

MyParallelOrg  
DEDUCCIÓN DE JERARQUÍAS  
ALTERNATIVAS EN UNA COMPAÑÍA EN  
BASE A INTERACCIONES DE  
USUARIOS

Anexo I. Plan de proyecto software



TRABAJO DE FIN DE GRADO  
INGENIERÍA INFORMÁTICA

Yoel Pérez Carrasco

Dirigido por Fernando De La Prieta Pintado y Walter Vinci

Salamanca, septiembre 2024

# Tabla de contenidos

1. Introducción.....	4
2. Estimación de esfuerzo.....	5
2.1. Complejidad de los Casos de uso.....	5
2.2. Complejidad de los Actores.....	6
2.3. Factores.....	7
2.3.1. Factores de Complejidad Técnica.....	7
2.3.2. Factores de Complejidad del Entorno.....	9
2.4. Análisis e Interpretación de resultados.....	11
3. Planificación Temporal.....	12
3.1. Elaboración de Calendario de Trabajo.....	12
3.2. Identificación de Tareas y Subtareas.....	12
3.3. Resultados.....	16
4. Bibliografía.....	17

## Lista de figuras

Figura 1: Fórmula Factores Complejidad Técnica.....	7
Figura 2: Factores Complejidad Técnica.....	7
Figura 3: Fórmula Factores Complejidad Técnica.....	9
Figura 4: Factores Complejidad Técnica.....	9
Figura 5: Estimación de Esfuerzo.....	11
Figura 6: Calendario laboral Planificación Temporal.....	12
Figura 7: Diagrama de Proceso Unificado [2].....	13
Figura 8: Fase Inicio.....	13
Figura 9: Fase Elaboración Iteración 1.....	14
Figura 10: Fase Elaboración Iteración 2.....	14
Figura 11: Fase Construcción Iteración 1.....	14
Figura 12: Fase Construcción Iteración 2.....	15
Figura 12: Fase Transición Iteración 1.....	15
Figura 13: Estadísticas del proyecto.....	16

## Lista de tablas

Tabla 1: Complejidad Casos de Uso.....	5
Tabla 2: Actor Usuario No Login.....	6
Tabla 3: Actor Worker.....	6
Tabla 4: Actor Admin.....	6
Tabla 5: Actor Manager.....	6
Tabla 6: Actor Usuario.....	6
Tabla 7: Actor <<System>>.....	6

# 1. Introducción

Este Anexo I presenta una descripción detallada del proceso de estimación del esfuerzo necesario para llevar a cabo el proyecto de desarrollo del Trabajo de Fin de Grado: "MyParallelOrg: Deducción de Jerarquías Alternativas en una Compañía en Base a Interacciones de Usuarios", así como de la planificación temporal que define las tareas a realizar y su distribución entre los recursos disponibles. El contenido de este anexo se organiza en dos partes principales:

- Estimación del Esfuerzo
- Planificación Temporal

Para la estimación del esfuerzo de desarrollo, se utiliza la métrica **UCP** (Use Case Points), la cual se basa en la funcionalidad del sistema representada a través de casos de uso. Esta metodología considera actores, escenarios y factores de complejidad técnica y ambiental. El cálculo de la estimación requiere determinar las siguientes variables:

- **UUCP** (Puntos de Caso de Uso Desajustados)
  - **UUCW** (Peso Desajustado de Casos de Uso)
  - **UAW** (Peso Desajustado de Actores)
- **TCF** (Factor de Complejidad Técnica)
- **ECF** (Factor de Complejidad del Entorno)

Una vez obtenidos estos valores, se calcula el UCP mediante la fórmula:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

El esfuerzo estimado se expresa a través de la siguiente ecuación:

$$\text{Esfuerzo} = \text{UCP} * F$$

donde "F" es un factor de conversión que representa el número de horas por persona por cada UCP.

