

# **ANEXO I: Plan de proyecto.**

*Chatbot aplicado a la resolución de consultas en asignaturas  
de Bases de Datos*

**Trabajo de Fin de Grado**

**INGENIERÍA INFORMÁTICA**



**VNiVERSiDAD  
D SALAMANCA**

**Septiembre de 2023**

**Autor:**

*Manuel Santa Isabel Mayo*

**Tutoras:**

*Ana Belén Gil González*

*Ana De Luis Reboredo*



## ÍNDICE DE CONTENIDO

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
<b>1. ESTIMACIÓN TEMPORAL.....</b>	<b>2</b>
<b>2.1. Complejidad actores del sistema (UAW) .....</b>	<b>3</b>
<b>2.2. Complejidad de los casos de uso (UUCW) .....</b>	<b>4</b>
<b>2.3. Factores de complejidad Técnica (TCF) .....</b>	<b>6</b>
<b>2.4. Factores de complejidad del Entorno (ECF).....</b>	<b>8</b>
<b>2.5. Cálculo de la estimación del esfuerzo .....</b>	<b>9</b>
<b>2. PLANIFICACIÓN TEMPORAL.....</b>	<b>12</b>
<b>3.1. Elaboración de un calendario de trabajo .....</b>	<b>13</b>
<b>3.2. Identificación de tareas/subtareas .....</b>	<b>14</b>
<b>3.3. División temporal y de recursos.....</b>	<b>24</b>
<b>3.4. Relación entre tareas .....</b>	<b>30</b>
<b>3.5. Diagrama de Gantt.....</b>	<b>34</b>



## ÍNDICE DE FIGURAS

Figura 1. Definición de actores y Casos de Uso en EZEstimate.....	9
Figura 2. Definición de factores de complejidad técnica y del entorno en EZEstimate .....	10
Figura 3. Cálculo de la estimación temporal en EZEstimate.....	10
Figura 4. Configuración del calendario en Microsoft Project.....	13
Figura 5. Diagrama de Gantt 1 .....	35
Figura 6. Diagrama de Gantt 2.....	36
Figura 7. Diagrama de Gantt 3.....	37
Figura 8. Diagrama de Gantt 4.....	38
Figura 9. Diagrama de Gantt 5.....	39
Figura 10. Diagrama de Gantt 6.....	40
Figura 11. Diagrama de Gantt 7.....	41
Figura 12. Diagrama de Gantt 8.....	42
Figura 13. Diagrama de Gantt 9.....	43



## ÍNDICE DE TABLAS

Tabla 1. Niveles de complejidad de actores .....	3
Tabla 2. Pesos de los actores del sistema .....	3
Tabla 3. Niveles de complejidad de los Casos de Uso .....	4
Tabla 4. Complejidad de los Casos de Uso del sistema.....	4
Tabla 5. Análisis de los factores de complejidad técnica del sistema .....	6
Tabla 6. Análisis de los factores de complejidad del entorno del sistema.....	8
Tabla 7. Definición de las tareas del proyecto .....	14
Tabla 8. Repartición del tiempo entre las diferentes iteraciones.....	24
Tabla 9. Asignación temporal y de recursos por tarea.....	24
Tabla 10. Relaciones entre tareas.....	30







# 1. INTRODUCCIÓN

El plan de proyecto es el documento que recoge todas las estimaciones y planificaciones para la creación de nuestro proyecto software.

Para la realización de estas estimaciones, nos basaremos en el esquema de Proceso Unificado. De esta forma podremos identificar los distintos casos de uso, su complejidad y su planificación de forma incremental e iterativa.

Después realizaremos una planificación temporal donde dividiremos nuestro proyecto en *tareas*. Todas las relaciones y asignaciones de dichas tareas se mostrarán en un diagrama de Gantt, mostrando de forma gráfica la planificación temporal que queremos realizar.

En resumen, buscamos una estimación y planificación a la hora de comenzar con nuestro proyecto de software.

## 2. ESTIMACIÓN TEMPORAL

Dentro de la estimación temporal vamos a realizar la **estimación de costes**.

Crearemos predicciones sobre la cantidad de esfuerzo que llevará construir el proyecto Software. En otras palabras, buscaremos tener una estimación de la duración del proyecto.

Usaremos un modelo de coste, que se basa en datos empíricos para proporcionar una estimación de la duración del proyecto.

Para poder aplicar este modelo, usaremos la métrica de los Puntos de Casos de Uso (**UCP**) basados en el esquema del Proceso Unificado.

Para poder realizar todos los cálculos de la estimación de costes con UCPs usaremos la herramienta **EZEstimate**.

Los UCP se dividirán en tres variables:

- **UUCP**
  - **UAW** Complejidad de los actores del sistema
  - **UUCW** Complejidad de los casos de uso
- **TCF** Factores de complejidad técnica
- **ECF** Factores de complejidad del entorno

## 2.1. Complejidad actores del sistema (UAW)

Clasificaremos los distintos actores que intervienen en nuestro sistema. Para ello se dividirán en tres niveles de complejidad:

Tabla 1. Niveles de complejidad de actores

<b>Simple</b>	Si el actor es un sistema y se comunica mediante una API.	<b>Peso 1</b>
<b>Medio</b>	Si el actor es un sistema y la aplicación se comunica con él mediante un protocolo (Internet).	<b>Peso 2</b>
<b>Complejo</b>	Persona con una interfaz gráfica	<b>Peso 3</b>

Ahora, en nuestro caso particular encontraremos cinco actores. Cuatro de ellos tendrán una interfaz gráfica, y el último será el propio sistema. Los actores **ACT-01 Normal**, **ACT-02 Suscriptor** y **ACT-03 Premium** van a diferenciarse entre ellos en el aspecto de límites de llamadas a ChatGPT, pero comparten las mismas características de interfaz y funcionalidades, por tanto, aunque son actores complejos, sólo los contaremos como una unidad.

El actor **ACT-04 Usuario no registrado** será un actor complejo. Y, por último, el actor **ACT-05 Sistema** se tratará de un sistema el cual engloba tanto llamadas a una API propia como una API externa. Se trata de un actor de complejidad simple.

Teniendo en cuenta esta clasificación y el contexto de nuestro proyecto, nuestros actores tendrán los siguientes pesos:

Tabla 2. Pesos de los actores del sistema

<b>Actor</b>	<b>Complejidad</b>	<b>Peso</b>
<b>ACT-01 Normal</b>	Complejo	3
<b>ACT-02 Suscriptor</b>	Complejo	-
<b>ACT-03 Premium</b>	Complejo	-
<b>ACT-04 Usuario no registrado</b>	Complejo	3
<b>ACT-05 Sistema</b>	Simple	1

## 2.2. Complejidad de los casos de uso (UUCW)

El siguiente Punto de Caso que analizaremos será la complejidad de los casos de uso. Para medir dicha complejidad, vamos a tomar como dato el número de transacciones que se realizan en un caso de uso.

Una transacción será la acción de un actor y la respuesta que conlleva. Una transacción no será el realizar una sola acción.

La complejidad de los casos de uso se clasifica de la siguiente manera:

*Tabla 3. Niveles de complejidad de los Casos de Uso*

<b>Simple</b>	Menos de 4 transacciones
<b>Medio</b>	Entre 4 a 7 transacciones
<b>Complejo</b>	Más de 7 transacciones

Teniendo clara esta clasificación, nuestros casos de uso tendrán la siguiente complejidad:

*Tabla 4. Complejidad de los Casos de Uso del sistema*

Caso de Uso	Transacciones	Complejidad
UC-01	1	Simple
UC-02	1	Simple
UC-03	1	Media
UC-04	2	Simple
UC-05	2	Simple
UC-06	3	Simple
UC-07	3	Simple
UC-08	1	Simple
UC-09	4	Media
UC-10	2	Simple
UC-11	2	Simple
UC-12	4	Media
UC-13	1	Simple
UC-14	3	Simple
UC-15	1	Simple
UC-16	1	Simple
UC-17	1	Simple
UC-18	2	Simple
UC-19	4	Media
UC-20	2	Simple
UC021	1	Simple
UC-22	2	Simple
UC-23	3	Simple
UC-24	4	Media
UC-25	5	Media

**Manuel Santa Isabel Mayo**

<b>UC-26</b>	3	Simple
<b>UC-27</b>	2	Simple

### 2.3. Factores de complejidad Técnica (TCF)

El tercer punto que tendremos en cuenta serán los factores de complejidad técnica. Se trata de una lista de trece factores. A cada factor se le asigna un **peso (W)** acorde a su impacto y una complejidad percibida (F) correspondiente a la **percepción de complejidad (F)** correspondiente a la percepción de complejidad que tiene el equipo de desarrollo. Esta percepción de complejidad tiene un valor del cero al cinco.

Siempre debemos justificar nuestra percepción de complejidad y por qué se aplica dentro de nuestro proyecto.

Tabla 5. Análisis de los factores de complejidad técnica del sistema

Factor de complejidad técnica	Peso	Percepción de complejidad	Justificación
<b>Sistema distribuido</b>	2	2	El sistema se distribuirá entre un servicio frontend de ReactJS y un backend Django. Además, deberá disponer de un servidor de bases de datos PostgreSQL. No es un proyecto con una distribución muy grande pero sí existe cierta complejidad. Además, solo se realizará un servidor local.
<b>Rendimiento</b>	2	2	El sistema debe resultar ágil al tratar con las peticiones de los usuarios, pero no es un factor tan importante. Las dudas de los usuarios se responderán nada más se reciba la respuesta de ChatGPT.
<b>Eficiencia del usuario final</b>	1	1	Al tratarse de una web de aprendizaje de bases de datos, no necesitamos ningún usuario con nivel de conocimiento.
<b>Procesamiento interno complejo</b>	1	4	El sistema deberá tratar los datos tanto de usuarios como de las preguntas y respuestas que hacemos a la API de ChatGPT. Debemos tratar cada una de las respuestas que recibimos.
<b>Reusabilidad</b>	1	5	El sistema está pensado para poder implementarse con otra asignatura de aprendizaje, como poder cambiar el modelo de ChatGPT por otro modelo que pueda aparecer en un futuro. Al

			final ChatGPT aunque engloba una parte muy importante del sistema, siempre puede cambiarse en un futuro por otra inteligencia artificial generativa.
<b>Facilidad de instalación</b>	0.5	2	Para un usuario no se requiere ninguna complejidad de instalación. No obstante, para un desarrollador se encuentra con cierto nivel de abstracción debido a todos los paquetes de NodeJS que se deben instalar.
<b>Facilidad de uso</b>	0.5	2	La interfaz explicará gran parte de cómo usar la web, entender el chatbot, etc. Hay una cierta complejidad dentro de cómo funciona ReactJS y cierto nivel también en la creación de la red neuronal.
<b>Portabilidad</b>	2	1	El sistema está pensado para funcionar en el SO Windows.
<b>Facilidad de cambio</b>	1	3	Añadir nuevas funcionalidades o características resulta relativamente sencillo debido al estar usando el framework de ReactJs
<b>Concurrencia</b>	1	1	Al tratarse de un servicio web probado en local, la concurrencia no es un factor a tener en cuenta.
<b>Características especiales de seguridad</b>	1	4	Debemos tener especial cuidado con las llamadas que un usuario haga a ChatGPT. Debemos crear prompts guiados por la ingeniería de prompts.
<b>Acceso directo a terceras personas</b>	1	0	El proyecto no está pensado para dar acceso a terceras personas.
<b>Se requiere entrenamiento especial del usuario</b>	1	0	El usuario no requiere de ningún entrenamiento para usar el servicio



## 2.4. Factores de complejidad del Entorno (ECF)

El cuarto y último punto que mediremos serán los factores del entorno. Se medirán de la misma manera que los puntos anteriores. En este caso solo serán ocho factores los que tendremos en cuenta.

