificación: inspección de código. Los datos sensibles estarán ofuscados y/o cifrados.
 - Validación: será imposible acceder a los datos sensibles a través del servidor web.

15. Procesos de cierre

El proceso de cierre se realizará de la siguiente forma:

- Reunión entre el responsable del proyecto y el cliente para analizar los resultados finales. Se listarán conclusiones del proyecto, viabilidad para incluirlo en el producto Critical Match y posibles caminos de mejora.
- Reunión entre el responsable del proyecto y el director de proyecto para analizar los resultados del proyecto de acuerdo con el plan definido.
- El responsable del proyecto escribirá un breve reporte, en forma de listado, de los aciertos y errores del proyecto. También incluirá los desafíos inesperados a los que se enfrentó y las acciones tomadas para superarlos.
- Preparación de una presentación del trabajo y defensa pública.
- Al final de la defensa pública el responsable del proyecto dedicará unos minutos en realizar el acto de agradecimiento a todos los interesados, en especial al equipo de trabajo y colaboradores. Los agradecimientos también quedarán reflejados en la presentación y en la memoria del trabajo final.
- Al tratarse de un proyecto personal, no se prevé que haya financiación alguna de los gastos correspondientes del proyecto.