En cuanto a la planificación temporal, se desarrolla un cronograma de trabajo, identificando las tareas siguiendo un enfoque basado en el Proceso Unificado, estableciendo las dependencias entre ellas. Este análisis permite determinar el camino crítico, identificando las tareas que no deben retrasarse para garantizar que el proyecto se complete dentro del plazo previsto. [1]

## 2. Estimación de esfuerzo

### 2.1. Complejidad de los Casos de uso

La complejidad de los casos de uso se evalúa en función del número de transacciones que cada caso de uso involucra. Una transacción se define como una interacción que comienza cuando el usuario realiza una acción y concluye cuando el sistema devuelve una respuesta. De acuerdo con esta clasificación, se asigna la complejidad de la siguiente manera:

- **Simple:** Si el caso de uso contiene hasta 3 transacciones.
- **Intermedio:** Si el caso de uso incluye entre 4 y 7 transacciones.
- **Complejo:** Si el caso de uso tiene más de 7 transacciones.

Este enfoque permite categorizar los casos de uso de acuerdo con su nivel de interacción y facilitar la estimación del esfuerzo necesario para su desarrollo, teniendo en cuenta la cantidad de operaciones involucradas en cada uno.[\[1\]](#)

Caso de Uso	Transacciones	Complejidad
UC-001 Registrar Usuario	1	Simple
UC-002 Logout	1	Simple
UC-003 Login Usuario	1	Simple
UC-004 Cambiar Contraseña	2	Simple
UC-005 Crear Usuario	1	Simple
UC-006 Ver Usuarios	2	Simple
UC-007 Editar Usuario	3	Simple
UC-008 Eliminar Usuario	2	Simple
UC-009 Leer Datos JSON	1	Simple
UC-010 Cargar grafo 3D	3	Simple
UC-011 Crear hub Herramientas	3	Simple
UC-012 Modificar Vista Visualización 3D	4	Simple
UC-013 Modificar Posicionamiento Cámara	2	Simple
UC-014 Ver ayuda	1	Simple
UC-015 Ver Gráfico De Empresa	4	Simple
UC-016 Autenticar API	3	Simple
UC-017 Obtener Datos API	2	Simple
UC-018 Procesar datos Github	1	Simple
UC-019 Ver Conexiones	2	Simple
UC-020 Exportar CSV Conexiones	1	Simple
UC-021 Generar CSV Conexiones	1	Simple
UC-022 Realizar Conexiones	1	Simple
UC-023 Procesar Conexiones	2	Simple
UC-024 Incluir Organización	1	Simple
UC-025 Procesar Organizaciones de Usuarios Github	1	Simple
UC-026 Listar Organizaciones	3	Simple

Tabla 1: Complejidad Casos de Uso

## 2.2. Complejidad de los Actores

Teniendo en cuenta las distintas complejidades que pueden tomar cada uno de los actores, se estipulan tres valores:

- **Simple:** Cuando el actor es un sistema y la aplicación interactúa con él a través de una API.
- **Intermedio:** Cuando el actor es un sistema y la aplicación se conecta utilizando un protocolo.
- **Complejo:** Cuando el actor es un usuario que interactúa a través de una interfaz gráfica [1].

ACT - 001	Usuario No Login
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

*Tabla 2: Actor Usuario No Login*

ACT - 002	Worker
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

*Tabla 3: Actor Worker*

ACT - 003	Admin
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

*Tabla 4: Actor Admin*

ACT - 004	Manager
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

*Tabla 5: Actor Manager*

ACT - 005	Usuario
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

*Tabla 6: Actor Usuario*

ACT - 006	<<System>>
Complejidad	Simple
Comentarios	Es un sistema que se comunica con la aplicación mediante una API

*Tabla 7: Actor <<System>>*

## 2.3. Factores

En el proceso de estimación del esfuerzo de desarrollo, se identifican un conjunto de factores que influyen directamente en la complejidad del proyecto. Estos factores se dividen en dos categorías: factores de complejidad técnica y factores de complejidad del entorno.

A cada uno de estos factores, tanto técnicos como de entorno, se le asigna un valor que representa la magnitud de su influencia en el proyecto. Este valor se determina en base a las características específicas del proyecto y las capacidades del equipo de desarrollo.