Estos ocho factores están más relacionados con el equipo que va a trabajar en el proyecto. Como hicimos anteriormente, todas las percepciones de complejidad deben estar justificadas.

Tabla 6. Análisis de los factores de complejidad del entorno del sistema

Factor del entorno	Peso	Complejidad percibida	Justificación
<b>Familiaridad con UML</b>	1.5	2	Experiencia con varios proyectos a lo largo de la carrera de Ingeniería con UML. No obstante, no se considera con una experiencia alta en la materia.
<b>Experiencia con la aplicación</b>	0.5	1	Casi nula experiencia con las herramientas que se usan en el proyecto, por tanto, hay una gran inversión de tiempo para aprender a usarlas de forma correcta.
<b>Experiencia en la orientación objetos</b>	1	4	Alto grado en programación orientada a objetos gracias al grado.
<b>Capacidad de los analistas</b>	0.5	2	Cierto grado de conocimiento para detectar posibles problemas, no obstante, la falta de conocimiento en las herramientas dificulta el análisis.
<b>Motivación</b>	1	5	Al tratarse de un proyecto con herramientas de inteligencia artificial la motivación es muy elevada.
<b>Estabilidad de los requisitos</b>	2	3	No se prevén grandes cambios en los requisitos establecidos. Al tratarse de un pro
<b>Trabajadores a tiempo parcial</b>	-1	1	El proyecto está formado por un solo desarrollador, por tanto, no existe ningún límite.
<b>Dificultad del lenguaje de programación</b>	1	4	Aunque Django es relativamente sencillo, el uso de ReactJS conlleva muchas horas de aprendizaje.

## 2.5. Cálculo de la estimación del esfuerzo

El último paso que tenemos que realizar es el cálculo de la estimación del esfuerzo. Para ello usaremos la herramienta **EZEstimate** junto con todos los valores que se han ido exponiendo en las tablas anteriores.

Lo primero que vamos a añadir dentro del cálculo serán los distintos actores y los casos de uso. Solo debemos añadir qué complejidad tienen cada uno de ellos.

The screenshot displays the EZEstimate software interface, divided into several sections:

- Summary:** Contains input fields for 'Total Modules' (1), 'Excel Report' (Generate Report button), 'Use cases' (Simple: 22, Average: 5, Complex: 0), and 'Actors' (Simple: 1, Average: 0, Complex: 2).
- Tech / Env Factors:** Includes buttons for 'Set Tech Factor' and 'Set Env Factors'.
- Use case / Actor List:** A table with columns for Id, Module, Type, Name, and complexity. It lists 25 items, including actors and use cases with their respective complexities.

Id	Module	Type	Name	complexity
1	DBCapy	Actor	UsarioNoRegist...	Complex
10	DBCapy	Usecase	UC-07	Simple
11	DBCapy	Usecase	UC-08	Simple
12	DBCapy	Usecase	UC-09	Average
13	DBCapy	Usecase	UC-10	Simple
14	DBCapy	Usecase	UC-11	Simple
15	DBCapy	Usecase	UC-12	Average
16	DBCapy	Usecase	UC-13	Simple
17	DBCapy	Usecase	UC-14	Simple
18	DBCapy	Usecase	UC-15	Simple
19	DBCapy	Usecase	UC-16	Simple
2	DBCapy	Actor	Sistema	Simple
20	DBCapy	Usecase	UC-17	Simple
21	DBCapy	Usecase	UC-18	Simple
22	DBCapy	Usecase	UC-19	Average
23	DBCapy	Usecase	UC-20	Simple
24	DBCapy	Usecase	UC-21	Simple
25	DBCapy	Usecase	UC-22	Simple

Figura 1. Definición de actores y Casos de Uso en EZEstimate

Lo siguiente que debemos añadir es la complejidad percibida que tenemos de los **ECF** y de **TCF**.

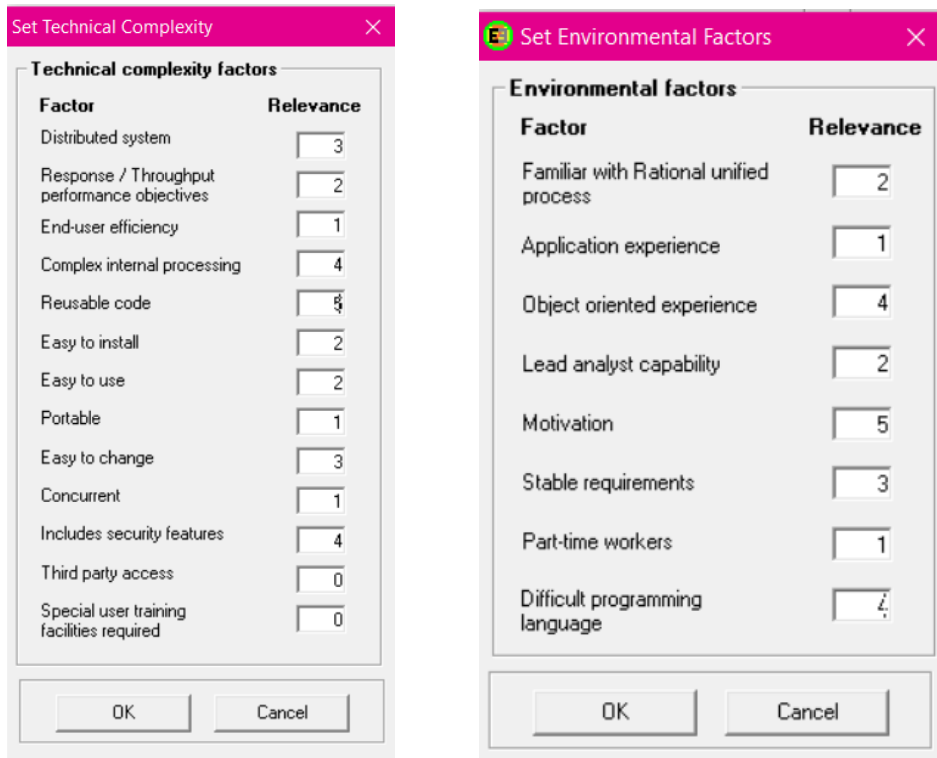


Figura 2. Definición de factores de complejidad técnica y del entorno en EZEstimate

Teniendo ya todos los datos, se puede proceder a hacer la estimación temporal. Para ello EZEestimate calculará primero los puntos de Caso de Uso, para luego poder convertirlos en una estimación de tiempo (medido en horas).

La fórmula será  $UCP * f$  (factor de conversión). Este factor de conversión simplemente es una estimación de en cuantas horas se realizará un UCP.

El resultado es el siguiente:

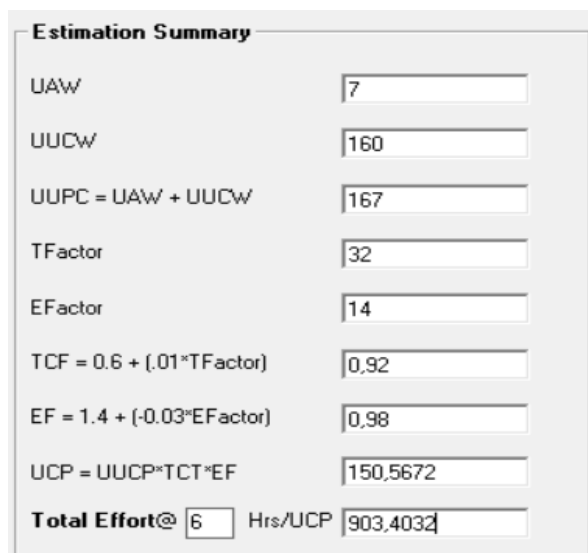


Figura 3. Cálculo de la estimación temporal en EZEstimate

***Manuel Santa Isabel Mayo***

Una estimación de 903 horas, las cuales son muy razonables a la propuesta de creación del proyecto de fin de grado.

El proyecto se pretende realizar en seis meses y suponiendo que se trabajaría un total de cuarenta horas semanales.

### 3. PLANIFICACIÓN TEMPORAL

Ya creada nuestra estimación temporal, vamos a realizar nuestra planificación. Para poder crear la planificación temporal, usaremos la herramienta **Project Profesional**. Con ella podremos visualizar nuestra planificación. Dividiremos el programa en diferentes tareas para asignar los tiempos y recursos.

Para hacer nuestra planificación, dividiremos este apartado en los siguientes puntos:

- Elaboración de un calendario de trabajo
- Identificación de tareas/subtareas
- División temporal y de recursos
- Relación entre tareas

La elaboración de cada uno de estos apartados concluirá con un gráfico de Gantt, el cual expone todas las tareas para la planificación.

### 3.1. Elaboración de un calendario de trabajo

Lo primero que debemos aclarar es lo que se mencionó anteriormente en el cálculo de la estimación del esfuerzo.

Se pretende trabajar cinco horas a la semana (se van a excluir los fines de semana) un total de ocho horas diarias. Esto se entiende como 40 horas semanales. Por tanto, añadiremos este horario dentro de nuestro proyecto Project. También debemos incluir las fechas que son festivas.

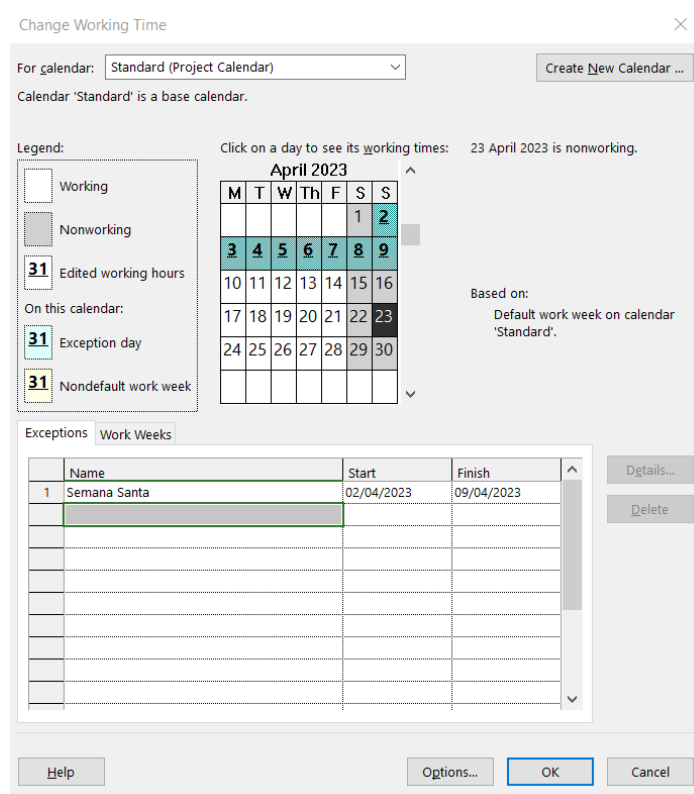


Figura 4. Configuración del calendario en Microsoft Project

Añadiremos al calendario Standard solo una fecha que consideraremos festiva que es Semana Santa. El resto del tiempo se dedicará a trabajar cinco días a la semana. No se incluye por ejemplo el periodo de exámenes finales u otras fechas dentro del calendario ya que el desarrollador no tiene dificultades para trabajar dichos días.

Al tratarse de un proyecto desarrollado por solo una persona y con una fecha límite en septiembre, pueden aparecer distintas semanas en las cuales el desarrollador trabaje los fines de semana y descansa dos días de entre diario. Lo que se pretende es siempre tener todas las semanas una media de cuarenta horas de trabajo

### 3.2. Identificación de tareas/subtareas

A la hora de plantear como dividir el proyecto en distintas tareas, debemos recordar que estamos trabajando con el modelo de Proceso Unificado.

Debemos tener en cuenta que el Proceso Unificado se basa en cuatro etapas que se corresponden con la vida y desarrollo de un proyecto: Inicio, Elaboración, Construcción y Transición. Cada una de estas se encuentra dividida en diferentes iteraciones. Una iteración está compuesta por tareas.

Para dividir las tareas lo organizaremos en seis clases distintas: Modelo de negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas.

De esta forma podremos diferenciar tareas más enfocadas a la investigación de tareas como la integración de una nueva parte a nuestro proyecto.

En el caso concreto del sistema web que estamos desarrollando, dividiremos todo el proyecto en 5 iteraciones. Las dos primeras iteraciones tendrán un gran peso a la hora de analizar e investigar herramientas, sacar requisitos y también crear análisis iniciales. Por ejemplo, nos centramos mucho en estas dos primeras iteraciones en la estructura, herramientas que tiene el chatbot, ya que es una de las piezas fundamentales de nuestra web.

En las dos siguientes iteraciones, tiene un peso más grande la implementación y el testeo de los subsistemas que contiene el sistema web.

Por último, la última iteración apenas encontraremos tareas, ya que se supone que en esta parte ya el proyecto está prácticamente testeado en todos los apartados, solo haciendo hincapié en las pruebas finales del sistema. También en esta iteración se pretende trabajar con el último subsistema que es el de estadísticas, ya que es un objetivo del proyecto que necesita una versión final para poder ponerse en práctica.

Tabla 7. Definición de las tareas del proyecto

Nombre		Descripción
0	Inicio Proyecto	Marca temporal que sirve como referencia para identificar el comienzo del proyecto
1	Iteración1	La primera etapa se hace una gran inversión en la investigación y en decidir hacia qué rumbo va a ir el proyecto. Al tratarse de una etapa...
	1.1 Modelo Negocio	
	1.1.1	Investigar proyectos similares existentes
		La primera tarea en la que vamos a trabajar es en la búsqueda de otros proyectores parecidos al nuestro. En nuestro caso buscamos plataformas de aprendizaje que usen chatbots como parte de la enseñanza.

1.1.2	Investigar para el uso de herramientas de Chatbot	Se investigarán distintas herramientas que puedan ser usadas como chatbot dentro de un servicio web.
1.1.3	Investigar sobre herramientas para el desarrollo de la web	Se investigarán los lenguajes, herramientas, etc. que puedan ser utilizadas en el entorno de creación de la web que deseamos.
<b>1.2 Requisitos</b>		
1.2.1	Reunión con cliente	Tendremos una reunión con los clientes (en este caso se trata de las dos tutoras del proyecto) para planear y organizar las distintas opciones para la creación del proyecto. También se deben mencionar los posibles objetivos que debe tener el proyecto.
1.2.2	Identificar los objetivos	Debemos fijar todos los objetivos previamente discutidos con las tutoras sobre nuestro proyecto. En especial podemos hablar de que se fijan un gran número de objetivos relacionados con la creación del servicio del chatbot, pero sin olvidarse del resto de componentes del sistema.
1.2.3	Determinar ámbito del sistema	Se determina el ámbito donde vamos a trabajar. En este caso se determina como la web está orientada a estudiantes que desean aprender la materia de BBDD pero con la particularidad de un tutor virtual disponible en toda su sesión.
<b>1.3 Análisis</b>		
1.3.1	Realización del primer boceto de la arquitectura	Se esquematiza de forma tentativa la estructura que queremos que nuestro sistema tenga. Aquí planeamos cómo vamos a dividir la web en dos secciones, cómo se interactuará con el chatbot, la relación entre materia práctica y teórica.
<b>1.4 Diseño</b>		
1.4.1	Distribución de los subsistemas	Se hace una distribución del sistema en diferentes subsistemas de forma tentativa.
<b>1.5 Implementación</b>		



	1.5.1	Descargar entorno de desarrollo	Se descarga tanto los entornos de trabajo de ReactJS, Node, Django y PostgreSQL para la creación de nuestra web.
	1.6 Pruebas		
	1.6.1	Pruebas sobre el entorno de desarrollo	Primeras pruebas de los entornos de trabajo. Se busca familiarizarnos con estas herramientas.
2	Hito 1		
3	Iteración 2		En esta segunda iteración nos centramos sobre todo en la idea de nuestro chatbot. Aunque todavía es una etapa en la que no hay tanto peso en cuanto a implementación y pruebas.
	3.1 Modelo Negocio		
	3.1.1	Investigar sobre redes neuronales de lenguaje Natural	Se decide que el chatbot que vamos a utilizar será en parte una red neuronal que procese lenguaje natural, debemos investigar cómo funciona esta herramienta y ver como implementarla dentro de nuestro sistema.
	3.1.2	Investigar sobre funcionalidad de ChatGPT	La segunda herramienta que compondrá el chatbot es la herramienta de inteligencia artificial ChatGPT, debemos investigar sus utilidades y sobre todo sus limitaciones. Debido a esto último, decidimos que vamos a hacer una limitación en cuanto al número de llamadas y a la longitud de nuestros mensajes.
	3.2 Requisitos		
	3.2.1	Definición del modelo inicial de requisitos	Primera definición de requisitos, ya teniendo en cuenta todos los objetivos y con la idea de cómo se forma el chatbot, vamos a redactar de forma tentativa los distintos requisitos que nuestro sistema va a tener.
	3.2.1	Definición inicial CU del subsistema <b>"Gestión de red neuronal"</b>	Realizar una primera definición del caso de uso "Gestión de red neuronal". El subsistema se encargará con todo lo relacionado con la herramienta de red neuronal que crearemos.

3.2.3	Definición inicial CU del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Realizar una primera definición del caso de uso “Gestión de ChatGPT desde Teoría”. El subsistema se encargará con todo lo relacionado con la herramienta ChatGPT.
3.3 Análisis		
3.3.1	Definición inicial análisis del subsistema “ <b>Gestión de red neuronal</b> ”	Realizar un primer análisis del caso de uso “Gestión de red neuronal”.
3.3.2	Definición inicial análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Realizar un primer análisis del caso de uso “Gestión de ChatGPT desde Teoría”.
3.4 Diseño		
3.4.1	Diseño tentativo del subsistema “ <b>Gestión de red neuronal</b> ”	Realizar un primer diseño del subsistema en cuestión, debido a que se trata de una primera iteración del diseño, este irá siendo definido con mayor exactitud en próximas iteraciones
3.4.2	Diseño tentativo del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Realizar un primer diseño del subsistema en cuestión, debido a que se trata de una primera iteración del diseño, este irá siendo definido con mayor exactitud en próximas iteraciones
3.4.3	Diseño arquitectura inicial	Partiendo de la iteración anterior 1.4.1 “Distribución de los subsistemas”, vamos a hacer el primer diseño tentativo de nuestro sistema. Este diseño al tratarse de un aproximamiento muy temprano puede verse cambiado a lo largo del proyecto.
3.5 Implementación		
3.5.1	Despliegue de una red neuronal de lenguaje natural	En esta primera implementación comenzaremos a crear un modelo de red neuronal que pueda ser utilizado como chatbot dentro de nuestra aplicación. Buscamos ante todo que sea determinista para poder enlazar nuestra red neuronal con la API de ChatGPT. No se completa como una implementación tan grande ya que solo estamos comenzando a implementarla, en las próximas iteraciones se mejorará junto a su resto de funcionalidades.

	3.5.2	Implementación de una interfaz de llamadas a ChatGPT	Esta segunda implementación es mucho más sencilla, no obstante, nos enfrentaremos a ella de una forma mucho más densa en un futuro con la mejor de ingeniería de prompts. De momento solo debemos implementar la llamada a la API.
	3.6 Pruebas		
	3.6.1	Primeras pruebas de llamadas ChatGPT	Las llamadas serán simples solo para comprobar el correcto funcionamiento de la implementación.
	3.6.2	Primeras pruebas con la RNN	En estas primeras pruebas con la red neuronal solo estamos buscando comprobar el funcionamiento de este modelo.
4	Hito 2		
5	Iteración 3		En esta tercera iteración ya habrá un mayor peso a la hora de implementar y probar nuestro bot. Nos centramos solo en las funciones del apartado de teoría, ya que son muy similares a las del apartado de práctica. Sobre todo, queremos un chatbot mucho más refinado y a lo largo de esta etapa vemos cómo evolucionan los dos subsistemas que lo involucran,
	5.1 Modelo Negocio		
	5.1.1	Profundización en conocimientos de redes neuronales del lenguaje	Buscamos investigar para mejorar la actual red neuronal.
	5.1.2	Investigación en la ingeniería de prompts	Buscamos refinar nuestras peticiones a ChatGPT, para ello haremos uso de la ingeniería de prompts.
	5.2 Requisitos		
	5.2.1	Revisiones con clientes	Se mostrará a las tutoras todo lo creado y hacia donde estamos orientando el proyecto.
	5.2.2	Refinamiento de los requisitos del subsistema “ <b>Gestión de red neuronal</b> ”	Refinamiento de los requisitos del subsistema “ <b>Gestión de red neuronal</b> ” en conclusión con la reunión con las tutoras y las tareas del modelo de negocio.
	5.2.3	Refinamiento de los requisitos del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Refinamiento de los requisitos del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ” en conclusión con la

		reunión con las tutoras y las tareas del modelo de negocio.
5.2.4	Definición inicial del resto de subsistemas	Partiendo del punto 3.2.1 “Definición del modelo inicial de requisitos” y con la reunión con las tutoras se realizará una definición menos abstracta sobre el resto de los subsistemas.
5.3 Análisis		
5.3.1	Realizar matriz de rastreabilidad	Se creará una matriz de rastreabilidad para poder analizar los distintos requisitos funcionales relacionados con los objetivos planteados para nuestro sistema.
5.3.2	Refinamiento del análisis del subsistema “ <b>Gestión de red neuronal</b> ”	Refinamiento del análisis del subsistema “ <b>Gestión de red neuronal</b> ” en función de las funciones obtenidas del modelo de negocio y de los requisitos.
5.3.3	Refinamiento del análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Refinamiento del análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ” en función de las funciones obtenidas del modelo de negocio y de los requisitos.
5.4 Diseño		
5.4.1	Refinamiento del diseño del subsistema “ <b>Gestión de red neuronal</b> ”	Refinamiento del diseño del subsistema “ <b>Gestión de red neuronal</b> ” en función de las funciones obtenidas del análisis.
5.4.2	Refinamiento del diseño del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Refinamiento del diseño del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ” en función de las funciones obtenidas del análisis.
5.5 Implementación		
5.5.1	Implementar el subsistema “ <b>Gestión de red neuronal</b> ”	Implementación del resto de las funciones de la red neuronal. Queremos tener una red neuronal refinada y que cumpla con los requisitos especificados.
5.5.2	Implementar el subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	Implementación del resto de las funciones de las llamadas a ChatGPT desde teoría. Sobre todo, queremos implementar además de todos los requisitos, lo aprendido en el modelo de negocio sobre la ingeniería de prompts.
5.6 Pruebas		