### 2.3.1. Factores de Complejidad Técnica

Se establecen trece factores que representan distintos aspectos técnicos del sistema. Estos factores reflejan las características del proyecto en términos de requisitos tecnológicos, arquitectura, rendimiento, seguridad, entre otros. Cada uno de estos factores se evalúa en función de cómo afecta al desarrollo del proyecto. Por ejemplo, la existencia de sistemas distribuidos o la necesidad de un rendimiento elevado incrementa la complejidad técnica y, por ende, el esfuerzo necesario. [1]

$$TCF = C_1 + C_2 \sum_{i=1}^{13} W_i F_i$$

Figura 1: Fórmula Factores Complejidad Técnica

Factores de complejidad técnica		
T <sub>1</sub> Sistemas distribuidos	T <sub>6</sub> Facilidad de instalación	T <sub>11</sub> Características especiales de seguridad
T <sub>2</sub> Rendimiento	T <sub>7</sub> Facilidad de uso	
T <sub>3</sub> Eficiencia del usuario final	T <sub>8</sub> Portabilidad	T <sub>12</sub> Acceso directo a terceras partes
T <sub>4</sub> Procesamiento interno complejo	T <sub>9</sub> Facilidad de cambio	T <sub>13</sub> Se requiere entrenamiento especial del usuario
T <sub>5</sub> Reusabilidad	T <sub>10</sub> Concurrencia	

Figura 2: Factores Complejidad Técnica

- **Sistemas Distribuidos.** Valor 3. Debido a que es un sistema distribuido, mediante llamadas a APIS externas, teniendo en cuenta la base de datos de carácter externo y el uso e implementación de dockers.
- **Rendimiento.** Valor 2. La aplicación web debe tener una respuesta rápida entre el usuario y el sistema. La parte del sistema encargada de mostrar los



enlaces entre workers debe llevarlo a cabo maximizando el rendimiento para no perder información.

- **Eficiencia del Usuario Final. Valor 1.** El usuario podrá completar su tarea de manera eficiente, sin necesidad de realizar una cantidad excesiva de pasos.
- **Procesamiento Interno Complejo. Valor 4.** La visualización en 3D presenta una complejidad significativa debido a que el procesamiento de grandes volúmenes de datos requiere una carga intensiva de nodos.
- **Reusabilidad. Valor 1.** El sistema solo será reutilizado para la autenticación, mientras que las demás funcionalidades son muy específicas y complejas, lo que limita su aplicabilidad en otros contextos.
- **Facilidad de Instalación. Valor 4.** La instalación del sistema es sencilla y se puede completar fácilmente mediante la ejecución de una serie de comandos.
- **Facilidad de Uso. Valor 4.** El diseño del sistema se enfocará en ser accesible e intuitivo para los usuarios finales, facilitando su interacción.
- **Portabilidad. Valor 3.** Gracias al uso de Docker y a la portabilidad de la aplicación web basada en ReactApp, el sistema podrá funcionar en diversos sistemas operativos y navegadores.
- **Facilidad de Cambio. Valor 2.** El sistema está dividido en módulos y controladores, lo que permite realizar modificaciones de manera rápida y eficiente.
- **Concurrencia. Valor 3.** El sistema ha sido diseñado para manejar múltiples usuarios al mismo tiempo, asegurando un funcionamiento concurrente.
- **Características Especiales de Seguridad. Valor 2.** El sistema debe proteger la información almacenada, permitiendo el acceso únicamente a usuarios autorizados para garantizar la privacidad.
- **Acceso Directo a Terceras Partes. Valor 0.** El sistema se ejecuta en un entorno cerrado, lo que significa que solo los usuarios internos tienen acceso al mismo.
- **Entrenamiento Especial del Usuario. Valor 0.** No se requiere capacitación especial para que los usuarios comiencen a utilizar el sistema desde el principio.