	5.6.1	Pruebas del subsistema “ <b>Gestión de red neuronal</b> ”	Probar como la red neuronal se integra correctamente, ver como da respuestas de forma rápida (la red neuronal no es perfecta, por tanto, tiene un cierto porcentaje de error). Lo más importante es que se cumpla los requisitos y poder enlazar nuestra red neuronal con ChatGPT.
	5.6.2	Pruebas del subsistema “ <b>Gestión de ChatGPT desde Teoría I</b> ”	Probar como las llamadas a ChatGPT dan los resultados esperados, además de una buena integración dentro del sistema.
	5.6.3	Pruebas de integración	Pruebas generales de integración de estos dos subsistemas dentro de la plataforma web.
6	Hito 3		
7	Iteración 4		En esta cuarta iteración se centrará por completo en funcionalidades dentro de la implementación y pruebas. El resto de las funciones no tienen tanto protagonismo. Vamos a trabajar en el resto de los subsistemas de nuestro proyecto para integrarlo junto al creado chatbot.
	7.1 Modelo Negocio		
	7.1.1	Investigar interfaces para estudiantes	Buscaremos diferentes interfaces y ejemplos para crear nuestra interfaz para usuarios (centrado en este caso en estudiantes).
	7.1.2	Explorar contenido para añadir dentro de la web	Exploraremos qué contenido queremos dentro de nuestra web. Se nos especificó anteriormente que debe ser materia sobre la asignatura de BBDD de la carrera de Ing. Informática de la USAL.
	7.2 Requisitos		
	7.2.1	Refinamiento requisitos subsistema “ <b>Contenido Web</b> ”	Refinamiento requisitos iniciales del subsistema “ <b>Contenido Web</b> ” en función de las conclusiones del modelo de negocio.
	7.2.2	Refinamiento requisitos subsistema “ <b>Gestión con chatbot desde práctica</b> ”	Refinamiento requisitos iniciales del subsistema “ <b>Gestión con chatbot desde práctica</b> ” en función de las conclusiones del modelo de negocio.
	7.2.3	Refinamiento requisitos subsistema “ <b>Gestión de usuarios</b> ”	Refinamiento requisitos iniciales del subsistema “ <b>Gestión de usuarios</b> ”

		en función de las conclusiones del modelo de negocio.
7.3 Análisis		
7.3.1	Refinamiento análisis subsistema “ <b>Contenido Web</b> ”	Refinamiento del análisis inicial del subsistema “ <b>Contenido Web</b> ” en función de las conclusiones del modelo de negocio.
7.3.2	Refinamiento análisis subsistema “ <b>Gestión con chatbot desde práctica</b> ”	Refinamiento del análisis inicial del subsistema “ <b>Gestión con chatbot desde práctica</b> ” en función de las conclusiones del modelo de negocio.
7.3.3	Refinamiento análisis subsistema “ <b>Gestión de usuarios</b> ”	Refinamiento del análisis inicial del subsistema “ <b>Gestión de usuarios</b> ” en función de las conclusiones del modelo de negocio.
7.4 Diseño		
7.4.1	Refinamiento diseño subsistema “ <b>Contenido Web</b> ”	Refinamiento del diseño inicial del subsistema “ <b>Contenido Web</b> ” en función de las conclusiones del modelo de negocio y el análisis.
7.4.2	Refinamiento diseño subsistema “ <b>Gestión con chatbot desde práctica</b> ”	Refinamiento del diseño inicial del subsistema “ <b>Gestión con chatbot desde práctica</b> ” en función de las conclusiones del modelo de negocio y el análisis.
7.4.3	Refinamiento diseño subsistema “ <b>Gestión de usuarios</b> ”	Refinamiento del diseño inicial del subsistema “ <b>Gestión de usuarios</b> ” en función de las conclusiones del modelo de negocio y el análisis.
7.5 Implementación		
7.5.1	Implementación subsistema “ <b>Contenido Web</b> ”	Implementación del subsistema de contenido web.
7.5.2	Implementación subsistema “ <b>Gestión con chatbot desde práctica</b> ”	Implementación del subsistema del chatbot desde práctica. Esta implementación es más sencilla ya habiendo dedicado la investigación de cómo usar ChatGPT.
7.5.3	Implementación subsistema “ <b>Gestión de usuarios</b> ”	Implementación del subsistema de gestión de usuarios.
7.5.4	Implementación de mejora de prompts	Mejora y refinamiento de los prompts. Implementando esta mejora en el nuevo subsistema añadido de chatbot en práctica.
7.5.5	Implementación de interfaz para usuario	Refinamiento y expansión de una interfaz para los usuarios. Al utilizar React, ya se incluía una interfaz, pero muy pobre. Ahora buscamos

		algo mucho más atractivo para los estudiantes.
	7.6 Pruebas	
	7.6.1	Pruebas de la integración de los nuevos subsistemas
		Pruebas de las diferentes integraciones y como afectan al resto de subsistemas. Sobre todo, a la hora de tener un usuario que pueda utilizar las funciones del sistema habiéndose registrado.
	7.6.2	Pruebas de usuarios con el chatbot
		Aunque se menciona en la anterior tarea, se hace hincapié en la tarea del uso de un usuario al chatbot (esto incluye el envío, recepción, guardado de mensajes).
8	Hito 4	
9	Iteración 5	
	9.1 Modelo Negocio	
	9.1.1	Revisión de posibles riesgos existentes en el sistema
		Visionado final de posibles riesgos que pueda sufrir nuestro proyecto ya terminado. Por ejemplo, contemplar posibles límites en el uso de ChatGPT que debemos tener en cuenta si decidimos escalar más el sistema.
	9.2 Requisitos	
	9.2.1	Refinamiento de los requisitos del subsistema " <b>Gestión estadísticas</b> "
		Refinamiento requisitos iniciales del subsistema " <b>Gestión estadísticas</b> " en función de las conclusiones del modelo de negocio.
	9.2.2	Refinamiento de los requisitos de riesgos existentes en el sistema
		Continuando con la tarea 9.1.1 del modelo de negocio, debemos refinar todos los posibles riesgos que nuestro sistema pueda tener.
	9.3 Análisis	
	9.3.1	Refinamiento del análisis del subsistema " <b>Gestión estadísticas</b> "
		Refinamiento del análisis inicial del subsistema " <b>Gestión estadísticas</b> " en función de las conclusiones del modelo de negocio.
	9.3.2	Revisión final del modelo del análisis
		Última revisión de nuestro modelo de análisis teniendo todo el proyecto finalizado.
	9.4 Diseño	
	9.4.1	Refinamiento del análisis del subsistema " <b>Gestión estadísticas</b> "
		Refinamiento del diseño inicial del subsistema " <b>Gestión estadísticas</b> " en función de las conclusiones del modelo de negocio y el análisis.

	9.4.2	Revisión final del diseño	Última revisión de nuestro diseño teniendo todo el proyecto finalizado.
	9.5 Implementación		
	9.5.1	Implementación de las estadísticas	Implementación del subsistema de estadísticas, el cual se añade al final del proyecto para que pueda ser usado por los usuarios con todo nuestro sistema funcional y probado.
	9.5.2	Implementación y despliegue de los cambios	Implementación de los posibles cambios que se han producido en la versión final.
	9.6 Pruebas		
	9.6.1	Últimas pruebas sobre el sistema final	Pruebas de revisión finales para verificar el cumplimiento de todos los requisitos y funciones que nuestro sistema ofrece.
10	Fin del proyecto		



### 3.3. División temporal y de recursos

Ya con todas nuestras tareas clasificadas y catalogadas, debemos asignarlas unos tiempos que sean proporcionales a la cantidad de esfuerzo que queremos invertir dentro de nuestro proyecto. El tiempo que se estimó con EZEstimate era de 903 horas, así que buscaremos un número similar en nuestra distribución.

Además, debemos siempre tener en cuenta las características especificadas de cada una de nuestras iteraciones.

Para hacernos una idea de cómo vamos a dividir los distintos días, realizamos la siguiente tabla.

*Tabla 8. Repartición del tiempo entre las diferentes iteraciones*

Iteración	Días
Iteración 1	6,25
Iteración 2	12,5
Iteración 3	37,5
Iteración 4	50
Iteración 5	6,25

Además de asignarlas un tiempo a cada tarea, se asigna un recurso humano (en este caso el desarrollador). Al solo ser un único desarrollador, vamos a invertir el 100% de nuestro recurso para cada una de nuestras tareas. Aunque muchas de las tareas se realizan de forma paralela, pondremos que se realizan al 100% ya que pretendemos trabajar a la vez en muchas de las tareas ya que se considera que están relacionadas.

*Tabla 9. Asignación temporal y de recursos por tarea*

Identificador	Nombre	Tiempo (días)	Cantidad de recurso
1.1.1	Investigar proyectos similares existentes	3	100%
1.1.2	Investigar para el uso de herramientas de Chatbot	3	100%
1.1.3	Investigar sobre herramientas para el desarrollo de la web	3	100%
1.2.1	Reunión con cliente	0,25	100%
1.2.2	Identificar los objetivos	1	100%
1.2.3	Determinar ámbito del sistema	1	100%

1.3.1	Realización del primer boceto de la arquitectura	1	100%
1.4.1	Distribución de los subsistemas	0,5	100%
1.5.1	Descargar entorno de desarrollo	0,25	100%
1.6.1	Pruebas sobre el entorno de desarrollo	0,5	100%
3.1.1	Investigar sobre redes neuronales de lenguaje Natural	2	100%
3.1.2	Investigar sobre funcionalidad de ChatGPT	2	100%
3.2.1	Definición del modelo inicial de requisitos	4	100%
3.2.1	Definición inicial CU del subsistema <b>“Gestión de red neuronal”</b>	4	100%
3.2.3	Definición inicial CU del subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	4	100%
3.3.1	Definición inicial análisis del subsistema <b>“Gestión de red neuronal”</b>	3	100%
3.3.2	Definición inicial análisis del subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	3	100%
3.4.1	Diseño tentativo del subsistema <b>“Gestión de red neuronal”</b>	1	100%
3.4.2	Diseño tentativo del subsistema <b>“Gestión de</b>	1	100%

	<b>ChatGPT desde Teoría</b>		
3.4.3	Diseño arquitectura inicial	1	
3.5.1	Despliegue de una red neuronal de lenguaje natural	2	100%
3.5.2	Implementación de una interfaz de llamadas a ChatGPT	2	100%
3.6.1	Primeras pruebas de llamadas ChatGPT	2	100%
3.6.2	Primeras pruebas con la RNN	2	100%
5.1.1	Profundización en conocimientos de redes neuronales del lenguaje	1	100%
5.1.2	Investigación en la ingeniería de prompts	1	100%
5.2.1	Revisiones con clientes	1	100%
5.2.2	Refinamiento de los requisitos del subsistema <b>“Gestión de red neuronal”</b>	1	100%
5.2.3	Refinamiento de los requisitos del subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	1	100%
5.2.4	Definición inicial del resto de subsistemas	1	100%
5.3.1	Realizar matriz de rastreabilidad	3	100%
5.3.2	Refinamiento del análisis del subsistema <b>“Gestión de red neuronal”</b>	3	100%
5.3.3	Refinamiento del análisis del subsistema	3	100%

	<b>“Gestión de ChatGPT desde Teoría”</b>		
5.4.1	Refinamiento del diseño del subsistema <b>“Gestión de red neuronal”</b>	5	100%
5.4.2	Refinamiento del diseño del subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	5	100%
5.5.1	Implementar el subsistema <b>“Gestión de red neuronal”</b>	8,5	100%
5.5.2	Implementar el subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	8,5	100%
5.6.1	Pruebas del subsistema <b>“Gestión de red neuronal”</b>	5	100%
5.6.2	Pruebas del subsistema <b>“Gestión de ChatGPT desde Teoría”</b>	5	100%
5.6.3	Pruebas de integración	5	100%
7.1.1	Investigar interfaces para estudiantes	1	100%
7.1.2	Explorar contenido para añadir dentro de la web	1	100%
7.2.1	Refinamiento requisitos subsistema <b>“Contenido Web”</b>	2	100%
7.2.2	Refinamiento requisitos subsistema	2	100%

	<b>“Gestión con chatbot desde práctica”</b>		
7.2.3	Refinamiento requisitos subsistema <b>“Gestión de usuarios”</b>	2	100%
7.3.1	Refinamiento análisis subsistema <b>“Contenido Web”</b>	3	100%
7.3.2	Refinamiento análisis subsistema <b>“Gestión con chatbot desde práctica”</b>	3	100%
7.3.3	Refinamiento análisis subsistema <b>“Gestión de usuarios”</b>	3	100%
7.4.1	Refinamiento diseño subsistema <b>“Contenido Web”</b>	5	100%
7.4.2	Refinamiento diseño subsistema <b>“Gestión con chatbot desde práctica”</b>	5	100%
7.4.3	Refinamiento diseño subsistema <b>“Gestión de usuarios”</b>	5	100%
7.5.1	Implementación subsistema <b>“Contenido Web”</b>	5	100%
7.5.2	Implementación subsistema <b>“Gestión con chatbot desde práctica”</b>	5	100%
7.5.3	Implementación subsistema <b>“Gestión de usuarios”</b>	5	100%
7.5.4	Implementación de mejora de prompts	2	100%

7.5.5	Implementación de interfaz para usuario	3	100%
7.6.1	Pruebas de la integración de los nuevos subsistemas	10	100%
7.6.2	Pruebas de usuarios con el chatbot	10	100%
9.1.1	Revisión de posibles riesgos existentes en el sistema	1	100%
9.2.1	Refinamiento de los requisitos del subsistema <b>"Gestión estadísticas"</b>	1	100%
9.2.2	Refinamiento de los requisitos de riesgos existentes en el sistema	1	100%
9.3.1	Refinamiento del análisis del subsistema <b>"Gestión estadísticas"</b>	1	100%
9.3.2	Revisión final del modelo del análisis	2	100%
9.4.1	Refinamiento del diseño del subsistema <b>"Gestión estadísticas"</b>	1	100%
9.4.2	Revisión final del diseño	1	100%
9.5.1	Implementación de las estadísticas	1	100%
9.5.2	Implementación y despliegue de los cambios	1	100%
9.6.1	Últimas pruebas sobre el sistema final	1	100%

### 3.4. Relación entre tareas

El último punto que tendremos en cuenta antes de realizar nuestro diagrama de Gantt es la relación que pueden existir entre las distintas tareas.

La gran mayoría están relacionadas de la siguiente forma: cuando una fase terminase, comenzará la siguiente, creando así una relación secuencial.

No obstante, cuando trabajamos con tareas dentro de un ámbito en concreto (por ejemplo, Modelo de Negocio), se suelen trabajar de forma paralela distintas funciones.

Este caso es posible al estar trabajando un solo desarrollador, ya que un mismo día de trabajo pueden dedicarse a combinar los distintos trabajos de forma conjunta. En otras palabras, podremos dedicarle el 100% a diferentes tareas al mismo tiempo ya que muchas se combinan y tienen un sentido conjunto.

También encontraremos tareas dentro del mismo ámbito que sí necesitan una relación secuencial, ya que el resultado final de una implica el inicio de la siguiente. Pero la mayoría de las tareas se trabajarán de manera paralela.

Las relaciones entre las distintas tareas son las siguientes

Tabla 10. Relaciones entre tareas

ID	Nombre	Relación
1.1.1	Investigar proyectos similares existentes	-
1.1.2	Investigar para el uso de herramientas de Chatbot	-
1.1.3	Investigar sobre herramientas para el desarrollo de la web	-
1.2.1	Reunión con cliente	1.1.3 Investigar sobre herramientas para el desarrollo de la web
1.2.2	Identificar los objetivos	1.1.3 Investigar sobre herramientas para el desarrollo de la web
1.2.3	Determinar ámbito del sistema	1.1.3 Investigar sobre herramientas para el desarrollo de la web
1.3.1	Realización del primer boceto de la arquitectura	1.2.3 Determinar ámbito del sistema
1.4.1	Distribución de los subsistemas	1.3.1 Realización del primer boceto de la arquitectura
1.5.1	Descargar entorno de desarrollo	1.4.1 Distribución de los subsistemas
1.6.1	Pruebas sobre el entorno de desarrollo	1.5.1 Descargar entorno de desarrollo
3.1.1	Investigar sobre redes neuronales de lenguaje Natural	1.6.1 Pruebas sobre el entorno de desarrollo
3.1.2	Investigar sobre funcionalidad de ChatGPT	1.6.1 Pruebas sobre el entorno de desarrollo

3.2.1	Definición del modelo inicial de requisitos	3.1.2 Investigar sobre funcionalidad de ChatGPT
3.2.1	Definición inicial CU del subsistema “ <b>Gestión de red neuronal</b> ”	3.1.2 Investigar sobre funcionalidad de ChatGPT
3.2.3	Definición inicial CU del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	3.1.2 Investigar sobre funcionalidad de ChatGPT
3.3.1	Definición inicial análisis del subsistema “ <b>Gestión de red neuronal</b> ”	3.2.3 Definición inicial CU del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
3.3.2	Definición inicial análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	3.2.3 Definición inicial CU del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
3.4.1	Diseño tentativo del subsistema “ <b>Gestión de red neuronal</b> ”	3.3.2 Definición inicial análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
3.4.2	Diseño tentativo del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	3.3.2 Definición inicial análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
3.4.3	Diseño arquitectura inicial	3.3.2 Definición inicial análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
3.5.1	Despliegue de una red neuronal de lenguaje natural	3.4.3 Diseño arquitectura inicial
3.5.2	Implementación de una interfaz de llamadas a ChatGPT	3.4.3 Diseño arquitectura inicial
3.6.1	Primeras pruebas de llamadas ChatGPT	3.5.2 Implementación de una interfaz de llamadas a ChatGPT
3.6.2	Primeras pruebas con la RNN	3.5.2 Implementación de una interfaz de llamadas a ChatGPT
5.1.1	Profundización en conocimientos de redes neuronales del lenguaje	3.6.2 Primeras pruebas con la RNN
5.1.2	Investigación en la ingeniería de prompts	3.6.2 Primeras pruebas con la RNN
5.2.1	Revisiones con clientes	5.1.2 Investigación en la ingeniería de prompts
5.2.2	Refinamiento de los requisitos del subsistema “ <b>Gestión de red neuronal</b> ”	5.1.2 Investigación en la ingeniería de prompts
5.2.3	Refinamiento de los requisitos del subsistema “ <b>Gestión de chatGPT desde Teoría</b> ”	5.1.2 Investigación en la ingeniería de prompts
5.2.4	Definición inicial del resto de subsistemas	5.1.2 Investigación en la ingeniería de prompts



5.3.1	Realizar matriz de rastreabilidad	5.2.4Definición inicial del resto de subsistemas
5.3.2	Refinamiento del análisis del subsistema “ <b>Gestión de red neuronal</b> ”	5.2.4Definición inicial del resto de subsistemas
5.3.3	Refinamiento del análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	5.2.4Definición inicial del resto de subsistemas
5.4.1	Refinamiento del diseño del subsistema “ <b>Gestión de red neuronal</b> ”	5.3.3Refinamiento del análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
5.4.2	Refinamiento del diseño del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	5.3.3Refinamiento del análisis del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
5.5.1	Implementar el subsistema “ <b>Gestión de red neuronal</b> ”	5.4.2Refinamiento del diseño del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
5.5.2	Implementar el subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	5.5.1Implementar el subsistema “ <b>Gestión de red neuronal</b> ”
5.6.1	Pruebas del subsistema “ <b>Gestión de red neuronal</b> ”	5.5.2Implementar el subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
5.6.2	Pruebas del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”	5.5.2Implementar el subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
5.6.3	Pruebas de integración	5.6.2Pruebas del subsistema “ <b>Gestión de ChatGPT desde Teoría</b> ”
7.1.1	Investigar interfaces para estudiantes	5.6.3Pruebas de integración
7.1.2	Explorar contenido para añadir dentro de la web	5.6.3Pruebas de integración
7.2.1	Refinamiento requisitos subsistema “ <b>Contenido Web</b> ”	7.1.2Explorar contenido para añadir dentro de la web
7.2.2	Refinamiento requisitos subsistema “ <b>Gestión con chatbot desde práctica</b> ”	7.1.2Explorar contenido para añadir dentro de la web
7.2.3	Refinamiento requisitos subsistema “ <b>Gestión de usuarios</b> ”	7.1.2Explorar contenido para añadir dentro de la web
7.3.1	Refinamiento análisis subsistema “ <b>Contenido Web</b> ”	7.2.3Refinamiento requisitos subsistema “ <b>Gestión de usuarios</b> ”
7.3.2	Refinamiento análisis subsistema “ <b>Gestión con chatbot desde práctica</b> ”	7.2.3Refinamiento requisitos subsistema “ <b>Gestión de usuarios</b> ”
7.3.3	Refinamiento análisis subsistema “ <b>Gestión de usuarios</b> ”	7.2.3Refinamiento requisitos subsistema “ <b>Gestión de usuarios</b> ”