### 2.3.2. Factores de Complejidad del Entorno

Además de los factores técnicos, se consideran ocho factores relacionados con el entorno en el que se desarrollará el proyecto. Estos factores están ligados a las condiciones del equipo de trabajo, su experiencia, familiaridad con las tecnologías empleadas, y otros elementos contextuales. Por ejemplo, si el equipo tiene poca experiencia con el lenguaje de programación utilizado o si los requisitos del proyecto son inestables, la complejidad del entorno aumenta, lo que puede dificultar el progreso y requerir un mayor esfuerzo. [1]

$$ECF = C_1 + C_2 \sum_{i=1}^8 W_i F_i$$

Figura 3: Fórmula Factores Complejidad Técnica

Factores de complejidad del entorno	
E <sub>1</sub> Familiaridad con UML	E <sub>5</sub> Experiencia en orientación a objetos
E <sub>2</sub> Trabajadores a tiempo parcial	E <sub>6</sub> Motivación
E <sub>3</sub> Capacidad de los analistas	E <sub>7</sub> Dificultad del lenguaje de programación
E <sub>4</sub> Experiencia en la aplicación	E <sub>8</sub> Estabilidad de los requisitos

Figura 4: Factores Complejidad Técnica

- **Conocimiento de UML.** Valor 2. Aunque se ha estudiado UML y se ha trabajado con este en ejercicios prácticos y actividades relacionadas con el Proceso Unificado, no se cuenta con experiencia en su aplicación en proyectos reales.
- **Desarrollador a Tiempo Completo.** Valor 0. El proyecto será realizado por un único desarrollador, quien dedicará todo su tiempo exclusivamente a esta tarea, sin compartirlo con otras actividades.
- **Habilidades de Análisis.** Valor 0. El desarrollador carece de experiencia en técnicas de análisis, ya que no ha trabajado previamente con estas herramientas en proyectos anteriores.
- **Conocimiento de la Aplicación.** Valor 0. El desarrollador está comenzando el proyecto desde cero, sin tener experiencia previa con la aplicación específica que se va a desarrollar.

- **Experiencia en Programación Orientada a Objetos.** Valor 3. Aunque el desarrollador ha trabajado en varios proyectos pequeños utilizando programación orientada a objetos, aún no ha participado en el desarrollo de un proyecto completo de mayor envergadura.
- **Motivación.** Valor 5. El desarrollador está altamente motivado, ya que enfrenta el reto de completar su primer proyecto integral, lo que representa una oportunidad significativa para su crecimiento profesional.
- **Complejidad del Lenguaje de Programación.** Valor 1. Aunque los lenguajes de programación que se emplearán son de alto nivel, el desarrollador tiene experiencia limitada en algunos de ellos, lo que podría representar un desafío adicional durante el desarrollo.
- **Definición de Requisitos.** Valor 4. Los requisitos del sistema están claramente establecidos desde el principio, con un alto grado de estabilidad. Cualquier cambio que surja a lo largo del desarrollo será mínimo y no afectará significativamente al alcance del proyecto.

## 2.4. Análisis e Interpretación de resultados

Después de asignar la complejidad a los actores, casos de uso, factores de complejidad técnica y de entorno, se procede a ingresar estos valores en la herramienta EZEstimate para calcular la estimación del esfuerzo. Para este cálculo, es necesario establecer la cantidad de horas por punto de caso de uso, que en este proyecto es de 4.

**Module**  
 Gestión de Organizaciones  
 Add Module Delete

**Summary**  
 Total Modules: 6  
 Use cases: Simple 26, Average 0, Complex 0  
 Actors: Simple 1, Average 0, Complex 5  
 Excel Report: Generate Report

**Add Actor / Use case**  
 Actor / Use case Name:   
 Select Type: Usecase  
 Complexity: Simple  
 Add

**Tech / Env Factors**  
 Set Tech Factor  
 Set Env Factors

**Estimation Summary**  
 UAW: 16  
 UUCW: 130  
 UUPC = UAW + UUCW: 146  
 TFactor: 33  
 EFactor: 18  
 TCF = 0.6 + (.01\*TFactor): 0.93  
 EF = 1.4 + (-0.03\*EFactor): 0.86  
 UCP = UUPC\*TCF\*EF: 116,7708  
 Total Effort@ 4 Hrs/UCP: 467,0832