7.4.1	Refinamiento diseño subsistema “ <b>Contenido Web</b> ”	7.3.3Refinamiento análisis subsistema “ <b>Gestión de usuarios</b> ”
7.4.2	Refinamiento diseño subsistema “ <b>Gestión con chatbot desde práctica</b> ”	7.3.3Refinamiento análisis subsistema “ <b>Gestión de usuarios</b> ”
7.4.3	Refinamiento diseño subsistema “ <b>Gestión de usuarios</b> ”	7.3.3Refinamiento análisis subsistema “ <b>Gestión de usuarios</b> ”
7.5.1	Implementación subsistema “ <b>Contenido Web</b> ”	7.4.2Refinamiento diseño subsistema “ <b>Gestión con chatbot desde práctica</b> ”
7.5.2	Implementación subsistema “ <b>Gestión con chatbot desde práctica</b> ”	7.5.1Implementación subsistema “ <b>Contenido Web</b> ”
7.5.3	Implementación subsistema “ <b>Gestión de usuarios</b> ”	7.5.2Implementación subsistema “ <b>Gestión con chatbot desde práctica</b> ”
7.5.4	Implementación de mejora de prompts	7.5.3Implementación subsistema “ <b>Gestión de usuarios</b> ”
7.5.5	Implementación de interfaz para usuario	7.5.4Implementación de mejora de prompts
7.6.1	Pruebas de la integración de los nuevos subsistemas	7.5.5Implementación de interfaz para usuario
7.6.2	Pruebas de usuarios con el chatbot	7.6.1Pruebas de la integración de los nuevos subsistemas
9.1.1	Revisión de posibles riesgos existentes en el sistema	7.6.2Pruebas de usuarios con el chatbot
9.2.1	Refinamiento de los requisitos del subsistema “ <b>Gestión estadísticas</b> ”	9.1.1Revisión de posibles riesgos existentes en el sistema
9.2.2	Refinamiento de los requisitos de riesgos existentes en el sistema	9.1.1Revisión de posibles riesgos existentes en el sistema
9.3.1	Refinamiento del análisis del subsistema “ <b>Gestión estadísticas</b> ”	9.3.1Refinamiento de los requisitos de riesgos existentes en el sistema
9.3.2	Revisión final del modelo del análisis	9.3.1Refinamiento de los requisitos de riesgos existentes en el sistema
9.4.1	Refinamiento del diseño del subsistema “ <b>Gestión estadísticas</b> ”	9.3.2Revisión final del modelo del análisis
9.4.2	Revisión final del diseño	9.4.1Refinamiento del diseño del subsistema “ <b>Gestión estadísticas</b> ”
9.5.1	Implementación de las estadísticas	9.4.2Revisión final del diseño
9.5.2	Implementación y despliegue de los cambios	9.5.1Implementación de las estadísticas
9.6.1	Últimas pruebas sobre el sistema final	9.5.2Implementación y despliegue de los cambios

### **3.5. Diagrama de Gantt**

Lo último que realizaremos es el diagrama de Gantt, el cual supone la visualización de todos los datos antes mostrados.

Para poder crear este diagrama meteremos todas las iteraciones, hitos y tareas dentro de nuestro proyecto de Windows Project.

# Manuel Santa Isabel Mayo

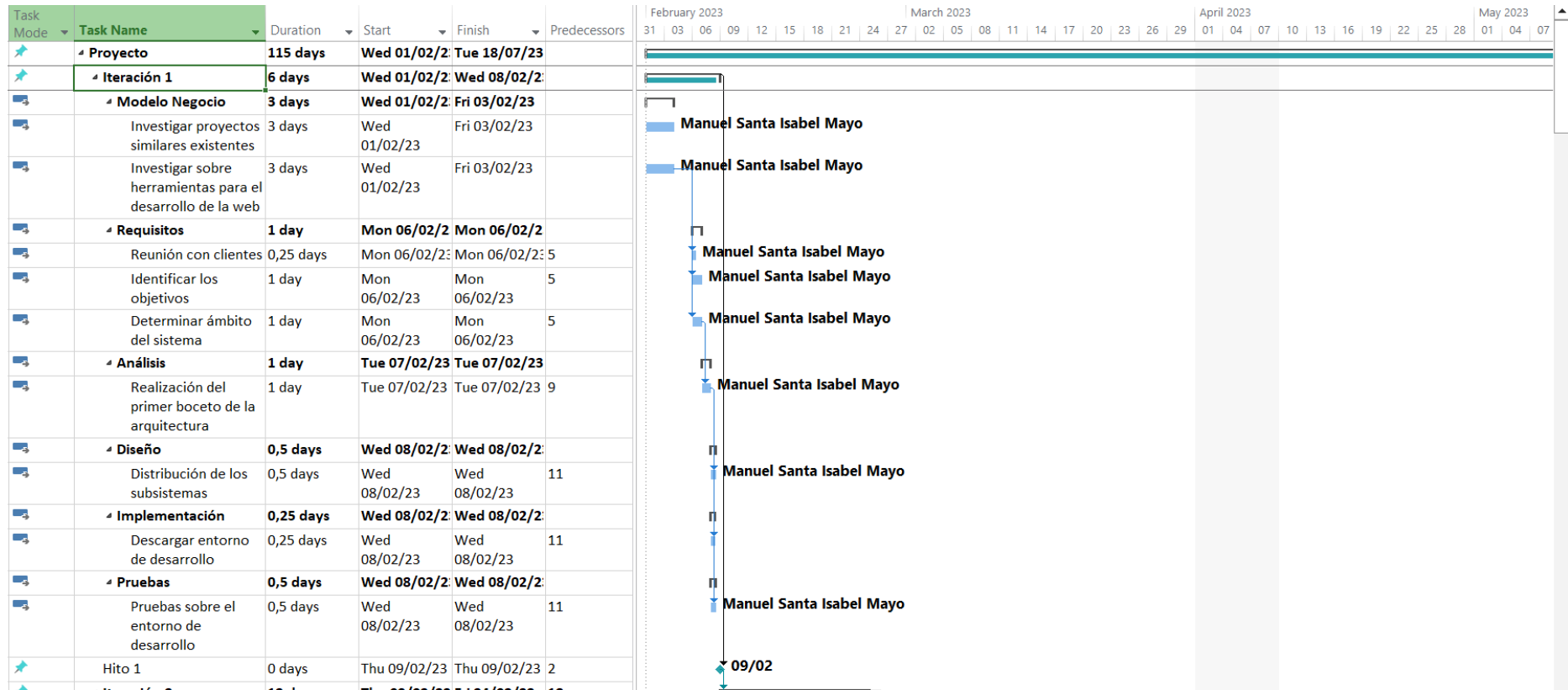


Figura 5. Diagrama de Gantt 1

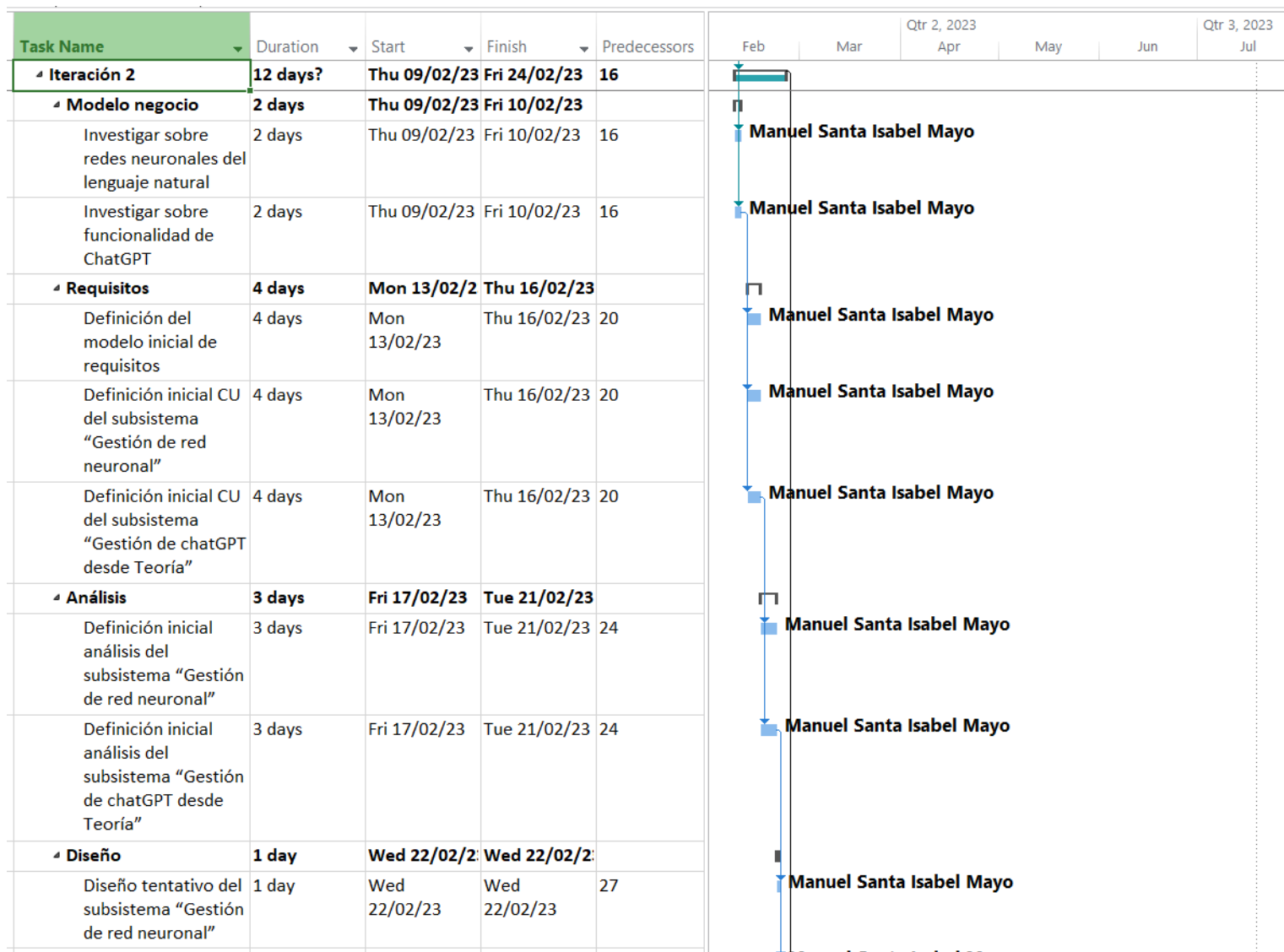


Figura 6. Diagrama de Gantt 2

**Manuel Santa Isabel Mayo**

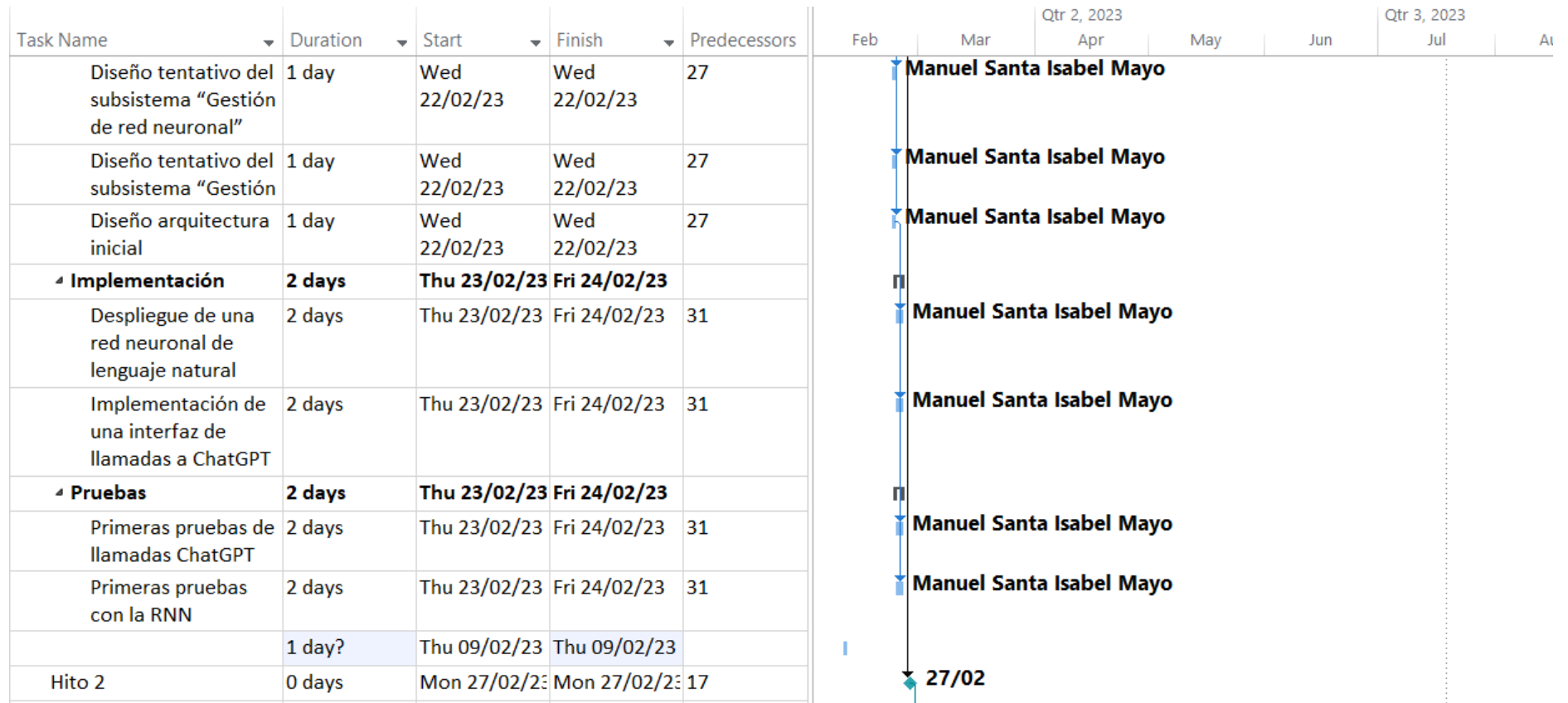


Figura 7. Diagrama de Gantt 3

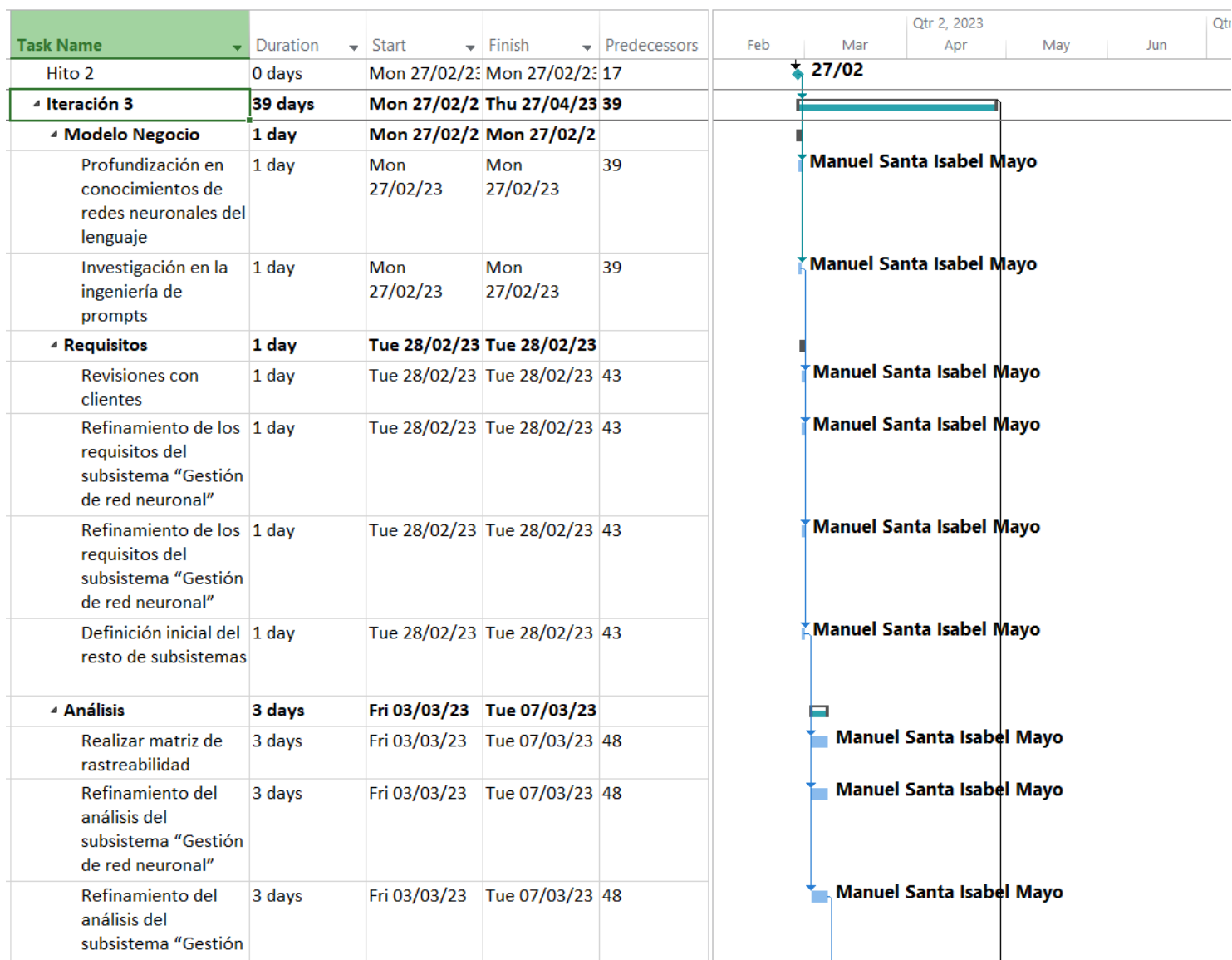


Figura 8. Diagrama de Gantt 4

## Manuel Santa Isabel Mayo

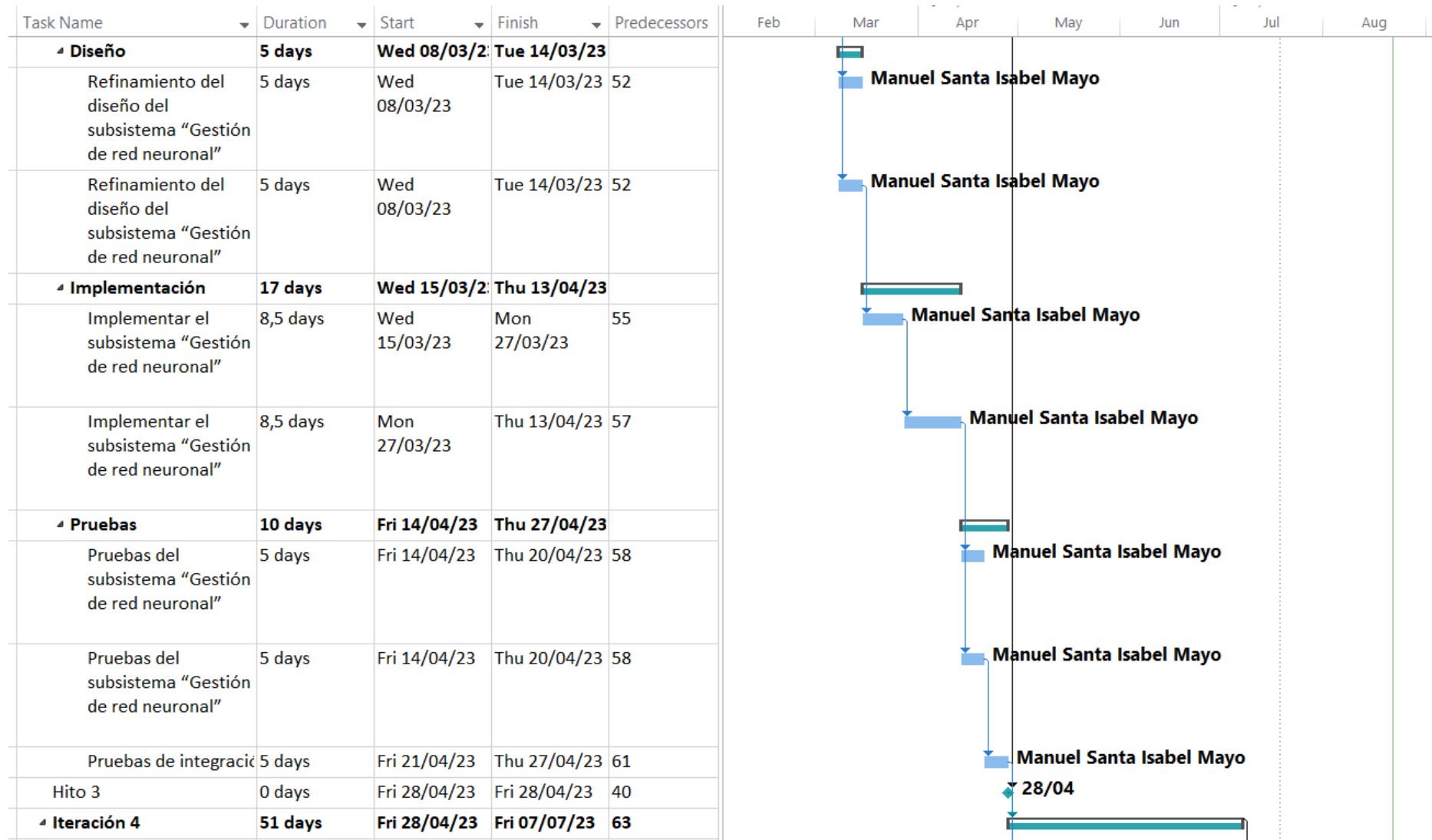


Figura 9. Diagrama de Gantt 5



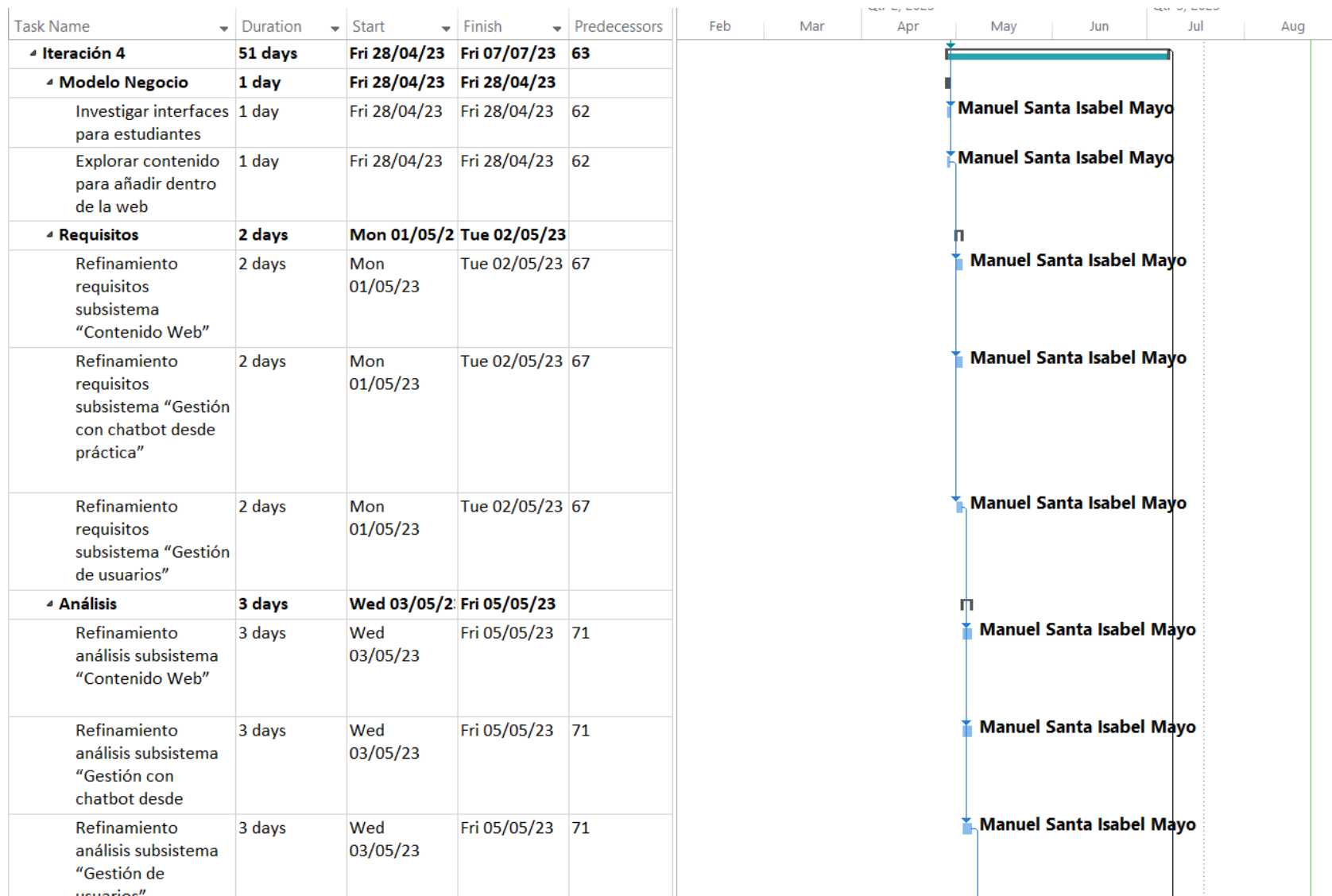


Figura 10. Diagrama de Gantt 6