**Use case / Actor List (Double click to delete)**

Id	Module	Type	Name	complexity
30	Gestión de Org...	Usecase	Incluir Organiza...	Simple
31	Gestión de Org...	Usecase	Procesar Organ...	Simple
32	Gestión de Org...	Usecase	Listar Organiza...	Simple
10	Gestión de Usu...	Actor	Admin	Complex
11	Gestión de Usu...	Actor	Manager	Complex
12	Gestión de Usu...	Actor	Worker	Complex
13	Gestión de Usu...	Actor	Usuario	Complex
5	Gestión de Usu...	Usecase	Crear Usuario	Simple
6	Gestión de Usu...	Usecase	Ver Usuarios	Simple
7	Gestión de Usu...	Usecase	Editar Usuario	Simple
8	Gestión de Usu...	Usecase	Eliminar Usuario	Simple
9	Gestión de Usu...	Usecase	Cambiar Contra...	Simple
14	Gestión de Visu...	Actor	System	Simple
15	Gestión de Visu...	Usecase	Leer Datos JSON	Simple
16	Gestión de Visu...	Usecase	Cargar grado 3D	Simple
17	Gestión de Visu...	Usecase	Crear HUB Herr...	Simple
18	Gestión de Visu...	Usecase	Modificar Vista ...	Simple
19	Gestión de Visu...	Usecase	Modificar Posici...	Simple

Figura 5: Estimación de Esfuerzo

El resultado obtenido es que el proyecto se estima desarrollar en 467,08 horas.

### 3. Planificación Temporal

#### 3.1. Elaboración de Calendario de Trabajo

Se ha realizado un planteamiento del Calendario de trabajo de tal forma que se estipularon un total de 8 horas diarias de trabajo, de 9:00 a 13:00 o de 15:00 a 19:00, teniendo en cuenta una serie de días festivos y que se ha dedicado los días de lunes a viernes, realizando descansos los sábados y domingos, tomando como día de inicio el 26 de abril de 2024.

Figura 6: Calendario laboral Planificación Temporal

#### 3.2. Identificación de Tareas y Subtareas

La aplicación se desarrolla siguiendo el Proceso Unificado, un marco de desarrollo de software que organiza el trabajo en diferentes fases para garantizar un enfoque estructurado y metódico. Este proceso se divide en cuatro fases principales: Inicio, Elaboración, Construcción y Transición.

1. **Fase de Inicio:** En esta etapa se establecen los objetivos generales del proyecto, se definen los requisitos iniciales y se realiza una evaluación de riesgos. Es crucial asegurar que el proyecto tiene una base sólida antes de avanzar.
2. **Fase de Elaboración:** Aquí se profundiza en el análisis y diseño del sistema. Se define la arquitectura del software, se afina la estimación de recursos y tiempos, y se elaboran los planes detallados para la fase de construcción. Esta fase permite reducir la incertidumbre técnica y organizativa.
3. **Fase de Construcción:** En esta etapa se desarrolla el software de acuerdo con los requisitos y la arquitectura definidos en las fases anteriores. Es la fase

donde se produce la mayor parte del código y se realizan pruebas iniciales para garantizar la funcionalidad del sistema.

**4. Fase de Transición:** Finalmente, en esta fase, se entrega el software a los usuarios finales. Incluye actividades de despliegue, capacitación y soporte inicial, asegurando que el sistema esté listo para operar en el entorno de producción.

Cada una de estas fases abarca varias disciplinas y puede incluir un número determinado de iteraciones. Las iteraciones permiten ajustar y refinar el desarrollo del software, asegurando que el producto final cumple con los requisitos y expectativas del cliente. Estas iteraciones son ciclos repetitivos dentro de cada fase que permiten al equipo de desarrollo revisar, mejorar y evolucionar el software de manera continua antes de pasar a la siguiente fase.