## Manuel Santa Isabel Mayo

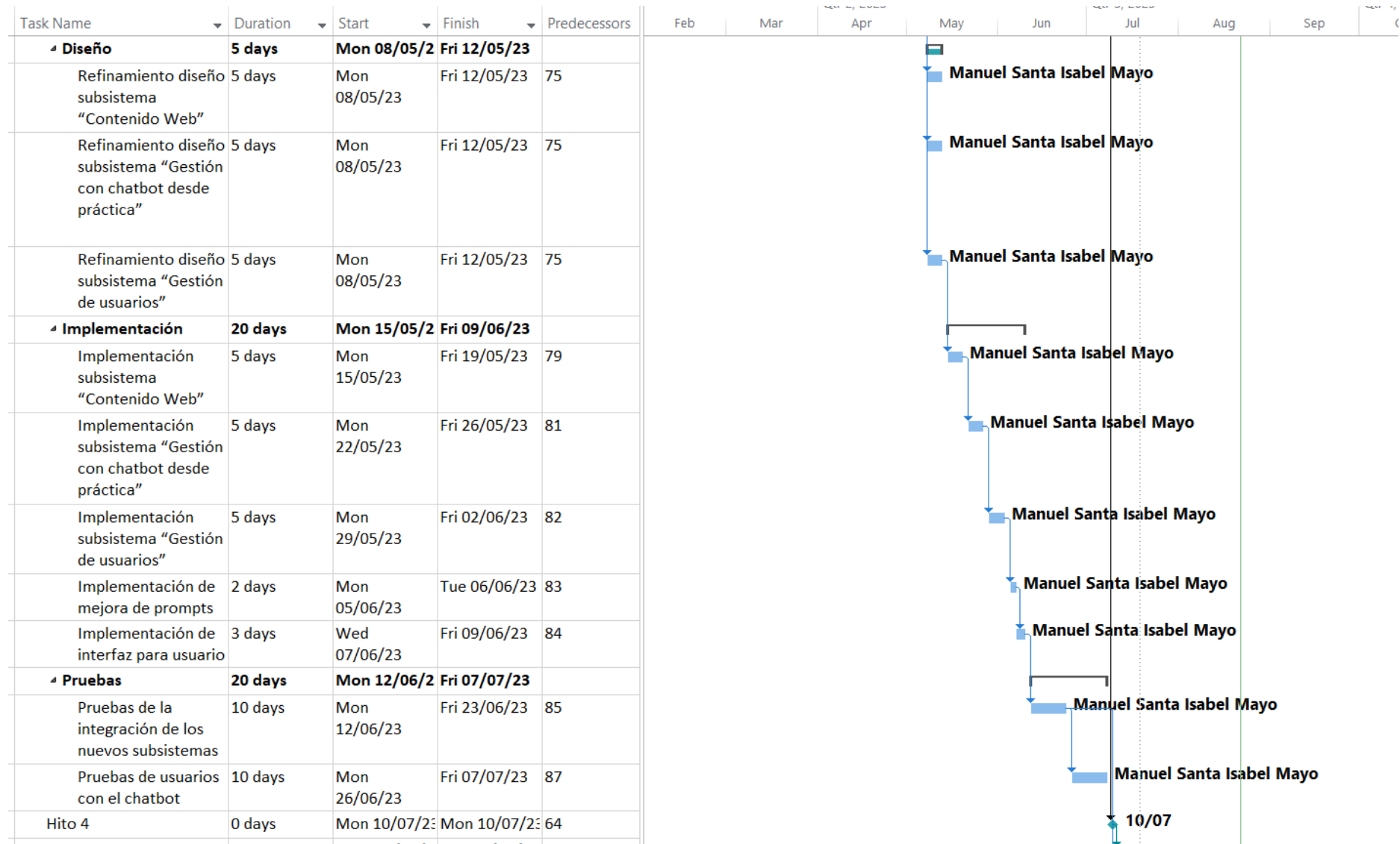


Figura 11. Diagrama de Gantt 7

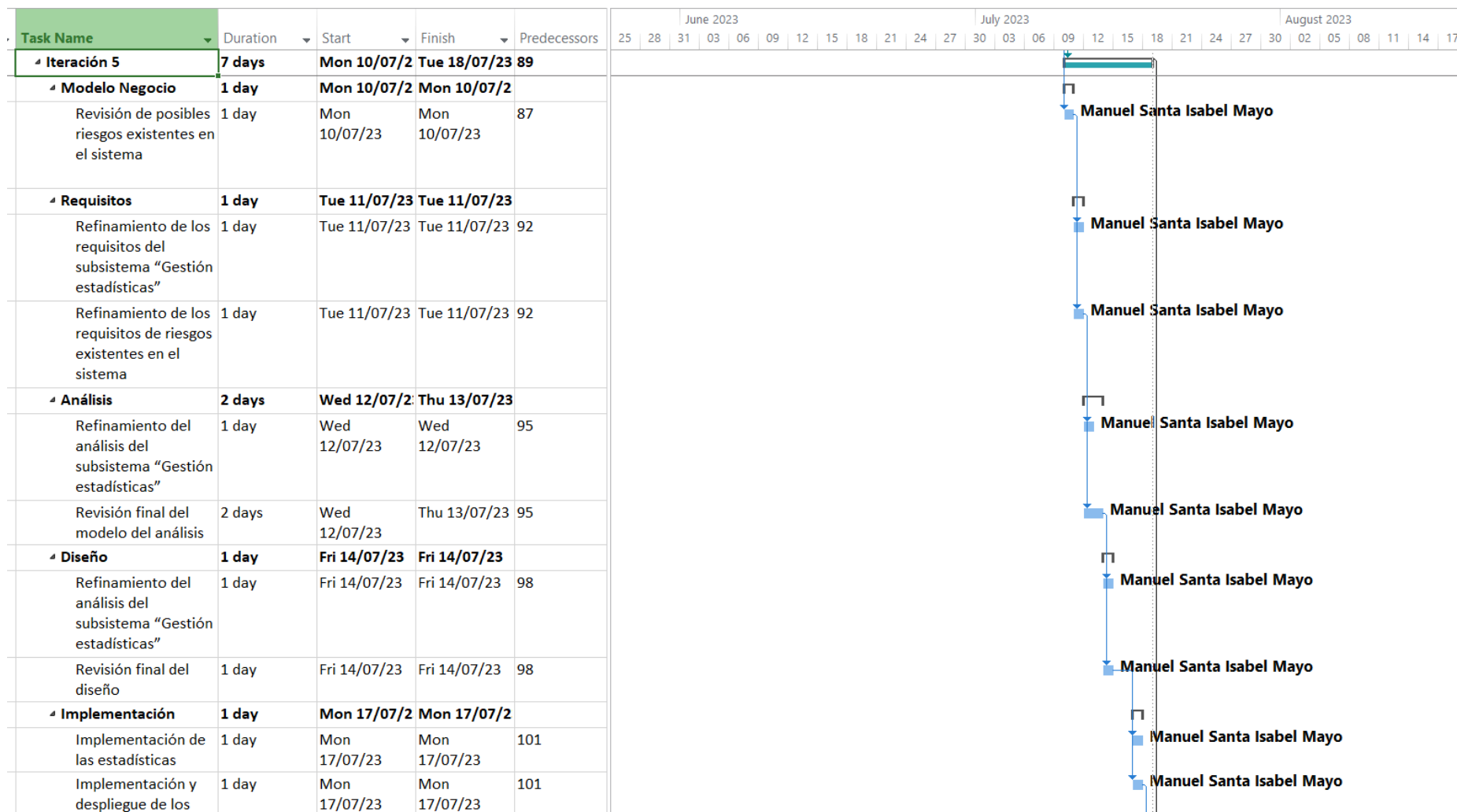


Figura 12. Diagrama de Gantt 8

## Manuel Santa Isabel Mayo

4 Implementación		1 day	Mon 17/07/23	Mon 17/07/23	
	Implementación de las estadísticas	1 day	Mon 17/07/23	Mon 17/07/23	101
	Implementación y despliegue de los cambios	1 day	Mon 17/07/23	Mon 17/07/23	101
4 Pruebas		1 day	Tue 18/07/23	Tue 18/07/23	
	Ultimas pruebas sobr	1 day	Tue 18/07/23	Tue 18/07/23	104
	Hito Final	0 days	Tue 18/07/23	Tue 18/07/23	90

Figura 13. Diagrama de Gantt 9