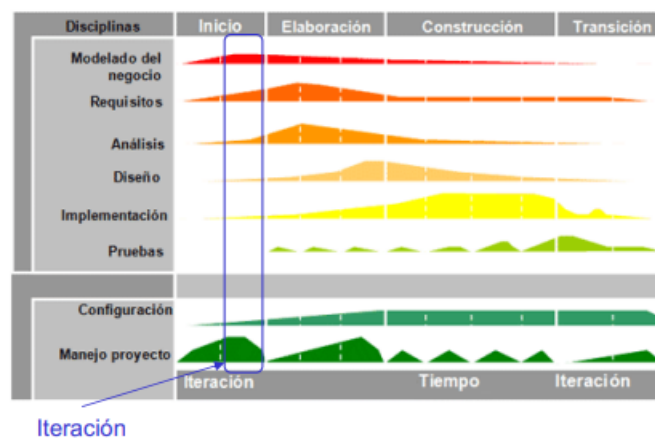


Figura 7: Diagrama de Proceso Unificado [2]

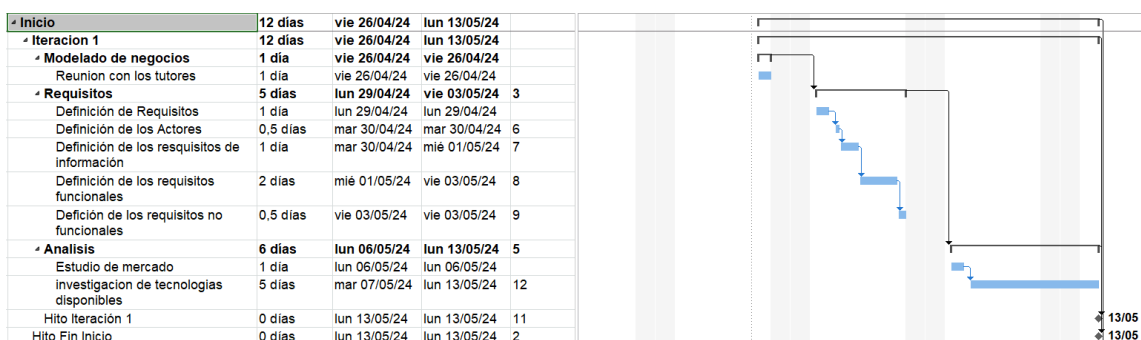


Figura 8: Fase Inicio

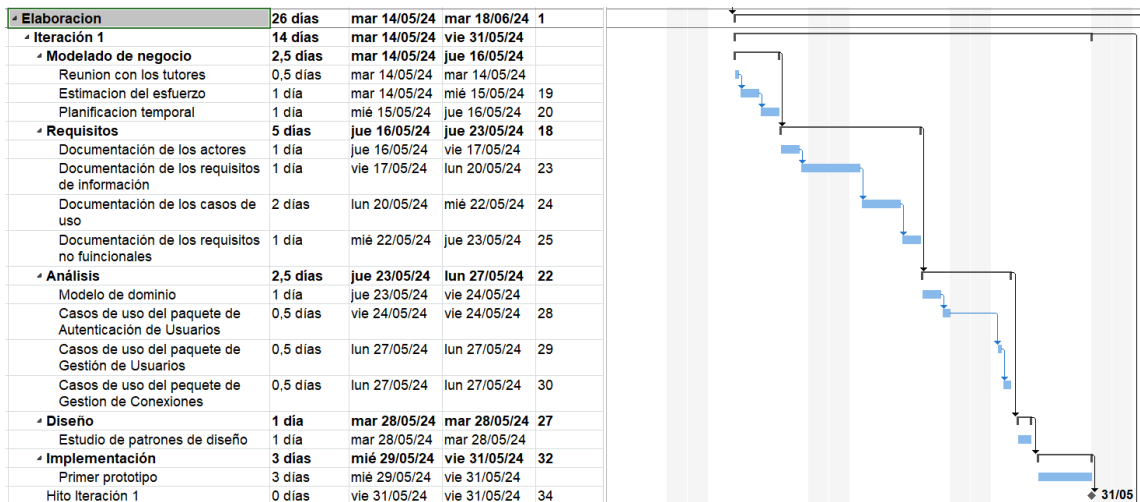


Figura 9: Fase Elaboración Iteración 1

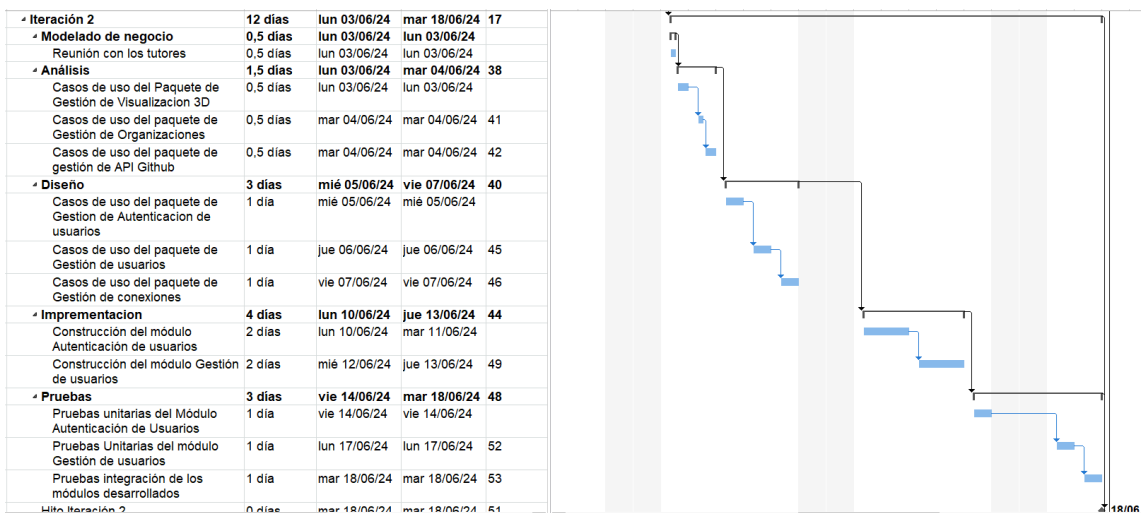


Figura 10: Fase Elaboración Iteración 2

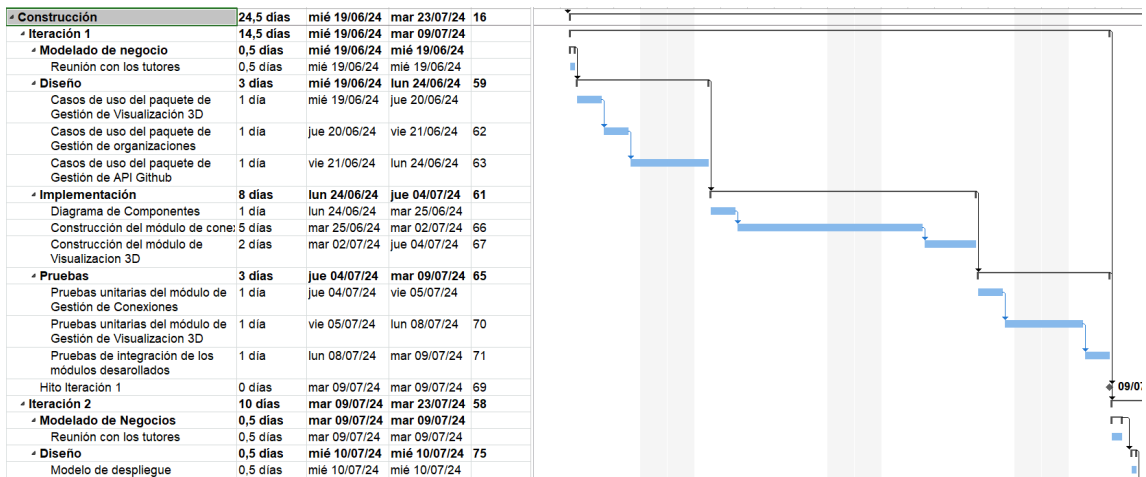


Figura 11: Fase Construcción Iteración 1

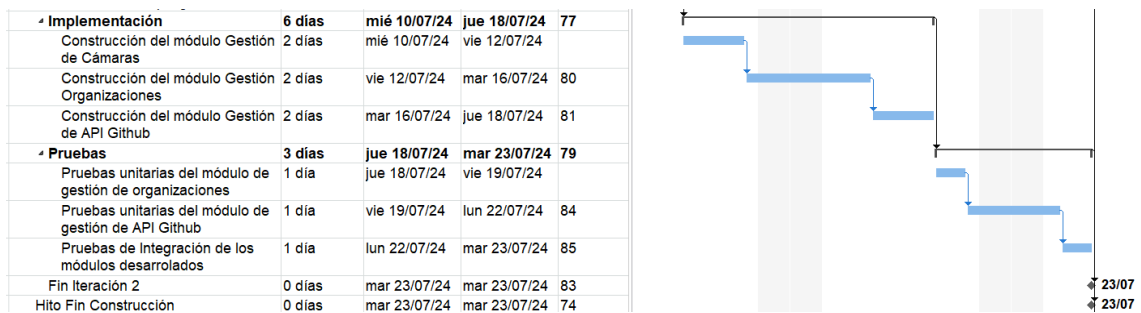


Figura 12: Fase Construcción Iteración 2

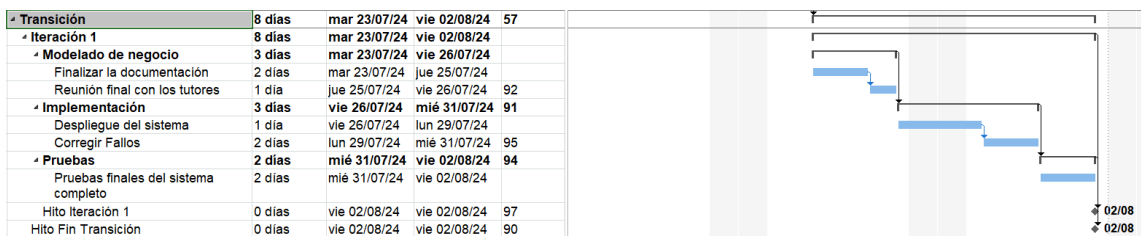


Figura 12: Fase Transición Iteración 1



### 3.3. Resultados

Estadísticas del proyecto 'Nuevo Microsoft Project Document.mpp' ✕

	Comienzo	Fin
Actual	vie 26/04/24	vie 02/08/24
Previsto	NOD	NOD
Real	NOD	NOD
Variación	0d	0d

	Duración	Trabajo	Costo
Actual	70,5d	500h	0,00 €
Previsto	0d	0h	0,00 €
Real	0d	0h	0,00 €
Restante	70,5d	500h	0,00 €

Porcentaje completado:

Duración: 0%

Trabajo: 0%

Cerrar

Figura 13: Estadísticas del proyecto

Tras la realización del cálculo de estadísticas temporales del proyecto se obtuvo un total de 500 horas, debido en gran medida a la complejidad del proyecto y al número de horas diarias destinadas a la realización del mismo, necesarias para lograr el objetivo previsto.

## 4. Bibliografía

1. GP-GII: Diapositivas. (s. f.). Recuperado 3 de septiembre de 2024, de <https://studium23.usal.es/mod/resource/view.php?id=126798>
2. GP-GII: Diapositivas. (s. f.). Recuperado 3 de septiembre de 2024, de <https://studium23.usal.es/mod/resource/view.php?id=126808>